

B.B.Mo'minov

DASTURLASH II



O‘ZBEKISTON RESPUBLIKASI
OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI

B.B.MO‘MINOV

DASTURLASH II

*O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta‘lim vazirligi tomonidan oliy
o‘quv yurtlarining talabalari uchun darslik sifatida tavsiya etilgan
(28.12.2020 yil 676 sonli buyrug‘i)*

TOSHKENT 2020

UO‘K: 004.4(075)

KBK 32.973.-01

Mo‘minov B.B., Dasturlash II. (Darslik). – T.: «Nihol print» OK, 2021. – 604 b.

ISBN 978–9943–7028–1–3

Darslikda noma'lum tiplar va noma'lum nomlar fazosi, STL kutubxonalari, oddiy, assosiativ va tartiblanmagan assosiativ konteynerlar hamda konteynerlar adapterlari tushunchalari, standart algoritmlar, iteratorlar va sonli sinflar bilan ishlash, sintaktik tahlil qilishni tashkil etishning samarali usullari va murakkab saralash algoritmlaridan unumli foydalanish, Visual C++ muhitida menyular va uskunalar paneli, komponenta tushunchasi va xususiyatlari, muloqot oynalari, Visual C++ ning grafik imkoniyatlari, OLE va MFC texnologiyalari hamda kichik loyihalarni yaratishning nazariy va amaliy asoslari keltirilgan.

UO‘K: 004.4(075)

KBK 32.973.-01

Taqrizchilar:

- Nuraliyev F.M.** – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti, Televizion texnologiyalari fakulteti dekani, t.f.d., dotsent;
- Raximov N.O.** – O‘zbekiston Milliy universiteti Jizzax filiali, o‘quv ishlari bo‘yicha direktor o‘rinbosari, t.f.d.

ISBN 987–9943– 7028–1–3

© B.B.Mo‘minov, 2021.

© «Nihol print» OK, 2021.

KIRISH

O‘zbekiston Respublikasi Prezidentining bir qator Qaror va Farmonlari Axborot-kommunikatsiya texnologiyalarini yanada jadal ruvojlantirish maqsadlarida qabul qilinmoqda. Ushbu qonunlardan “Raqamli iqtisodiyot va elektron hukumatni keng joriy etish chora-tadbirlari to‘g‘risi”dagi PQ-4699-son 28.04.2020 va “«Raqamli O‘zbekiston — 2030» strategiyasini tasdiqlash va uni samarali amalga oshirish chora-tadbirlari to‘g‘risi”dagi PF-6079-son 05.10.2020 yilgi qaror va farmonlarida takidlanganidek «Bir million dasturchi» loyihasi doirasida 500 ming nafar yoshlarni qamrab olish orqali kompyuter dasturlash asoslariga o‘qitish tashkillashtirish maqsadida Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universitetining barcha yo‘nalish talabalariga mo‘ljallangan “Dasturlash II” fani uchun darslikni yaratish dolzarb masala hisoblanadi.

Darslikning birinchi bobida noma’lum tiplar va noma’lum nomlar fazosi, STL kutubxonalari, oddiy konteynerlar, assosiativ va tartiblanmagan assosiativ konteynerlar hamda konteynerlar adapterlari tushunchlari va ulardan dasturlashda unumli foydalanish o‘rgatiladi.

Darslikning ikkinchi bobida standart algoritmlar, iteratorlar va sonli sinflar bilan ishlash hamda sintaktik tahlil qilishni tashkil etishning samarali usullari va murakkab saralash algoritmlaridan unumli foydalanish o‘rgatiladi.

Darslikning uchinchi bobida Visual C++ muhitida menyular va uskunalar paneli, komponenta tushunchasi va xususiyatlari, muloqot oynalari, Visual C++ning grafik imkoniyatlari, OLE va MFC texnologiyalari hamda kichik loyihalarni yaratish o‘rgatiladi.

Qo‘llanma bo‘yicha takliflar, undagi aniqlangan kamchiliklar bo‘yicha fikr-mulohazalaringizni mbbahodir@gmail.com elektron pochtasiga yuborishingizni so‘raymiz. Sizning bu bildirgan fikr-mulohazalaringiz qo‘llanmaning keyingi nashrlarida uni yanada mazmunliroq va kamchiliklardan xoli tarzda chop etishimizga albatta asqotadi.

Muallif

1-BOB. KONTEYNERLAR

1.1. Noma'lum tiplar va noma'lum nomlar fazosi.

 Tiplarni dinamik tarzda aniqlashning ba'zi usullari, `typeid` operatori, `typeid` kutubxonasi, `typeid` sinfi a'zolaridan foydalanish, dastur bajarilish davomida ko'rsatkichliva oddiy o'zgaruvchilarning tiplarini almashtirish operatorlari `dynamic_cast`, `const_cast`, `static_cast`, `reinterpret_cast` operatorlaridan foydalanish, noma'lum va foydalanuvchining nomlar fazolarini yaratish va dasturda ulardan foydalanish usullari, buferlashtirilgan kirishvachiqishni yaratish, IO amallarida formatlashning identifikatorlarivamonupulyatorlarini o'rnatish va o'qirish usullari keltirilgan bo'lib, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan.

Bilimlarni mustahkamlash uchun 25
tanazariy savol va amaliy ko'nikma va malakalarni rivojlantirish uchun 4
ta assistent topshirig'iva har assistentda 7 ta topshiriq, jami
28 ta topshiriq berilgan.

Butopshiriqlarni bajarish mavzu ni mustahkamlash uchun xizmat qiladi.

 **Kalitso'zlar.** Tipni aniqlash, tipni almashtirish, nomlar fazosi, buferlashtirilgan kirishvachiqish, `typeid`, `typeid`, `typeid` sinfi, `dynamic_cast`, `const_cast`, `static_cast`, `reinterpret_cast`, `namespace`, `istream`, oqim, formatlash identifikatori, monupulyatorli formatlash.

 **Bilish shart bo'lgan tushunchalar.** Tip tushunchasi, sinf va sinf ob'ekti, polimorf sinf, kasting amali, dastur fragmenti va soxasi, kutubxona, sinfi a'zolari, funksiyavako'rsatkich, identifikator, noma'lum otlarni kirishvachiqishi, oqim, dasturlashga oid dastlabki tushunchalar va C++ tilini o'qib o'rganish va tushunish uchun hitda ishlatishni bilish lozim.

Bilib olasiz.

Ixtiyoriy ko'rsatkichli o'zgaruvchining tipni aniqlash va tipni almashtirish usullari, `typeid`, `typeid`, `typeid` sinfi, `dynamic_cast`, `const_cast`, `static_cast`, `reinterpret_cast` yangi operatorlar, kalitso'z `namespace` va nomlar fazosini yaratish, ulardan dasturda foydalanish usullari va texnologiyalari, buferlashtirilgan kirishvachiqish orqalima'lum otlarni massivga joylatirish va `istream`, oqim, formatlash identifikatori, monupulyatorli formatlash usullari orqalilokalizatsiyalashgan IO operatorlarini yaratish va foydalanish usullari o'rganishingiz mumkin.

REJA

1. Tiplarnidynamiktarzdaaniqlash.
2. Tiplarnialmashtirishoperatorlari.
3. Yanginomlarfazosiyaratish.
4. Buferlashtirilgankiritishvachiqarish.

KIRISH

Noma'lumtiplarvanoma'lumfazolaruchundinamiktipleiniqlashvati pinialmashtirishoperatorlarizamonaviyob'ektagayonaltirilgandasturlashni iqlab-quvvatlovchi C++ dasturlashtiliningikkitavositlari: dinamiktipleiniqlash (runtime type identification - RTTI) vaqoshimchatipquyishoperatorlaribilanbog'liq. Buvositalarningbiri C++ spesifikatsiyasiqismiemasdi, lekinharikki runtime polimorfizmchunqo'lab-quvvatlashoshirishchunqoshilgan. RTTI dasturbajarishdavomidaob'ektturiniqlashimkoniyatimuvjudliginibildi radi. Bundamuhokamaqilinayotgantipturitashlashoperatorlari-buamalnibajarishchundasturchigaxavfsizyo'llarinitaklifqiladi. Ulardanbiri, dynamic_cast, bevosita RTTI aniqlashbilanbog'liq.

Tiplarnidynamiktarzdaaniqlash. Ushbuvositapolimorfbo'lmagantil larda (masalan, C) mavjudemasligisabablidinamikturdagiidentifikatsiyalashbilantanishbo'lm asligigiz mumkin, chunkiharbirob'ektningtipikompilyatsiyavaqtida (yani, dasturnibajarishda) ma'lumbo'lganiuchundasturbajarilishidama'lumotolishninghojatiyo'q. Biroq, C++ kabipolimorftillardaob'ektipikompilyatsiyavaqtidanoma'lumbo'lganhol atlarbo'lishim mumkin, chunkidasturbajarilaboshlagungaqadarbuob'ektninganiqtipi (tiplari) aniqlanmaydi. Ma'lumki, C++ sinfierarxiyasi, virtual funksiyalarvako'rsatkichlarni (pointer) asossinflargaishlatibpolimorfizmniamalgaoshiradi. Tayanchsinfgako'rsatkichbutayanchsinfa'zolarigayokiundanolinganharaq andayob'ekta'zolarigamurojaatqilishchunfoydalanishmumkin. Shuninguchunasossinfgako'rsatgichningistalganvaqtdaqandayturdagiob' ektgamurojaatqilishihardoimhamoldindanma'lumemas. Buvaqatdasturishlayotganida — dinamikturdagiidentifikatsiyavositlaridanbiri yordamidaaniqlanadi.

Dasturbajarilishidavomidaob'ektturiniolishchuntypeidoperatoridanfo ydalaniladi.

Dasturbajarilayotganvaqtdaob'ekttipiniolishchun<typeid>kutubxo nasinikiritishvattypeidoperatoridanfoydalanishmumkin. typeid

operatoridan foydalanishning eng keng tarqalgan yozilish sintaktisi quyidagi cha:

`typeid(object)`

Bunda, `object` elementini olish uchun ob'ekt anglatadi. Shuningdek, nafaqat ichki tipini, balki dasturchi tomonidan yaratilgan sinf tipini ham aniqlash mumkin. Buning uchun `typeid` operator ob'ekt tipini tasvirlaydigan `type_info` ob'ektiga murojaat qiladi.

`type_info` sinfda `public` modifikatori bilan quyidagi statik operatorlar aniqlangan bo'ldi.

```
bool operator == (const type_info &ob);  
bool operator != (const type_info &ob);
```

```
bool before(const type_info &ob);
```

```
const char *name();
```

Bunda qayta aniqlangan (overloaded) operatorlari `[==]` va `[!=]` tiplarini taqqoslash uchun ishlatiladi.

Chaqiruvchi ob'ekt parametrlar sifatida ishlatiladigan ob'ekt (element `ob`) oldidagi ketma-ketlik tartibidagi bo'lsa `before()` funksiyasi `true` qaytaradi. Bu funksiya asosan ichki foydalanish uchun mo'ljallangan.

Uning natijasi qiyosning sonli yoki sinfiy erarxik bilan hech qanday aloqasi yo'q. `name()` funksiyasi tip nomiga ko'rsatkich qaytadi.

`typeid` operatoridan foydalanishga birmisol keltiramiz.

1.1-dastur. Tiplarni taqqoslash¹

```
#include "stdafx.h"  
#include <iostream>  
#include <typeinfo>  
using namespace std;  
class myclass {  
    // . . .  
};  
int main(){  
    int i, j;  
    float f;  
    myclass ob;
```

¹Дастур Visual C++ muhitida tuzilgan.

```

cout << "i o'zgaruvchi tipi: " <<
typeid(i).name()<<endl;
cout << "f o'zgaruvchi tipi: " <<
typeid(f).name()<<endl;
cout << "ob o'zgaruvchi tipi: " <<
typeid(ob).name()<<endl<<endl;
if(typeid(i) == typeid(j))
    cout << " i va j tiplar bir xil.\n";
if(typeid(i) != typeid(f))
    cout << " i va f tiplar bir xil emas.\n";
system("PAUSE");
return 0;
}

```

1.1 – dasturnatijasi. Output

```

i o'zgaruvchi tipi: int
f o'zgaruvchi tipi: float
ob o'zgaruvchi tipi: class myclass
i va j tiplar bir xil.
i va f tiplar bir xil emas

```

1.1-dasturtahlili. Dasturda int tipigaoid i va j, floattipidaf, myclasstipidaobo'zgaruvchilarianiqlangan.

typeid()metodibilanularningtiplarianiqlangan.

Taqqsolashoperatorlaribilano'zraguvchilarningtiplaritekshirilganvanatijalarekrangachiqlangan.

Buesatiplarbo'yichadastlabkiishlovberishuchunlozim.

1.1-jadval. Type_info sinfa'zolarivavazifalari

№	Funksiyanomi	vazifasi
1	(constructor)	Sinfda standart vanusxakonstruktor mavjud emas.
2	(destructor) [virtual]	xavfsiztayanchsinfuchun orqaliob'ektnio'chirishmumkin (virtual funksiyasi)
3	operator=	O'zlashtirishamalidanfoydalanibko'chiribbo'lmaydi
4	operator== operator!=	Ob'ektlarbirxiltipgaliginitaqqoslashydi (public funksiya)

5	<u>before</u>	Joriytipdagiboshqa type_index ob'ekttomonidannusxasitipiniboryokiyo'qliginitekshira di.Ob'ektlar, aniqrog'itiplari, amalgaoshirilishibelgilangantartibdatashkiletildi. (public funksiya)
6	<u>hash_code</u>	Joriytipgamoskeladigantiplarniqaytaradi (public funksiya)
7	<u>name</u>	Joriytipninganiqlangannominiqaytaradi. (public funksiya)

Typeid
operatoripolimorftayanchsinfuchunko'rsatkichsifatidaqo'llaniladiganbo'lsa (eslabtibo'tamiz, polimorfsinfkamidabitta virtual funksiyanio'zichigaolgansinfgaaytiladi), uavtomatikravishdarealob'ekttipiniqaytaradivauasossidanolingansinf tayanchsinfob'ektyokiob'ektemasliginihamko'rsatadi.

Shuninguchun typeid
operatoridasossinfgako'rsatgichbilanmurojaatqilinganob'ekttipinidina mikravishdaaniqlashuchunfoydalanishmumkin.
Buxususiyatquyidagidasturdako'rsatilgan.

1.2-dastur. Typeid operatorinipolimorfsinflarierarxiyasigaqo'llanilishi.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
virtual void f() {}; // Asos sinf uchun polimorf
// . . .
};
class myClass_one: public Base {
// . . .
};
class myClass_two: public Base {
// ...
};
int main(){
Base *p, baseob;
```

```

myClass_one ob1;
myClass_two ob2;
p = &baseob;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
p = &ob1;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
p = &ob2;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
system("pause");
return 0;}

```

1.2 – dasturnatijasi. Output

```

O'zgaruvchi p tipdagi obyektga ishora: class Base
O'zgaruvchi p tipdagi obyektga ishora: class
myClass_one
O'zgaruvchi p tipdagi obyektga ishora: class
myClass_two

```

1.2-dasturtahlili. Dasturdadinamikko‘rsatkichliBasetipli p o‘zgaruvchisiyaratilganva 3 ta, base tipiga baseob, myClass_one tipiga ob1, myClass_two tipiga ob2 moso‘zgaruvchilaryaratilgan. Dasturdavomidadinamiko‘zgaruvchigaoddatiyo‘zgaruvchilarqiymatqilib berilgan.

Natijashuma’lumbo‘ldikidininamiko‘zgaruvchiningtipidanamikko‘rinishd ao‘zgardi.

Typeid operatoripolimorfurdagitayanchsinfgako‘rsatgichsifatidaqo‘llanilsa, haqiqiymanzillanganob’ektningtipinidasturbajarilayotganvaqtdaaniqlaydi.

Typeid operatoripolimorfb o‘lmaganiyerarxiksinflaruchunko‘rsatkichqo‘llaniladi ganbarchahollarda, tayanchtipiuchunko‘rsatikichbo‘ladi. Boshqaso‘zlarbilanaytganda,

buko'rsatkichaslidatipnianiqlashmumkinemas. Tajribasifatida 1.2-dasturdagiasossinfdagi f() virtual funksiyaniizohgaolingvanatijanitekshiribko'ring.

Dasturdabuo'zgarishniamalgaoshirgandanso'ngharbirturlitipliob'ektning tipi base debaniqlanishiniko'rasiz, chunkibu p ko'rsatgichgaegabo'lgantipdir.

Typeid operatoriodatdakengqirraliko'rsatkich (dereferenced pointer ya'ni, [*] operatorigaegabo'lganko'rsatkich) uchunqo'llanilganligisababli, bubuko'rsatkich null bo'libchiqqandavaziyatniboshqarishuchunmaxsusistisnoyaratiladi.

Buholda, typeid operatori bad_typeid istisnogao'tadi.

Polimorfsinflarierarxiasidagiob'ektlargahavolalarko'rsatgichlarka biishlaydi. Typeid operatoripolimorfsinfningmosyozuvlaruchunqo'llaniladiganbo'lsa, uaslidaob'ekttipiniqaytaradivauasosiytipio'rnigaolingantipdagiob'ektbo'lishimumkin.

Buusulo'zgaruvchilarnifunksiyalargamurojaatqilibo'tishdaengko'pqo'llaniladi. Masalan, quyidagi (1.3-dastur) dasturda WhatType() funksiyasiasosiytipdagiob'ektlargamosyozuvlarparametrinie'lonqiladi.

Demak, WhatType() funksiyasibazaviyturdagiob'ektlargamurojaatlarniyokiasossinfdanolinga nharqandaysinfob'ektlarigamurojaatlarnikeltiribchiqaradi.

Bundayparametrlargaqo'llaniladigan typeid operatorifunksiyagao'tganob'ektninghaqiqiyturiniqaytaradi.

1.3-dastur. Typeid operatorinihavolaliparametrigaqo'llash.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
virtual void f() {}; // Asos sinf uchun polimorf
// . . .
};

class myClass_one: public Base {
// . . .
};
class myClass_two: public Base {
// ...
```

```
};
void WhatType(Base &ob){
cout << " ob parametri ob'ekt tipiga havolasi: ";
cout << typeid(ob).name() << endl;
}
int main(){
Base baseob;
myClass_one ob1;
myClass_two ob2;
WhatType(baseob);
WhatType(ob1);
WhatType(ob2);
    system("pause");
return 0;
}
```

1.3 – dasturnatijasi. Output

```
ob parametri ob'ekt tipiga havolasi: class Base
  ob parametri ob'ekt tipiga havolasi: class
myClass_one
  ob parametri ob'ekt tipiga havolasi: class
myClass_two
```

1.3-dasturtahlili. Dasturda `WhatType` funksiyasi yaratilib, unga asossinftipidpgihavolaliparametrbelgilandi. Asosiyfunksiyada 3 ta, base tipiga baseob, myClass_one tipiga ob1, myClass_two tipiga ob2 moso‘zgaruvchilaryaratilgan.

Dasturdavomida `WhatType` funksiyasininghavolaliparametrgaoddatiyo‘zgaruvchilarqiymatqilibberilgan. Natijashuma’lumbo‘ldiki, dinamiko‘zgaruvchiningtipidanamikko‘rinishdao‘zgardi.

Dasturlashjarayonlarida o‘zgaruvchilarnitiplarini realtipbilantaqqosl ashgato‘g‘rikeladi. Buninguchun typeid operatori argumentsifatidatipinominiolishyordamida amalga oshiriladi. Uningyozilishiquyidagicha:

```
typeid(tipnomi)
```

Masalan, quyidagilistingamalqiladi.

```
Cout<<typeid(int).name();
```

Typeid

operatoriniushbuvariantdaishlatilishiningmaqsaditipnitaqqoslashuchunfoydalanishdir, (belgilanganma'lumotlartipinitalaydi) ammo, typeid ob'ektiniolishuchunemas.

Dinamiktipnianiqlashni (RTTI) qo'llash.
Dasturadinamiktipnianiqlash (identifikatsiya) vositasi (RTTI)
imkoniyatifoqdalibo'lishimumkinlinginikotibotimiz. Masalan,
geometrikshakllaruchuniyerarxihsinibberilganbo'lsin, undadoira,
uchburchakvotibburchakmaydonihisoblashusullarimavjud.
Ushbudasturdadoira,
uchburchakyokitibburchakniyaratishuchunmo'ljallanganfunksiyasi
factory() bo'lsin. Ushbufunksiyayaratilganob'ektgabir nusxaniqaytadi.
Ob'ektlarnihosilqiluvchifunksiyabotibob'ektfabrikasidebataladi.
Yaratilayotganob'ektningotibgaxostipi C++
tasodifiysongeneratorining Rand() funksiyasigakirishorqalianiqlanadi.
Demak, qandayturdagiob'ekthosilbo'lishinioldindanbilolmaymiz. 1.4-
dasturdaotibob'ektyaratadivaharbirturdagiyaratilganob'ektlarsoninihiso
blaydi. factory()
funksiyasinichaqirishdaharqandayshaklturinihosilqilishimumkinligisabab
li, dasturaslidayaratilganob'ekttipnianiqlashuchun typeid
operatoridanfoydalanadi.

1.4-dastur. Dinamiktibdagiidentifikatsiyavositasidanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
#include <cstdlib>
using namespace std;
class figure {
protected:
    double x, y;
public:
    figure(double i, double j) {
        x = i;
        y = j;
    }
    virtual double area() = 0;
};
class triangle : public figure {
```

```

public:
    triangle(double i, double j) : figure(i, j) {}
    double area() {
        return x * 0.5 * y;
    }
};
class rectangle : public figure {
public:
    rectangle(double i, double j) : figure (i, j)
    {}
    double area() { return x * y;}
};
class circle : public figure {
public:
    circle(double i, double j=0) : figure(i, j) {}
    double area() {return 3.14 * x * x;}
};
// figure sinf ob'ektlarini yaratish.
figure *factory(){
    switch(rand() % 3 ) {
        case 0: return new circle (rand()%15);
        case 1: return new triangle
(rand()%15,rand()%15);
        case 2: return new rectangle (rand()%15,
rand()%15);
    }
return 0;
};
int main(){
    figure *p; // asos sinfga ko'rsatkich
    int i;
    int t=0, r=0, c=0;
        // 5 ta ob'ektlarni hosil qilsh va sanash
    for(i=0; i<15; i++) {
        p = factory(); // ob'ektni hosil qilish
        cout << "Ob'ektning tipi: " <<
typeid(*p).name();
        cout << ". ";
    }
}

```

```

        // ob'ektni tekshirish orqali sanasj
        if(typeid(*p) == typeid(triangle)) t++;
        if(typeid(*p) == typeid(rectangle)) r++;
        if(typeid(*p) == typeid(circle)) c++;
        // Yuzasini hisoblash
        cout << " S= " << p->area() << endl;
    }
cout << endl;
cout << "Quyidagi ob'ektlar hosil qilindi:\n";
cout << " Uchburchaklar: " << t << endl;
cout << " To'rtburchaklar: " << r << endl;
cout << " Doiralalar: " << c << endl;
system("pause");
return 0;
}

```

1.4 – dasturnatijasi. Output

```

Ob'ektning tipi: class rectangle.  S= 8
Ob'ektning tipi: class triangle.  S= 28
Ob'ektning tipi: class circle.  S= 28.26
Ob'ektning tipi: class triangle.  S= 35
Ob'ektning tipi: class rectangle.  S= 12
Ob'ektning tipi: class triangle.  S= 55
Ob'ektning tipi: class rectangle.  S= 72
Ob'ektning tipi: class circle.  S= 254.34
Ob'ektning tipi: class rectangle.  S= 21
Ob'ektning tipi: class triangle.  S= 33
Ob'ektning tipi: class rectangle.  S= 10
Ob'ektning tipi: class circle.  S= 379.94
Ob'ektning tipi: class circle.  S= 254.34
Ob'ektning tipi: class triangle.  S= 4
Ob'ektning tipi: class triangle.  S= 36

```

```

Quyidagi ob'ektlar hosil qilindi:
Uchburchaklar: 6
To'rtburchaklar: 5
Doiralalar: 4

```

Typeid operatorini shablonsinflariga qo'llash. Typeid operatorini shablonsinflariga ham qo'llash mumkin albatta. Shablonsinfi asosida hosil bo'lgan ob'ektni amalga oshirishda uning umumlashgan ma'lumotlari uchun qanday ma'lumotlardan foydalanilganligi asosida qisman aniqlanadi. Shuning uchun, turlima ma'lumotlari yordamida yaratilgan bir xil shablonsinfi gategishli bo'lgan ob'ektlar tip xil bo'ladi.

1.5-dastur. shablonsinflar bilan typeid operatoridan foydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
template <class T>
class myclass {
    T a;
public:
    myclass(T i) { a = i; }
    // . . .
};
int main(){
    myclass<int> o1(28), o2(1);
    myclass<double> o3(19.80);
    cout << " o1 ob'ekt tipi: ";
    cout << typeid(o1).name() << endl;
    cout << " o2 ob'ekt tipi: ";
    cout << typeid(o2).name() << endl;
    cout << " o3 ob'ekt tipi: ";
    cout << typeid(o3).name() << endl;
    cout << endl;

    if(typeid(o1) == typeid(o2))
        cout << " o1 va o2 ob'ektlar bir xil
tip.\n";
    if(typeid(o1) == typeid(o3)) cout <<
"Xatolik\n";
    else cout << "o1 va o3 ob'ektlar bir xil tip
emas.\n";
    system("pause");
```



```
return 0;
}
```

1.5 – dasturnatijasi. Output

```
o1 ob'ekt tipi: class myclass<int>
o2 ob'ekt tipi: class myclass<int>
o3 ob'ekt tipi: class myclass<double>

o1 va o2 ob'ektlar bir xil tip.
o1 va o3 ob'ektlar bir xil tip emas.
```

1.5-dasturdanko‘rinibturibdiki, ikkiob‘ektlarbirxilshablonsinfob‘ektlaridabo‘lsa-da, ularningparametrlima‘lumotlarmosbo‘lmasa, ularturitengemas. Budasturda o1 ob‘ekti myclass <int >tipida, o3 ob‘ektiesamyclass <double>tipidabo‘ladi. Shundayqilib, buob‘ektlarturlixilliginiko‘rsatadi.

Tipeid operatorinishablonsinflarigaqo‘llashningyanabirmisolini, ya‘ni1.4-

dasturgeometrikshaklnianiqlashdasturiningo‘zgartirilganvariantiniko‘rib chiqamiz. Busafar figure sinfinishablonsinfigaaylantiriladi.

1.6-dastur. Figure ierarxiyasiningshablonvarianti.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
#include <cstdlib>
using namespace std;
template <class T>
class figure{
protected:
    T x, y;
public:
    figure(T i, T j) {
        x = i;
        y = j;
    }
    virtual T area() = 0;
};
template <class T>
```

```

class triangle : public figure<T>{
    public:
        triangle(T i, T j) : figure<T>(i, j) {}
        T area() {
            return x * 0.5 * y;
        }
};

template <class T>
class rectangle : public figure<T>{
    public:
        rectangle(T i, T j) : figure<T>(i, j) {}
        T area() {
            return x * y;
        }
};

template <class T>
class circle : public figure<T>{
    public:
        circle(T i, T j=0) : figure<T>(i, j) {}
        T area() {
            return 3.14 * x * x;
        }
};

figure<double> *generator(){
    switch(rand() % 3 ) {
        case 0: return new circle<double> (rand() %
28);
        case 1: return new triangle<double>(rand()
% 28, rand() % 28);
        case 2: return new rectangle<double>
(rand() % 28, rand() % 28);
    }
    return 0;
}

int main(){
    figure<double> *p;
    int i;
    int t=0, c=0, r=0;

```

```

        for(i=0; i<15; i++) {
            p = generator();
            cout << "Ob'ekt tipi: " <<
typeid(*p).name();
            cout << ". ";
            if(typeid(*p) == typeid(triangle<double>))
t++;
            if(typeid(*p) == typeid(rectangle<double>))
r++;
            if(typeid(*p) == typeid(circle<double>))
c++;
            cout << " S= " << p->area() << endl;
        }
        cout << endl;
        cout << "Ob'ektlarni sanash:\n";
        cout << " uchburshaklar: " << t << endl;
        cout << " o'rtburchaklar: " << r << endl;
        cout << " doiralar: " << c << endl;
        system("pause");
        return 0;
    }

```

1.6 – dasturnatijasi. Output

```

Ob'ekt tipi: class rectangle<double>. S= 90
Ob'ekt tipi: class triangle<double>. S= 136
Ob'ekt tipi: class circle<double>. S= 615.44
Ob'ekt tipi: class triangle<double>. S= 210
Ob'ekt tipi: class rectangle<double>. S= 351
Ob'ekt tipi: class triangle<double>. S= 202.5
Ob'ekt tipi: class rectangle<double>. S= 44
Ob'ekt tipi: class circle<double>. S= 803.84
Ob'ekt tipi: class rectangle<double>. S= 156
Ob'ekt tipi: class triangle<double>. S= 30
Ob'ekt tipi: class rectangle<double>. S= 225
Ob'ekt tipi: class circle<double>. S= 706.5
Ob'ekt tipi: class circle<double>. S= 1384.74
Ob'ekt tipi: class triangle<double>. S= 66.5
Ob'ekt tipi: class triangle<double>. S= 15

```

Ob'ektlarni sanash:
uchburshaklar: 6
o'rtburchaklar: 5
doiralar: 4

Dinamik tipni aniqlash har bir dasturda ishlatilmaydi. Ammo, dastur bajarilish vaqtida, polimorfturlar bilan ishlashda, undagi ob'ektlarning har qanday tipini aniqlashga imkonini beradi.

Tiplarni almashtirish operatorlari. C++
tilida beshta tipni almashtirish operatorlari mavjud.

Ulardan birinchi bu odatdagi (an'anaviy) uslubda ishlatiladigan `[]` (cast operatori) azaldan C++ ga qurilgan. U quyidagicha ishlatiladi:

```
float f = (float)5;  
bool b = (bool)5;
```

Tipni almashtirish to'rtta operatori (`dynamic_cast`, `const_cast`, `reinterpret_cast` va `static_cast`) 10 yil oldin C++ tilga qo'shildi. Bu operatorlarni tiplarni almashtirish bilan amallarni bajarish uchun zarur bo'lgan xilmi xilni yaratib beradi. Ularning har birini alohida ko'rib chiqaylik.

Dynamic_cast operatori. `Dynamic_cast` operatori dastur bajarilish davomida polimorfturda quyish amalini bajaradi. Yangi operatorlardan eng muhimi `dynamic_cast` tipidagi tipni almashtirish (`kasting`, `casting`) operatoridir. Dastur bajarilish davomida talif qilinayotgan amalning bajarilishini tekshiradi. Agar belgilangan amal chaqirilgan dasturda qo'shilmagan bo'lsa, hech qanday turdagi `kasting` amalga oshirilmaydi. `Dynamic_cast` operatorini qo'llashning umumiy formati quyidagicha:

```
dynamic_cast<type>(expr)
```

Bunda `type` elementining maqsadiga bo'lgan yangi tipni, `expr` elementining maqsadiga bo'lgan yangi tipni bildiradi.

Tipni to'rtta ko'rsatkich yoki mos yozuvlar bilan almashtirish kerak va `expr` ifodasi ko'rsatkich yoki mos yozuvlar uchun tashlanish kerak. Shunday qilib, `dynamic_cast` operatoridan birturda ko'rsatkichni boshqartiradigan ko'rsatkichga yoki birturda ko'rsatkichni boshqartiradigan ko'rsatkichga aylantirish uchun foydalanish mumkin.

Bu operator asosan polimorftiplar orasida dinamik tipni quyish amallari uchun ishlatiladi. Masalan, agar polimorftiplar B va D bo'lsin. `sinf D` va `sinf Volingach`, `dynamic_cast` operatoriyordamida,

hardoimko'rsatkichni Dgaaylantirishmumkin. Chunki *in pointer* tayanchsinf gako'rsatkich hardoim bazasidan olingan sinf ob'ektiniko'rsatis huchun foydalanish mumkin. Biroq, dynamic_cast operatorifaqatko'rsatkichaylantirish mumkin. D ob'ektko'rsatkichuchun murojaat qilinganda, albatta, sinf D ob'ektibo'ladi. Umuman olganda, dynamic_cast operatorifaqat polimorf tipdagikastinggaruxsat berilsa, muvaffaqiyatli bajariladi, ya'ni yangi tipdagiko'rsatik chushbu yangi tipdagi ob'ektgayoki undan olinga nob'ektgahavola qilish mumkin. Aksholda, agar belgilangan tipdagialmashtirish amalini bajarib bo'lmasa, bu amaldako'rsatgichlarishti rokesa dinamic_cast operatorining natijasi null qiymatga ay lanadi. Agarushbu amalni bajarilish muvaffaqitsiz bo'lsa, uzilishlarishti roketgan bo'lsa, bad_cast istisnosi avtomatik bajariladi.

Oddiy misol niko'ribchi qamiz. Base asossinfi polimorf sinf bo'lsin va myClass sinf esa Base sinfidan olingan deb faraz qilaylik.

```
Base *bp, b_ob;
myClass *dp, d_ob;
bp = &d_ob; // myClass sinf ob'ektga Base sinfchun ko'rsatkich.
dp = dynamic_cast< myClass *> (bp); // myClass
sinf ko'rsatgichga almashtirish
if(dp) cout << "Tipni almashtirish muvaffaqiyatli bo'ldi!";
```

Buyerdabp ko'rsatkichini (asossinf) dp ko'rsatkichiga (myClass sinf) almashtirish muvaffaqiyatli bo'ladi, chunki b paslida sinf ob'ektiga ishora qiladi.

myClass shuning uchunushbu misol ni bajarishda tipni almashtirish muvaffaqiyatli chi qdi. bp aslida sinf bazasini ob'ektga ishora qilgandi. Quyidagi misolda tipni almashtirishda myClass sinf ob'ekt bo'lsa, ko'rsatgich gatayanch sinfga ob'ektni tashlash uchun muvaffaqiyatsiz vanoto 'g'riolingan hisoblanadi.

```
Base *bp, b_ob;
myClass *dp, d_ob;
bp = &b_ob; // Base sinfgako'rsatkich sifatida Base sinf ob'ekt berilgan.
dp = dynamic_cast< myClass *> (bp); // xatolik
if(!dp) cout << "Tipni almashtirish muvaffaqiyatsiz ";
```

Ushbudasturfragmentidatipnialmashtirishamalimuvaaffaqiyatsizbo‘l
divatipnialmashtirishuchunmuvaaffaqiyatsizbo‘lganliginiekrangachiqarad
i.

dynamic_cast operatoridanfoydalanishdaquyidagi 1.7-
dasturdatipnialmashtirishholatlarianiqlaymiz.

1.7-dastur. dynamic_cast operatoridanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
    public:
        virtual void f() { cout << " Base sinfi
orqali.\n"; }
    // . . .
};
class myClass : public Base {
    public:
        void f() { cout << "myClass sinfi
orqali.\n"; }
};
int main(){
    Base *bp, b_ob;
    myClass *dp, d_ob;
    dp = dynamic_cast<myClass *> (&d_ob);

    if(dp) {
        cout << " tipni almashtirish " <<" (myClass*
=> myClass *) bajarildi. ";
        dp->f();
    } else cout <<"Xatolik\n";
    cout << endl;
    bp = dynamic_cast<Base *> (&d_ob);

    if(bp) {
        cout << " tipni almashtirish " <<"(myClass*
=> Base *) bajarildi. ";
        bp->f();
    }
}
```

```

    } else cout << "Xatolik\n";
    cout << endl;

    bp = dynamic_cast<Base *> (&b_ob);
    if(bp) {
        cout << " tipni almashtirish " <<"(Base* =>
Base *) bajarildi. ";
        bp->f();
    } else cout << "Xatolik\n";
    cout << endl;
    dp = dynamic_cast<myClass *> (&b_ob);
    if(dp) cout <<"Xatolik\n";
    else
        cout <<" tipni almashtirish "<<"(Base*=>
myClass*) bajarilmadi.\n";
    cout << endl;

    bp = &d_ob; // bp myClass ob'eyktiga
ko'rsatkich
    dp = dynamic_cast<myClass *> (bp);
    if(dp) {
        cout << " bp ni myClass tipiga almashtirish
bajarildi. ";
        dp->f();
    } else cout << "Xatolik\n";
    cout << endl;

    bp = &b_ob; // br Base ob'eyktiga ko'rsatkich
    dp = dynamic_cast<myClass *> (bp);
    if(dp) cout << "Xatolik";
    else cout <<" bp myClass tipiga almashtirish
bajarilmadi.\n";
    cout << endl;
    dp = &d_ob; // dp myClass ob'eyktiga
ko'rsatkich
    bp = dynamic_cast<Base *> (dp);

    if(bp) {

```

```

        cout <<" dp ni myClass tipiga almashtirish
bajarildi. ";
        bp->f();
    }
    else cout <<"Xatolik\n";
    system("pause");
return 0;
}

```

1.7 – dasturnatijasi. Output

```

tipni almashtirish (myClass* => myClass *)
bajarildi. myClass sinfi orqali.
tipni almashtirish (myClass* => Base *) bajarildi.
myClass sinfi orqali.
tipni almashtirish (Base* => Base *) bajarildi.
Base sinfi orqali.
tipni almashtirish (Base*=> myClass*) bajarilmadi.
bp ni myClass tipiga almashtirish bajarildi.
myClass sinfi orqali.
bp myClass tipiga almashtirish bajarilmadi.
dp ni myClass tipiga almashtirish bajarildi.
myClass sinfi orqali.

```

Dynamic_cast operatoridan typeid operatori o'ziga ishlatilishi mumkin. Masalan, Base sinfi polimorf va myClass sinfi uchun asos sinfi, deb faraz qilaylik. Quyidagi dastur fragmentini bajarishda, ob'ekt aslida myClass sinfi ob'ekt bo'lsada dp ko'rsatkich bp ko'rsatkich tomonidan ob'ekt manzili beriladi.

```

Base *bp;
Derived *dp;
// . . .
if(typeid(*bp) == typeid(Derived))
dp = (Derived *) bp;

```

Bu holda, odatiy tipni almashtirish ishlatildi. Bujudaxavfsiz, chunki if operatori typeid operatori yordamida typeid amalining haqiqiyligini bajarilishidan oldin tekshiradi. Typeid operatori va if operatori bilan almashtirish orqali yanada samarali roq qilish mumkin.

dynamic_cast:


```
dp = dynamic_cast<myClass *> (bp);
```

Dynamic_cast operatorifaqatob'ektningbelgilangantipiyokibelgilangantipialmashtirilgan tipibilanbirob'ektbo'lsamuvaffaqiyatlinatijaberadi, bualmashtirishtugagandanso'ng, dp ko'rsatkicholingnob'ektganullqiymatiniyokiko'rsatkichnihamo'zichiga oladi. Belgilangantipnitashlashamalito'g'ribajarilganbo'lsa dynamic_cast operatorifaqatmuvaffaqiyatlibajariladi, uningalmashtirishmantig'inimuayyanvaziyatlardasoddalashtirilganmumkin. Quyidagi 1.8-dastur typeid operatorini dynamic_cast operatoribilanalmashtirishmumkinliginiko'rsatilgan. Buyerdabirxilamallarto'plamiikki marta bajariladi, masalan, avval typeid operatori, so'ngra dynamic_cast operatoriyordamida.

1.8-dastur. Typeid operatorio'rniga dynamic_cast operatoridanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
public:
    virtual void f() {}
};

class MyClass : public Base {
public:
    void derivedOnly() {
        cout << "Bu MyClass ob'ekti.\n";
    }
};

int main(){
    Base *bp, b_ob;
    MyClass *dp, d_ob;
    // typeid operatoridan foydalanish
    bp = &b_ob;
    if(typeid(*bp) == typeid(MyClass)) {
        dp = (MyClass *) bp;
        dp->derivedOnly();
    }
}
```

```

    } else
        cout << " Base tipidagi ob'ekt MyClass
almashtirilmadi. \n";
        bp = &d_ob;
        if(typeid(*bp) == typeid(MyClass)) {
            dp = (MyClass *) bp;
            dp->derivedOnly();
        } else
            cout << "Xatolik, tip almashtirish yozilmagan!
\n";
        // dynamic_castoperatoridan foydalanish
        bp = &b_ob;
        dp = dynamic_cast<MyClass *> (bp);
        if(dp) dp->derivedOnly();
            else cout << " Base tipidagi ob'ekt MyClass
almashtirilmadi. \n";
            bp = &d_ob;
            dp = dynamic_cast<MyClass *> (bp);
            if(dp) dp->derivedOnly();
                else cout << "Xatolik, tip almashtirish
yozilmagan! \n";
                system("pause");
            return 0;
        }

```

1.8 – dasturnatijasi. Output

```

Base tipidagi ob'ekt MyClass almashtirilmadi.
Bu MyClass ob'ekti.
Base tipidagi ob'ekt MyClass almashtirilmadi.
Bu MyClass ob'ekti.

```

1.8-dasturdanko‘rinadiki, asossinfningko‘rasatkichini MyClass ko‘rsatkichigaalmashtirishda dynamic_cast operatoriniqo‘llashdasturmantig‘iniqulayroqqiladi. Yuqoridagidasturnatijasigaqarang. Shuningdek, dynamic_cast operatorinishablonsinflargahamqo‘llashmumkin.

const_cast operatori. Const_cast operatori const va/yoki volatile o‘zgartirgichlarniqaytaaniqlashuchunxizmatqiladi.

Const_cast operatori const va / yoki volatile o'zgartirgichlarni ochiq qayta aniqlash/joriy qilish uchun ishlatiladi. const yoki volatile ning atributlariga istisnalar sifatida, yangi tip joriy tip bilan mos bo'lishi kerak. Ko'pincha const_cast operatori const atributini olib tashlash uchun ishlatiladi. Uning umumiy formati quyidagicha:

<code>const_cast<type> (expr)</code>

Buyerda type elementi yangi tipga almatirish tipini o'rnatadi, expr elementies yangi tipga almatirilishi kerak bo'lgan o'zgaruvchi (ifoda) ni bildiradi.

1.9-dastur. Const_cast operatoridan foydalanish.

<pre>#include "stdafx.h" #include <iostream> #include <typeinfo> using namespace std; void f (const int *p){ int *v; v = const_cast<int *> (p); // const- o'zgaruvchini qayta almashtirish. *v = 1980; // yangi qiymat berish, modifikatsiyalash } int main(){ int x = 2004; cout <<" f() funksiya chaqirilmasdan oldin x ni qiymati: " << x<< endl; f (&x); cout <<" f() funksiya chaqirilgandan so'ng x ni qiymati: " << x<< endl; system("pause"); return 0; }</pre>

1.9 – dasturnatijasi. Output

```
f() funksiya chaqirilmasdan oldin x ni qiymati:
2004
f() funksiya chaqirilgandan so'ng x ni qiymati:
1980
```

Ko'rib turganingizdek, xo'zgaruvchi qabul qilgan parametr `const` ko'rsatgich sifatida o'rnatilgan `f()` funksiyasi tomonidan o'zgartirilgan.

`Const` atributini o'chirish uchun `const_cast` operatoridan foydalanish xavfli darajadagi vosita ekanligini alohida ta'kidlash lozim. Shuning uchun, juda diqqat bilan foydalanishni tavsiya qilamiz.

Faqat `const_cast` operatori `const` atributini o'chirish imkonligini bilib olingiz.

Boshqaso'zlar bilan aytganda, na `dynamic_cast`, `static_cast`, yoki `reinterpret_cast` ob'ekt `const` atributini o'zgartirish uchun foydalanish mumkin emas.

`static_cast` operatori. `Static_cast` operatori polimorf bo'lmagan tipdagi ko'rsatkichlarni almashtirish uchun foydalaniladi. Har qanday tipni almashtirish uchun foydalanish mumkin. Biroq, dastur bajarilgan vaqtda hech qanday tekshirish amalga oshirilmaydi.

`Static_cast` operatori quyidagi umumiy yozuv formatiga ega:

```
static_cast<type> (expr)
```

Buyerda `type` elementi yangi tipdagi almashtirish tipini belgilaydi va `expr`

elementi buyan tipga tip almashtirilish kerak bo'lgan ifodani bildiradi.

`Static_cast` operatori tipni almashtirishning o'rginal operatoridir. U faqat polimorf bo'lmagan tiplarni amalga oshiradi. Masalan, quyidagi 1.10-dastur ni ishlaganda `float` tipidagi xo'zgaruvchi `int` tipiga aylantiradi.

1.10-dastur. `Static_cast` operatoridan foydalanish.

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main(){
    int i;
    float f;
    f = 199.22F;
    i = static_cast<int> (f);
    cout << i;
    return 0;
```

```
system("pause");  
}
```

reinterpret_cast operatori. Reinterpret_cast operatorifundamentaltipnialmashtirishniamalgaoshiradi.

Reinterpret_cast operatoritipnitubdanfarqqiladigantipgaaylantiradi. Masalan, birko'rsatkichni int tipigahamdabirko'rsatkichnibir int qiymatiuchunko'rsatkichgaaylantirishhuchunfoydalanishingizmumkin.

Bundantashqari, tabiatanmoskelmaydiganko'rsatkichtiplarnitashlashuchunfoydalanishmumkin. Ushbuoperatorquyidagiumumiyformatgaega:

```
reinterpret_cast<type> (expr)
```

Buyerda type elementiyangitipdagitipnialmashtirishnibelgilaydiva expr elementibuyangitipgaalmatiriladiganifodanibildiradi.

Reinterpret_cast operatoridanfoydalanishquyidagi 1.11-dasturniko'ribchiqamiz.

1.11-dastur. Reinterpret_cast operatoridanfoydalanish.

```
#include "stdafx.h"  
#include <iostream>  
  
using namespace std;  
  
int main(){  
    int i;  
    char *p = "Salom Buxoro";  
    i = reinterpret_cast<int> (p); // ko'rsatkichni  
    int tipiga almashtirish.  
    cout << i<<endl;  
    system("pause");  
    return 0;  
}
```

Buyerda reinterpret_cast operatori p ko'rsatkichnibutunson int qiymatigaaylantiradi. Buo'zgartirishasosiytipnio'zgarishifodalaydi. Butipalmashtirishnioddikastingbilanhamamalgaoshirishmumkin.

Yangito'rtta cast operatorlaribilanan'anaviytipnialmashtirishamali (kasting) qandayfarqmavjud. Ba'zidasturchilarto'rt cast operatorlariyuqoridatasvirlanganbutunlayan'anaviytipnialmashtirishamalinigo'rnigadebo'ylashlarimumkin.

Ularning o'rniga har doim datiy tipni almashtirishning yangi vositalaridan foydalanish kerakmi? — degan savol qiziqtirish mumkin.

Aslida barcha dasturchilar uchun umumiy qoida yo'q.

Yangi operatorlar birmalum o'ttipini boshqasiga almashtirishni ancha xavfli. Mallar xavfsizligini yaxshilash uchun yaratilganligi sababli, ko'pgina C++ dasturchilar faqat shu maqsadda ishlatilish kerakligiga shonch hosil qilgan. Bungae'tiroz qilib bo'lmaydi.

Boshqa dasturchilar anaviy tipni almashtirish amaliko'pyillarda vomida ularga "sadoqat" bilan xizmat qilib bergan, undan shunday vozekechish oson emas. Masalan, oddiy vanisbatan xavfsiz tipdagikastinga mallarinibajarish uchun read() va write() funksiyalarini chaqirganda talab qilinadi.

Ammo shunday dasturlashdajarayonlari borki, ularda cast operatorlaridan foydalanish maqsadga muvofiqdir, masalan, polimorftiplar uchun cast amallarini bajarishda albatta dynamic_cast operatoridan foydalanish gato'g'rikeladi.

Yangi nomlar fazosi yaratish.

Nomlar fazosi haqida qisqachama'lumotga ega bo'lsangiz kerak.

Endi butun hunchanibatafsil roqko'ribchiqamiz va foydalanuvchining nomlar fazosi qanday yaratilishini ko'ribchiqamiz. Namespaces tushunchasi C++ dasturlashtirilgan nisbatan yangi paydobo'lgan tushunchalardan biridir.

Ular nomlarning o'zarobirxilligini oldini olish (kolliziyasizlik) uchun identifikatorlarning nomlarini lokalizatsiya qilish uchun mo'ljallangan. C++ dasturlash muhitida,

o'zgaruvchani nomlari aqlbovar qilmaydigan bir qator operatorlari, funksiyalari, metodlarivasi n mavjud.

Nomlar fazosi tushunchasi kiritilishidan oldin, bunomlarning barchasi bir xil global nomlar bo'lib, ko'plab nizolarni (xatoliklarni, o'zarobirxilliklarni) keltiribchiqargan. Agarda dasturda dasturchi o'zining toupper()

funksiyasini yaratgan bo'lsa, masalan, agar, u (uning parametrlari ro'y xatiga qarab) standart kutubxona toupper(), funksiyasini almashtirish mumkin. Ammo,

ikkala funksiyalarning nomlari bir xil global namespace da saqlangan bo'ladi.

Bir xil dasturlarni turli ishlabchiqaruvchilarning funksiyalarivasi nlar, kutubxonalar foydalangan dan omalar bir xil bo'lgan funksiyalardan turlibajarilishi da ziddiyatlargaduchkeladi. Bunday holda,

birkutubxonadaaniqlangannomlarboshqakutubxonadaaniqlangannomlarbilanto‘qnashishimumkin.

Barchamuammolarnomlarfazositushunchasiva namespace kalitso‘ztomonidanhalqilindi.

Bukalitso‘zyanginomlarfazosielonqilishorqalibarchafunksiya, sinfvakutubxonalarnimahalliyashtirishimkoniniberadi.

Bunomlarfazositurlikontekstlardabirxilnomdagifunksiyavasinflarziddiyatgaolibkelmasdanishlatishimkoniniberadi. std standartkutubxona (std - Standart Library) judako‘pfoydalanilganbo‘lishimumkin. Tilningdastlabkivariantlarida C++ kutubxonasi global nomlarfazosidabelgilangan.

Sezilarlinomlarziddiyatlarietimolinikamaytirishuchun std namespace nomlarfazosiyaratilgan. Bundantashqari,

ziddiyatlarnioldiniolishuchunnomlarfazrsiko‘laminimahalliyashtirishuchundasturdao‘znomfazongizniyaratishingizмумkin.

Bushaxsiysinfyokifunksiyakutubxonalaryaratishda, ayniqsa, muhimahamiyatgaega.

Namespace kalitso‘zinomlarfazosidasohalarniqo‘shibglobal namespace danajratishimkoniniberadi. Aslida,

birnomlarfazosidoirasibelgilaydi. Namespace kalitso‘zidanfoydalanishningasosiyshakli quyidako‘rsatilgan:

```
namespace Space{  
    //....  
}
```

Namespace kontentidaaniqlanganixtiyoriyharbirelement, bu namespace doirasibajariladi.

1.12-dastur. myNameSpace nomli namespace e‘lonnamuna.

```
namespace MyNameSpace {  
    int i,k;  
    void myfunc (int j) { cout <<j; }  
    class myclass {  
        public:  
            void set(int x) { i = x; }  
            int get() { return i; }  
    };  
};
```

Mynamespace nomli namespace da i va k o‘zgaruvchilarningnomlari, myfunc() funksiyasiva myclass sinfiyaratilgan.

Bu namespace ichida o'zgaruvchilarga, funksiyalarni ifodalash uchun 'g' dan to'g'ri murojaat qilishni ta'minlaydi. Biroq, namespace kalit so'z ma'lum bir dastur sohasini belgilaydi, chunki, namespace tashqaridan namespace ichida 'lonob' e'knlarni foydalanish imkoniyatini faqat ziddiyatlarni oldini olgan holda foydalanishga ruxsat beradi. Masalan, My namespace nomli namespace tarkibi ga kirmagan dasturning birligida i o'zgaruvchiga 28 qiymat berish uchun quyidagicha ifoda yozish lozim:

```
MyNameSpace::i = 28;
```

My namespace nomli namespace sohasiga kiradigan MyClass tipidagi ob'ektni odatdagidek 'lonqilish imkoniyatini ta'minlaydi, ammo kirmagan MyClass tipidagi ob'ektni quyidagicha 'lonqilish lozim:

```
MyNameSpace::myclass ob;
```

Shunday qilib, busohatlash qaridan nomlar fazosining nomini bilan kengaytirish operatori (::) bilan murojaat qilish mumkin.

Agar dasturda namespace a'zolariga tez murojaat bo'ladigan bo'lsa, hardimuning nomini kengaytirish operatori yozish dasturining ko'rinishiga va dastur chining ko'pyozishiga sabab bo'lishi mumkin. Ushbu muammoni hal qilish uchun foydalanuvchining nomlar fazosidan foydalanish fragmenti ishlab chiqilgan. Ushbu fragmenti katta asosiy shaklga ega:

```
using namespace MyNameSpace;
```

```
using MyNameSpace::myclass;
```

Fragmentning birinchi qatorida keltirilgan shaklda a'zolarini kirmoqchib o'lgan nomlar fazosining nomini bildiradi. Foydalanish fragmentida boshqacha yordamida yozilgan nomlar fazosidagi barcha a'zolariga joriynom mavjud deb olinadivato'g'ridan to'g'ri murojaat qilish mumkin, faqat nomini kengaytirish operatori harsafari yozish kerak bo'lmaydi.

Fragmentning ikkinchi qatorida using fragmentining ikkinchi shaklidan foydalanilganda faqat fragmentdako'rsatilgan nom dosha'zosini ko'rinadi. Masalan, yuqoridagi ifodadan My namespace nomli namespace bo'lsa, unda quyidagi fragmentlar va operatorlaridan foydalanish to'g'ri:

```
using MyNameSpace::k;
```



```
int main(){
    k = 10;
    return 0;
}
```

Fragmentasosida faqatko'zgaruvchiko'rinadi.
ko'rsatmato'g'ri.

Demak,

```
using namespace MyNameSpace;
```

```
int main(){
    k = 10;
    i = 10;
    myclass ob;
    myfunc(25);
    return 0;
}
```

Bu fragmentda MyNameSpace
fazoning barcha a'zolariko'rinadivadaDemak, ko'rsatmato'g'ri.

Shuningdek,
bir xil nom bilan bir nechta fazoni e'lon qilish imkoniyati mavjud.
Bu bir nechta fayllar ichiga fazolarni ajratish imkonini beradi,
yoki hatto bitta fayl ichida fazolarni ajratish imkonini.
Bu nida sturaching dasturga ishlatayotgan funksiyaviy sinflarini ajratish uchun
qo'llash mumkin. Quyida gimisolni ko'rib chiqamiz:

```
namespace NS { int a;}
namespace NS { int b;}

int main(){
    NS::a = 10;
    NS::b = 10;
    return 0;
}
```

Buyerda NS namespace ikki qismga bo'lingan. Shunga qaramay,
har bir qismi a'zolariga ham bir xil nom bilan murojaat qilingan, ya'ni NS
fazo.

Bir dastur fayli da bir nomli fazolarni e'lon qilish maqsadga muvofiq.
Ammo, dasturda 2
va undan ortiq dastur fayllari ishlatilsa va fazolarni birlashtirish kerak bo'lsa,
shunda fazoni ikki ga ajratish usulidan foydalanish mumkin.
Boshqaso'zlar bilan aytganda, bir namespace faqat boshqa namespace

bo‘lishimumkinemas.

Bubirnomlifazoe‘lonqilinishimumkinemasdeganma‘nonianglatadi, masalan, birfunksiyaichida.

Nomsizfazo - fazoningmaxsusturidir.

Muayyanfaylichidanoyobidentifikatorlar, a‘zolari ni yaratishimkoniniberadi.

Nomsizfazoningasosiyshakli quyidagicha bo‘ladi:

```
namespace {  
    const float f = 3.1415;  
    int myIf(int a, int b){return a>b?1:0;}  
}
```

Fazolarninomsiznomlashlarfaqatbittafayldoirasidama‘lumbo‘lgannoyobidentifikatorlarvafunksiya, sinflarnio‘rnatishimkoniniberadi. Shutarzda, noma‘lumnomlifazonio‘zichigaolganfaylichida, ushbufazoninga‘zolarigato‘g‘ridan-to‘g‘ri, hechqandaytushuntirishgamuhtojbo‘lmasdanfoydalanishmumkin.

Ammobufayldantashqaridabundayidentifikatorlarnoma‘lumbo‘libqoladi.

Sizkichikvao‘rtadasturlaruchuno‘zfazolaringizniyaratishingizshartemas. Ammo, agarqaytafoydalanishhuchunmo‘ljallanganfnuksiyalar, ssinflarva/yokisinflarkutubxonalaryaratishhuchunba‘zi namespace ichidasizningkodlaringizjoylashtirishyozgandasturingiznitushunarliroqqiladi.

1.13-

dasturdanomlarfazosiniyaratishvafoydalanishgamisolkeltirilgan.

1.13-dastur. Nomlarfazosiniyaratishvafoydalanish.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
// birinchi nomlar fazosi  
namespace NS1{  
    class demo{  
        int i;  
        public:  
            demo(int x) { i=x; }  
            void set(int x) { i=x; }  
            int get() { return i; }  
    };  
    char str[] = "Nomlar fazolarini o‘rnatish\n";  
    int counter;
```

```

}
// ikkinchi nomlar fazosi
namespace NS2 { int x, u; }

int main(){
    NS1::demo ob(10);
    cout << " ob = " << ob.get() << endl;
    ob.set(99);
    cout << "Endi ob = " << ob.get() << endl;

    using NS1::str; cout << str;

    using namespace NS1;
    for(counter = 1; counter<=10; counter++)
        cout << counter << ends;
    cout << endl;

    NS2::x = 10; NS2::u = 20;
    cout << "o'zgaruvchilar x, y = " << NS2::x << ", "
    << NS2::u << endl;

    using namespace NS2;
    demo x_ob(x), y_ob(u);
    cout << "x_ob, y_ob ob'ektlarning qiymati:
    " << x_ob.get(); cout << " " << y_ob.get() << endl;
    system("pause");
    return 0;
}

```

1.13 – dasturnatijasi. Output

```

ob = 10
Endi ob = 99
Nomlar fazolarini o'rnatish
1 2 3 4 5 6 7 8 9 10
o'zgaruvchilar x, y = 10, 20
x_ob, y_ob ob'ektlarning qiymati: 10 20

```

Dasturda birmuhimnuqtako'rsatilgan:
bir nechta nomlarni birgalikda ishlatganda,

birnomniboshqasialmashtirmaydi.

Hozirgifragmentdaba'zinomlarnikiritganingizda, uningnomlario'shapaytdaboshqanomdagingnomlarningo'zidabo'lishidanqat'inazar, bufragmentdaqo'shiladi. Shundayqilib, dasturbajarilayotganvaqtda, std, NS1 va NS2namespaces global namespace bo'libqo'shildi.

Yuqoridaaytibotilganidek, fayllarorasidayokibittafaylichidanomlarfazosiniajratishmumkin, keyinesabunomlarfazosininga'zolarinimazmunanbirlashtiriladi. Birgalikdanomlarfazosinibirlashtirishmisoliniko'ribchiqamiz.

1.14-dastur. Namespace larnibirlashtirish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
namespace Demo {
    int a; // Demo fazoda a o'zgaruvchi
}
int x; // Global fazoda x o'zgaruvchi
namespace Demo {
    int b; // Demo fazoda b o'zgaruvchi
}
using namespace Demo;
int main(){
    a = b = x = 101;
    cout << a << " " << a << " " << x << endl;
    system("pause");
    return 0;
}
```

1.14 – dasturnatijasi. Output

101 101 101

Bumisolda a va b o'zgaruvchilarihambirxilfazoda - Demo fazodava x o'zgaruvchisi global fazodae'lonqilingan.Ammo, using namespace Demo; fragmentibilan, a va b o'zgaruvchilarinivaglobalo'zgaruvchiniishlatishmumkin.

Yuqoridaaytibotilganidek, C++ standart o'z std fazosigaegavabutunkutubxonalariniqamraboladi.

Shuninguchunushbukitobdagibarchadasturlarquyidagiko'rsatmagaega:

```
using namespace std;
```

Bu fragment `std namespace` joriy qiladi va bevosita C++ standart kutubxonasida belgilangan funksiyalar va sinflar nomlarini kirishimkonini beradi. `std namespace` kengaytirish operatori yordamida har doim ishlatish maqsadga muvofiq emas.

Biroq, agar xohlasangiz, har bir identifikator dan oldin `std namespace` nomi ni va kengaytirish operatori ni qo'yishingiz mumkin, xatobo'lmaydi. Masalan, quyidagi 1.15-dasturda standart `std` kutubxonasi global doirada kiritilmagan.

1.15-dastur. Ochiq foydalanish uchun nom ko'rsatilgan.

```
#include "stdafx.h"
#include <iostream>
int main(){
    double val;
    std::cout << "Son kiriting:";
    std::cin >> val;
    std::cout << "Son:";
    std::cout << val << std::endl;
    system("pause");
    return 0;
}
```

Ushbu dasturda ko'rsatilganidek, standart kirish va chiqish oqimlari `cin` va `cout` operatorlaridan foydalanish uchun nomlaridan oldin ularning qaysi fazodanligini belgilash kerak.

Agar dasturingiz C++ standart kutubxonasidan keng foydalanish zarur bo'lmasa, `std namespace` ni global doirada kiritish shart emas. Ammo, agar sizning dasturingiz standart kutubxonaa'zolari uchun minglab murojaatlarni o'z ichiga olsa, hara'z uchun `std` fazosini chaqirish juda ham oson kechmaydi. Shuning uchun uni global kiritish nisbatan dasturchi son va `std` barcha standart identifikatorlarni o'z ichiga oladi.

Agar dasturingizda standart kutubxonadan faqat bir nechta a'zodan foydalansangiz, bu bir nechta a'zolari alohida belgilash uchun `using` fragmentidan foydalanish qulayroq bo'lishi mumkin. Ushbu yondashuvning afzalligishundaki, siz dasturda ko'rsatilgana'zolari ni fazosini ko'rsatmasdan kiritishingiz mumkin va ayni paytda butun standart kutubxonani global nomda kiritishingiz shart emas. Masalan:

```
#include "stdafx.h"
#include <iostream>

using std::cin;
using std::cout;
int main(){
    double val;
    cout << "Son kiriting:";
    cin >> val;
    cout << "Son:";
    cout << val << std::endl;
    system("pause");
    return 0;
}
```

Buyerda standart kirishvachiqishoqimlari cin va cout operatorlarito‘g‘ridan-to‘g‘ri ishlatilishim mumkin, ammo aynipaytda std fazoning boshqa a‘zolar joriy soxadantashqarida qoldiriladi. Maslan, endl;

Avvalroq, C++ tilining kutubxonasi global nomdabelgilanganligihaqida aytib o‘tilgan edi. Agareski C++ dasturini yangilashigiz kerak bo‘lsa, siz ham unda foydalanish uchun namespace std foydalanish kerak bo‘ladi, yokikengaytirish operatoriva std namespace nomini kiritish lozim. Agaryangi slub sarlavhalaribilaneski header fayllarnio‘rniga, ayniqsa, muhim ahamiyatga ega (*.h). Eski header fayllar global namespace o‘z mazmunini joylashtiradivayangi slub sarlavhalari uchun std namespace o‘z mazmunini joylashtiradi.

Agar, nomlar fazosini global sifatida bitta faylga e‘lon qilib, cheklamoqchibo‘lsangiz, Stili dastatikdebe‘lon qilish kerak edi, ya‘ni, statik identifikator bilan. Masalan, quyidagi ikki fayl bir xil dasturning bir qismini deb faraz qilaylik:

Birinchifayl	Ikkinchifayl
<pre>static int counter; void f1(){ counter == 99; // to‘g‘ri }</pre>	<pre>extern int counter; void f2() { counter = 10; // Xato }</pre>

counter o‘zgaruvchisi birinchifaylda aniqlanganligi uchun hamunibirinchifaylda ishlatilishigiz mumkin. Ikkinchifaylda counter

o'zgaruvchisi bilan fragmentdagi extern
 identifikatori niko'rsatishiga qaramay,
 bu o'zgaruvchini ishlatishga urinish xatolikka olib keladi. Birinchi fayldagi
 counter o'zgaruvchisi statistika deb e'lon qilib,
 uning ko'lamini shu fayl gacha cheklangan.

Statika identifikatori bilan global o'zgaruvchilarning deklaratsiyalari C++
 hali ham mavjud bo'lsa-da,
 bu qochish maqsadida eng yaxshi yo'l nomlar fazosidan foydalanish hisoblanadi,
 quyidagi misoldagi ko'rsatilganidek.

Birinchi fayl	Ikkinchi fayl
<pre>namespace { int counter; } void f1(){ counter == 99; // to'g'ri }</pre>	<pre>extern int counter; void f2() { counter = 10; // to'g'ri }</pre>

Buferlashtirilgan kiritish va chiqarish. C++ tilida ekran va faylga IO
 dantashqari kiritish va chiqarish vazifalarini lokalizatsiyalash sifatida massiv
 lardan foydalanadigan qator funksiyalarni qo'llab-quvvatlaydi.

Massivlarga asoslangan kiritish / chiqarish konseptual (array-based
 I/O) echimga o'xshasada, C tilining IO funksiyalari bilan analogdir (ayniqsa,
 sscanf() va sprintf() funksiyalari).
 Massivlarning yordamida kiritish/chiqarish C++ tilida juda moslashuvchan va foyda
 lidir,

chunki unda foydalanuvchi tomonidan yaratilgan o'zgaruvchilarning kiritish va
 chiqarishdagi ham foydalaniladi.

Massivlarning kiritish/chiqarishning barcha jihatlari niqamrabolish mumkin
 emas, ammo, eng muhim va tez
 ishlatiladigan xususiyatlari amalga oshirish mumkin.

Massivlarning yordamida

IO ni amalga oshirish uchun ham oqimlarni talab qilishini tushunish muhimdir.
 Oldingi ko'nikmalaringizdan C++ da IO amaliyotini bilishingiz mumkin.
 Biroq,

kiritish/chiqarishdagi parametrlar sifatida ob'ektlardan foydalanishning barcha
 fozalliklari haqida tushunchilarning ko'nikmalari muhimdir. Shuningdek,
 ma'lumot olish uchun,
 siz bir nechta yangi xususiyatlarni boshqaradigan funksiyalar bilan tanishishin

gizkerak.

Bufunksiyalaristalganoqimnima'lumxotiramaydonigabog'lashuchunmo'ljallangan. Buamallartugagandanso'ng, barcha IO xusisiyatlariallaqachondasturchibiladigan IO vazifalariyordamidaamalgaoshiriladideganko'nikmagaolibkeladi.

Kiritish/chiqarishob'ektlarisifatidamassivdanfoydalanishniboshlashdanoldin, <sstream>kutubxonasinidasturingizgakiritilganbo'lishikerak. Bukutubxonadagi istringstream, ostream, va stringstream sinflardanfoydalanishmumkin. Businflar, o'znavbatida, kiritish, chiqishvakirish/chiqishhuchunoqimlardanfoydalanishasoslangan. Ios sinfibusinflaruchuntayanchsinfdir, shuninguchun istringstream, ostream va istringstream sinflariningbarchafunksiyalarivamanipulyatorlari istringstream, ostream va stringstream sinflaridahammavjud. Satrmassivinichiqishhuchun ostream sinfkonstruktordanquyidagiasosiyshakldafoydalaniladi:

```
ostream chiqish_oqimi(char *bufer, streamsize size, openmode mode = ios::out);
```

Buyerda, chiqish_oqimimassivlisatrbilanbog'laydiganoqimbo'lib, buffer pointer (buferlanganko'rsatkich) orqaliberiladi. size parametrimassivningo'lchaminibelgilaydi. Mode - bustandartrejimparametri,odatdanormalchiqishholatigao'rnatiladi, lekinsiz ios sinfdabelgilanganharqandaychiqishrejimiindentifikatorio'rnatishingizmumkin.

Chiqishhuchunmassivochilganda, simvolarmassivgasig'gunchajoylashtiriladi.To'lganmassivbo'lishimumkininemas.

Ixtiyoriyto'lganmassivkiritish/chiqarishnixatolikkaolibkelishimumkin. Massivdagbelgilarnisonninianiqlashhuchun pcount() funksiya-a'zosiishlatiladi.

```
int pcount()
```

Funksiyafaqatbiroqimbilanaloqadabo'lib, shumunosabatibilanchaqiriladivaumassivgayozilganbelgilarsoniniqaytaradi. Shujumladan null tugatishnibildiradi.

Undanmassivgakiritishhuchun istringstream sinfkonstruktoriningquyidagishaklidanfoydalaniladi.

```
istringstream kirish_oqimi (const char * bufer) ;
```


Buyerda, bufer belgilari kiritiladigan massivga ko'rsatgich hisoblanadi va kirish_oqimi kirish oqimi bilan ko'rsatiladi. Biror massivdankirishma'lumotlarini o'qishda eof() funksiyasidan foydalaniladi va massivning oxiriga yetganida true qaytaradi.

Kirish/chiqish uchun massiv ochish uchun stringstream sinfi konstruktor quyidagicha shakldan foydalaniladi:

```
stringstream kch_oqimi(char * bufer, streamsize size,
openmode mode = ios : : in | ios: : out) ;
```

Buyerda, kch_oqimi – bu kirish/chiqish oqimi, belgilari o'z ichida belgilangan uzunligidagi massivga bufer ko'rsatkichi orqali kirish/chiqish ob'ekti sifatida ishlatiladi.

Yuqoridagi keltirilgan barcha IO vazifalari, shuningdek, oqim bilan ishlash, ikkilik, shujumladan tasodifiy kirish uchun muhimligini tushinish lozim.

Massivlar uchun oqim sinflaridan foydalanish C++ standartitomonidankam ko'lab-quvvatlanadi. Kelajakda C++ versiyalarida ko'lab-quvvatlanish ibtidoiyligi ham mumkin. Belgilarni massivlar oqimlari uchun mo'ljallangan bir xil vazifalar C++dagi sinf-konteyner asosida takomillashtirilgan.

1.

Massivga ma'lumotlarni yozish va chiqish uchun oddiy misol keltiramiz.

1.16-dastur. Massivga chiqarish.

```
#include "stdafx.h"

#include <iostream>
#include <sstream>
using namespace std;
int main(){
char buf[255]; // chiqarish uchun bufer
ostringstream ostr(buf, sizeof buf); // chiqarish
uchun massiv
ostr << "kirish/chiqish massiv oqimi bilan
ishlaydi.\n";
ostr << "oddiy kirish/chiqish kabi: \n " << 100;
ostr << ' ' << 123.23 << '\n';
// manipulyatorlardan ham foydalanish mumkin
ostr << hex << 100 << ' ';
```

```
// identifikator dam ham
ostr.setf(ios::scientific);
ostr << 123.45 << ' ';
ostr << ends;
// oxirgi natijani chiqarish
cout << buf;
system("pause");
return 0;
}
```

1.16 – dasturnatijasi. Output

```
kirish/chiqish massiv oqimi bilan ishlaydi.
oddiy kirish/chiqish kabi:
100 123.2364 1.234500e+002
```

Dasturda IO operatorlari qayta aniqlangan, hamda kirish/chiqish uchun ajralmas manipulatorlar, a'zo funksiyalari va massivdvn foydalanishdagi format identifikatorlari oqimlar uchun foydalanilgan. Bular manipulyatorlar uchun amal qiladigan lokal sinflar uchun IO operatorlari yaratishimkonini beradi.

Yuqoridagi dasturda ends manipulyatori massivni tugatishni belgisini qo'shish uchun ishlatiladi. Null belgi avtomatik ravishda massivni kiritilgan yoki yo'qligini amalga oshirishdastur chining o'ziga bog'liq. Shuning uchun, bu dastur chi uchun muhim, chunki massiv qatorining oxiriga null belgi yozish yaxshidir.

1.17-dastur. Massivni lugatlarini kiritish.

```
#include "stdafx.h"

#include <iostream>
#include <sstream>
using namespace std;
int main(){
char buf[] = "Salom Buxoro 1980 28.01 B";
istringstream istr(buf);
char str[80]; float f; char c; int i;
istr >> str >> i >> f >> c;
cout << str << ' ' << i << ' ' << f << ' ' << c <<
'\n';
```

```
system("pause");  
return 0;  
}
```

1.17 – dasturnatijasi. Output

SalomBuxoro 1980 28.01 B

Budasturbuferko'rsatkichidamassivdamavjudma'lumotlarnio'qiydi vaqaytaishlabekrangachiqaradi.

Agarmassivoqimbilanbog'langanbo'lsa, faylkabiishlatiladi. Masalan, massivma'lumotlarni buferko'rsatkichibilano'qishvadasturda eof() va get() foydalanishniifodalaydi

1.18-dastur. Massivlarasosida eof() va get() funksiyalarning IO dafoydalanish.

```
#include "stdafx.h"  
#include <iostream>  
#include <sstream>  
using namespace std;  
int main(){  
    char buf[] = "SalomBuxoro 1980 28.01 B";  
    istringstream istr(buf);  
    char c;  
    while(!istr.eof()) {  
        istr.get(c);  
        if (!istr.eof()) cout << c;  
    }  
    cout<<endl;  
    system("pause");  
    return 0;  
}
```

1.18 – dasturnatijasi. Output

SalomBuxoro 1980 28.01 B

Quyidagidasturmassivma'lumotlarnikiritishvachiqarishgaoid.

1.19-dastur. Massivlaryordamidakirish/chiqish.

```
#include "stdafx.h"  
#include <iostream>  
#include <sstream>  
using namespace std;
```

```

int main(){
    char iobuf[255];
    stringstream iostr (iobuf, sizeof iobuf) ;
    iostr << "tekshrish \n";
    iostr << 100 << hex << ' ' << 100 << ends;
    char str[80]; int i ;
    iostr.getline(str, 79);
    iostr >> dec >> i;
    cout << str << ' ' << i << ' ';
    iostr >> hex >> i;
    cout << hex << i << endl;
    system("pause");
    return 0;
}

```

1.19 – dasturnatijasi. Output

tekshrish 100 64

Buferlashtirilgankiritishvachiqarishda C++ da IO
 niboshqarishvazifalariuchuna'zofunksiyalarini,
 identifikatorlarnivamanipulyatorlarniformatlashorqalita'minlanadi.
 Identifikatorlar, funksiyalarvamanipulyatorlarbixilvazifanibajaradi -
 ularaxborotnikiritish/chiqarishhuchunmaxsusformatnio'rnatadi. C++
 daekrandan/gakiritish/chiqarishmosravishda cin va cout operatorlari
 yordamidaamalgaoshiriladi, ya'niformatlashmanipulyatorlariushbu IO
 operatorlaribilanbirgalikdaishlatiladi.
 Identifikatorvazifalarivaformatlashmanipulyatorlaror'tasidagifarqularg
 aqo'llaniladiganusuldir.

Endiformatlashobektlaridanfoydalanishyo'llariniko'ribchiqamiz.

Asosiyformatlasha'zofunksiyalari:

cout.fill('symbol'); to'ldiruvchibelginio'rnatadi. Bunda symbol
to'ldiruvchibelgivatirnoqichidaberiladi.

cout.width(width_field); maydonuzunliginiberish. width_field –
pozitsiyalarsoni, harbirpozitsiyadabittasimvolsig'adi.

cout.precision(number);

haqiqiysonlardanuqtadankeyingibelgilarsoniberish. number -
haqiqiysonlardanuqtadankeyingibelgilarsoni.

Funksiyalarganuqtaamaliorqalikirishiladivaqavslarichidakerakliarg
umentyoziladi. Fill()

funksiyasigaargumentnibittaqo'shtirnoqbilanbelgisifatidayokison(belgik

odi) sifatidayozishmumkin.
 kirish/chiqishoqimlarniformatlashuchunbittafunksiyayetarliemas.
 C++daindentifikatorlarbilanformatlashusullariniko‘ramiz.

Indentifikatorlarorqaliformatlashkirish/chiqishparametrlaridanbiri
 niyoqishyokio‘chirishimkoniniberadi. IO
 nindentifikatornio‘rnatishuchun setf()
 funksiyasiniishlatamizvaagarchiqishindentifikatornio‘chirmoqchibo‘lsan
 giz, unsetf() funksiyasidanfoydalaniladi.
 Quyidagiindentifikatornio‘rnatishvao‘chirishgadoirinstruksiyako‘rsatilga
 n.

Chiqishindentifikatornio‘rnatish

```
cout.setf( ios::name_flag );
```

name_flag - indentifikatornomi.

Chiqarishoperatorlariganuqtaorqaliruxsatberiladi. setf()
 funksiyasibittaargumentqabulqiladivaindentifikatornomidir.

Indentifikatorlar ios
 sinfigategishlibo‘lganligiuchunoldinshusinfnomiyoziladivakengaytirisha
 maliorqalikerakliindentifikatoratanlanadi.

Chiqishindentifikatornio‘chirish

```
cout.unsetf( ios::name_flag );
```

name_flag - indentifikatornomi.

Agarkirish/chiqishdabirnechtaidentifikatorlarniishlatishkerakbo‘lsa
 , Yoki [] mantiqiyamalidanfoydalanishmumkin.

Birnechtaidentifikatorlarnio‘rnatish

```
cout.setf( ios::name_flag1 | ios:: name_flag2 | ios:: name_flag_n );
```

Birnechtaidentifikatorlarnio‘chirish

```
cout.unsetf( ios::name_flag1 | ios:: name_flag2 | ios:: name_flag_n );
```

1.1-

jadvalasosiyformatlashidentifikatorlarinikeltirilganvaulardanfoydalanish
 gadoirmisollarko‘rsatilgan.

1.1-jadval. C++daformatlashidentifikatorlari.

Identifi- kator	Vazifasi	misol	Misolna tija-si
boolalpha	Mantko‘rinishidamanti qiykattaliknikiritish (true, false)	cout.setf(ios::boolalpha); bool log_false = 0, log_true = 1; cout << log_false << endl	false true

		<< log_true << endl;	
oct	Sakkizlik sanoq sistema sidakattaliklarni kiritish (avvaldec identifikator ni o'chiramiz va oct identifikator o'rnatamiz)	cout.unsetf(ios::dec); cout.setf(ios::oct); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 1438
dec	O'nlik sanoq sistemasid akattaliklarni kiritish/c hiqarish (identifikator joriy o'rnatilgan)	cout.setf(ios::dec); int value = 148; cout << value << endl;	148
hex	O'nollik sanoq sistema sidakattaliklarni kiritish (avvaldec identifikator ni o'chiramiz va oct identifikator o'rnatamiz)	cout.unsetf(ios::dec); cout.setf(ios::hex); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 6316
showbase	Sanoq sistemaga asoslangan indikatorni chiqarish	cout.unsetf(ios::dec); cout.setf(ios::oct ios::showbase); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 01438
uppercase	16 lik SSdakat harflarni chiqarish (joriy holda dagi chikharflar o'rnatilgan)	cout.unsetf(ios::dec); cout.setf(ios::hex ios::uppercase); int value; cin >> value; cout << value << endl;	kirish: 25510 chiqish: FF16
showpos	Musbat sonlar uchun + ishorasini chiqarish	cout.setf(ios::showpos); int value = 15; cout << value << endl;	+15
scientific	Eksponensial sonlarni chiqarish	cout.setf(ios::scientific); double value = 1024.165; cout << value << endl;	1.02416 5e+003
fixed	Fiksirlangan holda haqiqiy sonlarni chiqarish	double value = 1024.165;	1024.16 5

	(joriyholatda)	cout << value << endl;	
right	O'ngtomondantekslash	cout.width(40); cout << «cppstudio.com» << endl;	__cppst udio.co m
left	Chaptomondantekislas h	cout.setf(ios::left); cout.width(40); cout << «cppstudio.com» << endl;	cppstudi o.com__

Shuningdek, manipulyatorlar bilan formatlashning usullari ham bor. manipulyator — maxsus tip bo'lib, oqimlarga o'tgan ma'lumotlarni formatlash uchun kiritish/chiqish oqimlarni nazorat qiladi. Lekin, ko'pchilik manipulyatorlarining funksiyalari identifikatorlar bilan bir xil formatlash amallarini bajaradi. Ba'zida, identifikatorli yoki formatlash funksiyalaridan foydalanish osonroq bo'lsada, ba'zan formatlashda manipulyatorlaridan foydalanish qulayroq. Shuning uchun C++da kiritish / chiqishni formatlashning bir nechta vositalari bilan o'tirilgan. 1.2-jadvalda formatlashning manipulyatorlariga misollar keltirilgan.

1.2-jadval. Formatlashning manipulyatorlari.

Manipulyator	vazifasi	misob	natijasi
endl	Chiqarishdagi yangi qatorga o'tish	cout << «Salom:» << endl << «mbbGroup »;	website: cppstudio.c om
boolalpha	Mantiqiy kattalikni matn ko'rinishidagi chiqarish (true, false)	bool log_true = 1; cout << boolalpha << log_true << endl;	true
noboolalpha	Mantiqiy kattalikni son ko'rinishidagi chiqarish (true, false)	bool log_true =	1

		<pre>true; cout << noboolalpha << log_true << endl;</pre>	
oct	8 SSdasonlarnichiqarish	<pre>int value = 64; cout << oct << value << endl;</pre>	1008
dec	10 SSdasonlarnichiqarish	<pre>int value = 64; cout << dec << value << endl;</pre>	6410
hex	16 SSdasonlarnichiqarish	<pre>int value = 64; cout << hex << value << endl;</pre>	408
showbase	SSuchunasosiyindikatornichiqarish	<pre>int value = 64; cout << showbase <<hex << value << endl;</pre>	0x40
noshowbase	SSuchunasosiyindikatornichiqarish maslik (joriyholat).	<pre>int value = 64; cout << noshowbase << hex << value << endl;</pre>	40
uppercase	16 SSdakattaharflarnichiqarish (joriyholatdakichikharflar)	<pre>int value = 255; cout << uppercase << hex <<</pre>	FF16

		value << endl;	
nouppercase	16 Ssdakichikharflarnichiqarish (joriyholat)	int value = 255; cout << nouppercase << hex << value << endl;	ff16
showpos	Sonlarni + belgisibilanchiqarish	int value = 255; cout << showpos<< value << endl;	+255
noshowpos	Sonlarni + belgisizchiqarish (joriyholat)	int value = 255; cout <<noshowpos<< value << endl;	255
scientific	Sonlarnieksponensialko‘rinishda chiqarish	double value = 1024.165; cout << scientific << value << endl;	1.024165e+003
fixed	Fiksirlashholdasonlarnichiqarish (joriyholat).	double value = 1024.165; cout << fixed << value << endl;	1024.165
setw(int number)	Maydonuzunliginio‘rnatish, number — simvollarningpozitsiyalarisoni, parameterlimonupulyator	cout << setw(40) << “mbbGroup” << endl;	— mbbGroup

right	O'ngdantekslash	cout << setw(40) << right << « mbbGroup » << endl;	— mbbGroup
left	Chapdantekislash	cout << setw(40) << left << « mbbGroup » << endl;	mbbGroup —
setprecision(int count)	Verguldankeyikalitbelgiberish, count — verguldankeyingibelgilarsoni	cout << fixed << setprecision (3) << (13.5 / 2) << endl;	6.750
setfill(int symbol)	Qo'shimchabelgilarnio'rnatish	cout << setfill('0') << setw(4) << 15 << ends << endl;	0015

C++

dabuferlashtirilganformatlangankirish/chiqishdasturlashdagiengoddiyma
vzulardanbiri hisoblanadi.

Ma'lumformatlashvositalaridanqandayfoydalanishyuqoridagijadvallarda
ko'rsatilgan,

shuninguchunbumavzudahechqandayqiyinchilikbo'lmasligikerak.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Polimorfsinfqandayxususiyatgaegabo'lishilozim?

|

2. Dasturbajarilishidavomidaob'ektturiniolishuchunqandayoperatordanfoydalaniladi?
3. `type_info` sinfda `public` modifikatoribilanqaysistatikoperatorlaraniqlangan?
4. `Type_info` sinfa'zolaridagiqandayoperatordanfoydalanishtaqiqlangan?
5. `Typeid` operatorinihavolaliparametriga qo'llashmumkinmi?
6. C++ tilidanechatipnialmatirishoperatorlarimavjudvaularninomlari, ishlatishuslublariniayting.
7. `Dynamic_cast` operatoridasturbajarilishidavomidaqaysiturdagiquyishamalinibajaradi?
8. `Dynamic_cast` va `const_cast` tipalmashtirishoperatorlariningvazifalari, kamchiliklarivayutuqlarieslang.
9. O'zgarmase'lonqilinganidentifikatorningtipnialmashtirishmumkinmi. Mumkinbo'lsa qandayamalgaoshiriladi. Mumkinbo'lsa, nimasababdanizohlayubering.
10. Polimorfbo'lmaganvabo'lgansinfob'ektingtipnialmashtirishuchunqaysioperatorlardanfoydalaniladi, harbiriniizog'layubering.
11. To'rtta `cast` operatorlari `dynamic_cast`, `const_cast`, `static_cast`, `reinterpret_cast` bilan an'anaviy tipnialmashtirishamali (kasting) qandayfarqmavjud.
12. Yanginomlarfazosiyaratishchun kerak, misollarbilaneslabtushuntiribbering.
13. Jamoaviy dasturlashvaqtidasturdaganomlarningziddiyatlarni oldin olishchun nimavazifani amalga oshirish kerakvaziddiyatlarni bekorqilis hningengoddiyvaprofessionalusulinitushuntiring.
14. Nomlarfazosini 2 ga ajratishmumkinmi, mumkinbo'lsa, qandayvanimauchun nomlarfazosini 2 ga ajratishkerakbo'ladi.
15. `Using` kalitso'zni ishlatmasdanturib, nomlarfazosigamurojaat qilishmumkinmi? Mumkinbo'lsa, misolbilanasoslabbering.
16. C++ tilida ekranvafaylga IO dantashqarikiritishvachi qarishvazifalarini lokalizatsiyalashsifatida qanday vositadan foydalanadigan qator funksiyalarni qo'llab-quvvatlanadi.
17. Kiritish/chiqarishob'ektlarisifatidamassivdan foydalanishchun qanday header fayldan foydalanishlozim.
18. Massivgama'lumotlarni yozishvachi qishchun qanday tipdan foydalaniladi.

19. Massivlarasosida eof() va get() funksiyalarning IO dafoydalanishnimauchunzarur.
20. Kirish/Chiqishidentifikatornio‘rnatishvao‘chirishusullariaytibberi ngvamisollarorqaliasoslang.
21. Formatlashidentifikatorlarinimauchunkerakvauqandayishlatiladi.
22. Sanoqsystemagaasoslanganindikatornichiqarishuchunqaysiformatla shidentifikatoridanfoydalanamiz.
23. Chaptomondantekislabformatlashdaqaysiformatlashidentifikatorida nfoydalanamiz.
24. Monipulyatorlarbilanformatlashningusullarinimauchunkerakvauqa ndayishlatiladi.
25. Sanoqsystemagaasoslanganindikatornichiqarishuchunqaysimonipul yatorlarbilanformatlashningusuldanfoydalanamiz.



AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG‘I	
	Tipnianiqlashgaoidberilganquyidagidasturbo‘yichaberilgantopshi riqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarb osqichma – bosqichamalganoshriladi.
dastur	topshiriqlar
<pre>#include <iostream> #include <typeinfo> #include <cstdlib> using namespace std;</pre>	1. Figure sinfdagixatoliknianiqlang. Dasturdatuzatishkiriting.

<pre>class figure { protected: double x, y; public: figure(int i, double j) { x = i; y = j; } virtual double area() = 0; };</pre>	2. Figure sinfi uchun kub nomli meros xurrosini yaratish va uning haqiqiy tanlash operatoridan foydalanib, yaratilgan ob'ektlarining tiplarini kuba nisbatini topuvchi dasturtuzing.
<pre>class triangle : public figure { public: triangle(double i, double j) : figure(i, j) {} double area() { return x * 0.5 * y;</pre>	3. Dasturda ob'ektlarni yaratish sonidagi dinamik almash tirish uchun o'zgartirish kiriting va fragmentani qayta yozing.
<pre>}</pre>	4. typeid(*p) == typeid(triangle) fragmentiga almashtirish variantlarini aniqlang va dasturga yozib, tekshirib ko'ring.

<pre> } }; class rectangle : public figure { public: rectang le(double i, double j) : figure (i, j) {} double area() { return x * y;} }; class circle : public figure { public: circle(double i, double j=0) : figure(i, j) {} double area() {return 3.14 * x * x;} }; // figure sinf </pre>	5. dasturdatipnialmashtirishishlatilganmiyokitiplarn itaqqoslashmi? Vabuamaldasturningqaysifragmentlaridao‘zaksinitopgan.

ob'ektlarin
i yaratish.

```
figure
*factory(){
    switch(
rand() % 3
) {
    case 0:
return new
circle
(rand()%15)
;
    case 1:
return new
triangle
(rand()%15,
rand()%15);
    case 2:
return new
rectangle
(rand()%15,
rand()%15);
    }
return 0;
};
```

```
int main(){
    figure
*p; // asos
sinfga
ko'rsatkich
    int i;
    int
t=0, r=0,
c=0;
// 15 ta
ob'ektlarni
```

```

hosil qilsh
va sanash
    for(i=0
; i<15;
i++) {
        p =
factory();
// ob'ektni
hosil
qilish

        cout <<
"Ob'ektning
tipi: " <<
typeid(*p).
name();

        cout <<
". ";
        //
ob'ektni
tekshirish
orqali
sanasj

        if(type
id(*p) ==
typeid(tria
ngle)) t++;

        if(type
id(*p) ==
typeid(rect
angle))
r++;

        if(type
id(*p) ==

```



```

typeid(circle)) c++;
    //
    Yuzasini
    hisoblash

    cout <<
    " S= " <<
    p->area()
    << endl;
    }
    cout <<
    endl;
    cout <<
    "Quyidagi
    ob'ektlar
    hosil
    qilindi:\n"
    ;
    cout << "
    Uchburchakl
    ar: " << t
    << endl;
    cout << "
    To'rtburcha
    klar: " <<
    r << endl;
    cout << "
    Doiralar: "
    << c <<
    endl;
    system("pau
    se");
    return 0;
}

```

6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

IKKINCHI ASSISMENT TOPSHIRIG'I	
	Tipnialmas trishga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani.  Bunday dasturda giba'zi'zgartirishlarni topish orqali topshiriqlar bo'sqichma – bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>#include "stdafx.h" #include <iostream> #include <typeinfo> using namespace std; class Base { public: virtual void f() { cout << " Base sinfi orqali.\n"; } // . . . }; class myClass : public Base { public: void f() { cout << "myClass sinfi orqali.\n"; }</pre>	1. Dynamic_cast tipni almashtirishni ishlamasligi uchun dasturning qaysi fragmentini o'chirish kerak. _____ _____ _____
	2. int berilgan o'zgarishni tipni almashtirish dastur fragmentining qaysi qismida berilgan. Shu fragmentni float tipidan boshqatipga almashtirish fragmentini tuzing. _____ _____ _____ _____ _____
	3. static_cast<bool>(x); fragmentning qiymatini 0 chiqaradigan qilib dasturni o'zgartiring. _____ _____ _____ _____
	4. Dastur natijasida ekranga 2 marta Xatolik nishqarish uchun dasturning kerakli fragmentlariga o'zgartirishlarni yozing. _____ _____ _____

<pre> }; void funk (const int *p){ int *v; v = const_cast<i nt *> (p); *v = 1980; }; int main(){ Base *bp, b_ob; myClass *dp, d_ob; bp = dynamic_cast <Base *> (&d_ob); if(bp) { cout << " tipni almashtirish " << (myClass* => Base *) bajarildi. "; bp->f(); } else cout <<"Xatolik\n "; cout << </pre>	5. Dasturda reinterpret_cast<int>ning natijasi 65 chiqishi uchun qaysi fragmentni o'zgartirish kerak.

```

endl;
    dp =
dynamic_cast
<myClass *>
(&b_ob);
    if(dp)
cout
<<"Xatolik\n
";
    else
    cout <<"
tipni
almashtirish
"<<"(Base*=>
myClass*)
bajarilmadi.
\n";
    cout <<
endl;
    int x =
2004;
    cout <<"
x = " << x
<< endl <<
endl;
    funk(&x)
;
    cout <<"
x = " << x
<< endl;
    cout <<
endl;
    double
d;
    d =
static_cast<
bool>(x);
    cout <<"

```

```

x = " << dec
<< d <<
endl;
    cout <<
endl;
    char *p
= "Salom
Buxoro";
    x =
reinterpret_
cast<int>
(p);
    cout <<"
x = " << x
<< endl;
system("paus
e");
return 0;
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

UCHINCHI ASSISMENT TOPSHIRIG'I	
	Nomlar fazosidan foydalanishga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani. 👉 Bundagi dasturda bo'yicha 'zio' zgartirishlari topish orqali topshiriqlar bosqichma – bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>#include "stdafx.h" #include <iostream > using namespace std; namespace NS1{ class demo{ int i; public: c: demo(int x) { i=x; } void set(int x) { i=x; } int get() { return i; }</pre>	1. Dasturdagi std fazosining elementlari bilan using namespace std; fragmentidan foydalanmasdan qanday bo'yicha zgartirishlari kiritilishi aniqlang. _____ _____ _____
	2. NS1 fazodagi str ko'rsatkichining qiymatini bo'yicha zgartirib chiqarish fragmentini yozing. _____ _____ _____
	3. Dasturda yaratilgan fazolardan yopiq holda foydalanish uchun barcha fragmentlari bo'yicha zgartiring. _____ _____ _____ _____ _____
	4. Dasturda _____ 2 ga ajratilgan fazo yaratilgan va tekshirib ko'ring. _____ _____ _____

<pre> }; char str[] = "Nomlar fazolarin i oʻrnatish \n"; int counter; } namespace NS2 { int x, u; } int main(){ NS1:: demo ob(10); cout <<" ob = " << ob.get() << endl; ob.se t(99); cout << "Endi ob = "<< ob.get() << endl; using NS1::str; cout << str; </pre>	5. Dasturdagi takrorlanish jarayonika mayis tartibd asaralanib chiqishi uchun qanday oʻzgartirish kiritish kerak.

```
        using
namespace
NS1;
        for(c
ounter =
1;
counter<=
10;
counter++
)

        cout
<<
counter
<< ends;
        cout
<< endl;

        NS2::
x = 10;
NS2::u =
20;
        cout
<<
"o'zgaruv
chilar x,
y = "<<
NS2::x<<"
, " <<
NS2::u<<
endl;

        using
namespace
NS2;

demo
```



```

x_ob(x),
y_ob(u);

cout <<
"x_ob,
y_ob
ob'ektlar
ning
qiymati:
"<<
x_ob.get(
); cout<<
" " <<
y_ob.get(
) <<
endl;
system("p
ause");
return 0;
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

TO‘RTINCHI ASSISMENT TOPSHIRIG‘I	
	Buferlashtirilgankirishvachiqarishdanfoydalanishgaoidberilganquy idagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosida bajaring. 👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbos qichma – bosqichamalganoshiriladi.
dastur	topshiriqlar
<pre>#include "stdafx.h" #include <iostream> using namespace std; int main(){ char buf[] = "SalomBuxor o 1980 28.01 B"; istrstream istr(buf); char c; while(! istr.eof()) { istr.ge t(c); if (!istr.eof()) cout << c; } cout<<e ndl;</pre>	1. Dasturdagibuferlashtirilgankirish/chiqishoperatorl arningishlamasliginianiqlangvaxatosinituzatib, keraklijoygafragmentniyozing. _____ _____
	2. Birinchiekrangachiqarishoperatorininimaniekrang achiqaradivabirinchi buferlashtirilganchiqarishmassivg animaniyozadi. _____ _____ _____
	3. Dasturda <code>iostr.getline(str, 79);</code> fragmentningvazifasinianiqlang. _____ _____
	4. Dasturdamatnitekslashvasonlarniishorasibilanchiq arishfragmentiniajratingvatekshiribko‘ring. _____ _____ _____
	5. Dasturdagitakrorlanishjarayoniqandayvazifanibaj aradi. _____ _____ _____

```

char
str[80];
float f;
char ch;
int i;
istr >> str
>> i >> f
>> ch;
cout << str
<< ' ' << i
<< ' ' << f
<< ' ' <<
ch << '\n';

stringstream
iostr
(iobuf,
sizeof
iobuf) ;

    iostr
<< 100 <<
hex << ' '
<< 100 <<
ends;
    char
str[80];
int j ;
    iostr.g
etline(str,
79);
    iostr
>> dec >>
j;
    cout <<
str << ' '
<< j << '
';

```

<pre> iostr >> hex >> j; cout << hex << j << endl; cout.setf(i os::left); cout.width(28); cout << «mbbGroup» << endl; cout.setf(i os::showpos); int value = 15; cout << value << endl; system("pau se"); return 0; } </pre>	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

1.2. Konteynerlar (Kollektsiyalar).

 STL (Standard Template Library) kutubxonalaribilantanishib, uningto‘plamlarivaharto‘plamningvazifasi, ishlatishtmaqsadi, maxsusususullarivaalgoritmalarini, iteratorvadinamikstrukturaxususiyatlarini, iteratorlarningturlarini, iteratorlarningusullaridanfoydalanishni, barchato‘plamlardamavjudbo‘lganasosiyusullarni, STL kutubxonaalgoritmalarini, barchato‘plamelementlaritanlashvaulargaishlovberishusullari,

saralashusullari,
 to'plama'zolariustidama'lumarifmetikamallarnibajarishusullari,
 pridekatlar, oqimxavfsizligi, konteynersinflar, ketma-ketkonteynerlar,
 assotsiativkonteynerlar, konteynerlarningxususiyatlari,
 konteynersinflarningumumiyusullari, chiziqlikonreynerlar(array, vector,
 deque, list, forward_list), konteynerlarbilashishlashfunksiyalari,
 allocator ajratuvchiningvazifasi, biraloqaliro'yxat<forward_list>,
 ikkialoqaliro'yxat (ikkilanganro'yxat) <list>,
 ikkitomonlamanavbat<deque>, dinamikmassiv (vektor) <vector>,
 statikmassiv<

array>kabito'plamlarningfazifalarivausullarikeltirilganbo'lib,
 nazariybilimlarniasoslashuchundasturlartuzibko'rsatilgan.

Bilimlarnimustahkamlashuchun 20
 tanazariysavolvaamaliyko'nikmavamalakalarnirivojlantrishuchun 3
 taassimenttopshirig'ivaharassimentda 7 tatopshiriq, jami
 21tatopshiriqberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍Kalitso'zlar. STL kutubxonasi, to'plam, shablon, vector, list, map, set, multimap, multiset, string, wstring, stringstream, satrlloqim, iterator, dinamikma'lumotlartuzilmasi, pridekat, konteynersinflar, ketma-ketkonteynerlar, chiziqlikonreynerlar(array, vector, deque, list, forward_list), biraloqaliro'yxat<forward_list>, ikkialoqaliro'yxat (ikkilanganro'yxat) <list>, ikkitomonlamanavbat<deque>, dinamikmassiv (vektor) <vector>, statikmassiv< array>, allocator.

✓Bilishshartbo'lgantushunchalar. Tiptushunchasi, sinfvasinfob'ekti, to'plam, massiv, elementgamurojaat, funksiyavako'rsatkich, ma'lumotlarnikirishvachiqishi, oqim, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo'llabquvvatlovchimu hitdaishlashnibilishlozim.

👉Bilibolasiz. STL kutubxonasiningimkoniyati, vector, list, map, set, multimap, multiset, string, wstring, stringstream, satrlloqimkabito'plamlarniishlatishusullarivavazifalar, iteratorvadinamikma'lumotlartuzilmasiishlatishvaturlari, pridekat, konteynersinflar, ketma-ketkonteynerlar, chiziqlikonteynerlar(array, vector, deque, list, forward_list) xususiyatlarnivafunksiyalariniishlashtamoyllari, biraloqaliro'yxat<forward_list>, ikkialoqaliro'yxat (ikkilanganro'yxat) <list>, ikkitomonlamanavbat<deque>, dinamikmassiv (vektor)

<vector>, statikmassiv< array>kabikonteynerlarnidasturlashdagio‘rniva allocator orqaliyaratishvafoydalanishusullarinio‘rganishingiz mumkin.

REJA

- 1.STL kutubxonalari.
- 2.Konteynersinflar.
- 3.Chiziqlikonteynerlar (array, vector, deque, list, forward_list).

KIRISH

Dasturlashtexnologiyalariningrivjlanishidasturlashtillariniishlabchi qaruvchilaroldigajudajiddiymasallarnipaydoqilmoqda.

Shulardanbiribudasturlashtillariuchunturlixilto‘plamlarbilanishlashdir.

Masalan, xayotdanolibqaraydiganbo‘lsak, ammoqonuniyatiharxil. barchanarsalarqandaydirto‘plam, ammoqonuniyatiharxil.

Buto‘plamlarniqandaydasturlashkerakdeganmuammopaydobo‘ladi.

Shuninguchunbarchaob‘ektgayo‘naltirilgandasturlashtillarida STL, ya‘nistandardshablonlarkutubxonasi tushunchasikiritilgan. STL (Standard Template Library) kutubxonalari, to‘plamturlari, xususiyatlari, usullarivafunksiyalar,

konteynervaiteratorlarbo‘yichanazariymateriallarnikeltiramiz.

STL	(Standard	Template	Library)
kutubxonalari.	Shablonmexanizmlari		C++

kompilyatorigamoslabqurilganbo‘lib,

dasturchilargaumumiyydasturlashyordamidadasturfragmentlariniqisqartiri shgaimkonberadi. Tabiiyki,

bundaymexanizmlarniamalgaoshiruvchistandardkutubxonalarham mavjud

. Bugungikunda C++ dasturlashtilidaengsamarali STL kutubxonasihisoblanadi.

STL kutubxonasiningko‘plabtatbiqlarimavjudbo‘lib, ularningharbirianiqtandarddoirasidayaratilganbo‘lsa-da,

o‘zkengaytmalarigaega. Ammobundayyondashuvningbirkamchiligibor: dasturfragmentinihardoimturlikompilyatorlarbilanbirxiltarzdaishlamaydi

. Shuninguchun, dasturchiqanchalikmohirlikbilankutubxonaniyaratsavafoydalanisa,

o‘zigaxosbajarilishinitushunsaham,

imkonqadaran‘anaviyusullardanfoydalanishnitavsiyaqilamiz.

C++dasturlashtiliningkutubxonalaridanengmashhurto‘plamlarniko‘ribchiqaylik.

Ular har bir muhim vazifalar ni hal qilinishi mumkin doirasi uchun o'z shablون parametr lariga ega.

To'plamlar ni dastur fragmentida ishlatish uchun quyidagi fragmentdan foydalaniladi.

```
#include <T>
```

Bunda T – to'plamning nomi.

Odatda quyidagi to'plamlar ko'pishlatiladi.

vector – elementlar to'plami, o'lchamini o'zgartish kerak bo'lgan massivdasaqlanadigan elementlar to'plami (odatda ortib boradigan); Dastur gaulanish uslubi:

```
#include <vector>
```

list – elementlar to'plami, elementlar ni ikki tomon lamab o'lgan ro'yxat sifatida saqlaydigan to'plam; Dastur gaulanish uslubi:

```
#include <list>
```

map – elementlar to'plami, har bir element shakli <const kalit, T> juftlikdasaqlanadigan to'plam, bu oddiy bir juftlik <kalit, qiymat> juftligi (har bir kalit bitta qiymatni mos keladi). Kalit – taqqoslash amali uchun qiymatini tavsiflovchi ma'lum bir qiymat. Juftlikda kalit tez qidirishni amalga oshirish imkonini beradi, kalit asosida tartiblashtirilgan shaklida saqlanadi. Tartiblashni amalga oshirish uchun oldindan tashkil qilish qonuniyatini aniqlash kerak. Dastur gaulanish uslubi:

```
#include <map>
```

set – elementlar to'plami, faqat kalitlarning qiymatini bo'yicha tartiblangan to'plamdir, ya'ni taqqoslash amali qo'llaniladigan, ammo takrorlanmaydigan qiymatlar — har bir kalit to'plamda (inglizcha set-to'plam degan ma'noni beradi) faqat birmarta foydalaniladi; Dastur gaulanish uslubi:

```
#include <set>
```

multimap – map, juftlikda kalitlar unikal emas, takrorlanadigan to'plam. Agarkalit bo'yicha qidirsangiz, siz bitta qiymatni emas, balki bir xil kalit qiymatiga ega bo'lgan elementlar to'plamini olasiz. Dastur gaulanish uslubi:

```
#include <map>
```

multiset – set, juftlikda kalitlar unikal emas, takrorlanadigan to'plam. Agarkalit bo'yicha qidirsangiz, siz bitta qiymatni emas,

balkibirxilkalitqiymatigaegabo‘lganelementlarto‘plaminiolasiz.

Dasturgaulanishuslub:

```
#include<set>
```

Belgilarto‘plamibo‘lgansatr

(qator)nihamto‘plamsifatidaqarashmumkin.

Shuninguchunixtiyoriykutubxonasatrlarbilanishlashvaifodalashuchuno‘z iningsinflargaega. STLdasatrlar ASCII va Unicode formatlaridaifodalanadi.

string — to‘plam, ASCII formatidagibirbaytlibelgilarto‘plami;

wstring — to‘plam, Unicode formatidagiikkibaytlibelgilarto‘plami;

Dasturgaulanishuslub:

```
#include <string>
```

```
#include <wstring>
```

Shuningdek, belgilarto‘plamibo‘lgan, satrliqimlarnihamto‘plamsifatidaqarashmumkin. **stringstream-**

oddiy ma’lumotlartiplarini STL-string

ko‘rinishadasaqlashnitashkilqilishuchunishlatiladi.

Dasturgaulanishuslub:

```
#include <stringstream>
```

Dasturlarnitahlilqilishninishusinfdanboshlaymiz.

2.1-dastur. Satrliqimto‘plamsifatidaishlatish

```
#include "stdafx.h"
#include <iostream>
#include <stringstream>
#include <string>
using namespace std;

int main(){
    stringstream xstr;
    for (int i = 0; i < 10; i++)
    {
        xstr << "Demo " << i << endl;
    }
    cout << xstr.str() << endl;
    string str;
    str.assign(xstr.str (), xstr.pcount());
    cout << str.c_str() << endl;
```



```

    system("pause");
    return 0;
}

```

2.1-dasturnatijasi. Output	
Demo 0	
Demo 1	
Demo 2	
Demo 3	
Demo 4	
Demo 5	
Demo 6	
Demo 7	
Demo 8	
Demo 9	
	xxxx1111111111yu■yu■
Demo 0	
Demo 1	
Demo 2	
Demo 3	
Demo 4	
Demo 5	
Demo 6	
Demo 7	
Demo 8	
Demo 9	

Satrliqim - oqimoxiridabir null terminatorbilantugaydiganbuferdir.

Shuninguchununingbo‘shqolganelementlariixtiyoriybelgibilanto‘ldiriladi. Satrliqimninghaqiqiyqisminiolishuchun pcount() elementlarnisanagichidanfoydalanishmumkin. Keyinesa, satrliqimning "haqiqiyqismi" olinadivachopqilinadi.

Iteratorlarhamdinamikstrukturagaegato‘plamdir.

Iteratordinamikma’lumotlartuzilmalariniamalgaoshirishdajudamuhimtus hunchadir. Tushunishimizuchun,

iteratornima’lumbircheklovlarbilanbirko‘rsatkichsifatidaabstraktko‘rinis hdaaniqlashmumkin. Sirasiniaytganda,

iteratorumumiytushunchabo‘libvabirko‘rsatkichuchunob’ektto‘plamibo‘l

adi, ammmo'rsatkichbuiteratoremas.
Iteratorsinfiniquyidagichaquribolishmumkin:
2.2-dastur. Iteratorninginfiniquyirish.

```
class Iterator{
    T* pointer;
public:
    T* GetPointer (){
        return this -> pointer;
    }
    void SetPointer (T* pointer){
        this -> pointer = pointer;
    }
};
```

Iteratorningba'zibirformallashtirilganta'riflari:
Iteratorlarto'plamelementlarigakirishnita'minlaydi. Harbiraniq
STL sinfiuchuniteratorlarto'plamdasinfighidaalohidaaniqlanadi.

Iteratorlarninguchturimavjud:

- (forward) iterator - to'plamnikichikindeksdankattaroqindeksgaotkazishuchun;
- reverse iterator - to'plamnikattaindeksdankichikroqindeksliotkazishuchun;
- random access iterator - to'plamdantasofidiy, harqandayyonalishdatanlashuchun.

To'plamningyarimelementlarinio'chirishgadoirmisolkeltiramiz.

2.3-dastur. To'plamningyarimelementlarinio'chirish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

void printInt (int number);

int main (){

    vector<int> myVec;
    vector<int>::iterator first, last;
    for (long i=0; i<10; i++){
```

```

        myVec.push_back(i);
    }
    first = myVec.begin();
    last = myVec.begin() + 5;
    if (last >= myVec.end()){
        return - 1;
    }
    myVec.erase(first, last);
    for_each (myVec.begin(), myVec.end(),
printInt);
    system("pause");
    return 0;
}
void printInt (int number)
{
cout << number << endl;
}

```

2.3-dasturnatijasi. Output

```

5
6
7
8
9

```

2.3-dasturdagiba ‘zifunksiyalarni keyinroqtushuntiribo’ tamiz.

Shuni ham tushunish kerakki,

to‘plamning qandaydirelementi uchun iterator olishdato‘plamni ketma-keto‘zgartirish iteratorni yaroqsiz holga keltirish mumkin.

Iteratorning iteratsiyasida oldingavashungao‘xshashorqagao‘tish quyidagicha fragmentiasosidasodirbo‘ladi:

```

for (iterator element = begin(); element < end();
element++)
{ t = (*element); }
for (iterator element = end(); element < begin();
element--)
{ t = (*element); }

```

Tasodifiy tanlash iteratoridan foydalanganda,

masalan,

quyidagicha fragmentni yozish mumkin:

```
for (iterator element = begin(); element < end();
    element+=2)
{ t = (*element); }
```

Barchato‘plamlardamavjudbo‘lganasosiyusullarquyidagi 2.1-jadvalgakeltiramiz.

2.1-jadval. To‘plamusullari.

№	nomi	vazifasi
1	empty	To‘plamnibo‘shliginitekshiradi
2	size	To‘plamningo‘lchaminiqaytaradi
3	begin	To‘g‘riiteratorningbirinchielementiniko‘rsatadi
4	end	To‘g‘riiteratorningoxirgielementiniko‘rsatadi. Elementiyo‘qto‘plamgaoxirgisidankeyingao‘tadi
5	rbegin	Teskariiteratorningbirinchielementiniko‘rsatadi
6	rend	Teskariiteratorningoxirgielementiniko‘rsatadi.
7	clear	To‘plamnitozalash, barchaelementlarinio‘chiradi.
8	erase	Ajratilganelementlarnito‘plamdano‘chiradi.
9	capacity	To‘plamningsig‘iminiqaytaradi, ya‘nibuto‘plamuchunmumkinbo‘lganelementlarsoni (aslidato‘plamuchunqanchaxotiraaajratilganiniqaytaradi) ;

To‘plamningsig‘imi (hajmi), boshidaaytibotilganidek, kerakbo‘lgandao‘zgaradi, ya‘ni, agarto‘plamuchunajratilganbarchaxotiralarto‘lganbo‘lsa, undangielementqo‘shilganda, to‘plamningsig‘imioshiriladivao‘sishtanoldinundagibarchaqiymatlaryan gixotiramaydonigako‘chiriladi - budasturchilaruchunjudaqimmatamalbo‘libhisoblanadi. Dasturlashdato‘plamninghajmivaquvvatiishonchhosilqilishmuhimdasturl ashelementhisoblanadi. To‘plamningsig‘imiturlichaekanligigaishonchhosilqilishuchunquyidagid asturnikeltiramiz.

2.4-dastur. To‘plamningsig‘iminitekshirish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>

using namespace std;
int main(){
```

```
vector<int> vec;
cout << "Real size of array in vector: " <<
vec.capacity() << endl;
for (int j = 0; j < 25; j++){
    vec.push_back (10);
}
cout << "Real size of array in vector: " <<
vec.capacity() << endl;
system("pause");
return 0;
}
```

2.3-dasturnatijasi. Output

```
Real size of array in vector: 0
Real size of array in vector: 28
```

vector – engko‘pishlatiladiganto‘plamvektorhisoblanadi. Buto‘plamning operator[] operatorigaegaekanligijudaqulay. Odatiymassivkabiishlatiladi. Xuddishuningdek, buoperator map, deque, string i wstring to‘plamlarigahammavjud.

Vektorningquvvatidinamikravishdao‘zgarishinitushunishmuhimdir. Odatda, multiplikativyondashuvhajminioshirishuchunishlatiladi: zarurbo‘lsa, vektoruchunajratilganxotiraningmartasoniortadi, ya‘ni, yangielementqo‘shibto‘plamsig‘iminiogoshirishgasababbo‘lsa, operatsiontizimidasturiuchunyangixotiramaydoniajratadi. Masalan, ikkibarobarkattabo‘lishuchun, eskixotiramaydonibarchaelementlarinusxasiyangiqiymatqilibqo‘shish.

STL kutubxonaalgoritmlari. STL

kutubxonasiniishlabchiquvchilarshablonlima‘lumotlartuzilmalarimajmua sigaegakutubxonayaratishdao‘zlarigaanchajiddiyvazifaqo‘yishgan. STL kutubxonasito‘plamlaribilanishlashimkoniniberadigan, mashhuralgoritmlarnioptimaltatbiqlarivakattamajmuinio‘zichigaoladi. Barchaamalgaoshirilganfunksiyalarniuchguruhgabo‘lishmumkin:

1. Barchato‘plamelementlaritanlashvaulargaishlovberishusullari :

```
scount, count_if, find, find_if, adjacent_find, for_each, mismatch, equal, search, copy, copy_backward, swap, iter_swap, swap_ranges, fill, fill_n, generate, generate_n, repl
```

```
ace,  
replace_if, transform, remove, remove_if, remove_copy  
,  
remove_copy_if, unique, unique_copy, reverse, reverse  
_copy, rotate, rotate_copy, random_shuffle, partition  
,  
stable_partition
```

Bu usularni vazifalarini ixtiyoriy manbadan, splusplus.com
vebsahifasidan foydalanib bilish mumkin.

2. Saralash usullari:

```
sort, stable_sort, partial_sort, partial_sort_copy,  
nth_element, binary_search, lower_bound, upper_bound,  
  
equal_range, merge, inplace_merge, includes, set_unio  
n,  
set_intersection, set_difference,  
set_symmetric_difference, make_heap, push_heap, pop_h  
eap, 'sort_heap, min, max, min_element, max_element,  
lexographical_compare, next_permutation,  
prev_permutation
```

Bu usularni vazifalarini ixtiyoriy manbadan, splusplus.com
vebsahifasidan foydalanib bilish mumkin.

3. To'plama'zolari ustidama'lumarifmetika mallarni bajarish usullari:

```
Accumulate, inner_product, partial_sum,  
adjacent_difference
```

Bu usularni vazifalarini ixtiyoriy manbadan, splusplus.com
vebsahifasidan foydalanib bilish mumkin.

Ushbu usularni sanabo'tishdan maqsad STL
kutubxona sitomoni dantaqdim etilgan boy vositalar to'plamini bilantirish
hdir. Qo'shimchama'lumot olish uchun,
C++ dasturlashtirishga oid sinflarning tegishli hujjatlaridan olishib chiqish mum
kin.

Predikatlar. Ko'pgina STL
kutubxona algoritmlari uchun algoritmlar to'plamining muayyan zosibilanni
maqilish kerakligini aniqlaydigan shartni o'rnatishni taqdim etadi.
Predikat - bu funksiya, bir nechta parametrlarni oladi va Boolean
qiymatini qaytaradi (true/false). Standart predikatlar to'plamini ham mavjud.

Oqim xavfsizligi. Bu STL
butunlay xavfsiz kutubxona emasligini tushunish muhim ahamiyatga ega.

Lekin bu muammoni hal qilish juda oddiy:

ikkioqimlar bir xil to'plamdan foydalanayotgan bo'lsa, Mutex
seksiyasini amalga oshirish zarur.

STL cross-platform kutubxona hisoblanadi. Albatta, ushbu kutubxona kompilyatorning har qanday versiyasi uchun mavjudligiga mutlaq kafolat yo'q. Masalan, ukamdan-
kam hollarda mobil qurilmalarda amalga oshiriladi, chunki amalga oshirilgan ma'lumotlartuzilmalarining aksariyati xotira niteja masdan, tezlik foydasini tanlaydi hamda xotira mobil platformalarda eng qimmatli texnik resursdir, kompyuterda esa u hozir juda ko'p. Shuning uchun tez-tez STL kutubxonangizni lokalizatsiyasini yaratish kerak bo'ladi, masalan, ilova sinimobil platformaga ko'chirish uchun.

Konteynersinflar. Konteynersinflar muayyan tarzda tashkil qilingan ma'lumotlarni saqlash uchun mo'ljallangan sinflar.

Turli xil tipdagi ma'lumotlarni saqlash uchun bir xil turdagi konteynerdan foydalanishingiz mumkin.

Buxususiyat sinf shablonlari yordamida amalga oshiriladi, shuning uchun C++
kutubxonasining konteynersinflarini, shuningdek algoritmlarni va iteratorlarni o'z ichiga olgan qism standart shablonlarkutubxonasi (STL) debataladi.

Ma'lumotlarkonteynerlardasaqlanadivaularbilan turli amallarkonteynerusullarivamoslanuvchan algoritmlar bilan aniqlanadivabajariladi.

Iteratorlar bu ikki elementni bir-biriga bog'lagan holda ishlaydi.

Ular tufayli har qanday algoritm har qanday konteyner bilan ishlashim mumkin.

Professional dasturlashni kutubxonasinflarisiz foydalanish nita savvur qilishim mumkin emas, shuningdek alohida konteynerlarsiz ham. Ulardan foydalanish dasturlarning ishonchliligi, joriy qilish samaradorligi, moslashuvchanligi va ko'pchilik irraliligini oshirish hamda dastur tuzish vaqtinika maytirishga imkonini beradi.

Kutubxonani yaratish ko'pchilik va mashaqqatli talab qiladi, amao,
dastur yaratish vaqtida o'zini oqlaydi.

STL

kutubxona sidasturlarni yozishda ishlatiladigan asosiy ma'lumotlartuzilmalarini amalga oshiruvchi konteynerlarni o'z ichiga oladi: vektorlar, navbatlar, ro'yxatlar, lug'atlar va to'plamlar.

Konteynerlarni ikki turga bo'lishim mumkin: ketma-
ket va assotsiativ konteynerlar.

Ketma-ketkonteynerlar.

Ularuzluksizketma-

ketlikdao'xshashmiqdorlarningcheklisoninisaqlashnita'minlaydi.

Konteynerlarsifatidavektor (vector), ikkitomonlamanavbat (deque), ro'yxat (list) vabiraloqaliro'yxati (forward_list), shuningdekkonteynervariantlarasosidaadapterlar, stek (stack), navbat (queue) vaustuvorlikbilannavbat (priority_queue) sinflarinio'zichigaoladi.

Massivhamamallarbilancheklanganholdakonteynerningyanabirturidir.

Konteynerningharbirturima'lumotlarustidao'zamallarto'plaminita'minlaydi.

Siztanlagankonteynerturidasturdagima'lumotlarbilannimaqilishnixohlas hingizgabog'liq. Masalan, agarketma-ketliko'rtasidama'lumotlartez-tezjoylashtirishvao'chirishkerakbo'lsa, ro'yxatlardanfoydalanishkerak,ma'lumotlarnioxiridayokiboshida, birinchinavbatdajoylatrishkerakbo'lsa, ikkitomonlamanavbatdanfoydalanishmaqsadgamuvofiq.

Assotsiativkonteynerlar.

Assotsiativkonteynerlarasosiyma'lumotlargaakalitlarasosidatezkormuroja atqilishnita'minlaydi. Bukonteynerlarmuvozanatlidaraxtlargaasoslangan. Assotsiativkonteynerlarningbeshturimavjud: lug'atlar (map), ko'plug'atlar (multi) (multimap), to'plamlar (set), multito'plamlar (multiset) vabitlito'plam (bitset).

Dasturchistandartkutubxonadamavjudbo'lgansinflargaasoslanibo'zkonteynersinflariniyaratishimumkin.

Bungakirishishdanoldinmuhimtushunchanibilishinggizshart, ya'ni STLkutubxonasingfundamentaltushunchasibushablondir

Barchakonteynersinflaristandartlashtirilganinterfeysibilanta'minlangan.

Turlikonteynerlaruchunbirxilamallarningma'nosibirxilbo'lishitabiiydir.

Asosiyamallarkonteynerlarningbarchaturlariuchunqo'llaniladi.

Standartesafaqatkonteynerinterfeysinibelgilaydi, shuninguchunturlixildasturlarsamaradorlikdakattafarqqilishimumkin.

Barchakonteynerlaro'zxotirasinio'ziboshqaradi, shuninguchundasturchibuhaqidao'ylashninghojatiyo'q.

Deyarliharqandaykonteynerlarningquyidagixususiyatlaribor:

2.2-jadval. Konteynerxususiyatlari.

No	Xususiyatnomi	mazmuni
1	value_type	Konteynerelementiningtipi
2	size_type	Elementlarindeksiningtipi

3	<code>iterator</code>	Iterator
4	<code>const_iterator</code>	O'zgarasiterator
5	<code>reverse_iterator</code>	Teskariiterator
6	<code>const_reverse_iterator</code>	O'zgarateskariiterator
7	<code>reference</code>	Elementgahavola
8	<code>const_reference</code>	O'zgaraselementgahavola
9	<code>key_type</code>	Kalittipi (assotsiativkonteynerlaruchun)
10	<code>key_compare</code>	Taqqoslashmezoni (assotsiativkonteynerlaruchun)

Iteratorbirelementuchunko'rsatkichgaekvivalentidir. Ularkonteynerlarnito'g'riyokiteskariyo'nalishdako'rishuchunishlatiladi. Iteratordantalabqilinadiganbarchaamallarkonteynerelementigamurojaatqilishvauningkeyingielementigao'tishamaliniamalgaoshirishdir. Konteynerlarningelementlariningqiymatlario'zgaragandao'zgarasiteratorlardanfoydalaniladi.

Iteratorlaryordamidahaqiqiyma'lumotlartiplarihaqidao'ylamasdan, konteynerelementlargaurojaatqilishuchunfoydalanishmumkin. Buninguchun, harbirkonteynerquyidagi 2.3-jadvaldakeltirilganbirnechausullardanfoydalanishnitavsiyaqilinadi. Harbirkonteyneruchunbutiplarvausullarni, ularnirealizatsiyasiniamalgaoshirishgabog'liqtarzda belgilanadi. Shuningdek, ixtiyoriykonteynerlardaularninghajmihaqidama'lumotolishuchunusullari mavjud:

`size()` – elementlarsoni;
`max_size()` – konreynerningmaksimalo'lchami (1 milliardtaelementuchun);

`empty()` – mantiqiyusuli, konteynerbo'shliginitekshiradi;
Zaruriyatbo'lgandadasturchikonteynerningmaydonlarivausullariket ma-keto'zlashtiribboraveradi. Bizhamzaruriyattug'ilgandafoydalanamiz.

2.3-jadval. Konteynersinflarningumumiyusullari.

No	Usullar	Izoh
1	<code>iterator begin()</code> <code>const_iterator begin () const</code>	Birinchielementniko'rsatadi
2	<code>iterator end()</code>	Oxirgisidankeyingielementniko'rsatadi

	<code>const_iterator end()</code> <code>const</code>	
3	<code>reverse_iterator rbegin()</code> <code>const_reverse_iterator rbegin () const</code>	Teskariketma-ketlikdabirinchielementniko‘rsatadi
4	<code>reverse_iterator rend()</code> <code>const_reverse_iterator rend () const</code>	Teskariketma-ketlikdaoxirgisidankeyingielementniko‘rsatadi

Chiziqli konteynerlar (*array*, *vector*, *deque*, *list*, *forward_list*). Standart konteynerlarni ikki taguri gachiziqli va assotsiativ konteynerlarga bo‘lishi bilamiz.

O‘z navbatida chiziqli konteynerlarni bog‘langan ro‘yxat (*forward_list* va *list*) va tasodifiy kirish konteynerlariga (ekrin konteyner deb ham yuritiladi) (*deque*, *vector* va *array*) bo‘linadi.

Assotsiativ konteynerlar quyidagi variantlarning kombinatsiyasi bo‘lgan sakkiz ta konteyner bilan ifodalanadi (tegishli standart sinflar nomlarining qavslari chida beriladi): to‘plam (**set*) yoki lug‘at (**map*), elementlarni takrorlashga imkon beruvchi (**multi**) yoki ruxsat bermaydigan, tartibga solingan solinmagan (*unordered**).

Barcha konteynerlar `iterator` va `const_iterator` tiplari o‘z ichiga olish uchun o‘z navbatida o‘qish-yozish va faqat o‘qish iteratorlarini aniqlaydi.

Konteyner tarkibida olgan iteratorlar oralig‘ini `begin()` va `end()` (o‘qish uchun iterator), shuningdek `cbegin()` va `cend()` (o‘qish uchun `const_iterator`) funksiyalari yordamida olish mumkin. Konteynerlarni kitomona iteratorlar bilan ham teskari o‘tish uchun `rbegin()`, `rend()`, `crbegin()` va `crend()` funksiyalar mavjud.

Barcha konteynerlarni `empty` funksiyasi tomonidan bo‘shchun tekshiri sh mumkin. Qachonki `cont.begin() == cont.end()` bo‘lsa, o‘zgarmas konteyner bo‘sh hisoblanadi. `end()` konteynerning oxirgi elementi keyingisi har qanday elementga havola qiluvchi iteratorni qaytaradi. Elementlarni `size()` funksiyasi yordamida olish mumkin (*forward_list* bundan mustasno). Konteyner tarkibidagi funksiyalarni tozalash uchun `clear()` funksiyasi dan foydalaniladi (*array* bundan mustasno).

Iteratorda begin(), end(), cbegin(), cend(), rbegin(), rend(), crbegin() va crend() erkinfunksiyalariningshablonlarinianiqlashmumkin. Bularbirxilnomlifunksiyalargahavolaqilinadi, shuningdek, bundayaniqlanganfunksiyalarstatikmassivva std::valarray uchunaniqlandi. for(:) takrorlanishoperatorixuddishandayshakligatayanadi. Masalan, cr birkonteynerbo'lsin (shuningdek, statikmassivhambo'lishimumkin), shundadasturda for() fragmentiquyidagichaishlatiladi.

```
for (T x : cr) work(x);
```

Semantikjihatdanquyidagifragmentgateng

```
{
    using std::begin;
    using std::end;
    const auto e = end(cr);
    for (auto p = begin(cr); p != e; ++p)
    { T x = *p; work(x); }
}
```

BuyerdaTtipikonteynerelementlariningtipigamoskelishishartemasva autoasosidakonstruksiya yoki havola litip hambo'lishimumkin (engko'pishlatiladiganvariantlarauto&vaauto&&).

Chiziqlikonteynerlarni (array danboshqa) belgilanganqiymatlardan assign() funksiyasinichaqiribto'ldirishmumkin (eskiqiymalario'chiriladi) va resize() funksiyasibilano'lchamlarini (hajmikamaygandaelementlaroxiridano'chiriladivahajmiortganda, yangielementlaroxiridanqo'shiladi) aniqlashmumkin.

Konteynerningbirinchielementigato'g'ridan-to'g'rimurojaat (tartiblanmaganassotsiativkonteynerlardantashqari) front()funksiyayordamidaamalgaoshiriladi.

Iteratorlariikkitomonlamabo'lganbarchakonteynerlarningoxirgielementg amurojaatuchunback()funksiyasi, shuningdek, teskariyo'naltirilganiteratorlarreverse_iterator vaconst_reverse_iterator funksiyalaridanfoydalaniladi.

Tegishliintervallarnirbegin(), rend() i crbegin(), crend()funksiyalariyor damidaolishmumkin.

Barchakonteynerlarnitenglikvatengsizlikuchuntaqqoslashmumkinva ularningmazmunini swap() funksiyasiyordamidaalmashtirishmumkin. Chegaralanmaganassotsiativlardantashqaribarchakonteynerlarni<, <= , >va>= operatorlarileksikografikjihatdantaqqoslashmumkin.

Standartkonteynerlardadinamikxotiraniboshqarishuchun Allocator
ajratuvchimaxsussinflardanfoydalanadi. Allocator
xotiraniboshqarishningminimalbirliginibelgilaydiganvabirqatoryordamc
hita‘riflarnitaqdimetadiganelementtipigabog‘liqdir.

Buvazifato‘rtasosiyfunksiyalariyordamidaamalgaoshiriladi:
elementlariberilganqatoruchunxotiraajratishdaallocate,
xotiranitozalashuchundeallocate, konstruktorqurishuchunconstruct,
vadestruktoruchundestroy.

Allokatorboshqatipdagielementlaruchunanalogallokatorolishda
rebind metafunksiyasiniamalgaoshirishikerak.

Alloc – elementlarninganiqtiplariuchunallokatorberilganbo‘lsin. U
tiplielementlaruchununingallokatoriniolishvariantiquyidagidasturfragme
ntidakeltirilgan.

```
using AllocForU = typename Alloc::template  
rebind<U>::other;
```

C++dastandartkutubxona (<memory>sarlavhafayl bilan)
new/new[] va delete/delete[] operatorlariniqo‘llashorqali allocator<T
>nita‘minlaydi.

Bundanmodelvao‘zingizningallokatorlaringgizniyozishdafoydalanish
mumkin.

Biraloqaliro‘yxat<forward_list>. forward_list<T, a =
allocator<T>>birtomonlamaiteratororqali T
tipdagielementlarga kirishnita‘minlaydi.

Ro‘yxatningtugunlariniyaratishuchunA::rebindorqaliyaratilganallokator
dayfoydalaniladi. Biraloqaliro‘yxatningo‘zigaxosligishundaki,
faqatbelgilanganjoriytugundankeyinelementlarniqo‘shishvao‘chirishmu
mkin:

insert_after - elementqo‘shish.

emplace_after – yangielementyaratish.

Bundabelgilanganparametrlaruchunkonstruktorchaqiriladi.

erase_after - elemento‘chirish.

Birinchielementdanoldindeganko‘shimchaaniqlanganpozitsiyamavj
ud. Bubebefore_begin va cbefore_begin (const_iterator
qaytaruvchilaruchunvariant) funksiyalariyordamidabajariladi.
Shuningdek, fl.push_front(item) funksiyasidanfoydalanib,
elementlarnioldindanjoylashtirishmumkin (yuqoridagigaekvivalent),
masalan, quyidagichadasturfragmenti:

```
fl.insert_after(fl.before_begin(), item)
```

Konstruktorgaharqandayparametrlarni joylashtirish uchun
fl.emplace_front(...) funksiyasi ishlatiladi.
Bunga ekvivalent sifatida quyidagi dastur fragmentini yozish mumkin:

```
fl.emplace_after(fl.before_begin(), ...)
```

pop_front funksiyasi ro'yxatdan birinchi elementni o'chiradi.

C++

standart kutubxona sidaro'yxat konteynerlar xususiyati uchun samarali bo'lgan amallarni bajarishga faqat iteratorlarda foydalanishga ishonchsizligi yuqori darajali amallarni qo'llab-quvvatlash hisoblanadi:

merge - ikki tartiblangan ro'yxatni bir-biriga birlashtiradi, elementlarni o'chirilmaydilekino'ngdancha pro'yxatga o'tiladi

splice_after

berilgan ro'yxatni ko'rsatilgan elementdan keyin joylashtiradi.

remove

berilgan elementni o'chirib tashlab barcha elementlarni o'chiradi.

remove - berilgan predikatasosidabarcha elementlarni o'chiradi.

reverse - elementlarni tartibiga murosaj qiladi.

unique - barcha ketma-ket dublikatlarni o'chiradi.

sort - ro'yxatni joyida tartiblaydi.

Umuman olganda,

bu funksiyalar qandaydir standart algoritmlarga o'xshaydi, lekin juda tez va qulay ishlaydi.

Ularga murojaat qilish uchun umumiy ruqsat olish kerak, masalan, sort() funksiyasi uchun std::sort(from, to). Ammo, tasodifiy kirish iteratorlari kabitalab qilingan ro'yxatlar uchun amal qilmaydi.

Ikkialoqli ro'yxat (ikkilangan ro'yxat) <list>. list<T, allocator<T>> ikki tomonlama iterator orqali tipdagi elementlarga kirishni ta'minlaydi.

Bir aloqli ro'yxatdan farqli o'laroq,

elementlarni belgilangan pozitsiyadan oldin joylashtirildi

(insert, emplace, splice funksiyalari),

oxiridan qo'shish (push_back, emplace_back, pop_back),

va iterator ko'rsatayotgan elementni o'chirish

(erase)

funksiyalar bilan amalga oshiriladi. *_after funksiyasi mavjud emas.

Ikkimonlamali <deque>. deque<T, allocator<T>>, maxsus kirish iterator orqali elementlar uchun kirish imkonini beradi.

Faqat ro'yxat kabi,

usamarali qo'shish va har ikki tomonidan ma'lumotlarni olib tashlash imkonini beradi.

eradi. Indeks gakerish uchun kitafunksiya ishlatiladi: []
operatori va at (indeks). Birinchisidan farqli o'laroq,
ikkinchisi indeksni tekshiradi va qiymati haqiqiy bo'lmasa,
out_of_range istisnoga murojaat qiladi.

Ikki tomonlama navbat konteyneri uchun ro'yxatni bir xil tarzda har qanda
yholatda elementlarni kiritish va o'chirish imkonini beradi.
Lekin ikki tomonlama navbatdab u amallar arkonteyner hajmi,
vaqtichiziqlib o'lishini talab qilish imkonin. Bundantashqari,
oldindan iterator gasaqlanganelementlarni tartibini kiritish va o'chirishni buz
ish imkoninshuning uchun xotirada saqlanganelementlarning ko'rsatkichlar
ini yoddasaqlash maqsadga muvofiq
(Agar iteratorlar ro'yxat kabi saqlanayotgan bo'lsa,
elementlar gabo'lganko'rsatkichlar o'chirilmaydi).

Dinamik massiv (vektor) <vector>. Vektor <T, a = allocator
<T>> elementlar gamaxsus kirish iteratori orqali kirishni ta'minlaydi. Deque
dan farqli o'laroq,
ukiritish va boshida elementlarni o'chirish uchun ruxsat bermaydi.

Dinamik massiv xotiraning uzluksiz bo'limidaki tma-
ketsaqlanganelementlarning joylashishini kafolatlaydi, uning manzili data
funksiyasi bilan qaytariladi.

Elementlarni qo'shish va oldindan ajratilgan saqlashni tugatishda elementlar
ko'chiriladigan joyga kattahajmdagi yangi dinamik qator ajratish imkonin.
Eski saqlabo'chiriladi, vabarchasaqlangan iteratorlaryo'qoladi,
o'chirilgan ob'ektlar uchun ko'rsatkich bo'lib qoladi.

Dinamik massiv reserve funksiyasi yordamida yetarli hajmda saqlash uchun
oldindan massivni tayyorlash imkonini beradi
(uko'chibo'tishda elementlarni olib kelish imkonin).

Saqlash hajmini capacity funksiyasi yordamida bilib olish imkonin.
shrink_to_fit funksiyasi haqiqiy ishlatiladigan hajmini ajratadi va elementla
rni foydalanilmagan xotiradan olib tashlaydi.

Statik massiv <array>. array <T,
N> iteratorning tasodifiy kirish orqali elementlar gakerish imkonini beradi va
statik array T[N] to'plamini hisoblanadi, kerakli manzili data
funksiyasi tomonidan olinish imkonin. T[N] massivdan farqliravishda
array <T, N> - qiymatlariko'chirilishi
(qiymat bo'yicha funksiyalar gao'tishi)
imkonin bo'lgan va qiymatlari avtomatik ravishda ko'rsatkichlarga aylantiril
maydiganto'la huquqlitidir. array <T,
N> dan tayanchsinfi sifatida foydalanish imkonin (lekin ehtiyo'tkorlik bilan,

faqatharqandaykonteynerkabi,
 standartkonteynerlarbumaqsaduchunmo‘ljallanganemas, chunki,
 xususan, virtual funksiyalarnio‘zichigaolmaydi).
 deque va vectorkabibirxiltarzdaindeksitomonidanelementlarga murojaat
 qilishmumkin. Elementlarsoninio‘zgartirishmumkinemas, shuninguchun
 array
 elementlarnikiritishyokio‘chirishuchunhechqandayfunksiyalaraniqlanma
 ydi. fillfunksiyasimassivniqiymatnuxsalaribilanto‘ldiradi.

2.4-jadval. Ketma-ketkonteynerlaruchunasosiyfunksiyalar

Headers		<array>	<vector>	<deque>	<forward_list>	<list>
Members		array	vector	deque	forward_list	list
	<i>constructor</i>	<i>implicit</i>	<u>vector</u>	deque	forward_list	list
	<i>destructor</i>	<i>implicit</i>	~vector	~deque	~forward_list	~list
	operator=	<i>implicit</i>	operator=	operator=	operator=	operator=
iterators	begin	begin	begin	begin	begin before_begin	begin
	end	end	end	end	end	end
	rbegin	rbegin	rbegin	rbegin		rbegin
	rend	rend	rend	rend		rend
const iterators	cbegin	cbegin	cbegin	cbegin	cbegin cbefore_begin	cbegin
	cend	cend	cend	cend	cend	cend
	crbegin	crbegin	crbegin	crbegin		crbegin
	crend	crend	crend	crend		crend
capacity	size	size	size	size		size
	max_size	max_size	max_size	max_size	max_size	max_size
	empty	empty	empty	empty	empty	empty
	resize		resize	resize	resize	resize
	shrink_to_		shrink_to_	shrink_to_		

	fit		fit	fit		
	capacity		capacity			
	reserve		reserve			
element access	front	front	front	front	front	front
	back	back	back	back		back
	operator[]	operator[]	operator[]	operator[]		
	at	at	at	at		
modifiers	assign		assign	assign	assign	assign
	emplace		emplace	emplace	emplace_after	emplace
	insert		insert	insert	insert_after	insert
	erase		erase	erase	erase_after	erase
	emplace_back		emplace_back	emplace_back		emplace_back
	push_back		push_back	push_back		push_back
	pop_back		pop_back	pop_back		pop_back
	emplace_front			emplace_front	emplace_front	emplace_front
	push_front			push_front	push_front	push_front
	pop_front			pop_front	pop_front	pop_front
	clear		clear	clear	clear	clear
	swap	swap	swap	swap	swap	swap
list operations	splice				splice_after	splice
	remove				remove	remove
	remove_if				remove_if	remove_if
	unique				unique	unique
	merge				merge	merge
	sort				sort	sort
	reverse				reverse	reverse
observers	get_allocator		get_allocator	get_allocator	get_allocator	get_allocator
	data	data	data			

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. C++

- dasturlashtilida eng samarali qaysi kutubxona sihisoblanadi va nima uchun?
2. Kalitlarning qiymati bo'yicha tartiblangan qanday to'plamlarni bilasiz?
3. Bir va ikki baytli belgilarni to'plamini ma'noni bildirgan va ularning formatlari bo'yicha nimalarni bilasiz?
4. Iteratorlarni qanday to'plam va nima uchun?
5. Har biraniq STL
sinfi uchun iteratorlarni to'plamini qanday aniqlanadi va turlarini nech
hata? Har bir turini tushuntirib bering?
6. STL kutubxona si to'plamlari bilan ishlashim ko'ni beradigan,
mashhur algoritmlarni optimal tatbiqlarini vakattamajmu'ini o'z ichiga ol
adi. Bular algoritmlar nechta guruhga bo'linadi va qaysilar?
7. Konteynerlarni farqlaydigan sinflar sihisoblanadi?
8. Konteynerlarni nechta turiga bo'lishim mumkin va qaysilar?
9. Har qanday konteynerlarning bo'lishi shart bo'lgan xususiyatlarni sanab
boring?
10. Ixtiyoriy konteynerlarni daularning hajmi haqidagi ma'lumotni olish uch
un qanday usullar mavjud?
11. Iteratorlarda begin(), end(), cbegin(), cend(), rbegin(), rend(),
crbegin() va crend() erkin funksiyalarini nimalarini aniqlashim mumkin?
12. Chiziqli konteynerlarni belgilangan qiymatlardan qaysi funksiya
siniga qirib to'ldirishim mumkin?
13. Barcha konteynerlarni tenglik va tengsizlik uchun taqqoslashim mu
mkin va ularning mazmunini qanday funksiyalar yordamida almashtirish
mumkin?
14. Allocator nima uchun ishlatiladi?
15. Allocator
xotirani boshqarishning minimal birligini belgilaydigan va bir qator yor
damchilarni taqdim etadigan element tipiga bog'liqdir.

Buvazifanechtavaqandayasosiyfunksiyalariyordamidaamalgaoshiriladi?

16. Biraloqaliro'yxatningnimalariniyaratishuchunA::rebindorqali yaratilganallokatordayfoydalaniladi?
17. Qaysi funksiyasidanfoydalanib, elementlarnioldindandjoylashtirishmumkin.
18. Ikkialoqaliro'yxatvaikkitemonlamanavbatningfarqlarinisan abbering?
19. Dinamikvastatikmassivlarningo'xshashtomonlarinimisollar yordamidatushuntiribbering?
20. Staiikmassivelementlarnikiritishyokio'chirishuchunqandayfunksiyalaraniqlangan.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	STL kutubxonasiga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajar.  Bundagi dasturda <code>zio</code> zgartirishlarni topish orqali topshiriqlar bosqichma – bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>#include <iostream> #include <vector> using namespace std; int main(){ vector v; int i;</pre>	1. Dasturda STL kutubxonasining nechasi fivauning funksiyalar ishlatilgan. _____ _____ _____
	2. Dasturdagi STL sinfni o'zgartirib dasturni qayta dastur tuzing. Quyidagi qismlar o'zgartirib qaytda qiling. _____ _____ _____

<pre> cout << v.size() << endl; for(i=0; i<10; i++) v.push_back(i) ; cout << v.size() << endl; for(i=0; i<10; i++) cout << v[i] << " "; cout << endl; cout << v.front() << endl; cout << v.back() << endl; vector<int >::iterator p = v.begin(); while (p != v.end()) { cout << *p << " "; p++; } </pre>	3. Dasturdavektorelementlarigaxaqiqiytipliqimat larniberishuchundasturfragmentaniqyozing. 4. cout << v.front() << endl;fragmentgaalmatirishvariantlarinianiqlangvada sturgayozib, tekshiribko‘ring. 5. vector<int>::iterator p = v.begin();dasturningfragmentidanima o‘zaksinit opgan.

<pre> system("pa use"); return 0; } </pre>	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

IKKINChIASSIMENTTOPShIRIG‘I	
	Konteynersinflargaoidberilganquyidagidasturbo‘yichaberilganto pshiriqlarkeraklifragmentlariasosidabajaring.  Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarb osqichma – bosqichamalganoshriladi.
dastur	topshiriqlar
<pre> #include "stdafx.h" #include <iostream> int main() { std::vector<int> myVector; for (int count=0; count < 5; ++count) myVector.push_bac k(count); std::vector<int>: </pre>	1. Dasturdagiengkattaxonitoping. _____ _____ _____
	2. Dasturdaaniqlangan myVector, myList, mySet konteynerlarningelementlaribirxilbo‘lishiuchun keraklidasturfragmentlariniyozing. _____ _____ _____ _____
	3. itV qandaykonteynervanimauchun. _____ _____ _____

<pre> :const_iterator itV; itV = myVector.begin(); while (itV != myVector.end()) { std::cout << *itV << " "; ++itV; } std::cout << '\n'; std::list<int > myList; for (int count=0; count < 5; ++count) myList.push_back(count); std::list<int>::c onst_iterator itL; itL = myList.begin(); while (itL != myList.end()) { std::cout << *itL << " "; ++itL; } </pre>	4. Dasturdadasturchiningyanabiroddiyammo ko'rolxatosinitoping. 5. Dasturdagibarchakonteynerlarniyarimele mentlarinichiqarishdasturfragmentlarinituzing.

```

        std::cout <<
'\n';

        std::set<int>
mySet;

mySet.insert(28);

mySet.insert(01);

mySet.insert(80);

mySet.insert(9);

mySet.insert(3);

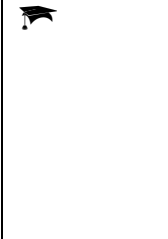

std::set<int>::co
nst_iterator itS;
    itS =
mySet.begin();
    while (itS !=
mySet.end())
    {
        std::cout
<< *itS << " ";
        ++itS;
    }

    std::cout <<
'\n';
    system("pause
");
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

UChINChIASSISMENTTOPShIRIG'I



Ketma-
ketkonteynerlardan foydalanishga oid berilgan quyidagi dasturbo'y icha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani.
👉 Bunda dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma – bosqich amalga oshiriladi.

dastur	topshiriqlar
<pre>#include "stdafx.h" #include <iostream></pre>	1. Dasturdagi yangi tiplarni aniqlang.
<pre>#include <vector> #include <cstring> #include <string> using namespace std; struct student { char fio[80]; short age; student(const char* fio, short age) : age(age) { strncpy_s(this ->fio, fio,</pre>	2. Dasturdagi ketma-ket konteynerlardasturfragmentini yozing.
<pre>strncpy_s(this ->fio, fio,</pre>	3. Dasturdagi keraksiz bo'lgan dasturfragmentlarini o'chirib tashlang

```
sizeof(this->fio));
    }
};

inline
ostream&
operator
<<(ostream&
out, const
student& obj)
{
    return out <<
obj.fio <<
"\t: " <<
obj.age;
}
struct
trimatrix :
vector<
vector<double>
> {
    trimatrix(int
n) {
        while (n > 0)
        {

vector<double>
row(n);
        for (int i =
0; i < n; i++)
            row[i] = n -
i;

push_back(row)
;
            n--;
        }
    }
};
```

4. vector konteyneriniboshqaketmaketkonteynergaalmatirishchun kerakli dastur fragmentlari yozing.

[illegible]

5. Dasturdagiketma-
ketkonteynerlaruchunasosiyfunksiyalarniyozing.

```

    }
};
inline
ostream&
operator
<<(ostream&
out, const
trimatrix&
obj) {
    for (int i =
0; i <
obj.size();
i++) {
        for (int j =
0; j <
obj[i].size();
j++)
            cout <<
obj[i][j] << "
";
        cout << endl;
    }
    return out;
}
int main(int
argc, char
**argv) {

    setlocale(LC_A
LL, "rus");

    vector<student
> group;

    group.push_bac
k(student("Bol
tavev B.B.",

```

```

18));

group.push_back(student("Aminov A.A.",
21));

group.push_back(student("Karimov A.Sh.",
19));

    for (int i =
0; i <
group.size();
i++)
    cout <<
group[i] <<
endl;

    cout <<
trimatrix(4);

    int k, n;
    vector<bool>
a(n = rand() +
1, true);
    a[0] = a[1] =
false;
    for (k = 2; k
< n; k++)
        for
(vector<bool>:
:iterator j =
a.begin() + k
+ k;
            j <
a.end()); j +=

```

```

k)
    *j =
false;
    const int
line = 10;
    k = 0;
    for
(vector<bool>:
iterator j =
a.begin(); j <
a.end(); j++)
    if (*j) {
        cout << j -
a.begin() <<
"\t";
        if (0 == ++k
% line) cout
<< endl;
    }
    if (k % line
!= 0) cout <<
endl;

system("pause"
);
    return 0;
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

1.3. Assosiativvatartiblanmaganassosiativkonteynerlar

 Assosiativ konteynerlarningasosiyfoydalanishko‘nikmalari, assosiativkonteynerlar (set, map, multiset, multimap) vatartiblanmaganassosiativkonteynerlar (unordered_set, unordered_map, unordered_multiset, unordered_multimap) bilanyaqindantanishib, ularningxususiyatlar, funksiyalari, sinflari, yaratishvafoydalanishuslublari, amallari, talablari, vazifalarivausullarikeltirilganbo‘lib,

dasturchining to'plamlar bilan ishlashdagitanlashim koniyati bo'yicha tajrib axulosalari, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan. Shuningdek,

assosiativ konteynerlarda qidirish ustunligi va unidastur data qqoslash bo'yicha hamiosllar keltirilgan. Bilimlarni mustahkamlash uchun 32 tanazariy savol va amaliy ko'nikma hamda malakalarni rivojlantirish uchun 4 ta assisment topshiriq'ivahar bir assismentda 10(12) ta topshiriq, jami 44 ta topshiriq berilgan.

Yuqoridagi topshiriqlarni bajarish mavzu mustahkamlash uchun xizmat qiladi.

✍️ Kalit so'zlar. to'plam, shablon, map, set, multimap, multiset, unordered_set, unordered_map, unordered_multiset, unordered_multimap, satrli oqim, iterator, dinamik ma'lumotlarni tuzilmasi, pridekat, konteyner sinflari, allocator, equal_range, operator==, operator=, tartiblangan assosiativ konteyner, tartiblangan assosiativ konteyner, unikalkalit, xesh funksiya, xeshlash, konstruktor, modifikator, qidirish usullari, pair, try - catch funksiyasi.

📑 Bilish shart bo'lgan tushunchalar. Tiptushunchasi, sinf va sinf ob'ekti, to'plam, statik va dinamik massiv, element gamurojaat va ko'rsatkich, funksiyalar va ko'rsatkich, konteyner, ketma-ket konteynerlar, xesh, tasodifiy qiymat, istisno holat, ma'lumotlarni kirish va chiqishi, oqim, dasturlashga oid dastlabki tushunchalar va C++ tilini o'qib o'tirish va talovchilarni hitda ishlashni bilish lozim.

🔗 Bilib olasiz.

Assosiativ konteynerlarning asosiy foydalanish ko'nikmalari, assosiativ konteynerlar (set, map, multiset, multimap), tartiblangan assosiativ konteynerlar (unordered_set, unordered_map, unordered_multiset, unordered_multimap) bilan yaqindan tanishib, ularning xususiyatlar, funksiyalari, sinflari, yaratish va foydalanish usullari, amallari, talablari, vazifalarini va usullarini, dasturchining assosiativ to'plamlar bilan ishlashdagitanlashim koniyati bo'yicha tajrib axulosalari, xeshlash, tartiblangan va tartiblangan konteynerlarning sinflari xususiyatlari, usullarini, dasturlashdagio'rin va allocator orqali yaratish va foydalanish usullarini o'rganishingiz mumkin.

REJA

1. Assosiativ konteynerlar (set, map, multiset, multimap).

2. Tartiblanmagan assosiativ konteynerlar (*unordered_set*, *unordered_map*, *unordered_multiset*, *unordered_multimap*).

KIRISH

Konteynerlar vaketma-ket konteynerlar bo'yicha nazariy va amaliy imkoniyatlarini bilamiz. Shuning uchun konteynerlarning ikkinchi turidagi assosiativ konteynerlar ni boshqa konteynerlar farqlib o'lishi kerak. Assosiativ konteynerlar kalitasidama'lumotlarni to'plam dantezizlab topi shunchun ishlatiladi. Tartiblangan konteynerlar muvozanatlashgan binardaraxt ustiga quriladigan at'iy tartibga tayanadi ("kichik" amali [<] operatori asosida hisoblangan). Ikkita element bir-biridan kichik bo'lmasa, ekvivalent hisoblanadi.

Buning kutubxonasi 4 ta asosiy set (to'plam), *multiset* (multito'plam, ko'pto'plam), *map* (lug'at) va *multimap* (mulilug'at, ko'plug'at) assosiativ konteynerlar bilan ishlashga qaratilgan. Bu konteynerlar *Key* kalitini o'zlariga asosiy parametrlar sifatida o'ladigan *Compare* munosabat bo'yicha tartiblaydi.

Tartiblash *Key* parametrlar bilan amal shoshiriladi. Shuningdek, *map* va *multimap* erkin tipida *Key* bilan assosiativlanadigan (sharikdir). *Compare* ob'ekti ning tipiga qo'slanilgan konteyner ob'ekti (*comparison object*) deb atiladi.

Shuning uchun bu konteynerlarning umumiy xususiyati va amallaridan boshlaymiz. Barcha assosiativ konteynerlar quyidagi amallarni qo'llab-quvvatlaydi:

count — elementlar sonini qaytaradi, belgilangan kalit bo'yicha elementlar sonini qaytaradi;

find — element gacha tekshirib o'tilgan iterator ni qaytaradi, agar bunday bo'lmasa *end()* funksiyasini vazifasini bajaradi.

equal_range - berilgan intervaldagi barcha elementlar uchun iteratorlar juftligini qaytaradi.

Kalitlarning tenglik xaqida munosabatlarning ekvivalentligi Kalitlar uchun shartli taqqoslash (*not operator*) tahlil qilamiz.

key1 va *key2* kalitlari o'zaro teng hisoblanadi, agar taqqoslanadigan ob'ektlar uchun *comp* chiniqi matqaytarsa, ya'ni:

$\text{comp}(k1, k2) == \text{false} \ \&\& \ \text{comp}(k2, k1) == \text{false}$

Assosiativ konteynerlar har qanday kalit qiymat uchun bitta eng katta elementning qiymatini saqlasa, unikal (takrorlanmaydigan) kalitlarni (*unique keys*) qo'llab-quvvatlaydi. Akshod aularteng qiymatlik kalitlarni (*equal keys*) ham qo'llab-quvvatlaydi.

set va *map* konteynerlarunikalkalitlarni,
multiset va *multimap* konteynerlartengqiymatlikalitlarbilanishlashgamo‘l
jallangan.

set va *multiset* konteynerlaruchunaniqlanadigantipvakalittipibirxilb
o‘ladi.

map va *multimap* konteynerlaruchunpair<const Key,
T>jufligibo‘ladi.

Assosiativkonteynerlarningiteratorlariikkito monlama iteratorlarsinfi
gakiradi. *Insert* konteynerhavolalarivaiteratorlarninginkorqilmaydi.
Elementlarnio‘chirishda *erase* faqat iteratorvahavolalarniinkorqiladi.

3.1-jadvalda assosiativkonteynerlarningtalablarikeltirilgan
(konteynerlarga o‘shimacha). Undaxbu assosiativsinf, a – X
ningqiymati, Agar X unikalkalitniko‘llabquvvatlasa, a_{uniq} – X
ningqiymati, Agar X birnechtakalitniko‘llabquvvatlasa, a_{eq} – X
ningqiymati, i va j –
iteratorning kirishtalablariniqanoatlantiruvchi value_type elementniko‘r
satadi (beradi), $[i, j)$ - interval, p – a uchunruxsatberilganiterator, q – a
uchuno‘zgartiriladiganiterator, $[q1, q2)$ - a daru xsatberilganinterval, t -
 $X::\text{value\_type}$ ningqiymati, k - $X::\text{key\_type}$ ningqiymati.

3.1-jadvalda assosiativkonteynerlarningtalablari.

Ifoda	Qaytariladigantipi	tasdiq/ oldingiyokikeyingiholatigaizo x	murakkabligi
$X::\text{key\_type}$	Key	.	Kompilyatsiyavaq ti
$X::\text{key\_compare}$	Compare	less<key_type>joriyholat	Kompilyatsiyavaq ti
$X::\text{value\_compare}$	binarpredik atturi	xuddi, key_compare uchun set va mu ltiset; munosabatitartiblanganjuftlik, map va multimap uchunchaqirilganbirinchielem entdek (Key)	Kompilyatsiyavaq ti
$X(c)$	.	Bo‘shkonteyneryaratish;	Doimiy

X a(c);		taqqoslashob'ektisifatidafoydalanishuchun.	
X() X a;	.	Bo'shkonteyneryaratish; taqqoslashob'ektisifatidaCompare() danfoydalanishuchun	Doimiy
X(i,j,c) X a(i,j,c);	.	Bo'shkonteyneryaratish [i, j)intervaldanqiymatqo'shish; taqqoslashob'ektisifatidafoydalanishuchun..	NlogN (N - idan jgacha); chiziqli, agar [i, j) value_comp()bilansaralanganbo'lsa
X(i,j) X a(i,j);	.	Bo'shkonteyneryaratish [i, j)intervaldanqiymatqo'shish; taqqoslashob'ektisifatidaCompare() danfoydalanishuchun	NlogN (N - idan jgacha); chiziqli, agar [i, j) value_comp()bilansaralanganbo'lsa
a.key_comp()	X::key_compare	Taqqoslashob'ektiniqaytaradi	doimiy
a.value_compare()	X::value_compare	Taqqoslashob'ektiuchunyaratilgan value_compareob'ektniqaytaradi	doimiy
a_unique.insert(t)	pair<iterator, bool>	Agarkonteynerdatkalitlielementbo'lmasa, tniqo'shish, juftlikdakomponentning boolqiymatqo'shishbajarilganligini, iteratorkalitlielementniko'rsatadi	logarifmik
a_eq.insert(t)	iterator	tniqo'shadi va iteratorniqaytaradi, qo'shishilganelementniko'rsatadi.	logarifmik

a.insert(p, t)	iterator	tniqo‘shish, agar faqat vafaqat unikalitlik konteynerda kalitga teng kalitli element bo‘lmasa; hardim nusxalar bilan konteynerlarga tniqo‘shish. hardim tkalitli elementni ko‘rsatuvchi iteratorni qaytaradi. p iterator – qo‘shishqa erdan boshlanishini ko‘rsatuvchi izoh	umuman logarifmik, lekin domiyga intiladi, agar t to‘g‘ri p niyonga qo‘yilgan bo‘lsa.
a.insert(i, j)	Natija ishlatilmaydi	[i, j) intervaldankonteynerga elementlarni qo‘shadi;	Umuman, $N \log(\text{size}() + N)$ (N – idan j gacha); chiziqli, agar [i, j) value_comp() bilan kelishgan holda saralangan bo‘lsa
a.erase(k)	size_type	konteynerda k kalitga teng bo‘lgan barcha elementlarni o‘chiradi. o‘chirilgan elementlarning soni niqaytaradi.	$\log(\text{size}()) + \text{count}(k)$
a.erase(q)	Natija ishlatilmaydi	q bilan ko‘rsatilgan elementni o‘chiradi	domiyga intiladi
a.erase(q1, q2)	Natija ishlatilmaydi	[q1, q2) intervaldagi barcha elementlarni o‘chiradi.	$\log(\text{size}()) + N$, N - q1 dan q2 gacha.
a.find(k)	iterator; const_iterator – auchuno‘zgarimas	k kalitli elementni ko‘rsatuvchi iterator yoki agar bunday element topilmasa a.end() qaytaradi	logarifmik
a.count(k)	size_type	k kalitli elementlarning soni niqaytaradi.	$\log(\text{size}()) + \text{count}(k)$

a.lower_bound(k)	iterator; const_iterator auchun o'zgarmas	k kalitli elementdankichikbo'lmagan birinchi elementni ko'rsatuvchi iteratorni qaytaradi	logarifmik
a.upper_bound(k)	iterator; const_iterator auchun o'zgarmas	k kalitli elementdankatta bo'lgan birinchi elementni ko'rsatuvchi iteratorni qaytaradi	logarifmik
a.equal_range(k)	pair<iterator, iterator>; pair<const_iterator, const_iterator> auchun o'zgarmas	make_pair(lower_bound(k), upper_bound(k)) ga ekvivalent	logarifmik

Assotsiativ konteyner iteratorlarining asosiy xususiyatishundaki, ular konteynerlar orqali o'sis tartibidagi kalitlartartibida iteratsiyalanadi, buyerdan o'sis tartibi ularni yaratish uchun ishlatilgan taqqoslash bilan belgilanadi.

Ikki iaixtiyoriy o'zguruvchan I va j iteratorlar uchun I dan j gacha masofa musbat bo'ladi, ya'ni $value\_comp(*j, *i) == false$. Unik kalitli assotsiativ konteynerlar uchun kuchli roq $value\_comp(*i, *j) == true$ holat saqlanib turibdi.

Barcha tartiblangan konteynerlar ikki tomonlama iteratorlari yordamida elementlarga murojaat qilish imkonini beradi.

Kalita sosida qidirishdaberilgan kichik bo'lmagan birinchi elementni `lower_bound()` funksiyasi va birinchi kattaelementni `upper_bound()` funksiyasi yordamida amalga oshiriladi (3.1-jadvalga qarang). Agar `myMap` konteyner berilgan bo'lsa, unda `myMap.equal_range(key)` funksiyasi bilan ekvivalent bo'ladi (3.1-dasturga qarang).

3.1-dastur. Tartiblangan konteynerlar funksiyalari.

```
// created by Mbbahodir
#include "stdafx.h"

#include <iostream>
```

```

#include <map>

using namespace std;
int main ()
{
    map<char,int> mymap;
    map<char,int>::iterator itlow,itup;

    pair<map<char,int>::iterator,map<char,int>::iterator>
    ret, make_ret;

    mymap ['a']=101;    mymap ['c']=303;    mymap
    ['f']=404;    mymap ['b']=202;

    cout << mymap.count('c') << endl;

    cout << mymap.find('c')->second << endl;

    ret = mymap.equal_range('b');

    char key = 'b';
    make_ret =
    make_pair(mymap.lower_bound(key),mymap.upper_bound(key));

    cout << "Kichik bo'lmagan element: ";
    cout << ret.first->first << " => " << ret.first-
    >second << '\t';
    cout << make_ret.first->first << " => " <<
    make_ret.first->second << '\t';
    itlow = mymap.lower_bound ('b');
    cout << itlow->first << " => " << itlow->second <<
    endl;

    cout << "katta bo'lgan birinchi element: ";
    cout << ret.second->first << " => " << ret.second-
    >second << '\t';
    cout << make_ret.second->first << " => " <<

```

```

make_ret.second->second << '\t';
    itup = mymap.upper_bound ('b');
    cout << itup->first << " => " << itup->second <<
endl;

system("pause");
return 0;
}

```

3.1-dastur. Output
1 303 Kichik bo'lmagan element: b => 202 b => 202 b => 202 katta bo'lgan birinchi element: c => 303 c => 303 c => 303

3.1-dasturtahlili. Dasturda 4 taelementdaniboratmap konteynerberilgan. 's' kalitlielementsoni count() funksiyabilananiqlagan. Xuddishukalitbilanuningqiymati find() funksiyabilantopilgan. equal_range('b'), mymap.lower_bound('b'); mymap.upper_bound('b'); make_pair(mymap.lower_bound(key), mymap.upper_bound(key));funksiyalariningekvivaletishlariko'rsatilgan.

Takrorlanmaydigan (unikal)
tartiblanganelementlarto'plami<set>. set<K, C = less<K>, A = allocator<K>> -
Bukonteynerjoyidaqiymatlarnio'zgartirishgayo'lqo'ymaydi. <set> -
buassosiativkonteynerbo'lib, unikalkalitniqo'llabquvvatlaydi
(birxilqaymatlikalitlarniqo'llabquvvatlamaydi)
vakalityordamidatezqidirishimkoniniberadi.
set<K>::iterator va set<K>::const_iterator
iteratorlarfunksionalekvivaletthisoblanadivao'zgarganda const
K& qiymatqaytaradi. Set
to'plamdaqiymatnio'zgartirishuchunavvalo'chiriblashlabso'ngyangiqiy
matniyokio'zgartirilganvariantiqo'shiladi.
<set>ningshablioni:

```

template <class Key, class Compare = less<Key>,
         template <class U> class Allocator = allocator>

```

<set>sinfningochiqxususiyatlari, operatorlarivafunksiyalari:

1. Typedefoperatorlari:

```

typedef Key key_type;
typedef Key value_type;
typedef Allocator<Key>::pointer pointer;
typedef Allocator<Key>::reference reference;
typedef Allocator<Key>::const_reference
const_reference;
typedef Compare key_compare;
typedef Compare value_compare;
typedef iterator;
typedef iterator const_iterator;
typedef size_type;
typedef difference_type;
typedef reverse_iterator;
typedef const_reverse_iterator;

```

2. Xotiraniajratishvabo‘shatishoperatorlari (allocation/deallocation):

```

set(const Compare& comp = Compare());
template <class InputIterator>
set(InputIterator first, InputIterator last,
    const Compare& comp = Compare());
set(const set<Key, Compare, Allocator>& x);
~set();
set<Key, Compare, Allocator>& operator=(const
set<Key, Compare,
    Allocator>& x);
void swap(set<Key, Compare, Allocator>& x);

```

3. Ruxsatberishvositalarininingoperatorlari (accessors):

```

key_compare key_comp() const;
value_compare value_comp() const;
iterator begin() const;
iterator end() const;
reverse_iterator rbegin() const;
reverse_iterator rend() const;
bool empty() const;
size_type size() const;
size_type max_size() const;

```

4. Qo‘shishvao‘chirishoperatorlari (insert/erase):

```

pair<iterator, bool> insert(const value_type& x);
iterator insert(iterator position, const value_type& x);

```

```
template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);
```

5. To'plamamallari (set operations):

```
iterator find(const key_type& x) const;
size_type count(const key_type& x) const;
iterator lower_bound(const key_type& x) const;
iterator upper_bound(const key_type& x) const;
pair<iterator, iterator> equal_range(const
key_type& x) const;
```

6. Taqqoslashoperatorlari:

```
template <class Key, class Compare, class Allocator>
bool operator==(const set<Key, Compare, Allocator>&
x,
    const set<Key, Compare, Allocator>& y);

template <class Key, class Compare, class Allocator>
bool operator<(const set<Key, Compare, Allocator>& x,
    const set<Key, Compare, Allocator>& y);
```

<set> to'plamda:

iterator (**iterator**)bu — const value_type
ko'rsatuvchidomi yikkitom n lama iterator.

Aniq tipni belgilash realizatsiya qilishg abog'liq va Allocator da aniqlanadi.

const_iterator - ham iterator (iterator) kabidir.

size_type - ishorasiz butun tip,
Aniq tipni belgilash realizatsiya qilishg abog'liq va Allocator da aniqlanadi.

difference_type - ishorali butun tip,
Aniq tipni belgilash realizatsiya qilishg abog'liq va Allocator da aniqlanadi.

Elementlarni qo'shishi insert()
funksiya sitomonidan amalga oshiriladivabu funksiyaning bir nechta variantl
ar mavjud:

-juft iteratorlarni kitirishda interval dan qiymat qo'shish;

```
template <class InputIterator>
void insert (InputIterator first,
InputIterator last);
```

-qo'yish mumkin bo'lgan joyga ko'rsatish orqali bitta qiymat qo'shish (agaryangi element bevosita oldin pozitsiya ni egallasa, o'rnatish o'zgarish vaqt mobaynida amalga oshiriladi);

```
pair<iterator,bool> insert (const value_type& val);
```

-berilgan qiymatni qo'shish (asosiy usul);

```
iterator insert (iterator position, const value_type& val);
```

Oxirgi variant insert funksiyasi bir juftli (iterator, Boolean) qaytaradi.

Juftlikdagi birinchi element qo'shish joyni yoki topilgan qiymatini, ikkinchi joylashtirilgan qiymatni qo'shish bajarilganligi (true) yoki qo'shish vaqtida topilgan elementning mavjudligini (false) bildiradi.

Elementlarni o'chirish erase funksiyasi orqali amalga oshiriladi. Bu funksiya qiymat yoki berilgan elementlarni o'chirish yoki iterator yoki iteratorlar intervalini qabul qiladi.

```
(1) size_type erase (const value_type& val);  
(2) void erase (iterator position);  
(3) void erase (iterator first, iterator last);
```

Birinchi parametrga elementni o'chiradi, qolgan ikki parametrga elementni ko'rsatadigan iterator qaytaradi. Birinchi variant o'chirilgan elementlarning sonini beradi (set bo'lsa, 1 yoki 0 qiymat beradi).

3.2-dastur. Set to'plamga elementlarni qo'shish.

```
// created by MBBahodir  
#include "stdafx.h"  
#include <iostream>  
#include <set>  
using namespace std;  
  
void erase_subs(set<int>&s, const int &subs)  
{  
    set<int>::iterator p, pe, pd;  
    p = s.begin(), pe = s.end();  
  
    while (p != pe)  
    {
```

```

        if (p == s.find(subs))
            p = s.erase(p);
        else
            ++p;
    }
}

int main ()
{
    set<int> myset;
    set<int>::iterator it;
    pair<set<int>::iterator,bool> ret;

    for (int i=1; i<=5; ++i) myset.insert(i*10);

    ret = myset.insert(20);

    if (ret.second == false) it=ret.first;

    myset.insert (it,25);
    myset.insert (it,24);
    myset.insert (it,26);

    int myints[] = {5,10,15,60,28};
    myset.insert(myints,myints+5);

    cout << "myset contains after insert:";
    for (it=myset.begin(); it!=myset.end(); ++it)
        cout << ' ' << *it;
    cout << '\n';

    // set to 'plam tltmentlarini o'chirish
    it = myset.begin();
    ++it;

    myset.erase (it);

    myset.erase (40);

```

```

it = myset.find (60);
myset.erase (it, myset.end());

cout << "myset contains after erase :";
for (it=myset.begin(); it!=myset.end(); ++it)
    cout << ' ' << *it;
cout << '\n';

erase_subs(myset,24);

cout << "myset contains after erase_subs :";
for (it=myset.begin(); it!=myset.end(); ++it)
    cout << ' ' << *it;
cout << '\n';
system("pause");
return 0;
}

```

3.2-dastur. Output

```

myset contains after insert: 5 10 15 20 24 25 26 28 30 40 50 60
myset contains after erase : 5 15 20 24 25 26 28 30 50
myset contains after erase_subs : 5 15 20 25 26 28 30 50

```

Tartiblanganmultito‘plam<set> - <multiset>. <set>to‘plamdanfarqlio‘lararoq, insert funksiyasiqiymatqo‘shishuchunqo‘shiladiganqiymatniko‘rsatuvchiiterat orqaytaradi. <multiset> - assosiativkonteynetbo‘lib, tengqiymatlikalitlarnisaqlaydi (mumkinqadarbirkalitqiymatlielmentlarto‘plaminisaqlaydi) vakalitorqalitezqidirishimkoniniberadi. <multiset>ningshabloni (<set>nishablonibilanbixil):

```

template <class Key, class Compare = less<Key>,
          template <class U> class Allocator = allocator>

```

<multiset>sinfningochiqxususiyatlari, operatorlarivafunksiyalari:

1. Typedefoperatorlari - <set>nikibixil.
2. Xotiraniajratishvabo‘shatishoperatorlari (allocation/deallocation):

```

multiset(const Compare& comp = Compare());

```



```

template <class InputIterator>
multiset(InputIterator first, InputIterator last,
         const Compare& comp == Compare());
multiset(const multiset<Key, Compare, Allocator>& x);
~multiset();
multiset<Key, Compare, Allocator>& operator=(const
multiset<Key,
                                     Compare, Allocator>& x);
void swap(multiset<Key, Compare, Allocator>& x);

```

3. Ruqsatberishvositalarininingoperatorlari (accessors) -

<set>nikibilanbirxil.

4. Qo'shishvao'chirishoperatorlari (insert/erase):

```

iterator insert(const value_type& x);
iterator insert(iterator position, const value_type& x);
template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);

```

5. To'plamamallari (set operations) - <set>nikibirxil.

6. Taqqoslashoperatorlari:

```

template <class Key, class Compare, class Allocator>
bool operator==(const multiset<Key, Compare, Allocator>& x,
               const multiset<Key, Compare, Allocator>& y);

template <class Key, class Compare, class Allocator>
bool operator<(const multiset<Key, Compare, Allocator>& x,
              const multiset<Key, Compare, Allocator>& y);

```

<multiset>to'plamda:

iterator (**iterator**)bu — const value_type
ko'rsatuvchidomiyikkitomnlamaiterator.

Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

const_iterator - hamiterator (iterator) kabidir.

size_type - ishorasizbutuntip,
Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

difference_type - ishoralibutuntip,
Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

<set> va <multiset> to'plamlarning xususiyatlari, operatorlar va funksiyalar umummanoligandabir xil. Shuning uchun 2
tato'plamni birga o'rganish maqsadga muvofiq.

set va *multiset*

uchun foydalaniladigan iteratorlar o'zgarasizlikni ta'minlaydigan bo'lganligi uchun *algorithm* kutubxonasida mazkur tipni qo'llamaydigan funksiyalar yaratish mumkin emas.

Shuning uchun joriy sinfda aniqlangan massiv elementlari bilan ishlaydigan usullari, funksiyalardan (umumlashgan algoritmlar o'rnida) foydalanish kerak.

Yuqoridagi kodlab o'tilganidek, *multiset* to'plamning *set* to'plamdan farqi faqat kalit bir nechta bir xil qiymatliklarni saqlashidir. Shuning uchun ixtiyoriy sohada ishlatishda *multiset* sinfining ishlatilishi *set* sinfiniki dan farq qilmaydi.

set va *multiset* sinflarining ob'ektlarining tipik kalit bilan yonma-yan bittashablonli parametrlari shu mumkin.

Bu shablon taqqoslash (*comp*) funksiyasidir.

Agar shundan funksiya mavjud bo'lsa, oshkormas `less<>` funksiyasi bilan beriladi (`amali`).

set (*multiset*) sinflarining ob'ektlarini quyidagi konstruktorga yaratish mumkin:

Bo'sh to'plam konstruktoriga - `set<type, comp> ar;` yoki `set<type, comp> ar(Comp);`

Nusxalash konstruktoriga - `set<type, comp> ar(other);`

Iteratorlarni yordamida qo'shish konstruktoriga - `set<type, comp> ar(first, last);` yoki `set<type, comp> ar(first, last, Somp);`

Ro'yxatni saqlashda initsializatsiya qilish konstruktoriga - `set<type, comp> ar {init};` yoki `set<type, comp> ar(init);` yoki `set<type, comp> ar(init, Comp);`

Bu konstruktorga `Comp` konteynerlarning kalitlarini taqqoslash funksiyasi (ixtiyoriy). Agar dasturchi o'zining aloqador funksiyasini yaratmasa, qo'shish uchun taqqoslash funksiyasining yonida konstruktorga majburiy bo'lmagan `Allocator()` funksiyasi bor.

set (*multiset*) sinflarining ob'ektlarini o'chirish uchun destruktorga - `ar.~set();`

Hajmlar va o'lcham bilan ishlash usullari: `empty()` – true qaytaradi, agar to'plam bo'sh bo'lsa false, `size()` -

to'plamdagi elementlar sonini qaytaradi, `max_size()` – mumkin bo'lgan elementlar sonini qaytaradi.

Modifikatorlari: `clear()` – konteynerni tozalaydi, `insert()` – element qo'shish, `erase()` – element o'chirish, `swap()` – o'ralmashtirish, `emplace()` – joydagi elementni yaratish.

Qidirish usullari: `set` to'plam bilan ishlaganda qidirish muhim xisoblanadigan aspekt bo'lib hisoblanadi.

Qidirish nisamaralita shakil qilish uchun bir nechta o'zining usullariga ega:

`count(key)`, `find(key)`, `equal_range(key)`, `lower_bound(key)`, `upper_bound(key)`

Taqqoslash va birlashtirish amallari: `set` to'plam uchun barcha taqqoslash amallari amal qiladi. `ning ketma-ket konteynerlarga amal qilgan barcha amal o'rinli.`

`[=]` `(operator=)` operatorni elementlar to'plamini olish uchun ishlatiladi. Masalan, `other` to'plamning nusxasini yangi bir `other_one` to'plamga nusxalash uchun ishlatiladi (`other_one = other`).

Semantik nusxalashni amalga oshirish usuli `other` to'plamning nusxasini yangi bir `other_one` to'plamga to'liq nusxalash uchun ishlatiladi va `other` to'plam o'chirilmaydi, ammo uni holatini aniqlab bo'lmaydi `other_one = move(other)`.

`<set>` va `<multiset>` doirasi masalalar va dasturlarni keltiramiz.

1-masala. `[10, 50]` intervalda `N` ta ketma-ket sonlartasofidiy berilgan. `{10, 20, 30, 40, 50}` berilgan sonlardan joriy ketma-ketlikdanechtadan bor.

3.3(a)-dastur. 1-masalaning dasturi.

```
#include "stdafx.h"

#include <iostream>
#include <vector>
#include <set>
#include <random>
#include <ctime>
using namespace std;

int main() {
    default_random_engine rnd(time(0));
    uniform_int_distribution<unsigned> g(10, 50);
    vector<int> myVektor;
```

```

int n, k = 0;
cout << "n = "; cin >> n;
for (int i = 0; i < n; i++) {
    int d = g(rnd);
    myVektor.push_back(d);
    cout << myVektor[i] << " ";
}
set<int> mySet;
for (int i=1; i<=5; ++i) mySet.insert(i*10);
cout << endl;
for (int i = 0; i < n; i++)
    if (mySet.count(myVektor[i])) k++;
cout << k << " ta element topildi." << endl;
system("pause");
return 0;
}

```

3.3(a)-dastur. Output

```

n = 255
47 50 18 29 44 47 31 31 25 19 40 32 43 29 38 22 10 39 47 10 43 46 26
47 18 50 30 24 34 47 31 13 35 41 29 26 25 26 19 46 18 17 23 32 18 47
37 14 13 13 32 28 14 42 41 29 21 19 23 38 12 31 26 48 34 44 20 21 34
13 43 42 30 26 44 10 34 22 15 26 23 26 30 22 44 41 44 12 36 33 29 37
20 17 10 28 33 26 19 12 50 31 49 33 50 41 32 42 38 35 27 23 18 25 28
31 12 36 18 42 36 44 24 37 10 34 49 26 29 11 39 33 31 15 11 42 37 40
49 34 41 20 50 39 19 29 30 24 38 47 14 41 37 21 49 21 42 14 37 45 12
40 17 46 17 10 45 43 37 19 25 39 11 49 33 17 11 12 10 22 19 26 12 47
34 16 21 33 25 44 26 40 28 38 35 15 31 26 10 32 40 42 45 32 17 20 41
30 36 13 23 24 44 19 23 29 21 38 30 26 20 23 33 31 39 20 22 50 34 18
24 21 30 21 42 13 22 41 19 35 40 14 34 37 16 14 35 17 44 40 13 37 28
45 24
34 ta element topildi.

```

Oldinroqjudatez-
tezelementlarnikiritishvao‘chirishalgoritmlarnifoydalanishkerakemasdeb
aytgan. Vektorvato‘plambilanishlashsamaradorliginitaqqoslaylik.
To‘plamvavektorningmassivlarinitasodifiysonlarbilanto‘ldiradiganvabel
gilanganqiymatniqidiradigandasturyarataylik.

Vektorvato‘plambilanishlashsamaradorliginitaqqoslash.

```
// Created by MBBahodir
```

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <set>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include <random>
```

```
#include <ctime>
```

```
#include <chrono>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```
    default_random_engine rnd(time(0));
```

```
    uniform_int_distribution<unsigned> g(1, 10000);
```

```
    cout << "n = "; cin >> n;
```

```
    const int k = 100;
```

```
    int a = n;
```

```
    auto start = chrono::system_clock::now();
```

```
    vector<int> ar1;
```

```
    while (a--) ar1.push_back(g(rnd));
```

```
    auto result1 = find(ar1.begin(), ar1.end(), k);
```

```
    auto end = chrono::system_clock::now();
```

```
    auto elapsed = end - start;
```

```
    cout << "vector => "
```

```
<< elapsed.count()
```

```
<< endl;
```

```
    a = n;
```

```
    start = chrono::system_clock::now();
```

```
    set<int> ar2;
```

```
    while (a--) ar2.insert(g(rnd));
```

```

        auto result2 = ar2.find(k);
        end = chrono::system_clock::now();
        elapsed = end - start;
        cout << "set => " << elapsed.count() << endl;

        system("pause");
        return 0;
    }

```

3.3(b)-dastur. Output
n = 10000 vector => 289989 set => 1160000

To‘plamning ishlashivektordan sezilarli darajada past bo‘ladi. Shuni yodd tutingki, to‘plamni to‘ldirish amal isaralash bilan amalga oshiriladi va bualbatta, qimmatbaho amal. Lekin, faqat qidirish uchun, har ikki konteynerlarni ishlash solishtiradigan bo‘lsa, vaziyat <set> foydasig aozgaradi:

3.3(c)-dastur. Output
n = 10000 vector => 70042 set => 20044

Bundan qanday xulosachiqarish mumkin? set yordamida (yoki multiset)

takrorlanish jarayonlarini dasturlashda ketmaketlikdanelementlarni olish, yokitez-

tezo‘chirish va elementlarni kiritish bunoto‘g‘ri yondashuv hisoblanadi. Set to‘plam ob‘ektini yaratish uchun takrorlanish jarayonlarini asosidashakllantiril ganketmaketlik foydalanish kerak yoki tayyor to‘plam uchun insert funksiyasidan foydalanish maqsadga muvofiq.

Ammo, dastur tez-tezelementlarni qidirish ishlatilsa, set raqobatgacha qo‘lmaydi.

pair - yordamchisi inf. Boshqa assotsiativ konteynerlar bilan ishlashni yanada ko‘ribchi qish uchun pair yordamchisi hablonsinf bilan tanishishimiz kerak. Buning birlik sifatida ikki taob‘ektlarni saqlashimkoniyatini beradi. pair STD

ningturlijoylarida, xususan minmax() algoritmda, set equal_range() sinfiningusulida, shuningdekjuft (key, value) bo‘lganassosiativkonteynerlarelementlaribilanishlashuchunishlatiladi.

pair

birjuftob‘ektyaratishuchunquyidagikonstruktorsintaktikiishlatiladi:

```
pair<T1, T2> p1;  
pair<T1, T2> p2(val1, val2);  
pair<T1, T2> p3(p1);
```

Amalari:

p.first – havolabo‘yichajuftlikningbirinchielementigamurojaat.

p.second – havolabo‘yichajuftlikningikkinchielementigamurojaat.

p->first – ko‘rsatkichbo‘yichajuftlikningbirinchielementigamurojaat.

p->second – ko‘rsatkichbo‘yichajuftlikningikkinchielementigamurojaat.

p = other_p – qiymatqilibberish (C++

qoidasigaasosanoshkormastipalmatirishyordamida).

p.swap(other_p) yoki swap(p, other_p) –p va other_p

laruchuno‘rinalmashtirish.

<, <=, >, >=, ==, != – taqqoslashamallari

make_pair(val1, val2) – juftlikniberadi.

3.4-dastur. Pairkonstruktorivaamallariniishlatish.

```
// Created by MBBahodir  
#include "stdafx.h"  
  
#include <iostream>  
using namespace std;  
  
int main() {  
    int a1 = 16, b1 = 11;  
    int a2 = 18, b2 = 03;  
    pair<int, int> par1(a1, b1);  
    pair<int, int> *par2 = new pair<int, int>(a2,  
b2);  
    cout << par1.first << '.' << par1.second <<  
endl;  
    cout << par2->first << '.' << par2->second <<  
endl;  
    system("pause");  
    return 0;
```

}

3.4-dastur. Output
16.11
18.3

Assosiativkonreynlaruchunmap va multimap sinflari. map assotsiativkonteyneri (xarita, lug‘at) o‘z-o‘zinimuvozanatlovchidaraxtga (qizil-qoradaraxt)

asoslanganto‘plamlarbilanbirxiltarzdaamalgaoshiriladi.

To‘plamvalug‘ato‘rtasidagiasosiyfarqshundaki, map massivikalitlardantashkiltopganelementlarjuftlariningtartibliassotsiativm assivi (konteyneri) vaularningmosqiyamatlaridaniborat.

Lug‘atdakalitlariunikal (takrorlanmaydigan) va multimap birnechtanusxadagikalitlaribo‘ladi.

Lug‘atlardakalitvaqiyamaturlarifundamentaltipyokiabstrakttipbo‘lishimumkin. Agarkalittipiabstrakttipbo‘lsa,

uninguchunsaralashyo‘nalishinibelgilovchikomparatorfunksiyasi (taqqoslashfunksiyasi) aniqlanishikerak.

Kalitdoimiyqiyamatgaegavakalitqiyamatini o‘zgartirishmumkinemas.

Juftlikningikkinchielementqiyamatini (asosiyqiyamatini) o‘zgartirishmumkin.

Map (yoki multimap) sinflaribilanishlashniboshlashuchundastursarlavhasiga (ikkalasinfbilanbirgalikda) quyidagidasturfragmentiyoziildi (buoldinhambarchaassosiativkonteynerlaruchunaytilganedi):

```
#include <map>
```

Konstruktorlar. Lug‘atob’ektlarishablonparametrlarini: kalittipivakalitningqiyamatitipi (key vaT), taqqoslash (comp)funksiyasinio‘zichigaoladi. Agarbundayfunksiyabo‘lmasa, uoshkormasless<>funksiya (operation <>) tomonidano‘rnatiladi.

Quyidagikonstruktorlaryordamidamapsinfob’ektlariniyaratiladi:

Oddiy map konreynerkonstruktorlari: map<Key, T, comp> ar; yoki map<Key, T, comp> ar(Comp);

Nusxalashuchunkonstruktor: map<Key, T, comp> ar(other);

Iteratorlaryordamidaqo‘shishuchunkonstruktorlar: map<Key, T, comp> ar(first, last); yokimap<Key, T, comp> ar(first, last, Somp);

Ro‘yxatasosidainitsializatsiyaqilishkonstruktorlari: map<Key, T, comp> ar {init}; yokimap<Key, T, comp> ar(init); yokimap<Key, T,

comp> ar(init, Comp);Eslatma.

Ro'yxatni initsializatsiya qilishda har bir juftlik alohida figurali qavs ichiga olinishi kerak!

```
map<string, int> ar {{"a1", 10}, {"www", 17}, {"j8", 100}};
```

Buyerda Comp konteyner kalitlarini solishtirish uchun funksiyasi (ixtiyoriy). Bundantashqari, taqqoslash funksiyani yonida konstruktor ixtiyoriy Allocator() vazifasini o'z ichiga oladi (soddaligi uchun yozilmaydi), dasturchiuning allocator vazifasini ham belgilaydi.

Map sinfining destruktori: ar.~map();

Lug'atusullari. Lug'atusullari set to'plam usullari bilan bir xil bo'lganligi uchun ularni takrorlab keltirmaymiz. Bu usullar qanday ishlashini misol yordamida ko'rsatamiz. Aytaylik, quyidagi chalug'at yaratishimiz kerak: kalitgatasodifiy (string), qiymatgatasodifiy (integer) o'rnatiladi. So'ngra key1 va key2 orasidagi barcha key = > qiymat juftliklarining chiqishini talab qilinsin. Dasturoxirida key 3 kalit uchun maksimal qiymatni chiqarishni talab qilinadi (dasturning o'zidagi izohlarga qarang).

3.5-dastur. Lug'atusullaridan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <random>
#include <ctime>
#include <string>
#include <map>
#include <vector>

using namespace std;

int main() {
    map<string, int> rat;
    vector<string> lit;
    lit.push_back("a1"); lit.push_back("a2");
    lit.push_back("a3");
    lit.push_back("b1"); lit.push_back("b2");
    lit.push_back("b3");
    lit.push_back("c1"); lit.push_back("c2");
```

```

lit.push_back("c3");
default_random_engine rnd(time(0));
uniform_int_distribution<int> d(-10, 20);
uniform_int_distribution<int> c(0, 8);

//=====1=====
=====//
    // lugʻatni toʻldirish: tasofidiy kalit tasodifiy
olinadi    //

//=====
=====//
    int n;
    cout << "Lugʻat elementlari sonini kirit: ";
    cin >> n;
    while (n-- > 0) {
        string S = lit[c(rnd)];
        int D = d(rnd);
        pair<string, int> tpr = make_pair(S, D);
        rat.insert(tpr);
        // emplace uchun juftlik yaratish shartmas:
        //rat.emplace(S, D);
    }

//=====2=====
=====//
    // 2 ta har xil amal bilan kalit va qiymat
qoshish    //

//=====
=====//
    rat.emplace("d1", 10);
    rat.insert(pair<const string, int>("d3", -15));
    rat.insert(pair<const string, int>("d3", 6));

//=====3=====
=====//
    // birinchi va ikkinchi kalit orasidagi

```

```

elementlarni chiqarish//

//=====
=====//
    string el_l, el_r;
    cout << "kalitlarni kirit:\n";
    cout << "1-kalit: "; cin >> el_l;
    cout << "2-kalit: "; cin >> el_r;
    auto fst = rat.lower_bound(el_l);
    auto lst = rat.upper_bound(el_r);
    while (fst != lst) {
        cout << fst -> first
<< " => "
<< fst -> second
<< endl;
        fst++;
    }

//=====4=====
=====//
    // kalitga mos eng katta elementni aniqlash
//

//=====
=====//
    string tmp;
    cout << "Kalit kirit: ";
    cin >> tmp;
    auto first = rat.equal_range(tmp).first;
    auto last = rat.equal_range(tmp).second;
    if (first == last)
        cout << 0 << endl;
    else
        while (first != last) {
            auto r = first -> second;
            if (r >= 10)
                cout << r << " ";
            first++;

```

```

}
    system("pause");
return 0;
}

```

3.5-dastur. Birinchisinov Output

Lugʻat elementlari sonini kirit: 10
a2 => 14
a3 => 13
b2 => -8
b3 => 3
c2 => -4
c3 => -7
d1 => 10
d3 => -15
kalitlarni kirit:
1-kalit: a1
2-kalit: c2
a2 => 14
a3 => 13
b2 => -8
b3 => 3
c2 => -4
Kalit kirit: c4
0

3.5-dastur. Ikkinchisinov Output

Lugʻat elementlari sonini kirit: 9
a1 => 11
a2 => 10
b1 => 2
b2 => -10
c1 => 3
c2 => 9
d1 => 10
d3 => -15
kalitlarni kirit:
1-kalit: a1
2-kalit: c2

```
a1 => 11
a2 => 10
b1 => 2
b2 => -10
c1 => 3
c2 => 9
Kalit kirit: d3
```

Elementlarga murojaat.

Lugʻatlarda elementlarga kirishning yana ikkita usuli mavjud (iteratorlar bilan ishlashdantashqari). Birinchi usul `at(key)` usulini qoʻllashdir (`bilsangizkerak`). Biroq, indeksni argument sifatida qabul qilmaydi, lekin kalitning qiymatini qabul qiladi.

Bu usul kalitga ekvivalent boʻlgan kalit bilan mos elementni qiymatiga mos yozuvlar qaytaradi. Agar bu element mavjud boʻlmasa, `out_of_range` istisno funksiyasining qiymati qaytariladi. Boshqacha aytganda, massivdabunday kalit bor-yoʻqligini tekshiradi. Ikkinchi usul `overloaded[]` funksiyadan foydalanishga asoslangan.

Lekin quyidagi sabablarga koʻra ehtiyojli korlik bilan foydalanish zarur.

`at(key)` mavjud boʻlmasa, standart konstruktoryordamida kalit bilan yangi element qator qoʻshiladi.

Aksholdaelementga koʻrsatkich qaytariladi. Eslatib oʻtish, indekslash funksiyasini assotsiativ konteynerlarning ikkita turiga (`map` va `unordered_map`) foydalanish mumkin.

Ushbu usullarni ishlab chiquvchi lartomonidan qoʻshilishining sabablari aniq : dastur juda qisqavachiroyl boʻladi.

`at()` funksiyasi istisno qaytarгани uchun `try - catch` funksiyasi bilan quyidagicha boshqarish mumkin:

```
map<string, int> rat;
try {
    rat.at("U");
}
// istesnoni ushlab olish
catch (std::out_of_range) {
    // uni qayta ishlash
    rat["U"] = 15;
}
```

Dasturfragmentida "U" kalitlijuftliktopilmasa, uni yaratib, yangiqiyamat beradi.

rat["U"] = 15; (hiyla, ruschafishka)
indekslashamaliu faqatasosiyqiyamatini olish emas,
balki bu qiyamatni o'zgartirish mumkin degan ma'noni anglatadi,
bir qiyamatini qaytaradi. Boshqacha qilib aytganda,
ushbu dastur fragmentidagi amallarini bajarish mumkin:

```
rat["b1"]=1980;  
cout << rat["b1"] << endl;  
rat["b1"] +=1;  
cout << rat["b1"] << endl;  
--rat["b1"];  
cout << rat["b1"] << endl;
```

map<K, T, C = less<K>, A = allocator<pair<const K, T>>> to'plam
pair<const K, T> tipidagi qiyamatni saqlaydi. Bunda K kalit,
tanlashimkonini beradiv T saqlanadigan qiyamat.
Shuning uchun iterator dan foydalangan nimizda first kalit qiyamatni va
second saqlangan qiyamatni qaytaradi, o'zgartirishimkonini beradi.
Yuqorida aytibo'tilgandek operator[] asosiy amallardan hisoblanadi
(vazifasini bilsangiz kerak). Agar map<K, T> m bo'lsa va m[k] = t
ma'noji hatidan quyidagi fragment gateng:

```
m.insert(make_pair(k, T())).first->second = t
```

Bu kabielementlarni qo'shish samarali hisoblanib, T
ob'ektni yaratishga imkon bermaydi (yuqoridagi fragmentda aniq T()
konstruktor chaqirilgan).

Agar operator[] noqulay va samarasiz deb hisoblasangiz,
yanaunga ikkita alternativ variantlar bor. Birinchisi find
funksiyasidan foydalanib, kalit bo'yicha (key, value)
juftligiga mos iteratorni olish yoki kalit lug'atdabo'lmasa, end()
funksiyadan foydalanish mumkin. Ikkinchisi at() funksiyasidan,
ya'ni kalit asosida qiyamatni qaytaradi. Agarkalit bo'lmasa,
istisno holatga o'tadi.

Map sinfining umumiy shabloni quyidagicha:

```
template <class Key, class T, class Compare = less<Key>,  
          template <class U> class Allocator = allocator>
```

Map sinfining operatorlari, xususiyatlari,
funksiyalariva usullarini qo'ribchiqamiz.

Typedef operatorlari :

```
typedef Key key_type;
```

```

typedef pair<const Key, T> value_type;
typedef Compare key_compare;
class value_compare
    : public binary_function<value_type, value_type, bool>
{
    friend class map;
protected:
    Compare comp;
    value_compare(Compare c) : comp(c) {}
public:
    bool operator()(const value_type& x, const value_type&
y) {
        return comp(x.first, y.first);
    }
};
typedef iterator;
typedef const_iterator;
typedef Allocator<value_type>::pointer pointer;
typedef Allocator<value_type>::reference reference;
typedef Allocator<value_type>::const_reference
const_reference;
typedef size_type;
typedef difference_type;
typedef reverse_iterator;
typedef const_reverse_iterator;

```

Xotiraniajratishvabo‘shatish (xolliqilish) operatorlari
(allocation/deallocation):

```

map(const Compare& comp = Compare());
template <class InputIterator>
map(InputIterator first, InputIterator last,
    const Compare& comp = Compare());
map(const map<Key, T, Compare, Allocator>& x);
~map();
map<Key, T, Compare, Allocator>&
    operator=(const map<Key, T, Compare, Allocator>& x);
void swap(map<Key, T, Compare, Allocator>& x);

```

Ruxsatetishvositalari (accessors):

```

key_compare key_comp() const;
value_compare value_comp() const;

```

```

iterator begin()
const_iterator begin() const;
iterator end();
const_iterator end() const;
reverse_iterator rbegin();
const_reverse_iterator rbegin();
reverse_iterator rend();
const_reverse_iterator rend();
bool empty() const;
size_type size() const;
size_type max_size() const;
Allocator<T>::reference operator[](const key_type& x);

```

Qo'shishvao'chirishfunksiyalar (insert/erase):

```

pair<iterator, bool> insert(const value_type& x);
iterator insert(iterator position, const value_type& x);
template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);

```

Mapsinfusulari:

```

iterator find(const key_type& x);
const_iterator find(const key_type& x) const;
size_type count(const key_type& x) const;
iterator lower_bound(const key_type& x);
const_iterator lower_bound(const key_type& x) const;
iterator upper_bound(const key_type& x);
const_iterator upper_bound(const key_type& x) const;
pair<iterator, iterator> equal_range(const key_type& x);
pair<const_iterator, const_iterator> equal_range(const key_type& x) const;

```

Mantiqiyoperatorlari:

```

template <class Key, class T, class Compare, class Allocator>
bool operator==(const map<Key, T, Compare, Allocator>& x,
                const map<Key, T, Compare, Allocator>& y);

template <class Key, class T, class Compare, class Allocator>
bool operator<(const map<Key, T, Compare, Allocator>& x,

```



```
const map<Key, T, Compare, Allocator>& y);
```

iterator- ikkitomonlama iterator bo'lib value_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosilabelgilanadi.

const_iterator- doimiyikkitomonlama iterator bo'lib const value_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi. Bunda iterator va const_iterator uchunkonstruktorbor, bukafolatlanadi.

size_type – butunishorasiztip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

difference_type - butunishoralitip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

Assotsiativkonteynerlaruchunstandartusullarto'plamidantashqari, lug'atAllocator::reference operator[](const key_type&)amalinita'minlaydi. Masalan, m lug'atda k kalituyulan m[k] kuyidagifragmentbilanekvmvalent:

```
(*((m.insert(make_pair(k, T()))).first)).second
```

Assotsiativkonteynerlarasosiyxususiyatlari

Headers		<set>		<map>	
Members		<u>set</u>	<u>multiset</u>	<u>map</u>	<u>multimap</u>
	constructo r	<u>set</u>	<u>multiset</u>	<u>map</u>	<u>multimap</u>
	destructor	<u>~set</u>	<u>~multise t</u>	<u>~map</u>	<u>~multima p</u>
	assignment	<u>operator =</u>	<u>operator =</u>	<u>operator =</u>	<u>operator =</u>
iterator s	begin	<u>begin</u>	<u>begin</u>	<u>begin</u>	<u>begin</u>
	end	<u>end</u>	<u>end</u>	<u>end</u>	<u>end</u>
	rbegin	<u>rbegin</u>	<u>rbegin</u>	<u>rbegin</u>	<u>rbegin</u>
	rend	<u>rend</u>	<u>rend</u>	<u>rend</u>	<u>rend</u>
const iterator s	cbegin	<u>cbegin</u>	<u>cbegin</u>	<u>cbegin</u>	<u>cbegin</u>
	cend	<u>cend</u>	<u>cend</u>	<u>cend</u>	<u>cend</u>
	crbegin	<u>crbegin</u>	<u>crbegin</u>	<u>crbegin</u>	<u>crbegin</u>
	crend	<u>crend</u>	<u>crend</u>	<u>crend</u>	<u>crend</u>
capacity	size	<u>size</u>	<u>size</u>	<u>size</u>	<u>size</u>
	max_size	<u>max size</u>	<u>max size</u>	<u>max size</u>	<u>max size</u>
	empty	<u>empty</u>	<u>empty</u>	<u>empty</u>	

Tartiblangan unikalkalitga egabo'lmagan map – multimap sinfi hisoblanadi.

Businfooperator[] standart operatorni qo'llab quvvatlamaydi, va multiset juftligini eslatadi. Bunda qidirish faqat birinchi maydon, ya'ni kalit maydon bilan amalga oshiriladi. multimap sinfini o'rganib chiqish uchun, misol sifatida matnli faylga yozilgan lug'atga murojaat mallarini ko'rsatamiz.

Faraz qilaylik lug'at so'zlar juftligidan tashkil topgan va birinchi so'z kalit bo'lsin va takrorlanishi mumkin.

“so'z - tarjimasi” tipini aniqlash.
Bunda so'z juftligini o'lamiz va lug'at element deb qaraymiz.

```
using Entry = pair<string, string>;
```

Lug'at tipini aniqlash

```
using Dictionary = multimap<string, string>;
```

Standart kirish va chiqish operatorlari quyidagicha aniqlaymiz.

```
namespace std
{
    istream& operator>>(istream &is, Entry &en)
    {
        return is >> en.first >> en.second;
    }

    ostream& operator<<(ostream &os, const Entry &en)
    {
        return os << en.first << ' ' << en.second;
    }

    istream& operator>>(istream &is, Dictionary &d)
    {
        istream_iterator<Entry> begin(is), end;
        d = Dictionary(begin, end);
        return is;
    }

    ostream& operator<<(ostream &os, const Dictionary &d)
    {
        ostream_iterator<Entry> out(os, "\n");
```

```

    copy(d.begin(), d.end(), out);
return os;
}
}

```

Lugʻatgakirishuchunamal.
Belgilanganlugʻatuchunyangilugʻatchiqaradi.

```

Dictionary reverse(const Dictionary &d)
{
    Dictionary result;
    for (const auto &el : d)
        result.emplace(el.second, el.first);
return result;
}

```

Lugʻatelementigamurojaatvaoʻqish.

```

void dictionary_reverse(istream &from, ostream &to)
{
    Dictionary read;
    from >> read;
    to << reverse(read);
}

```

Std sinfidaaniqlanganistream_iterator va ostream_iterator tiplaridanfoydalanganlikuchun [<<] va [>>] operatorlarishufazoagamoslabyaratilgan.

Bufoydalanuvchigakirishvachiqishoperatorlarivastandardtiplardan foydalanishruhiyatigataʻsirqilmaydi.

Multimarsinfningkonstruktori map sinfinikibilanbixil.

Multimarsinfning map sinfinikidanfarfqiladiganoperatorlari:

```

friend class multimap;
protected:
    Compare comp;
    value_compare(Compare c) : comp(c) {}
public:
    bool operator()(const value_type& x, const value_type&
y) {
        return comp(x.first, y.first);
    }
};

```

Mantiqiyoperatorlari:

```

template <class Key, class T, class Compare, class Allocator>

```

```
bool operator==(const multimap<Key, T, Compare, Allocator>& x,
                const multimap<Key, T, Compare, Allocator>& y);
```

```
template <class Key, class T, class Compare, class Allocator>
bool operator<(const multimap<Key, T, Compare, Allocator>& x,
              const multimap<Key, T, Compare, Allocator>& y);
```

iterator- ikkitomonlama iterator bo'lib value_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosilabelgilanadi.

const_iterator- doimiyikkitomonlama iterator bo'lib const value_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi. Bunda iterator va const_iterator uchunkonstruktorbor, bukafolatlanadi.

size_type – butunishorasiztip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

difference_type - butunishoralitip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

Tartiblanmaganassosiativkonteynerlar<unordered_set>va <unordered_map>.Tartiblanmagankonteynerlar<unordered_set>, <unordered_map>xesh – jadvalasosidaqurilgan (odatdabuxeshjadvallarekvivalentelementlarro'yxati - bloklar) vaxeshfunksiyalarga (joriyholatidastandarttiplaruchun<functional>sinfidan std::hash<T> bilananiqlanadi) vatenglikamaliga (joriyholatda [==] operatori) tayanadi.

Ikkitaelementuchunxeshlaritengbo'lsavatenglikamalichin (true) qiymatqaytarsa, ularekvivalenthisoblanadi. Tartiblanmagankonteynerlaribirtomolamaiteratorlarbilanelementlarigam urojaatqilishkerak.

Tartiblanmaganstandartassosiativkonteynerlargaquyidagishablonsi nflarkiradi:

```
unordered_set<K, H = hash<K>, E = equal_to<K>, A =
allocator<K>>,
unordered_map<K, T, H = hash<K>, E = equal_to<K>, A =
allocator<pair<const K, T>>>,
unordered_multiset,
unordered_multimap.
```

Bularningfunksionalligitartiblanganassosiativkonteynerlarga o'xshaydi.

Tartiblanmagankonteynerlardaxeshjadvalnishakllantirishningo‘zig
axosligilower_bound va upper_bound funksiyalarigaegamasligidir.
Ammo, berilgankalitgamosbirintervaldagielementlarniolishuchun
equal_range funksiyasiishlatiladi.

O‘rtachatariblanmagankonteynerlardaelemetnlarniqo‘shish,
qidirishvao‘chirisho‘rtachadoimiyvaqttalabvatartiblanganlorganisbatanse
zilarlidarajadatezroqbo‘lishimumkin. Biroq,

engyomonholatdahamvaqtgachiziqlierishishmumkin.

Samaradorligiheshfunksiyasiniamalgaoshirishsifatigabog‘liq,
lekinbutunlaybundayvaqtajratishnioldiniolishmumkinemas.

Shuninguchun, interaktivdasturlarda (masalan, o‘yinlar),
tartibgasolinmagankonteynerlardavriykechikishlartufaylitartiblanganlard
anko‘rayomonroqbo‘lishimumkin.

Kattasondagielementlarqo‘shishuchun,
keraklivaqtdabirqatorkiritishda "rehash"gaoshirishmumkin
(ko‘pvaqtoladiganajratilgansaqlashvaqaytatarqatishelementlarhajminios
hirish) rehash funksiyasinichaqiribto‘g‘rihisoblanadi (vektoruchun
reserve funksiyasigao‘xshaydi).

Ayrimvazifalardasaralashuchunvaqtko‘pajratilsaavtomatiksaralash
nimuhimkamchilikdebhisoblashmumkin. Buholda,
tartiblanganto‘plamvalug‘atlarsinflarinitartiblanmaganassotsiativkonteyn
erlarsinflar (unordered_set, unordered_multiset, unordered_map va
unordered_multimap) tomonidano‘zgartirilishimumkin.

Tartiblanmagankonteynerlardaqidirish,
qo‘shishvao‘chirishamallario‘rtachadoimiyvaqtmurakkabligigaegavauO
(1) gatengdir.

Tartiblanmagankonteynerlarxeshjadvallarsifatidaamalgaoshiriladi.
Xesh-jadvaldaamalnibajarishxesh-funksiyada (xeshlash)
kalitdanboshlanadi. Olinganxeshqiymati (shuningdek,
xeshyokixeshkodi) massivdaindeksrolinibajaradi.
Xeshjadvalioddiylug‘atgao‘xshaydi,
undatomma‘nodaxeshqiymatidebhisoblashmumkin.

Tartiblanmagankonteynerlarningxeshjadvallaridayacheykalariborv
aularcheksizko‘pelementlarnio‘zichigaolishimumkin. Buyacheeykani
segment debataymiz. Aslida, segment (xeshbilanbog‘liq)
ulanganro‘yxatdir.

To‘plamhajmiganisbatansaqqlanadiganelementlarsonimoskelsa,
(xeshfunksiyasiuchunmumkinbo‘lganelementlarsoni)
xeshjadvalnito‘ldirish (load factor)

koeffitsientidebyuritiladivao‘rtachaamalijrovaqtbelgilaydiganmuhimpara metrhisoblanadi.

Unordered_set va unordered_multiset sinflaribilanishlashniboshlashuchun unordered_set sarlavhasini (ikkalasinfbilanbirgalikda) quyidagichaqo‘shishkerak:

```
#include <unordered_set>
```

unordered_map va unordered_multimap sinflarihamxuddiyuqorida gidekqo‘shiladi:

```
#include <unordered_map>
```

Tartiblanmagankonteynerlarningtiplariquyidagishablonparametrlar inio‘zichigaoladi:

- const Key – kalittipi (unordered_set, unordered_multiset, unordered_map, unordered_multimap).

- T – qiymatningtipi (unordered_map, unordered_multimap).

- alloc –allokatorfunksiya (majburiyemas, erkin).

- size_type bucket_count – initsializatsiyaqilishdafoydalanishuchunminimallyacheykalarsoni (agarko‘rsatilmaganbo‘lsa, joriyholatbo‘yichaolinadi).

- hash – xesh-funksiya (erkin).

- equal – taqqoslashfueksiyasi, kalitlarnisolishtirishuchunishlatiladi (erkin).

Quyidagikonstruktorlarasosidaunordered_setva unordered_mapsni fl ariningob’ektiniquradi:

```
unordered_set<Key> ar;  
unordered_set<Key, hash, equal> ar;  
unordered_map<Key, T> ar;  
unordered_map<Key, T, hash, equal> ar;
```

Agar hash xeshfunksiyayozilmaganbo‘lsa, joriyholatbo‘yicha hash<key_type>funksiyasiishlatiladi.

Agarequal taqqoslashfunksiyasiyozilmaganbo‘lsa, joriyholatbo‘yicha equal_to<> (operator==) funksiyasiishlatiladi.

Soddaligiuchun Allocator () va bucket_count yozilmaydi.

Nusxalashkonstruktrlari:

```
unordered_set<Key> ar(other);  
unordered_map<Key, T> ar(other);
```

Iteratorlarasosidaqo‘shish:

```
unordered_set<Key> ar(first, last);  
unordered_set<Key, hash, equal> ar(first, last);  
unordered_map<Key, T> ar(first, last);
```

```
unordered_map<Key, T, hash, equal> ar(first, last);
```

Ro'yxatbo'yichainitsializatsiyaqilish:

```
unordered_set<Key> ar {init};  
unordered_set<Key> ar(init);  
unordered_set<Key, hash, equal> ar(init);  
unordered_map<Key, T> ar {init};  
unordered_map<Key, T> ar(init);  
unordered_map<Key, T, hash, equal> ar(init);
```

unordered_set va unordered_map

sinfob'ektlarinio'chirishhundestruktorlar:

```
ar.~unordered_set();  
ar.~unordered_map();
```

Tartiblanmagankonteynerlardaislatilmaydiganusullar:

- [$>$], [$<$], [$>=$], [$<=$] taqqoslashamallari ([$=$] va [$!=$] amallarniqo'llabquvvatlaydi)
- lower_bound va upper_bound
- Ikkitomonlamaiteratorlar: rbegin, crbegin, rend, crend

Tartiblanmagankonteynerlarningusullari:

Xeshlashuslubi

(Buusullartartibsizkonteynerlarningishlashiniboshqarishimkoniniberadi):

- load_factor() – to'ldirishkoeffitsientiniqaytaradi (yuqorigaqarang)
- max_load_factor() –

Agarargumentsizfoydalanilsa, maksimalto'ldirishqiymatiuchunfloat tipiniqaytaradi, agarargumentsifatida float tipifoydalanilsa, maksimalto'ldirishqiymatiuchuno'rnatiladi.

Agarterminatoromiliushbuchegaradanoshsa,

konteyneravtomatikravishdasegmentlarsoninioshirishimumkin.

-rehash(size_type count) – konteyneruchunqaytaxeshlasho'tkazadi, ya'niyacheykalarsonini (bucket_count) bucket_count $\geq$ count va bucket_count $>$ size/max_load_factor bo'lishiuchun.

-reserve(size_type count) – songayacheykalarsonnio'rnatadi, konteynerniqaytaxeshlasho'kazmaslikvamaksimalto'ldirishkoeffitsientiu chun, oxirgiyacheykaningcount hisoblanadi.

Maksimalsamaradorlikkaerishishuchunhardoimmaksimalyacheykalarsoninianiqlashkerak. Tavsiyaetilganqiymatlar 0.7 dan 0.8 gachaoraliqdabo'ladi.

Elementqidirishuchunstandartmaksimalyacheykalarsonio'rnatiladi (1.0 gateng) vadastuochilartomonidan 0.7 o'rnatiladi.

Xesh funksiyasining o'z konteyner faoliyatini yaxshilash mumkin, ammo bu mavzudoi rasidanchi qib ketishga olib keladi.

Interfeys segmentlari.

Segment elementlari bilan bevosita ishlash uchun

(bog'langan ro'yxat sifatida) quyidagi usullardan foydalaniladi:

- `begin()`, `end()`, `cbegin()`, `cend()` – iteratorlar. E'tibor bering, bular segmentlar uchun, massiv uchun emas!

Konteynerlar uchun shularning analoglari yaratilgan.

- `size_type bucket_count()` – massivdagi segmentlar sonini qaytaradi.

- `size_type max_bucket_count()` –

muu mumkin bo'lgan maksimal elementlar soni

(tizing avajoriy qilishga bog'liq).

- `size_type bucket_size(size_type n)` –

ni indeksli segmentdagi elementlar soni.

- `size_type bucket(Key)` – kalitasosid segment sonini qaytaradi.

Bu usullar asosida xesh jadvalni vizual ko'rinishni tasavvur qilish mumkin.

3.6-dastur. Tartiblanmagankonteynerlardan foydalanish.

```
#include "stdafx.h"

#include <iostream>
#include <iomanip>
#include <unordered_set>
#include <random>
#include <ctime>
#include <chrono>
using namespace std;

int main() {
    default_random_engine rnd(time(0));
    uniform_int_distribution<unsigned> g(100, 999);
    unordered_multiset<int> un;
    size_t n = 60;
    un.reserve(n);
    while (n--) un.emplace(g(rnd));
    cout << "Konteynerdagi segmentlar soni: "
```



```

<< unm.bucket_count()
<< "\n"
<< "Maksimal to'ldirishlar ko'ffitsenti: "
<< unm.max_load_factor()
<< endl;
    auto start = chrono::system_clock::now();
    auto res = unm.find(200);
    auto end = chrono::system_clock::now();
    auto elapsed = end - start;
    cout << "Natija => "
<< elapsed.count()
<< endl;
    // O'zimiz tomondan Maksimal to'ldirishlar
    ko'ffitsenti o'rnatamiz
    unm.max_load_factor(0.7);
    unm.rehash(200);
    cout << "Konteynerdagi segmentlar soni: "
<< unm.bucket_count()
<< "\n"
<< "Maksimal to'ldirishlar ko'ffitsenti: "
<< unm.max_load_factor()
<< endl;
    start = chrono::system_clock::now();
    res = unm.find(200);
    end = chrono::system_clock::now();
    elapsed = end - start;
    cout << "Natija => "
<< elapsed.count()
<< endl;
    cout << " Hash tuzilmasi " << endl;
    for (size_t i = 0; i < unm.bucket_count(); i++)
    {
        cout << "Segment N:" << setw(3) << i << "
=> ";
        // Iteratorga lokal segmentlarni olish
        auto first = unm.cbegin(i);
        auto last = unm.cend(i);
        while (first != last) {

```

```

        cout << *first++ << " ";
    }
    cout << endl;
}
system("pause");
return 0;
}

```

3.6-dastur. Output

```

Konteynerdagi segmentlar soni: 32
Maksimal to'ldirishlar ko'ffitsenti: 1
Natija => 30023
Konteynerdagi segmentlar soni: 256
Maksimal to'ldirishlar ko'ffitsenti: 0.7
Natija => 10004
Segment N: 0 =>          Segment N: 1 =>
Segment N: 2 =>          Segment N: 3 =>
Segment N: 4 =>
Segment N: 5 =>          Segment N: 6 =>
Segment N: 7 =>          Segment N: 8 =>
Segment N: 9 => 646
Segment N: 10 =>         Segment N: 11 =>
Segment N: 12 =>         Segment N: 13 =>
Segment N: 14 =>
Segment N: 15 =>         Segment N: 16 =>
Segment N: 17 =>         Segment N: 18 =>
Segment N: 19 =>
Segment N: 20 =>         Segment N: 21 =>
Segment N: 22 =>         Segment N: 23 =>
Segment N: 24 => 940
Segment N: 25 =>         Segment N: 26 =>
Segment N: 27 =>         Segment N: 28 =>
Segment N: 29 =>
Segment N: 30 => 717     Segment N: 31 =>
Segment N: 32 =>         Segment N: 33 =>
Segment N: 34 =>
Segment N: 35 =>         Segment N: 36 =>
Segment N: 37 =>         Segment N: 38 =>
Segment N: 39 =>

```

Segment N: 40 =>	Segment N: 41 =>
Segment N: 42 =>	Segment N: 43 =>
Segment N: 44 =>	
Segment N: 45 => 841 738	
Segment N: 46 =>	Segment N: 47 =>
Segment N: 48 =>	Segment N: 49 =>
Segment N: 50 =>	Segment N: 51 =>
Segment N: 52 => 768	Segment N: 53 =>
Segment N: 54 =>	Segment N: 55 =>
Segment N: 56 =>	Segment N: 57 =>
Segment N: 58 =>	Segment N: 59 =>
Segment N: 60 => 631	Segment N: 61 =>
Segment N: 62 => 316	Segment N: 63 =>
Segment N: 64 =>	Segment N: 65 =>
Segment N: 66 =>	Segment N: 67 =>
Segment N: 68 =>	Segment N: 69 =>
Segment N: 70 =>	Segment N: 71 =>
Segment N: 72 =>	Segment N: 73 =>
Segment N: 74 =>	Segment N: 75 =>
Segment N: 76 =>	Segment N: 77 =>
Segment N: 78 =>	Segment N: 79 =>
Segment N: 80 =>	Segment N: 81 =>
Segment N: 82 =>	Segment N: 83 =>
Segment N: 84 =>	Segment N: 85 =>
Segment N: 86 =>	Segment N: 87 =>
Segment N: 88 =>	Segment N: 89 =>
Segment N: 90 =>	Segment N: 91 =>
Segment N: 92 => 498	Segment N: 93 =>
Segment N: 94 => 187 476	
Segment N: 95 =>	Segment N: 96 =>
Segment N: 97 =>	Segment N: 98 => 280
Segment N: 99 =>	Segment N:100 => 225
Segment N:101 =>	Segment N:102 =>
Segment N:103 =>	Segment N:104 =>
Segment N:105 => 973	Segment N:106 =>
Segment N:107 =>	Segment N:108 =>
Segment N:109 =>	Segment N:110 =>
Segment N:111 =>	Segment N:112 =>Segment

N:113 =>	Segment	N:114 =>	Segment
N:115 =>	Segment	N:116 => 960	Segment
N:117 =>	Segment	N:118 =>	Segment
N:119 =>	Segment	N:120 =>	Segment
N:121 => 630	Segment	N:122 =>	Segment
N:123 =>	Segment	N:124 => 567	Segment
N:125 =>	Segment	N:126 =>	Segment
N:127 =>	Segment	N:128 =>	Segment
N:129 =>	Segment	N:130 =>	Segment
N:131 =>	Segment	N:132 =>	Segment
N:133 => 351	Segment	N:134 => 227	Segment
N:135 =>	Segment	N:136 =>	Segment
N:137 =>	Segment	N:138 =>	Segment
N:139 =>	Segment	N:140 =>	Segment
N:141 =>	Segment	N:142 =>	Segment
N:143 =>	Segment	N:144 =>	Segment
N:145 =>	Segment	N:146 => 761	Segment
N:147 =>	Segment	N:148 => 145	Segment
N:149 =>	Segment	N:150 =>	Segment
N:151 =>	Segment	N:152 =>	Segment
N:153 =>	Segment	N:154 =>	Segment
N:155 =>	Segment	N:156 =>	Segment
N:157 =>	Segment	N:158 =>	Segment
N:159 =>	Segment	N:160 =>	Segment
N:161 =>	Segment	N:162 =>	Segment
N:163 =>	Segment	N:164 =>	Segment
N:165 =>	Segment	N:166 =>	Segment
N:167 => 242	Segment	N:168 =>	Segment
N:169 =>	Segment	N:170 =>	Segment
N:171 =>	Segment	N:172 =>	Segment
N:173 =>	Segment	N:174 =>	Segment
N:175 =>	Segment	N:176 =>	Segment
N:177 =>	Segment	N:178 =>	Segment
N:179 =>	Segment	N:180 => 458	Segment
N:181 =>	Segment	N:182 =>	Segment
N:183 =>	Segment	N:184 =>	Segment
N:185 =>	Segment	N:186 =>	Segment
N:187 =>	Segment	N:188 =>Segment	N:189 =>

Segment N:190 =>	Segment N:191 =>
Segment N:192 => 852	Segment N:193 =>
Segment N:194 =>	Segment N:195 =>
Segment N:196 =>	Segment N:197 => 287
Segment N:198 =>	Segment N:199 =>
Segment N:200 =>	Segment N:201 =>
Segment N:202 =>	Segment N:203 =>
Segment N:204 =>	Segment N:205 =>
Segment N:206 =>	Segment N:207 =>
Segment N:208 =>	Segment N:209 => 483
Segment N:210 => 697	Segment N:211 =>
Segment N:212 =>	Segment N:213 =>
Segment N:214 =>	Segment N:215 => 967
Segment N:216 =>	Segment N:217 =>
Segment N:218 =>	Segment N:219 =>
Segment N:220 =>	Segment N:221 =>
Segment N:222 =>	Segment N:223 =>
Segment N:224 =>	Segment N:225 =>
Segment N:226 =>	Segment N:227 =>
Segment N:228 =>	Segment N:229 => 666
Segment N:230 => 484	Segment N:231 =>
Segment N:232 => 603	Segment N:233 =>
Segment N:234 =>	Segment N:235 => 254
Segment N:236 =>	Segment N:237 =>
Segment N:238 =>	Segment N:239 =>
Segment N:240 =>	Segment N:241 =>
Segment N:242 =>	Segment N:243 =>
Segment N:244 =>	Segment N:245 =>
Segment N:246 =>	Segment N:247 =>
Segment N:248 =>	Segment N:249 =>
Segment N:250 =>	Segment N:251 =>
Segment N:252 =>	Segment N:253 =>
Segment N:254 =>	Segment N:255 =>

Tartiblanmagankonteynerlarning sinflarimurakkab tuzilishiga, ammostandart vositalar bilan ham tartiblangankonteynerlardan kambo‘l magan mukammal qilishni ko‘rsatadi.

Tartiblangankonteynerlari yoki tartiblanmagankonteynerlar orasida gitano vazifa bog‘liq bo‘ladi.

Assotsiativ konteynerlarning asosiy usullari keng tarqalganligi sababli, dasturning bir bajarilishidan boshqasiga o'tish oson va testlarni o'tkazganingizdan so'ng tanlovingizni bir yoki bir nechta turlik konteynerda amalga oshirib, tanlov qilishingiz mumkin.

Tartiblanmagan konteynerlarning sinflari xususiyatlari.

Headers		<unordered_set>		<unordered_map>	
Members		unordered_set	unordered_multiset	unordered_map	unordered_multimap
	<i>constructor</i>	unordered_set	unordered_multiset	unordered_map	unordered_multimap
	<i>destructor</i>	~unordered_set	~unordered_multiset	~unordered_map	~unordered_multimap
	<i>assignment</i>	operator=	operator=	operator=	operator=
iterators	begin	begin	begin	begin	begin
	end	end	end	end	end
	rbegin				
	rend				
const iterators	cbegin	cbegin	cbegin	cbegin	cbegin
	cend	cend	cend	cend	cend
	crbegin				
	crend				
capacity	size	size	size	size	size
	max_size	max_size	max_size	max_size	max_size
	empty				

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Assotsiativ konteynerlarga qaysi sinflar kiradi?

2. C++

barcha assotsiativ konteynerlar qanday amallarni qo'llab-quvvatlaydi?

3. Key1 va key2 kalitlari o'zaro teng hisoblanadi,

agartaq qoslanadigan ob'ektlar uchun qaysi funksiyachini qiymat qaytaradi?

4. Qanday konteynerlar uchun aniqlanadigan tip va kalitni bir xil bo'ladi?

5. Assosiativ konteynerlarning iteratorlarine chatomolam iteratorlarsini fikirladi?
6. Kalitasida qidirishdaberilgan kichik bo'lmagan birinchi elementni qanday funksiyavabirinchi kattaelementni qanday funksiyayordamida amalga oshiriladi?
7. Set
to'plamda qiymatni o'zgartirish uchun avval nima funksiyani bajarib, so'ngi yangi qiymatni yoki o'zgartirilgan varianti qo'shiladi?
8. Xotirani ajratish va bo'shatish operatorlari set
sinf uchun sanab bering?
9. Ishoralib tutun tip, aniq tipni belgilash realizatsiya qilishga bog'liq va Allocator da aniqlanadi. Gap qaysi funksiyahaqida.
10. To'plam - assosiativ konteyner to'plam, teng qiymatli kalitlarni saqlaydi (mumkin qadar birkalit qiymatli elementlarni to'plamini saqlaydi) va kalitni o'qitish qidirish imkonini beradi. Gap qaysi to'plam haqidabormoqda.
11. set va multiset sinflarining ob'ektlarining tipikali bilan yonma-yon bittashablonli parametrolish imkon. Bittashablon qaysi funksiya?
12. [=] (operator=) operatorni nima uchun ishlatiladi?
13. To'plam vektorning massivlarini tasodifiy sonlar bilan to'ldiradigan va belgilangan qiymatni qidiradigan dasturtuzing.
14. Pair qandaysinfi va nima uchun kerak?
15. p.first –
havola bo'yicha juftlikning nechanchi elementiga murojaat?
16. Lug'atda kalitlari unikal (takrorlanmaydigan) va qaysi sinfdan iborat nusxa dagikalitlari bo'ladi.
17. Oddiy map konteyner konstruktorlari: map<Key, T, comp> ar; va map<Key, T, comp> ar(Comp);
ni ishlatish tartibini tushuntirib bering.
18. Ro'yxatni initsializatsiya qilishda har bir juftlik alohida qanday qasbi uchun amalga olinishi kerak.
19. Lug'atlarda elementlarga kirishning yana ikkita usuli mavjud (iteratorlar bilan ishlatishdantashqari).
Birinchi va ikkinchi usularni tushuntirib bering.
20. Indeksash funksiyasini assosiativ konteyner sinflarining qaysi sinflariga foydalanish imkon.
21. at() funksiyasi istisno qaytargani uchun try - catch funksiyasi bilan boshqarish imkonmi?

22. operator[] noqulayvasamarasizdebhisoblasangiz, yanaungaikkitaalternativvariantlarbor. Buvariantlarniayting.
23. Qaysito'plamdaqidirishfaqatbirinchimaydon, ya'nikalitmaydonbilanamalgaoshiriladi.
24. Qandaykonteynerlarxesh — jadvalasosidaqurilganvaxeshfunksiyalargavatenglikamaligatayanadi.
25. Ayrimvazifalardasaralashuchunvaqtiko'pajratilsaavtomatiksaralashnimuhimkamchilikdebhisoblashmumkin. Bundadasturchiqandayyo'ltutadi.
26. Tartiblanmagankonteynerlarningxeshjadvallaridanimaborvaularcheksizko'pelementlarnio'zichigaolishimumkin.
27. Tartiblanmagankonteynerlarningtiplariuchunshablonparametrlarinisanabbering.
28. Tartiblanmagankonteynerlardaagarequaltaqqoslashfunksiyasiyozilmaganbo'lsa, joriyholatbo'yichaqaysifunksiyaishlatiladi.
29. Xeshlashuslubinimauchunkerak.
30. rehash(size_type count) — konteyneruchunqaytaxeshlasho'tkazadi, ya'niyacheykalarinisoniniqaysishartdabo'lishiuchun.
31. Segmentelementlaribilanbevositaishlashuchunqandayusullardanfoydalaniladi.
32. Tartiblangankonteynerlaryokitartiblanmagankonteynerlarorasi dagitanlovnimagabog'liqbo'ladi.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
🎓	Assosiativkonteynerlarga (set, multiset) oidberilganquyidagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba'zio'zgartirishlarnitopishorqalitopshiriqlar bosqichma – bosqichamalganoshiriladi.
dastur	topshiriqlar

<pre>// Created by MBBahodir #include "stdafx.h" #include <iostream> #include <set> using namespace std; void myPrint(set<int>&my set){ for (set<int>::iterator it=myset.begin(); it!=myset.end(); ++it) cout << ' ' << *it; cout << '\n'; } void myPrintmulti(multis et<int>&myset){ for (multiset<int>::ite rator it=myset.begin(); it!=myset.end(); ++it) cout << ' ' << *it; cout << '\n';</pre>	1. Dasturdanechtaglobalfunksiyayaratilgan.
	2. Dasturdanechtaset to‘plammavjud.
	3. Dasturdanechtamultiset to‘plammavjud.
	4. Set vamultiset to‘plamlarningfarqidasturningqaysifragmentidako‘rsatilgan .
	5. Nimauchunoxirgi myPrint(myset);funksiyaishlamaydi.
	6. reverse_iteratornio‘rnigaiteratornio‘rn atsa, nimabo‘ladi. Xatobo‘ladi. Xatonituzatishuchunnimaqilishkerak.

<pre> } void mySizeSets(set<int> &myset){ cout << int (myset.size()) << '\n'; } int main () { set<int> myset; set<int>::iterator it; int myints[]={ 12,82,37,64,15,15 }; set<int> first (myints,myints+6); multiset<int> mfirst (myints,myints+6); set<int> second; second = first; myPrint(first); myPrintmulti(mfirst); mySizeSets(second); mySizeSets(first); </pre>	7. myset.swap(second);nio‘rnigasecond.swap(first); yozilsanimahodisasodirbo‘ladi. 8. Set vamultisetningoperator=dasturningqaysiqismlaridafoydalanilganvanimauchun.

```

    for (int i=1;
i<=5; i++)
myset.insert(i*10);

    myPrint(myset);

it=myset.find(20);
    myset.erase (it);
    myset.erase
(myset.find(40));

myset.swap(second);

    for (
set<int>::reverse_i
terator
rit=myset.rbegin();
rit !=
myset.rend();
++rit)
        std::cout << '
' << *rit;
        cout << '\n';

    while
(!myset.empty())
    {
        cout << ' ' <<
*myset.begin();

myset.erase(myset.b
egin());
    }

    myPrint(myset);

```


<pre>tring,int>&myMultimap){ for (auto& x : myMultimap) cout << " [" << x.first << ':' << x.second << ']'; cout << endl; } int main () { map<char,int> mymap, mymap_one; map<char,int>::iterator it; pair<map<char,int>::ite rator,map<char,int>::it erator> ret; mymap['a']=10; mymap_one['x'] = 100; mymap['b']=20; mymap_one['y'] = 200; mymap['c']=30; mymap_one['z'] = 300; multimap<std::string,in t> mymultimap; mymultimap.emplace("Ali ",1.50);</pre>	_____
	3. mapvamultimap to‘plamlarningfarqidasturningqaysif ragmentidako‘rsatilgan . _____ _____
	4. Qachonoxirgi myPrint(mymap);funksiyaishlamay di. _____ _____ _____
	5. mymultimapnio‘rnigamapnio‘r natsa, nimabo‘ladi. Xatobo‘ladi. Xatonituzatishuchunnimaqilishkera k. _____ _____ _____
	6.mymap.swap(mymap_one) nio‘rnigamymap_one.swap(mymap)yozilsanimahodisasodirbo‘ladi. _____ _____ _____

<pre> mymultimap.emplace("Val i",2.71); mymultimap.emplace("Sol i",1.40); mymultimap.emplace("Sod iq",3.14); it = mymap.find('b'); if (it != mymap.end()) mymap.erase (it); myPrint(mymap); myPrint(mymap_one); /* while (!mymap.empty()) { std::cout << mymap.begin()->first << " => " << mymap.begin()->second << '\n'; mymap.erase(mymap.begin ()); } */ mymap.swap(mymap_one); myPrint(mymap_one); myPrint(mymap); </pre>	7. map vamultimapningoperator=dasturning qaysiqismlaridafoydalanilganvanima uchun.
	8. Dasturdagiizohgaolingandastur fragmentiolibtashlansa, dasturdaqandayo‘zgarishlarbo‘ladiva natijadachi.

<pre> ret = mymap.equal_range('b'); cout << ret.first- >first << " => " << ret.first->second << endl; cout << ret.second- >first << " => " << ret.second->second << endl; myPrintmulti(mymultimap); system("pause"); return 0; } </pre>	
9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

UCHINCHI ASSISMENT TOPSHIRIG‘I	
	Tartiblanmaganassosiativkonteynerlarga (unordered_set, unordered_multiset) oidberilganquyidagidasturbo‘yichaberilgantopshiriqlar keraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalito pshiriqlarbosqichma – bosqichamalganoshriladi.
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include <iostream> #include <string> </pre>	1. Dasturdanechtaglobalfunksiyayaratilgan.

<pre> #include <unordered_set> using namespace std; template<class T> T cmerge (T a, T b) { T t(a); t.insert(b.begin(),b .end()); return t; } void myPrint(unordered_se t<string>&myset){ for (const string& x: myset) cout << " " << x; cout << endl; } void myPrintmultiset(unor dered_multiset<strin g>&mymultiset){ for (const std::string& x: mymultiset) std::cout << " " << x; std::cout << std::endl; } int main () { </pre>	2. Dasturdanechta unordered_set to‘plammavjud.
	3. Dasturdanechtaunordered_multiset to‘plammavjud.
	4. DasturdamyPrint(myset_one)funksiy asiyaqirilgandanatijaqandaybo‘ladivanim asababdan.
	5. To‘plamdanberilganqiymatniizlashd asturfragmentinianiqlangvamymset_oneuch unhamizlashfunksiyasiniyarating.
	6. myset.rehash(12)nimavazifanibajara di.

<pre>unordered_set<string> myset, myset_str; unordered_set<int> myset_one;</pre>	7. myset_str.swap(myset); nio'rni gamyset.swap(myset_str); yozilsanimahodisasodirbo'ladi.
<pre>unordered_multiset<string> first, second, third; myset.emplace("Laseti"); myset_one.insert(1); myset.emplace("Tico"); myset_one.insert(15); ; myset.emplace("Damas"); myset_one.insert(45); ; myPrint(myset); for (const int& x: myset_one) cout << " " << x; cout << endl; string input; cout << "Mashina nomini kirit: "; getline(cin, input); unordered_set<::string>::const_iterator got =</pre>	8. operator= dasturning qaysiqismlarida foydalanilgan va nima uchun. 9. myPrintMultiset(second); funksiyasie kranganimanatijanichiqaradi.
	10. third = cmerge (first, second); funksiyasini mavazifani bajaradigan natijasig anday.

```

myset.find(input);

    if ( got ==
myset.end() )
        cout <<
"to'plamda yo'q";
    else
        cout << *got <<
" nomli mashina
bor";
        cout << endl;

        myset.rehash(12);

myset.insert("mashin
a");

myset.insert("uy");

myset.insert("daraxt
");

myset.insert("eshik"
);

myset.insert("choyna
k");

    cout << "joriy
bucket_count: " <<
myset.bucket_count()
<< endl;

myset_str.swap(myset
);

```

```
    cout << "first
=>";
    myPrint(myset);

    cout << "second
=>";

myPrint(myset_str);

first.insert("www.uz
");
first.insert("tuit.u
z");
first.insert("edu.uz
");

second.insert("www.u
z");
second.insert("uzedu
.uz");
second.insert("edu.u
z");
    third = cmerge
(first, second);
    first = third;

    std::cout <<
"first =>";

myPrintmultiset(firs
t);

    second.clear();

second.insert("bed")
;
```

<pre> second.insert("bed") ; second.insert("wardr obe"); second.insert("night stand"); cout << "second =>"; myPrintmultiset(seco nd); string mystring = "red"; second.insert (mystring); second.insert (mystring+"dish"); cout << "second =>"; myPrintmultiset(seco nd); system("pause"); return 0; } </pre>	
11. Dasturda jamibo‘lib, nechao‘zgartirish kiritildi. _____ 12. Shudasturning analogini yaratish sizga mustaqil vazifa dir.	

TO‘RTINCHI ASSISMENT TOPSHIRIG‘I	
	Tartiblanmagan assosiativ konteynerlarga (<code>unordered_map</code> , <code>unordered_multimap</code>)

	oidberilganquyidagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba'zio'zgartirishlarnitopishorqalitopshiriqlarbosqichma — bosqichamalganoshiriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include <iostream> #include <string> #include <unordered_map> using namespace std; template<class T> T cmerge (T a, T b) { T t(a); t.insert(b.begin(),b.end()); return t; } void myPrint(unordered_map<int,string>&mymap){ for (auto& x: mymap) cout << "[" << x.first << "," << x.second << "]" ; cout << endl; } void myPrintmultimap(unordered_multimap<int,string>&mymultimap){ for (auto& x: mymultimap) cout << "["</pre>	1.Dasturdanechtaglobalfunksiya yaratilgan.
	2. Dasturdanechta unordered_map to'plam mavjud.
	3. Dasturdanechtaunordered_multimap to'plam mavjud.

<pre> << x.first << "," << x.second << "]" ; cout << endl; } int main () { unordered_map<int,string> mymap, mymap_str; unordered_map<int,int> mymap_one; unordered_multimap<int, string> first, second, third; mymap.emplace (80686,"Laseti"); mymap_one.insert(make_p air<int,int>(1,2)); mymap.emplace (10010,"Tico"); mymap_one.insert(make_p air<int,int>(15,16)); mymap.emplace (70707,"Damas"); mymap_one.insert(make_p air<int,int>(45,46)); myPrint(mymap); for (auto x: mymap_one) cout << "[" << x.first << "," << x.second << "]" ; </pre>	4. DasturdamyPrint(mymap_one) funksiyasiyaaqirilgandanatijaqanday bo‘ladivanimasababdan. 5. To‘plamdanberilganqiymatniiz lashdasturfragmentinianiqlangvamy map_oneuchunhamizlashfunksiyasi niyarating. 6. mymap.rehash(20)nimavazifan ibajaradi. 7.mymap_str.swap(mymap);n io‘rnigamymap.swap(mymap_str); yozilsanimahodisasodirbo‘ladi.

<pre> cout << endl; int input; cout << "Mashina raqamini kirit: "; cin >> input; unordered_map<int, string>::const_iterator got = mymap.find(input); if (got == mymap.end()) cout << "to'plamda yo'q"; else cout << "[" << got->first << "," << got->second << "]" << " raqamli mashina bor"; cout << endl; mymap.rehash(20); mymap[1] = "mashina"; mymap[10] = "uy"; mymap[15] = "daraxt"; mymap[20] = "eshik"; mymap[25] = "choynak"; cout << "joriy bucket_count: " << mymap.bucket_count() << endl; </pre>	8. operator= dasturning qaysiqismlarida foydalanilgan vanima uchun. 9. myPrintmultimap(second); funksiya iekrangan imanat ijanichiqaradi. 10. third = cmerge (first, second); funksiyasini mavazifani bajaradi vanat ijasiganday.

```

mymap_str.swap(mymap);

    cout << "first =>";
    myPrint(mymap);

    cout << "second =>";
    myPrint(mymap_str);

    pair<int,string>
mypair (1,"www.uz");
    pair<int,string>
mypair_one
(2,"tuit.uz");

    first.insert
(mypair);

second.insert(mypair_on
e);
    first.insert
(pair<int,string>(50,"G
OOG"));

second.insert(pair<int,
string>(501,"GGOD"));
    first.insert
(second.begin(),
second.end());

    third = cmerge
(first, second);
    first = third;

    cout << "first =>";

myPrintmultimap(first);

    second.clear();

```


<pre> second.insert(pair<int, string>(80,"BUXORO")); second.insert(pair<int, string>(01,"TOSHKENT")) ; second.insert(pair<int, string>(70,"QARSHI")); second.insert(make_pair (75,"TERMIZ")); cout << "second =>"; myPrintmultimap(second) ; system("pause"); return 0; } </pre>	
	11.Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 12.Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

1.4.Konteynerlarningadapterlari.

 Adapterdarning muhim tushunchalari asosida tek, navbat, ustivorna navbat, ikki tomonlama navbat sinf ob'ektlari va ularning xususiyatlari, funksiyalari, dasturlashda o'zlarini tutishlari, konteynerlar bilan ishlashga mo'ljallangan asosiy algoritmlar va ularning dasturlashdagi ahamiyati, funkto funktsiyalarini yaratish va foydalanish uslublari, amallari, talablari, vazifalarini va usullarini keltirilgan bo'lib, dasturchining adapterlar, algoritmlar va funkto funktsiyalar bilan ishlashdagi tanlash imkoniyati bo'yicha tajribaxulosalari, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan. Bilimlarni mustahkamlash uchun

tanazariysavolvaamaliyko'nikmavamalakalarnirivojlantrishuchun 5
taassimenttopshirig'ivaharassimentda 10(9,7) tatopshiriq, jami
46tatopshiriqberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍**Kalitso'zlar.**Konteyner, adapter, shablon, sinf, moslashtirish, stack (stek), queue (navbat), priority_queue (ustuvorbilannavbat), deque (ikkitomonlamanavbat), ma'lumotlartuzilmasi, push_back, pop_back, pop_front, LIFO, FILO, LIFO, for_each(), find(), funksionalob'ektlar, Operator(), Standart ML vafunktor, Haskell vafunktor.

✓**Bilishshartbo'lgantushunchalar.**Tiptushunchasi, sinfvasinfob'ekti, to'plam, statikvadinamikmassiv, elementgamurojaat, funksiyavako'rsatkich, konteyner, ketma-ketkonteynerlar, xesh, tasodifiy, istisnoholat, ma'lumotlarnikirishvachiqishi, oqim, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo'llabquvvatlovchimu hitdaishlashnibilishlozim.

✎**Bilibolasiz.** stack (stek), queue (navbat), priority_queue (ustuvorbilannavbat), deque (ikkitomonlamanavbat) sinfob'ektlarivaularningxususiyatlari, funksiyalari, dasturlashdao'zlarinitutishlari (LIFO, FILO, LIFO), konteynerlarbilanishlashgamo'ljallanganasosiyalgoritmlar (for_each(), find()) vaularningdasturlashdagiahamiyati, funktorfunksiyalarniyaratishvafoydalanishuslublari, amallari, talablari, vazifalarivausullarinio'rganishingiz mumkin.

REJA

1. Stack, queue, priority_queue, deque.
2. Konteynerlarbilanishlashalgoritmlari.
3. Funktorlarningqo'llanilishi.

KIRISH

Adapterlarstandartishablonkutubxonasiningalohidatoifasidir (vakilidir). Adapterlaryangitushunchalaryokiilovalaremas, balkimavjudkutubxonatushunchalariningo'zigaxos, tez-tezishlatiladiganmaqsadlaruchunmoslashtirishlari. Bumoslashtirish (adaptatsiyalash) ko'pinchaadapterningso'rovlarnnigasosiytushunchasigaasoslanib, funksiyalarinicheklashorqaliamalgaoshiriladi. Kutubxonakonteynerlarningadapterlari, iteratorlarvafunksiyalarnio'zichigaoladi.

Konteynerlarning adapterlari. Konteynerlarning adapterlari asosida adapterlarni hosil qilishning eng so'ngi yo'limi soliqaliti tushuntirishdir. Bular stack (stek), queue (navbat), priority_queue (ustuvor bilan navbat) misol bo'ladi. Buni soliqaliti tushuntirish bo'ldiki, adapterlar:

- juda keng vatez-tez ishlatiladigan ma'lumotlarni tuzilmalari;
- har qanday realizatsiya alohida bajarilish hakerak emas.

Ular nining funksiyalarini ta'minlash uchun STL standart konteyneri har qanday moslashtirishda tayanch sifatida push_back, pop_back, yoki pop_front amallardan (adapterni tuzilgacha qarab) foydalanish mumkin;

- adapter uchun tayanch konteyner nining to'plamidan qo'shimcha amallarni chiqarib tashlanish hakerak (vahimalarni yaratmaslik uchun, masalan, indekslash amali uchun, konteyner vektor bo'lsa, stek moslashtirish ishlatiladi);

Konteynerlarning adapterlari uchun sintaktik ta'riflarni soliqaliti qalitiqarib chiqishni yaxshilash natijasi beradi, shuning uchun misollar bilan tushuntirish bo'lamiz.

Birinchi navbatda konteynerlarning adapterlari uchun <stack>, <queue> kabi sinflarni dasturga qo'shish lozim.

Stek (stack, ma'lumotlarni yig'indisi): stekda uning elementlariga faqat bir uchidan murojaat qilish mumkinligi bilan xarakterlanadi va stekning yuqori qismida bataladi. Bu LIFO (Last In — First Out, birinchi kirgan oxir chiqadi) tamoyili bo'yicha faoliyat ko'rsatuvchi ma'lumotlarni to'plamidir.

Stekka oid deyarli barcha funksiyalar niko'rsatadigan dastur keltirilgan (4.1-dastur).

4.1-dastur. Stekka oid funksiyalardan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <vector>
#include <stack>
#include <string>
#include <array>
using namespace std;

int main() {
```

```

stack<int> st, st_one, st_two;
stack<string,vector<string>> vst;

vst.emplace("str1");
vst.emplace("str2");
vst.push("eostr");

    for (int i=0; i<5; ++i) st.push(i);

    for (int i = 0; i < 15; i++)
st_one.emplace(i*10);

    st_two = st;
    st.swap(st_one);

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    while (!vst.empty()) {
        cout << vst.top() << " : stack dagi o'rni " <<
vst.size() << endl;
        vst.pop();
    }

    int size = st.size();
    while (size--) {
        cout << st.top() << " : stack dagi o'rni " <<
st.size() << endl;
        st.pop();
    }

    while (st.size() != 0){
        cout << st_two.top() << " : stack dagi o'rni "
<< st_two.size() << endl;
        st_two.pop();
    }

```

```

while (!st.empty()) {
    cout << st.top() << " : stack dagi oʻrni " <<
st.size() << endl;
    st.pop();
}

cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

cout << " stack st " << (st.empty() ? "" : "not ")
<< " empty" << endl;
cout << " stack vst" << (vst.empty() ? "" : "not
") << " empty" << endl;
cout << " stack st_one " << (st_one.empty() ? "" :
"not ") << " empty" << endl;
system("pause");
return 0;
}

```

4.1-dastur. Output.

```

15 5 5 3
eostr : stack dagi oʻrni 3
str2 : stack dagi oʻrni 2
str1 : stack dagi oʻrni 1
140 : stack dagi oʻrni 15
130 : stack dagi oʻrni 14
120 : stack dagi oʻrni 13
110 : stack dagi oʻrni 12
100 : stack dagi oʻrni 11
90 : stack dagi oʻrni 10
80 : stack dagi oʻrni 9
70 : stack dagi oʻrni 8
60 : stack dagi oʻrni 7
50 : stack dagi oʻrni 6
40 : stack dagi oʻrni 5
30 : stack dagi oʻrni 4
20 : stack dagi oʻrni 3
10 : stack dagi oʻrni 2
0 : stack dagi oʻrni 1

```

0 5 5 0

```
stack st empty  
stack vst empty  
stack st_one not empty
```

4.1

dasturdakeltirilganstekadapteriningfunksiyalaribo'yichadasturnitahlilqili shorqaliba'zixulosalarkeltiramiz:

- Stack<string>elementlarisatrbo'lgano'zgaruvchinie'lonqiladi. String ob'ektlaro'zlari STL konteynerlarekanligigae'tiborbering. Shundayqilib, stekkonteynerlarningharqandaye elementlarinio'zichigaolgankonteynerlar niolishimumkin (buboshqa STL konteynerlariuchunxosdir).

- stack<string>e'lonqilishnimisollarningko'pchiligidastekkatavsifsifatidak o'rishmumkin.

Ko'pchilikdasturchibuniboshqabo'lishibilmasligivatassafurqilmasligimu mkin. Lekinbixilta'rifmavjud stack<string, vector<string>>satrlaribirsteki, bundatayanchsinflars ifatidafoydalanish. Ba'zanso'rashdi, nimauchun stack<vektor<string>>yozishmumkinemas (birinchi stringnio'chirganholda)? Chunki stack konstruktoriniamalgaoshirishdabuyo'lbilgilanganvabujudamumkin.

Ammo, butunlayboshqaturdagitavsifbo'ladi: string vektorlarito'plami stack<vektor<string>>umumanboshqastekbo'ladi (konteynerlarningtarkibiyjoylashtirishhaqidagieslatma).

- Judako'pholardasteknivektorko'rinishidainitsializatsiyaqilinda, ammobufoqatC++11variantidamavjud. 4.1-dasturdaemplace() funksiyasidanstekkaelementqo'shishuchunfoydalanilgan.

- Keyingifragmentlarda, deyarlibarchaamallarni (usullarini) ko'rishmumkin: stack:push() – stekkabittaelementqo'shish, top() – stekningengyuqoridaturganelementigako'rsatkicholish, pop() – yuqorielementinichiqaribtashlash, size() – stekningjoriyhajmi, empty() – stekningbo'shliginitekshirish, swap() – ikkistekningalmashtirish, operatop= stekniqiymatqilibboshqabirtekkaberish (o'zlashtirish).

- Dasturdanko'rishmumkinki, stekadapteritayanchkonteynergaxosusullarniyo'qotdi (at(), operator[]

vahokazo.), lekin push(), pop()
funksiyalarniqaytaaniqlashorqaliorttirilganhisoblanadi.

Steknitushinibolib,
o'xshashfunksiyalarninavbatgajoriyqilishqiyinmasalaemas.
Stekdanfarqlio'laroq, navbat FIFO (First In — First Out
birinchikirganbirinchichiqadi)
tamoyilibo'yichafaoliyatyuritadiganma'lumotlarto'plamidir.
(Buusulnibiruchigaoqibkirish,
keyinboshqauchidanoqibchiqadiganquvurgao'xshatishmumkin).

4.1-

dasturdatilsemantikasitalabibo'yichatahrirlarniqilibnavbatuchundeyarlio'
zgarishsizqoldiramizva 4.2 dasturniolamiz.

4.2-dastur. Navbatsinfiningfunksiyalari.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <list>
#include <queue>
#include <string>
#include <array>
using namespace std;

int main() {

    queue<int> st, st_one, st_two;
    queue<string, list<string>> vst;

    vst.emplace("str1");
    vst.emplace("str2");
    vst.push("eostr");

    for (int i=0; i<5; ++i) st.push(i);

    for (int i = 0; i < 15; i++)
st_one.emplace(i*10);

    st_two = st;
    st.swap(st_one);
```

```

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    while (!vst.empty()) {
        cout << vst.front() << " : stack dagi o'rni "
<< vst.size() << endl;
        vst.pop();
    }

    int size = st.size();
    while (size--) {
        cout << st.front() << " : stack dagi o'rni "
<< st.size() << endl;
        st.pop();
    }

    while (st.size() != 0){
        cout << st_two.front() << " : stack dagi o'rni
" << st_two.size() << endl;
        st_two.pop();
    }

    while (!st.empty()) {
        cout << st.front() << " : stack dagi o'rni "
<< st.size() << endl;
        st.pop();
    }

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    cout << " stack st " << (st.empty() ? "" : "not ")
<< " empty" << endl;
    cout << " stack vst" << (vst.empty() ? "" : "not
") << " empty" << endl;
    cout << " stack st_one " << (st_one.empty() ? "" :
"not ") << " empty" << endl;

```



```

    system("pause");
return 0;
}

```

4.2-dastur.Output

```

15 5 5 3
str1 : stack dagi oʻrni 3
str2 : stack dagi oʻrni 2
eostr : stack dagi oʻrni 1
0 : stack dagi oʻrni 15
10 : stack dagi oʻrni 14
20 : stack dagi oʻrni 13
30 : stack dagi oʻrni 12
40 : stack dagi oʻrni 11
50 : stack dagi oʻrni 10
60 : stack dagi oʻrni 9
70 : stack dagi oʻrni 8
80 : stack dagi oʻrni 7
90 : stack dagi oʻrni 6
100 : stack dagi oʻrni 5
110 : stack dagi oʻrni 4
120 : stack dagi oʻrni 3
130 : stack dagi oʻrni 2
140 : stack dagi oʻrni 1
0 5 5 0
stack st empty
stack vst empty
stack st_one not empty

```

4.1-dasturda oʻzgartirish talab qilinganlarini oʻzgartirib, quyidagixulosalarga olib keladi:

- navbatda `pop_front()` usulivektorkonteynerustiga qurib boʻlmaydi, lekin uni roʻyxat konteyner uchun qurish mumkin.
- Deque sinfi uchun `front()`, `push_back()`, `pop_front()`, usullari har qanday konteynerlarda ishlatiladi (faqat `pop_front()` usuliborlarida).
- shuning uchun navbat uchun yasala digan obʻektda, yaʼni deque sinfida `top()` usul oʻrniga `front()` usuli ishlatiladi.

Maʼlum birtartibda qayta ishlanish hakerakmaʼlumotlarni uchun umumiy maʼlumotlarni tuzilishiga asoslangan toʻplamibu – stekvanavbatdir (stack,

deque). Masalan, biror funksiya o'z navbatida uchinchifunksiyanicha qiradigan boshqa funksiya anicha qirsa, uchinchifunksiya birinchifunksiyanimas, ikkinchifunksiyaga qaytish muhimdir.

Bu ma'lumotlarni qayta shlashtartibini amalga oshirish uchun dasturchi o'zining navbat funksiyasini tashkil qilish lozim. Stekka qo'shilgan oxirgi qiymat birinchi bajariladi va aksincha, stekka qo'shilgan birinchi qiymat oxirgi bajariladi. Shunday qilib, ma'lumotlarni tuzilmasining o'zicha qirishlarning to'g'ri belgilangan tartibdab erilishini ta'minlaydi.

Konseptual, ma'lumotlarni tuzilishi - stek juda oddiy: u ma'lumotlarni kiritish yoki chiqarish uchun ma'lum birtartibni belgilaydi. Har safar birlamchi element qo'shiladi, buni stekka yuqori element bo'lib tutaydi. Stekdan olib tashlanadigan yagona element stekning yuqori qismida joylashgan elementdir. Shunday qilib, stekka "birinchi kirish, oxirgi chiqish — FILO" yoki "oxirgi kirish, birinchi chiqish — LIFO" deb ham aytiladi. Demak, Stekka qo'shilgan birinchi element undan oxirgi chiqariladi.

Xo'sh, buni qanday qilib? Nima uchun stek kerak? Yuqorida aytib o'tganimizdek, funksiyalarni (qiymatlarni) chaqirishni tashkil qilish uchun qulay yordam beradigan aslida, stek chaqirish (call stack) tez-tez ishlatiladigan yoki boshqa funksiyalarning qaytish qiymatini kutayotgan funksiyalar ro'yxatiga murojaat qilish uchun ishlatiladigan amaldir.

Shuningdek, stek informatika fanining fundamental tushunchalarining bir qismidir. Agar birinchi kelib, oxirish ketish tamoyili asosida fikrlasangiz, unda stek terminini ishlatish lozim.

Bunda tashqari, bunday (stek kabi) navbatlarni o'zaro qaytarishni ta'minlaydi, nazariy informatika dasturlash, bunday push-down funksiyalarini avajdagi hamkorlik bag usullarida.

Stek assosiativ usullarga ega:

Push – stekka element qo'shish..

Pop – stekdan element o'chirish

Top – elementni ko'rish.

LIFO - stek xarakati.

FILO - stek xarakati (ekvivalent LIFO ga).

Stack (stek)

Ma'lumotlarni tuzilmasini amalga oshirish uchun yaratish masalani qaraymiz.

Umumlashgan dasturlash usuliga mos keladigan umumiy stack, ya'ni Stack sinfi uchun shablon yaratish degani.

Agar shablonlar bilan tanish bo'lmasangiz,
 funksiyalar shablonlariga oid nazariy ma'lumotlarga qarang,
 shablonlar bilan ishlash mexanizmi batafsil bilish lozim.

Agar shablonlar bilan tanish,
 lekin sinf shablonlar bilan ishlashni unutmagan bo'lsangiz,
 sinf shablonlari haqidagi nazariy materiallarni o'qib o'rganing.

Bu stack shablonlar bilan amalga oshirilgan kutubxonaning deyarli har qanday ma'lumotlari uchun foydalanish mumkin. Bundan tashqari, stack hajmi dasturiy ro'sidavomida jadalbelgilanadi. Stackga qo'shimcha funksiyalar ham qo'shildi peek(), bu stackning yuqori qismidan n- elementini qaytaradi.

4.3-dastur. Stack shablonsini

```
#ifndef STACK_H
#define STACK_H
// Created by MBBahodir
#include <cassert>
#include <iostream>

#include <iomanip>

template <typename T>
class Stack{
private:
    T *stackPtr;           // stackka
    ko'satkich
    const int size;        // stackdagi
    maksimal elementlar soni
    int top;               // stackning
    joriy elementi
public:
    Stack(int = 10);        // stackning
    joriy holatda elementlar soni 10 ta
    Stack(const Stack<T>&); // nusxalash
    konstruktori
    ~Stack();              // destruktors

    inline void push(const T & ); // elementni
    stackning yuqori qismiga joylashtirish
};
```

```

        inline T pop(); // stack
ichidan eng yuqoridagi elementni o'chirish
        inline void printStack(); // stackni
ekranga chiqarish
        inline const T &Peek(int ) const; // stackning
yuqorisidagidam keyingi n-element
        inline int getStackSize() const; // stack
o'lchamini olish
        inline T *getPtr() const; // stackka
ko'rsatkich olish
        inline int getTop() const; // joriy
elementni olish
};

// Stack sinfi shablon funksiyalarini realizatsiyz
qilish

// konstruktorSteka
template <typename T>
Stack<T>::Stack(int maxSize): size(maxSize) //
o'zgarmaslarni initsalizatsiyalash
{
    stackPtr = new T[size]; // xotira ajratish
    top = 0; // joriy elementno 0 bilan toldirish;
}

// nusxalsh konstruktori
template <typename T>
Stack<T>::Stack(const Stack<T>& otherStack) :
    size(otherStack.getStackSize()) //
o'zgarmaslarni initsalizatsiyalash
{
    stackPtr = new T[size]; // yangi xotira
ajratish
    top = otherStack.getTop();

    for(int ix = 0; ix < top; ix++)
        stackPtr[ix] = otherStack.getPtr()[ix];
}

```

```

}

// destruktor
template <typename T>
Stack<T>::~~Stack()
{
    delete [] stackPtr; // ochirish
}

// element qo'shish
template <typename T>
inline void Stack<T>::push(const T &value)
{
    // o'lchamni o'zgartirish
    assert(top < size); // joriy element soni
    o'lchamdan kichik bo'lishi kerak

    stackPtr[top++] = value; // elementni
    joylashtirish
}

// elementni o'chirish
template <typename T>
inline T Stack<T>::pop()
{
    // o'lchamni tekshirish
    assert(top > 0); // joriy element 0 katta
    bo'lishi kerak

    stackPtr[--top]; // elementni o'chirish
    return 0;
}

// yuqori elementdan n-elementni qaytaradi
template <class T>
inline const T &Stack<T>::Peek(int nom) const
{
    //

```

```

    assert(nom <= top);

    return stackPtr[top - nom]; // n-elementni
qaytarish
}

// ekranga chiqarish
template <typename T>
inline void Stack<T>::printStats()
{
    for (int ix = top - 1; ix >= 0; ix--)
        cout << "|" << setw(4) << stackPtr[ix] <<
endl;
}

// o'lchamni qaytarish
template <typename T>
inline int Stack<T>::getStackSize() const
{
    return size;
}

// ko'rsatkich qaytarish (nusxalash konstruktori
uchun)
template <typename T>
inline T *Stack<T>::getPtr() const
{
    return stackPtr;
}

// o'lchamni qaytarish
template <typename T>
inline int Stack<T>::getTop() const
{
    return top;
}

#endif // STACK_H

```

Stack shablonsinfi *.h (header) fayldayaratildi. Shablonsinfinterfeysivaamalgaoshirishhambirxilfayldabo'lishikerak, bo'lmasa, shungao'xshashmazmunbilanxatoberishimumkin.

error undefined reference to "method of a class template»

Ya'ni, xatolik, shablonsinfusullarihavolalardaaniqsizlik.

Shablonsinfinterfeysiningfragmentlarida 9 -dan 28gachayangilandi.

Sinfningbarchausullariizohlarnio'zichigaoladivameningfikrimcha, ularningfunkiyalarinialohidata'riflashmantiqiyto'g'riemas. Stack shablonsinfningbarchausullari inline funksiyalarisifatidae'lonqilinadi. Businfishinitezlashtirishmaqsadidaamalgaoshiriladi.

Sinfningajralmasfunksiyalaritashqifunksiyalargaqaragandatezroqishlaydi.

Shabloninterfeysidanso'ng, stack sinfusullariamalgaoshiriladi, kutubxonafragmentlarida 32-117gachayangilandi. Agar stack sinf, shablonlarvasinflarusullariniamalgaoshirishhaqidabilsangizshablonsinfni yaratishdamurakkabhechnarsayo'q. SinfdaiKKitakonstrkutorbor,

birinchisifragmentning 32-33 qatorlarida - bustandartkonstruktorhisoblanadi. Fragmentning 41-50 qatordaginusxakonstruktorhisoblanadi.

Birob'ektniboshqasiganusxalashuchunkkerak. Peek usuli (funksiyasi), fragmentning 80-88 fatorlarida stack elementlarniko'rishimkoniniberadi. Faqatelementraqaminikiritishingizkerak, hisoblash stack yuqorisidagielementdanboshlanadivakeraklisiniqaytaradi.

Boshqafunksiyalarixizmatchifunksiyalarhisoblanadi, ulartashqisinfnafoydalanishuchunmo'ljallangan, ya'ni prinstack() funksiyasidantashqari, bufunksiya stack elementlariniekrangachiqaradi.

Endi stack shablonsinfuchundasturtuzaylikvahardoimgidek, stack sinfshablonnisinovbo'ladihamdaasosiyfunksiyalaridanfoydalanishniko'r amiz.

4.4-dastur. Shablonsinfndanfoydalanish.

```
// Created by MBBahodir
include "stdafx.h"

#include <iostream>

using namespace std;

#include "stack.h"
```

```

int main()
{
    Stack<char> stackSymbol(5);
    int ct = 0;
    char ch;

    while (ct++ < 5){
        cin >> ch;
        stackSymbol.push(ch); // element qo'shish
    }

    cout << endl;

    stackSymbol.printStack(); // ekranga chiqarish

    cout << "\n\n Element o'chirish\n";
    stackSymbol.pop();

    stackSymbol.printStack(); // ekranga chiqarish
    Stack<char> newStack(stackSymbol);
    cout << "\n\nNusxalash konstruktori!\n";
    newStack.printStack();
    cout << "2 tartibdagi element: "<<
newStack.Peek(2) << endl;
    system("pause");
    return 0;
}

```

4.4-dastur. Output.

```

b
a
h
o
q

|   q
|   o
|   h

```



```

|   a
|   b

Element o'chirish
|   o
|   h
|   a
|   b

Nusxalash konstruktori!
|   o
|   h
|   a
|   b
2 tartibdagi element: h

```

4.4-dasturda Bir stack ob'ektini yaratilgan, stack hajmi 5 gateng va stack ko'pio'zicha olishi mumkin bo'lgan elementlar soni 5 gateng. stackni while takrorlanish operatori bilan to'ldiriladi. Ekranda stack elementlarini chiqariladi, keyin stackdan bitta element o'chiriladi va yana ekranda stack elementlarini chiqariladi. Natija asosida u o'zgaradi, aniq bitta element o'chirildi.

Shuningdek, dasturda nusxa konstruktor ishlatiladi. Peek() funksiyasi stackning ikkinchi elementini qaytardi.

Stack shablonsini foydalanish yaratilgan bo'lgan, masalan, int ma'lumotlar turidasi nabo'rcangiz bo'ladi. Ishonchim komilki, hammanarsa to'g'ri ishlaydi.

Navbat (queue). Navbat ma'lumotlar tuzilmasi bo'lib (yuqorida aytib o'tilganidek) bu LIFO tamoyilini qo'llab quvvatlaydi (last in-last out: oxirgi kirgan oxirgi chiqadi). C++ allaqachon tayyor STL konteyner-navbat (queue) mavjud.

Navbatda, birinchi element kiritilgan bo'lsa, u ham birinchi chiqariladi. 4 ta elementni qo'shsangiz, birinchi qo'shilgan element birinchi, ikkinchi qo'shilgan element ikkinchi chiqadi.

Navbat qanday ishlashi tushunish uchun do'kon navbatini tasavvur qilishingiz mumkin. Sizes auning o'rtasidagi ibsiz, shunda kassaning oldidagi ibsiz,

avvaloldingizdagibarchaodamlargaxizmatqilinishi.
 Lekinoxirginavbatdao'tgankishiuchun,
 o'zitashqaribarchaodamlargaxizmatqilishkerakemas.

1	9	8	0	1	2	8

Buyuqoridakeltirilganro'yxatgaqarang.
 Bulardanbirortasinichiqarishuchunshuketma-ketlikasosidachiqarib,
 keraklisigato'xtaladi. Masalan, 0 sonninichiqarish, uchunavvl 3
 tasonchiqariladisongkeraklison.

Stack shablonsinfidabir peek()
 (buindeksbo'yichaelementniqaytaradi),
 bunavbatshablonidamuayyanelementniko'rishmumkinemas.

Agarbarchanavbatelementlargakirishkerakbo'lsa, lekin,
 navbatdabuniamalgaoshirishmumkin.

Keyinroqbuqandayamalgaoshirishniko'rsatibo'tamiz.

C++danavbatyaratish. Agar C++
 danavbatshabloniniishlatmoqchibo'lsangiz,
 avval<queue>kutubxonasiniulashingizkerak.

Keyinnavbatnie'lonqilishuchunquyidagisintaktikniishlatishingizkerak:

```
queue<type><name>;
```

Birinchi queue so'ziniyozishimizkerak.

Type ganavbatnito'ldirishuchunkkeraklitipniko'rsatishimizlozim.

Navbatob'ektiningnominiko'rsatishamizkerak.

Masalan,

```
queue<int>q;
```

Navbatusullarivafunksiyalari:

Navbatdafunksiyabubirxilfunksiyadir, lekinufaqat STL
 konteynerlaribilanishlaydi. Bufunksiyalarga

push(), pop(), front(), back(), empty() funksiyalarkiradi.

push() – navbatgayangielementqo'shish.

pop() – navbatdanbirinchielementnio'chirish,
 ikkinchielementbirinchielemento'rnigasuriladi.

front() – navbatdagibirinchielemenigamurojaat

back() - navbatdagibirinchielemenigamurojaat

empty() – navbatbo'shyokibo'shmasliginitekshiradi.

Agarbo'shbo'lsa true, aksholda false qiymatqaytaradi.

Bukeltirilganfunksiyalargadoir 4.5-dasturinekeltiramiz.

4.5-dastur. Navbatfunksiyalaridanfoydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <queue>
using namespace std;

int main() {
    queue <int> q;

    for (int h = 0; h < 7; h++) {
        q.push(rand()%100);
    }

    cout << "Navbatdigi birinchi element " <<
q.front() << endl;
    cout << "Navbatdigi oxirgi element " << q.back()
<< endl;
    q.pop();

    cout << "Navbatdigi birinchi element
(o'chirishdan so'ng): " << q.front() << endl;

    cout << "Navbat bo'sh " << (q.empty())?"":"emas")
<< endl;

    system("pause");
    return 0;
}
```

4.5-dastur.Output.

```
Navbatdigi birinchi element 41
Navbatdigi oxirgi element 78
Navbatdigi birinchi element (o'chirishdan so'ng):
67
Navbat bo'sh emas
```

Massivyordamidanaavbatlarni yaratish. Yuqorida aytganimizdek, navbatni massiv orqali amalga oshirish mumkin. Odatda, agar dasturchi bunday navbatni yaratasa, massiv navbat deb ataladi. Shuningdek, buni massiv deb ham aytish mumkin, ammo massiv so'zi C++ dabor. Shuning uchun, navbat shabloni deb aytish ham mumkin va bir xil tarzda unicha qiriladi.

Buning uchun 2 ta ko'shimcha o'zgaruvchi yaratish kerak. Birinchisi start bo'lsin va navbatning birinchi elementi ni bildirsin. Ikkinchisi ends bo'lsin va navbatning oxirgi elementi ni bildirsin.

```
queue[ends] -  
ifodasini ishlatib navbatning oxirgi elementini qiymatini qaytaradi.
```

```
queue[start] -  
ifodasini ishlatib navbatning oxirgi elementini qiymatini qaytaradi.
```

Navbatdagi birinchi elementni o'chirish uchun start ni birga kamaytirish kerak xolos, navbatni bo'shligi nite kshirish uchun start=ends mantiqiy shart nite kshirishning o'zi kifoyadir. Agar mantiqiy shart true qaytarsa navbat bo'sh, aks holda bo'sh emas.

4.6-dastur. Massivyordamidanaavbatlardan foydalanish.

```
// Created by MBBahodir  
#include "stdafx.h"  
#include <iostream>  
#include <queue>  
using namespace std;  
  
int main() {  
    //queue <int> q;  
    int q[3];  
    int start = 0, ends = 0;  
    for (int h = 0; h < 3; h++) {  
        int a; cin >> a;  
        q[start++] = a;  
    }  
  
    cout << "Navbatdagi birinchi element " << q[start  
- 1] << endl;  
  
    start--; // bir elementni o'chirish  
  
    cout << " Navbatdagi birinchi element
```

```

(o'chirilgandan so'ng) " << q[start - 1] << endl;

    cout << " Navbatdagi oxirgi element " << q[ends]
<< endl;

    cout << "Navbat bo'sh " << ((start == ends)?"":"
emas ") << endl;

system("pause");
    return 0;
}

```

4.6-dastur.Output

```

9
7
5
Navbatdagi birinchi element 5
Navbatdagi birinchi element (o'chirilgandan so'ng)
7
Navbatdagi oxirgi element 9
Navbat bo'sh emas

```

Ustuvornavbat (Priority_queue). Ustuvornavbat (Priority_queue) bilannavbat (queue) - oddiynavbatdek, lekinustuvornavbatgayangielementqo'shilasa, shundanavbatkamayishtartibidasaralanadi. Shutarikaustuvornavbatdaengkattaelementbirinchio'ringachiqadi. Ustuvornavbatshablonidanquyidasintaktikorqalifoydalanishkerak:

```
priority_queue<type><name>;
```

Birinchipriority_queueso'ziniyozishimizkerak.

Type ganavbatnito'ldirishchunkeraklitipniko'rsatishimizlozim.

Navbatob'ektiningnominiko'rsatishimizkerak.

Ustivornavbatdayangielementqo'shishuchun push() funksiyasi, birinchio'rindagielementigamurojaatqilishuchun top() funksiyasiishlatiladi. Front() – funksiyasiishlatilmaydi. Shuningdek, back() – oxirgielementgamurojaatfunksiyasihamishlatilmaydi.

4.7-dastur. Ustuvornavbatni (Priority_queue) yaratishvausullaridanfoydalanish.

```
// Created by MBBahodir
```

```

#include "stdafx.h"
#include <iostream>
#include <queue>
using namespace std;

int main() {
    priority_queue<int> priority_q,
    priority_q_one;

    for (int h = 0; h < 7; h++) {
        priority_q.push(rand()%100);
    }

    priority_q_one = priority_q;

    cout << "priority_q navbatdigi birinchi element "
    << priority_q.top() << endl;
    cout << " priority_q_one navbatdigi birinchi
    element " << priority_q_one.top() << endl;
    priority_q.pop();

    cout << "priority_q navbatdigi birinchi element
    (o'chirishdan so'ng): " << priority_q.top() <<
    endl;

    cout << "priority_q navbat bo'sh " <<
    (priority_q.empty()?"":"emas") << endl;

    while (!priority_q_one.empty())
    {
        cout << "priority_q_one navbatdigi element :
    " << priority_q_one.top() << endl;
        priority_q_one.pop();
    }
    cout << " priority_q_one navbat bo'sh " <<
    (priority_q_one.empty()?"":"emas") << endl;

    system("pause");
}

```

```

return 0;
}

```

4.7-dastur. Output
priority_q navbatdigi birinchi element 78 priority_q_one avbatdigi birinchi element 78 priority_q navbatdigi birinchi element (o‘chirishdan so‘ng): 69 priority_q navbat bo‘sh emas priority_q_one navbatdigi element : 78 priority_q_one navbatdigi element : 69 priority_q_one navbatdigi element : 67 priority_q_one navbatdigi element : 41 priority_q_one navbatdigi element : 34 priority_q_one navbatdigi element : 24 priority_q_one navbatdigi element : 0 priority_q_one navbat bo‘sh

Navbatlarni yaratishni ikki usulni kishrib chiqdik. Queue
sinf shabloni va massiv asosida. Aslida ko‘proq queue
sinf shablonidan foydalangan maqul.
Chunki massivga nisbat tanzimlash qiyin.

Ikkita monolamanavbat (deque, double-ended queue). Deque –
shunday ma'lumotlarni tuzilmasiga aytiladiki,
undama'lumotlarni qo‘shish va o‘chirish holdidan va orqadan amalga oshiriladi.
Deque ham xuddi queue dekk xotirada saqlanadi. Dek (Deque)
bushunday dinamik massivki,
u ikki tomondan kengayish va boshqanavbatlar va to‘plamlarga qaraganda tez
roq.

Bundantashqari,
u har qanday holatda o‘zi elementlarni boshida yoki oxirida kiritish uchun ruqsat
beradi,
lekin o‘z boshimchalik holatda elementlarni tez kiritish imkoniyati saqlab qolish
nimasligi mumkin. Vektor o‘xshash amallari kam bo‘lishi mumkin.

Dekk konteyner bilan tanisharekanmiz uni masibilanyaxshi,
boshqa konteynerlardan fandan farq qiladidegansavollar paydobo‘lishi tabiiy
yalbatta. Dekning umumiy tavsifi dan boshlaymiz. STL da bu deko‘zinima?
Tushuntirish uchun, uzoqqa bormaymiz, jismoniy tarbiya darsini olamiz.
Ko‘pchiligimiz "bo‘ybo‘yicha saflanish" nimaligini bilamiz.
Shunday qilib, haqiqiy Dek shunday qurilish mexanizmi ga.

Undashundaybirqatorhosilbo‘ladikivabuqatoruningoxiridavauningboshid
anto‘ldiriladi, zarurbo‘lsa, bundaketma-
ketelementlarnio‘rnialmashtirishihammumkin.

Qanchatezroqturishkerakbo‘lsa, shunchayaxshivastek,
navbatma‘lumotlartuzilmasigaqaraganda? Albatta,
yoboshidayoki qatorningoxiridavaqatorningo‘rtasidatxminansizningbo‘
ybalandligigaqarashvakeyinsolishtirishvao‘rinlarnio‘zgartirishkerak.
Biroq, birqatoro‘rtasidajoylashtirilganbo‘lsa, butunelementlarsuriladi.
Buhayotdandekkaqiyosimisoledi.

Shakllantirilayotganto‘planningboshidayokioxiridaelementlarniqo‘
shishkerakbo‘lganda,
ba‘zielementlarnio‘rtagaqo‘yishkerakbo‘ladidekniishlatishmantiqanto‘g‘
rikeladi (mosravishdashutipdagidektanlashkerak). Dekdaketma-
ketliko‘rtasidaqo‘shishvao‘chirishqo‘llab-quvvatlashbo‘lsa-da,
buamallarchiziqilivaqtidaamalgaoshiriladi.
Dasturdabukabiamallarko‘pbajarishkerakbo‘lsa,
ro‘yxatlarengyaxshitanlovbo‘ladi.

Aslida, Dek-
buboshidayokioxiridaelementlarningtezroqqo‘shilishiimkonibo‘lganvekt
ordir. Shubilanbirga,
Dekningboshqaamallarivektorlarningtegishliamallaribilanbirxilishlaydig
anvaqtga (amallarko‘naysasekinroq) ega.

Dek (navbat queue) bilanvavektor (vector) o‘rtasidagifarqlar:

1. Deklarningboshigavaoxirigahamelementlarniqo‘shishtezkoramald
ir, ammovektorlaruchunfaqatoxirigaqo‘shishtezkoramalhisoblanadi.
Buqo‘shishdoimiyvaqttalabetadi.

2. Dekningfunksiyalarvaelementlarniqo‘shishganisbattanvektordagis
hungao‘xshashfunksiyalarodatda, birozsekin.

3. Dekningiteratorlarimaxsustipdagiaqlliko‘rsatkichlargaegabo‘lishi
kerak,
chunkidoimiyko‘rsatkichlarularturliqismlario‘rtasidabiridanikkinchisiga
o‘tishkerak.

4. Xotirabloklarihajmigachegaralaribortizimlaridaikkitomonlamanav
batdekxotiraniorti qg‘ibilanajratadi,
chunkivektorolishimumkinbo‘lganma‘lumotlarganisbattanko‘p.
Shundayqilib, max_size ()
funksiyasiningnatijasivektorlarganisbatanikkitomonlamanavbatlarga
kattabo‘lishimumkin

5. Ikkitomonlamanavbatlarimkoniyatlarniboshqarishnita‘minlamayd
ivaxotiraqaytaajratilgandaungajoyaniqlanmaydi. Xususan,
boshidayokioxiridaelementlarniqo‘shishboshqaharqandayqo‘shishyokio‘
chirishkabiikkitomonlamanavbatdaelementlarbilanbog‘liqbarchako‘rsatk
ichlarni, murojaatlarniiteratorlarbekorqilmaydi. Biroq,
xotiraningtaqmislanishivektorlargaqaragandayaxshiroqishlashimumkin,
chunkiularodatdaichkituzilishigako‘ra,
ikkitomonlamanavbatlarxotiraniqaytajoylashtirishdabarchaelementlarini
ngnuxalashshartemas.

6. Deklardaishlatilmaydiganxotirabloklaribandqilinmaydivarealo‘lch
amdanso‘ngxotiraningbo‘shjoylariozodqilinadi.

BuesaDekningxotirahajmikamaytirishgaimkonberadi (ammo,
qachonvaqandayqilibbusodirbo‘lshiuniamalgaoshirishgabog‘liq).

Dekkata‘luqlibo‘lganvektorlarningafzalligi.

1. To‘plamningo‘rtasidaelementlarniqo‘shishvao‘chirishsekinroqam
alhisoblanadi.

Chunkibiro‘zgarishuchunbarchaelementlarningo‘rinlarinialmashtirishker
akbo‘ladi.

2. Vektorlarvadeklarningiteratorlarierkinruxsatturigamansub.

3. Deksinfivektorsinfiningfunksiyalaribilanbixilfunksiyalargaega.

Dekningkamchiliklariga

1. Tez-

tezqo‘shishyokio‘chirishdekningboshidayokioxiridasodirbo‘lishi.

2. Konteynerelementlarigahavolaqilaolmaysiz.

3. Konteynerishlatilmayqolgandaxotiranixoliqilishamalgaoshirilishi
gaC++qafolatbermaydi (amalgaoshirishtamoyiligaqarab,
xolibolishivabo‘lmasligihammumkin)

Inobatgaolishinikerakbo‘lganholatlar:

1. Dekningat()bo‘qabirorbirfunksiyasiiteratorvaindeksnitekshiraol
maydi.

2. Qo‘shishvao‘chirishqo‘shimchaxotiratalabqilishmumkin.

Shuninguchunqo‘shishvao‘chirishgadoiramallarabajarilgandadekningbos
hvaelementlariuchunko‘rsatkichlar,

havolalarvaiteratorlarnianiqko‘rsataolmaydi. Bungaistisnobolib,

Dekningboshigayokioxirigaelementqo‘shisholinadi.

Boshigayokioxirigaqo‘shilgandako‘rsatkichvahavolalaraniqbo‘ladi,
ammoiteratoroyo‘q.

3. Dekvavektorninginterfeyslarideyarlibixil. Farqishundaki,
ba‘zibirfunksiyalashyaratilganbo‘lishimumkin. Masalan, deksinfi

capacity, reserve kabifunksiyalargaegaemas.
Buningsababidekvektorlarga o'xshabishlashini yaxshilash kerak emas.

4. Deklar bilanishlaganda,
ayniq sama'lumotlarni qo'shishda ko'rsatkich va iteratorlarga bohliq dastur fr
agmenini yozishda ehtiyo't bo'lish va mustaqil yozish kerak mas.

Umuman olganda, shaxsiy deklar bu vektorlardir,
ular bilanishlash ham vektorlarkabi amalga oshiriladi. Ammo qachon dekni,
qachon vektorni tanlashni bilish kerak.

Ma'lumotlarni tuzilmasining boshiga va oxiriga qo'shish uchun optimizats
iyalangan (yaxshilangan, ixcham qilingan) vektor – bu dekdir.
Bunda qo'shishni qo'zgarmas vaqt da bajariladi va nusxalash konstruktorini
birmarta chaqiradi.

Agar elementdekning o'rtasiga qo'shilsa,
eng yomonibu amal vaqt talab qiladi,
chunki qo'shish nuqtadan boshigacha va oxirigacha bo'lgan masofalar orasid
akichikaniqlash lozim.

insert, push_front i push_back
funksiyalaridek daga ham iteratorlarni bekor qilish imomkin.
Shuningdekdekning o'rtasiga ma'lumot qo'shilganda havo larni xam bekor
qiladi.

Berilgan navbatdagi elementlarni guruhlash masalasini qaraylik.
Guruhlashni amalga oshirish uchun guruhlash shartini tanlaymiz va shu asosid
anavbatichidagi guruhlaymiz, shartga tegishli bo'lganlarni navbat boshidan,
bo'lmaganlar oxiridan joylatiriladi.

4.8-dastur. Navbvt dagi elementlarni guruhlash.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <deque>
#include <algorithm>
using namespace std;

bool mypred(const int x){
    return x <= 51;    // guruhlash uchun shart
}

int main(){

    int Arr[]={1,78,89,23,51,49,100,18,50};
```

```

deque<int> d(&Arr[0],&Arr[9]);

cout << "Joriy deque: " << endl;
for (deque<int>::iterator
it=d.begin();it!=d.end();it++) std::cout<<*it<<" | ";
cout << endl;

stable_partition(d.begin(),d.end(),mypred);

cout << "Natija: " << endl;
for (deque<int>::iterator
it=d.begin();it!=d.end();it++) std::cout<<*it<<" | ";
cout << endl;
system("pause");
return 0;
}

```

4.8-dastur.Output

Joriy deque:
1 | 78 | 89 | 23 | 51 | 49 | 100 | 18 | 50 |
Natija:
1 | 23 | 51 | 49 | 18 | 50 | 78 | 89 | 100 |

4.7-dasturdagifragmentasosiyguruhlashshartihisoblanadi.

```

bool mypred(const int x){
    return x <= 51;    // guruhlash uchun shart
}

```

Bundaagarx <= 51 bo'lsa true, aks holda false qaytaradi.
Bungaekvivalentsifatidaquyidagichafragmentham yozish mumkin.

```

bool mypred(const int x){
    if (x <= 51) return true;    // guruhlash uchun
shart
    return false;
}

```

Dekuchunkeltirilganmateriallardastlabkio'rganuvchilaruchun yetarl
idebhisoblaymiz.

Masalaningshartigaqarabdekyokivektornitanlashdasturchiningmahoratig
abog'liq.

Dekningfunksiyalariga quyidagilarkiradi:

4.1-jadval. Dekningfunksiyalari

push_front	Boshidanyangielementqo'shish
push_back	Oxiridanyangielementqo'shish
pop_front	Birinchielementni olish
pop_back	Oxirgielementni olish
front	Birinchielementqiymatiniko'rish
back	Oxirgielementqiymatiniko'rish
size	Elementlar soniniko'rish
clear	Barchaelementlarnio'chirish

Konteynerlar bilan ishlash algoritmlari. Barcha konteynerlar umumiy yondashuvni tufayli, amalda qiziq tirgan asosiy algoritmlar konteynerlarning har qanday turi uchun qo'llanilishi mumkin bo'lgan umumiy shaklda amalga oshirilishi mumkinligi tufayli STL konteynerlar uzoq amaliy foydalanish uchun birar zigulik ixtiro bo'ldi.

Algoritmlar kutubxonaning eng kattava ommabop qismidir. Judako'palgoritmlar mavjudki, ularning hammasini batafsil bayon qilish uchun kattakitob kerak. Quyida ularni guruhlarga ajratib, ularning nomlari bilan ataymiz (ularning hammasi emas) va ular dan faqat ayrimlari misol tariqasida keltiramiz.

for_each() algoritmi. for_each() eng ko'p ishlatiladigan algoritmdir. Konteyner elementlar guruhi (ehtimol, barchaelementlari) uchun ketma-ket harakatni amalga oshiradi. U har qanday STL konteynerda foydalanish mumkin. Quyidagi 4.9-dasturda for_each() algoritmi massiv vektor uchun ishlatilgan.

4.9-dastur. for_each() algoritmidan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

inline ostream& operator <<( ostream& out, const
vector< unsigned >& obj ) {
```

```

        cout << "< ";
        for( auto& p: obj )
            cout << p << " ";
        return out << ">";
    }

void pow2( unsigned& i ) { i *= i; }

int main( void ) {
    const int examples = 4;
    for( int i = 0; i < examples; i++ ) {
        unsigned ai[] = { 1, 2, 3, 4 , 5, 6, 7, 8, 9
    },
        ni = sizeof( ai ) / sizeof( ai[ 0 ] );
        vector< unsigned > vi( ai, ai + ni );
        cout << vi;
        switch( i ) {
            case 0:
                for_each( vi.begin(), vi.end(), pow2 );
                cout << " => " << vi << endl;
                break;
            case 1:
                for_each( ai, ai + ni, pow2 );
                cout << " => " << vector< unsigned >(
ai, ai + ni ) << endl;
                break;
            case 2:
                for( auto& i : ai ) pow2( i );
                cout << " => " << vector< unsigned >(
ai, ai + ni ) << endl;
                break;
            case 3:
                for_each( vi.begin() + 2, vi.end() - 2,
pow2 );
                cout << " => " << vi << endl;
                break;
        }
    }
}

```

```

    system("pause");
return 0;
}

```

4.9-dastur.Output.

```

< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 2 9 16 25 36 49 8 9 >

```

Dasturning birinchi global (auto&x : ...)
takrorlanish fragmenti ishlatilgan (C++11
standartitomonidan joriy qilingan).

Bu kabir fragmentni massivlar vakonteynerlarga ham qo'llanilishim mumkin
(bu vektorning oqimgachiqarish operator variant funksiyasida ko'rsatilgan).

Bu takrorlanish fragmentiaslida STL
kutubxonayoki algoritmlarni qismi emas, lekinu for_each()
algoritmbilan bir xil ta'sir ko'rsatadi,
ya'ni to'plam barcha elementlar uchun izchil qo'llashim mumkin.

Buyuqoridagi dastur barcha algoritmlarni berilgan interval asosida tash
kiliyasosiy mantiq qismini ko'rsatadi (shart emas, barcha konteynerlar).
Boshiva oxiri bilan cheklangan iteratorga (ko'pincha, birinchi berilgan 2
ta parametrlar) funksiya, funkto'rvapredikat (bu funksiyalarga -
qaysi direlementlar ibo'yicha saralashim ko'nibo'lsavam mantiqiy qiymat qayta
radigan funksiyalar kiritadi) navbat bilan qo'llaniladi.

Find(). Navbatdagi algoritmi - bu find(). Bu nomidan ma'lumki,
to'plamdagi elementlarni qidirish funksiyasidir. E'tibor bersak,

judako'p konteynerlar bunday find() -
funksiyasiga ega va u ob'ekt uchun obj.find(...) sietaksisida ishlatiladi.

Shu vaqtda algoritm quyidagicha funksiyani find(obj:iteator, ...
) ishlatish uradi. Aslida, bu bitta algoritmi emas,

balki ularning butun birkatta guruhimaveud,

ularto'plam elementlariniba'zi atribut,

shart bilan tanlashlari asosida birlashtirilishim mumkin.

Bu guruh predikatlariga **find(), find_if(), find_if_not(), find_first_of(), find_end(), adjacent_find()** funksiyalar kiritadi.

Shuningdek bu guruhda kengaytirilgan algoritmlar ham tegishli dirva bularga
count(), count_if(), search(), binary_search(), min(), max(),
minmax_element(), min_element(), max_element(), equal()
kabi birkatta algoritmlar kiritadi.

To'plamga islov berish (aralashtirish) bir shartliguruhalgoritmlar mavjud va ular joylarda elementlarni qaytatartibg asoluvchi va qiymatlarni o'zgartiruvchi algoritmlardir: **fill(), replace_copy(), reverse(), rotate(), rotate_copy(), shuffle(), random_shuffle(), transform(), replace(), replace_if()** **vaboshqa funksiyalar.**

Ikkita to'plamlar ustida bajariladigan algoritmlar guruhi ham mavjud. Bular asosan nusxalash, tarkibi niko'chirib o'tish (xartipdagikonteynerlar uchun bo'lishi mumkin. Masalan, **vector<>** ni **set<>**ga) funksiyalarini bajaradi. Bular **gacopy(), copy_if(), move(), swap_ranges(), remove_copy(), remove_copy_if(), merge(), set_intersection(), set_difference()** **vaboshqa algoritmlar.**

Nihoyat, juda maxsus guruhalgoritmlari, to'plam ichida elementlarni turlis aralash bilan bog'liq funksiyalar **sort(), stable_sort(), is_sorted(), is_sorted_until()** **vaboshqa algoritmlar.** Ushbu qiziqarli guruhi ni keyinroq, alohida batafsil ko'rib chiqish kerak.

Kutubxonada vaqto'tish bilan ortib borayotgan algoritmlarning ko'pligi va ularning aksariyati adabiyotda hech qanday ta'riflanmasligiga qaramasdan, tabiiysavoltug'iladi: bu xilma-xillikni qanday tushunish kerak? Bu qiyinchiliklar qanday eng o'lumumkin:

- Barcha STL ob'ektlar (konteynerlar, algoritmlar) shablonsintaktik bo'yicha tasvirlangan. Shuning uchun, ularning tavsiflari, ularning header fayllarida dastur fragmentlari sifatida tuzilgan bo'lishi kerak.

- Kerakli algoritmlar uchun header fayllarini standart </usr/include/c++> katalogiga o'ting va **stl_algo*** kabilar lavha fayllarini toping - u da barcha algoritm funksiyasi prototiplarini (o'xshashlarini) topasiz. Bundan tashqari, har bir prototipialgoritm maqsadini tushuntirib va parametrlarini tushuntirishning batafsil izohi dindanyozib ko'yilgan.

- Bir necha asosiy STL algoritmlarini foydalanishga doir misollarni ko'rib chiqing — kutubxona lartizimida ularning ko'pchiligi bor. O'xshashlik bilan boshqa barcha algoritmlarni yaratish mumkin.

Shablonsinfi kutubxonalar shablon jihatidan **template** termin bilan belgilangan, chunki o'sha, kompilyatsiya sintaktik xatobo'yicha xabarlar nichi qaradi. Birinchi siyoz maxabarlar fragmentlari bo'yicha va ikkinchi xatolarni izlashu

chun. Bushablonkabikuchlimexanizmitomonidanqayytaishlanidi (tekshiriladi) vadasturchibuuchuntayyorbo‘lishikerak.

Yuqoridaaytibotilganidek, misollarnio‘rganishjudako‘psavollargajavobberadi, shuninguchundasturkeltiramiz.

Endio‘rganganalgoritmlarnidiqqatbilaneslang (harbirinisharhlab, kutubxonasinflaridantashqaridabo‘lganjudako‘p STL algoritmlarimavjud, ammularningbarchasibir-birigao‘xshabishlaydi):

4.10-dastur. Standartalgoritmlardanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <cstring>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
using namespace std;

inline ostream& operator <<( ostream& out, const
vector<char>& obj ) {
    for( auto p: obj ) cout << p;
    return out;
}

int main( void ) {
    char s[] = "Muhammad al-Xorazmiy nomidagi
Toshkent Axborot Texnologiyalari Universiteti"; //
nazvanieknigi
    vector<string> vs;
    vs.push_back("Muminov");
    vs.push_back("Bahodir");
    // copy & find :
    vector<char> v1( strlen( s ) );
    copy( s, s + v1.size(), v1.begin() );
    int nb = 0;
    for( auto is = find( v1.begin(), v1.end(), ' '
); is != v1.end();
        is = find( ++is, v1.end(), ' ' ) )
```



```

nb++;
    cout << "bo'sh joylar soni: " << nb << " (" <<
nb + 1
<< " so'z)" << endl;
    // min & max :
    auto mm = minmax_element( v1.begin(), v1.end()
);
    cout << "belgilar uzunligi: '" << *mm.first <<
"' ... '"
<< *mm.second << "'" << endl;
    // fill & reverse & rotate & shuffle :
    vector<char> suv( vs[ 0 ].size() );
    copy( vs[ 0 ].begin(), vs[ 0 ].end(),
suv.begin() );
    cout << suv << endl;
    random_shuffle( suv.begin(), suv.end() );
    cout << suv << endl;
    reverse( suv.begin(), suv.end() );
    cout << suv << endl;
    rotate( suv.begin(), suv.begin() + suv.size() /
2, suv.end() );
    cout << suv << endl;
    // set_intersection & set_difference
    set< char > sus, pns;
    for( char s: vector<char>( vs[ 0 ].begin(), vs[
0 ].end() ) )
        sus.insert( s );
    for( char s: vector<char>( vs[ 1 ].begin(), vs[
1 ].end() ) )
        pns.insert( s );
    vector<char> outi( 100 ), outd( 100 );
    auto ret = set_intersection( sus.begin(),
sus.end(), pns.begin(),
                                pns.end(),
outi.begin() );
    cout << "umumiy belgi " << ( ret - outi.begin()
) << " : "
<< outi << endl;

```

```

    ret = set_difference( sus.begin(), sus.end(),
pns.begin(),
                                pns.end(), outd.begin() );
    cout << "unikal belgi " << ( ret - outd.begin()
) << " : "
<< outd << endl;
    system("pause");
    return 0;
}

```

4.10-dastur. Output

```

bo'sh joylar soni: 6 (7 so'z)
belgilar uzunligi: ' ' ... 'z'
Muminov
nuvmMoi
ioMmvun
mvunioM
umumiy belgi 2 : io
unikal belgi 5 : Mmnuv

```

Dasturda char uchunkonteynerlardan foydalanilgan. Deyarlibarchaberilgankonteynerlaruchunturlixilalgoritmlaramalgaoshirilgan. (ixcham, ammozerikarlifragmentlar).

Funktorlarningko‘llanilishi.

“funktionalob’ektlar”

Funktorso‘ziC++da ningqisqartmasidanolingan.

C++dafunktionalob’ekt

operator()bilananiqlangansinfningnusxasihisoblanadi.

AgarC++sinfioperator()aniqlasangiz, ufunksiyakabiishladi, ammoholatnisaqlayolmaydi. Masalan, 4.11-dasturgaqarang.

4.11-dastur. Operator()niqaytaishlatish.

```

#include "stdafx.h"

#include <iostream>
#include <string>

using namespace std;

class SimpleFunctor {
    string name_;

```

```

public:
    SimpleFunctor(const char *name) : name_(name)
    {}
    void operator()() { cout << "Salom, " << name_
<< " ishlar zo'rmi!" << endl; }
};

int main() {
    SimpleFunctor sf("Akbar");
    sf();
    system("pause");
    return 0;
}

```

4.11-dastur. output

Salom, Akbar ishlar zo'rmi! _____

Dasturdasfob'ektbo'lishidanqa'tiynazarsf() funksiyasinichaqiramiz. Chunki SimpleFunctop sinfida operator() aniqlangan.

Ko'phollarda STL kutubxonasi ningtaqqoslashvaxatolikfunksiyasiningalgoritmlaridafunktor larC++dapredikatlarsifatidaishlatiladi. Farazqilaymiz, sonlarketmaketligiberilgan.

Buketmaketlikdagibarchajuftsonlarvatoqsonlaryig'indisinihisoblashkera kbo'lsin. Bumasalaniyechishuchunfunktorva for_each algoritmidanfoydalanishmumkin.

4.12-dastur. Funktorva for_each algoritmidanfoydalanish.

```

#include "stdafx.h"
#include <algorithm>
#include <iostream>
#include <list>
using namespace std;
class EvenOddFunctor {
    int even_;
    int odd_;
public:
    EvenOddFunctor() : even_(0), odd_(0) {}
    void operator()(int x) {

```

```

        if (x%2 == 0) even_ += x;
        else odd_ += x;
    }
    int even_sum() const { return even_; }
    int odd_sum() const { return odd_; }
};

int main() {
    EvenOddFunctor evenodd;

    int my_list[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    evenodd = for_each(my_list,
my_list+sizeof(my_list)/sizeof(my_list[0]),
                    evenodd);

    cout << " Juftlar yig'indisi: " <<
evenodd.even_sum() << endl;
    cout << " Toqtlar yig'indisi: " <<
evenodd.odd_sum() << endl;
    system("pause");
    return 0;
}

```

4.12-dastur. output

Juftlar yig'indisi: 30

Toqtlar yig'indisi: 25

Dasturda `for_each` da `EvenOddFunctor` sinf ob'ektiberilgan. `for_each` `my_list` bo'yichaketmaketo'tgandafunktorfunksiyanicha qirmoqda. Hartakrorlanishda `evenodd` funktornusxasini qaytaradi. `Evenodd` funktoresajuft yoki toq elementlarni yig'indisini saqlaydi.

Standart

MLd funktorlar.

ML

dagi funksiyalar interfeyslarning umumiy tadbirlari mavjud va OYD jihatidan shakllantirish qiyin masallardan hisoblanadi. ML standart jihatidan funktorlar ML modul tizimining bir qismi dirvaular tuzilmalarni yaratishga imkon beradi.

Masalan, plagintizimini yozish masalasini olaylik. Aytaylik barcha pluginlar soddaroqbo'lishi uchun, faqat perform funksiyasini o'zicha o'ladigan zarur interfeysini amalga oshiri shkerakbo'lsin. MLda avval pluginlar uchun maxsus tasdiqlovchi (imzolovchini) ni yaratish zarur.

```
Signature Plugin =
sig
  val perform: unit -> unit
end;
```

Pluginlar uchun tasdiqlovchi interfeysni aniqladik, ikki ta pluginni, aytaylik, Loudplugin va Silentpluginni tuzilmalar orqali amalga oshirishimiz mumkin. LoudPluginni amalga oshirish:

```
structure LoudPlugin :> Plugin =
struct
  fun perform() = print "KATTA HARFLAR!\n"
end;
```

Silentpluginni amalga oshirish:

```
structure SilentPlugin :> Plugin =
struct
  fun perform () = print "kichik harlar\n"
end;
```

Endi funktorlarga yaqinlashamiz. ML dagi funktorlarni tuzilmalar ni argumentsifatida qabul qilishadi, shuning uchun **Plugin** argumentsifatida berilgan.

```
functor Performer(P : Plugin) =
struct
  fun job () = P.perform ()
end;
```

Bu funktor Plugin ni P argumentsifatida qabul qiladi va P pluginni perform funksiyasini chaqiradigan job funksiyasi uchun ishlatadi.

Performer funktorni ishlatamiz (Funktorni tuzilma qaytarishini bilamiz).

```
structure LoudPerformer = Performer(LoudPlugin);
structure SilentPerformer = Performer(SilentPlugin);

LoudPerformer.job ();
SilentPerformer.job ();
```

Standard ML uchun b koddii misol edi.
Ammo chuqurroq o'rganish uchun Stilini ham bilishtalab qilinadi.

Haskell da funkto'rlar.

Haskell da funkto'rlar haqiqiy funkto'rlar bo'lishi bilan farqlanadi.

Haskell da funkto'rlar kate'goriya nazariyasidan matematik funkto'rlarga juda o'xshaydi (Kate'goriya nazariyasi - matematikaning ob'ektlarining ichki tuzilishiga bog'liq bo'lmagan matematik ob'ektlar o'rtasidagi munosabatlarning xususiyatlarini o'rganadigan bo'limi).

Kate'goriya nazariyasida funkto'rlar kate'goriyalar o'rtasidashunday taqsimlani shakerakki,

kate'goriya strukturasiga qalib qolsin yoki boshqa qilibaytganda, u ikki kate'goriya orasidagi omomorfizm bo'lsin.

Buta'rif Haskell da oddiy tip dagi sinfsifatida amalga oshiriladi.

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b
```

ML ga qarab, masalan, Haskell tipi `lugin sinfiga` o'xshaydi, joriy `Sinfn`ing nusxasiga bo'lish uchun qaysi amallarni qanday qilib yaratishni aniqlaydi. Bu holatda, birgina `Functor` tiplar bo'yicha aniqlanmagan, balki `f` konstruktorasosida aniqlangan. Bundan shuni bilish mumkin ki `Functor` - `fmap` funksiyasini yaratib beradi. `fmap` funksiyasi tipini qabul qiladiv `b` tipini qaytaradi. `fa` (`f` konstruktorasosida qurilgan tip, aqaro'llaniladi) qabul qilinadigan tip va `fb` qaytariladigan tip.

Unit ushuni shuchun, ba'zi konreynerda har birelement uchun amal qiladigan funksiyasi sifatida `fmap` funksiyasini o'ylab ko'ring. Funkto'rlarning eng oddii misoli muntazam ro'yxatlar va `map` funksiyasiga bo'lib, u ro'yxatdagi har birelement uchun funksiyani qo'llaydi.

```
Prelude> map (+1) [1,2,3,4,5]
[2,3,4,5,6]
```

Bu oddii misolda `fmap` funksiyasi faqat `map` va tipidagi konstruktor `[]` - list tipidagi konstruktor dir. Shuning uchun, ro'yxatlar uchun, masalan, `Functor` sifatida belgilangan.

```
instance Functor [] where
  fmap = map
```

Bu `map` o'rniga `fmap` yordamida fragmentni shakllantirib olsak, albatta, to'g'ri bo'ladi.

```
Prelude> fmap (+1) [1,2,3,4,5]
```

```
[2,3,4,5,6]
```

Lekin funktorta 'rifituzilishini asrab-avaylash haqida hech narsa demaydi, unutmang!
Shuning uchun har qanday normal funktorlarning matematik funksiyalarta 'rifigak
iruvchi funktorlar qonunlarini qanoatlantirish kerak.
Fmapning ikki qoidalaribor:

```
fmap id = id  
fmap (g. h) = fmap g. fmap h
```

Birinchiqoida da konteynerdagi har bir element uchun bir xil funksiya ni xira-ritalash hech qanday ta'siri yo'q deb aytilgan. Ikkinchi qoida ga ko'ra, konteynerdagi har bir element ustida ikki ta funksiyaning tarkibi birinchi funksiyani ko'rsatish bilan, so'ngra ikkinchi siniko'rsatish bilan bir xil bo'ladi.

Ularni yaqqol ko'rsatish uchun funktorlarga yanabirmisoldaraxtlar ustida amallar hisoblanadi. Birkonteynersifatidadaraxt, shajara (ierarxiya, ichma ich joylashgan ob'ektlar) berilgan bo'lsin. Daraxt tuzilishini saqlab qolish uchun, daraxt xususiyatlariga fmap funksiyasini qo'llaymiz.

Buning uchun birinchi daraxtni belgilab boshlaymiz.

```
data Tree a = Node (Tree a) (Tree a)  
            | Leaf a  
            deriving Show
```

Bu fragmentda daraxt turi ikki daraxtning (chap va o'ng tugunlariga) tuguni (node) yokibarg (leaf) ekanligi aytiladi. **deriving Show** ifoda si show funksiyasi orqali ko'rishim ko'ni beradi. Endi Tree daraxti ustida funktor ni aniqlayolamiz.

```
instance Functor Tree where  
    fmap g (Leaf v) = Leaf (g v)  
    fmap g (Node l r) = Node (fmap g l) (fmap g r)
```

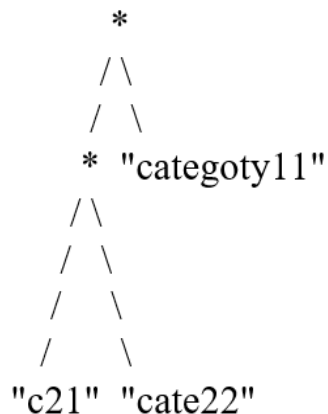
Bu fragmentda fmap tipidagi funksiyasi argumentlari sifatida Leaf tipidagi v qiymatni qabul qiladivaleaf tipidagi gni vga qiymat qilib beradi. Ikkinchi qoida da fmap tipiga oid g funksiyasi Node tipidagi va r qiymatlarini qabul qiladivaleaf tipidagi gni va r argumentlarini qiymat qilib beradi.

Fmap funksiyasini tree ob'ektlari bilan ishlash uchun tayyorlaymiz. Tree sinfini satrli (string) ro'yxat bilan quramiz va har bir ro'yxat uzunligini length() funksiyasi bilan aniqlaymiz.

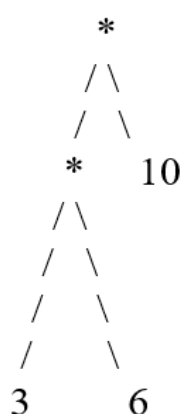
```
Prelude> let tree = (Node (Node (Leaf "c21") (Leaf "cate22")) (Leaf  
"category11"))
```

```
Prelude> fmap length tree
Node (Node (Leaf 3) (Leaf 6)) (Leaf 10)
```

Quyidagichatree quriladi:



(a)



(b)

4.1-rasm. (a) tree ko‘rinishi, (b) length() bo‘yicha ko‘rinishi.

Aslida, Haskell da funktoirlar fundamental funktoir sinfi hisoblanadi va applicative functors vastrelkalar barchasibular ham funktoirlardir. Haskell funksiyalar murakkab hisoblanadi, agar bu funktoirlarni haqiqiy xayot qonuniyatlariga bog‘lab o‘rganmoqchibo‘lsangiz, alohida fan va manbalarni o‘rganish lozim.

Prolog funktoirlari. Bu hammasidan oson funktoirlardir. Ikki tamisol qaraymiz. Birinchisi atom - butuzilmaning birinchisidir. Ifodani qaraymiz:

```
?- likes(olma, nok)
```

Birinchisi atom funktoiridir – likes

Ikkinchisi - funktoir deb nomlangan ajralmas predmet.

Bu argumentlari va funktoir tuzilmasini qaytaradi. masalan,

```
?- functor(likes(olma, nok), Functor, Arity).
```

Functor = likes

Arity = 2

Bu Prolog funktoirlariga misol.

Funktoirlarni yaratish dasturchilarga xos masalbat.

Chunki bugungi kundatayyor foydalanish uchun ko‘plab funktoirlar yaratilgan.

Ular asosan instrument yaratuvchilar, ya‘ni dasturchilar uchun dastur tuzuvchilari yaratadi apm mantiqiy dasturlash funktoirlarining o‘rnini muhim ahamiyat kasbetadi.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Adapterlarnimavaqaysisinflarkiradi?
2. Konteynerlarningadapterlariqaysisinflarkiradi.
3. Judakengvatez-tezishlatiladiganma'lumotlartuzilmalaribu – nima?
4. Konteyneradapterlariningfunksiyalarinita'minlashuchun STL standartkonteyneriharqandaymoslashtirishdatayanchsifatidaqaysia mallardan (adapterturigaqarab) foydalanishmumkin.
5. Birinchinavbvtidakonteynerlarningadapterlariuchunqandaysinflarni dasturgaqo'shishlozim.
6. Steknimavaunimasibilanharakterlanadi.
7. Birinchikirganoxirichiqaditamoyilibo'yichafaoliyatko'rsatuvchima'lumotlarto'plamibunima.
8. Navbvtnimavaunimasibilanharakterlanadi.
9. BirinchikirganBirinchichiqaditamoyilibo'yichafaoliyatko'rsatuvchi ma'lumotlarto'plamibunima.
10. Navbatningqaysiusulivektorkonteynerustigaquribbo'lmaydi, lekinuniro'yxatkonteyneruchunqurishmumkin.
11. Deque sinfiuchun*front()*, *push_back()*, *pop_front()*, usullariharqandaykonteynerlardaqachonishlatiladi?
12. Deque sinfidakaysiusulio'rniga *front()* usuliishlatiladi?
13. Ma'lumbirtartibdaqaytaishlanishikerakma'lumotlarniuchunu mumiyma'lumotlartuzilishigaasoslanganto'plamibu – qandayto'plamlar?
14. Ma'lumotlartuzilishi - stekjudaoddiy: uqandaytartibnibelgilaydi?
15. Stekningassosiativusullarinisanabbering?
16. Navbatda, birinchiementkiritilganbo'lsa, uqachonchiqariladi?
17. Agar C++ danavbatshabloniniishlatmoqchibo'lsangiz, avvalqandaykutubxonasiniulashingizkerak.
18. *empty()* – navbatuchunnimavazifanibajaradi.
19. Massivyordamidanaavbatlarniyaratishmumkinmi?, mumkinbo'lsaizohlabbering.
20. Ustuvornavbat (*Priority_queue*) bilannavbat (*queue*) ob'ektlariningfarqinimada.

21. Ustivornavbvtdayangielementqo‘shishuchun push() funksiyasi, birinchio‘rindagielementigamurojaatqilishuchunesaqaysifunksiyasi ishlatiladi.
22. Ikkitomonlamanavbatvaodiiynavbatningfarqinimada?
23. Dekvavektorningo‘xshashtomonlarivafarqlari.
24. Ikkitomonlamanavbatningavzalliklarinisanabbering.
25. Nimauchunikkitomonlamanavbatdusturchigakerak. 3 taasoskeltiring.
26. Dekningmaummolarinibilasizmi?
Deknavbatqachondasturchigapandberishimumkin.
27. Konteynerelementlarguruhi (ehtimol, barchaelementlari) uchunketmaketharakatniamalgaoshiradiganalgoritmninganiqlang.
28. To‘plamdagielementlarniqidirishfunksyasi find()gata‘rifbering.
29. To‘plamgaishlovberish (aralashtirish)uchunbirshartliguruhalgoritmlarmavjudvaularjoylardaelementlarniqaytatartibgasoluvchivaqiymatlarnio‘zgartiruvchialgoritmlardir. Ularisanabbering.
30. Ikkito‘plamlarustidabajariladiganalgoritmlarguruhiqaysialgoritmlarmisolbo‘laoladi.
31. Funktorso‘ziC++daqaysiso‘zningqisqartmasidanolingan.
32. C++dafunksionalob‘ekt — qaysioperatorbilananiqlangansinfningnusxasihisoblanadi.
33. Ko‘phollarda STL kutubxonasiiningtaqqoslashvaxatolikfunksiyasiningalgoritmlaridafunktorlarC++danimalarsifatidaishlatiladi.
34. ML standartjiatidanfunktorlar ML modultiziminiingbirqismidirvaularnimalariniyaratishgaimkonberadi.
35. Haskelldagifunktorlarkategoriyanazariyasidanqaysifunktorlar gajudao‘xshaydi.



AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG‘I	
	Stack stekkonteynerlaradarterigaoidberilganquyidagidasturbo‘yichabe —

	rilgantopshiriqlarkeraklifragmentlariasosidabajaring. 📢 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarb osqichma – bosqichamalganoshiriladi.	
dastur	topshiriqlar	
<pre>// Created by MBBahodir #include "stdafx.h" #include <iostream> #include <vector> #include <stack> #include <string> #include <array> #include <string> using namespace std; int main() { stack<int> st, st_one, st_two; stack<string, vector<string>> vst; vst.emplace</pre>	1. Dasturdanechtaglobalfunksiyayaratilgan. 	
	2. Dasturdanechtastack to‘plammavjud. 	
	3. Dasturdavst.push("belgi" - "+" to_string(int(i)) "+" to_string(i));qandayvazifanibajaradi. 	
	4. Ikkixiltipdagistekto‘plamelementlarinialma shtirishdasturfragmentiniyozing. 	
	5. (!vst.empty()) qachonchinvaqachonyolg‘onqiymatlarniqaytara di. 	

<pre> ("str1"); vst.emplace ("str2"); vst.push("e ostr"); for (int i = 0; i < 255; i++) { vst.push("b elgi - " + to_string(int(i)) + to_string(i)); } for (int i=0; i<50; ++i) st.push(i*15); for (int i = 0; i < 105; i++) st_one.emplace(i*10); st_two = st; st.swap(st_ one); cout << st.size() << " " << st_one.size() << " " << st_two.size() </pre>	6. Stekdanelementlarni ishlash funksiyasini tuzing. 7. while (st.size() != 0){ cout << st_two.top() << " : stack dagi o'rni " << st_two.size() << endl; st_two.pop(); } Dastur fragmentini stek elementlarini saqlab qolgan xolda ishlatish funksiyasini tuzing. 8. Stek vektor bilan farqlarini aniqlovchi dastur tuzing.

```

<< " " <<
vst.size() <<
endl;

    while
(!vst.empty())
{
        cout <<
vst.top() << "
: stack dagi
o'rni " <<
vst.size() <<
endl;

        vst.pop();
    }

    int size =
st.size();
    while
(size--) {
        cout <<
st.top() << " :
stack dagi
o'rni " <<
st.size() <<
endl;

        st.pop();
    }

    while
(st.size() !=
0){
        cout <<
st_two.top() <<
" : stack dagi
o'rni " <<

```

```

st_two.size()
<< endl;

    st_two.pop(
);
    }

    while
(!st.empty()) {
        cout <<
st.top() << " :
stack dagi
o'rni " <<
st.size() <<
endl;

        st.pop();
    }

    cout <<
st.size() << "
" <<
st_one.size()
<< " " <<
st_two.size()
<< " " <<
vst.size() <<
endl;

    cout << "
stack st " <<
(st.empty() ?
"" : "not ") <<
" empty" <<
endl;
    cout << "
stack vst" <<
(vst.empty() ?

```

<pre> "" : "not ") << " empty" << endl; cout << " stack st_one " << (st_one.empty() ? "" : "not ") << " empty" << endl; system("pau se"); return 0; } </pre>	
9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

IKKINCHI ASSISMENT TOPSHIRIG‘I	
	queue — navbatkonteyneradapterigaoidberilganquyidagida sturbo‘yichaberilgantopshiriqlarkeraklifragmentl ariasosidabajaring. ☝ Bundadasturdagiba‘zio‘zgartirishlarnitopishorq alitopshiriqlarbosqichma — bosqichamalganoshriladi.
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include <iostream> #include <list> #include <queue> #include <string> #include <array> using namespace std; </pre>	1. Dasturdanechtaglobalfunksiyay aratilgan. _____ _____ _____ _____ _____ _____

<pre> int main() { queue<int> st, st_one, st_two; queue<string, list<s tring>> vst; vst.emplace("str1"); vst.emplace("str2"); vst.push("eostr"); for (int i=0; i<5; ++i) st.push(i); for (int i = 0; i < 15; i++) st_one.emplace(i*10); st_two = st; st.swap(st_one); cout << st.size() << " " << st_one.size() << " " << st_two.size() << " " << vst.size() << endl; while (!vst.empty()) { cout << vst.front() << " : stack dagi o'rni " << vst.size() << endl; vst.pop(); } </pre>	2. Dasturdanechtaqueuesinfining ob'ektingto'plammavjud.
	3. Dasturdagiortiqchakutubxonal arnio'chiribtashlangvasabablariniayt ing.
	4. Navbtdaqachonoxirgielementq oladi. Dasturfragmentiasosidako'rsatibber ing.
	5. (st.size() != 0) shartnigmatiqantakrorlanishgashartqi libqo'yishto'g'rimivanimuchun. Anologiniyozing.

<pre> int size = st.size(); while (size--) { cout << st.front() << " : stack dagi o'rni " << st.size() << endl; st.pop(); } while (st.size() != 0){ cout << st_two.front() << " : stack dagi o'rni " << st_two.size() << endl; st_two.pop(); } /* while (!st.empty()) { cout << st.front() << " : stack dagi o'rni " << st.size() << endl; st.pop(); } */ cout << st.size() << " " << st_one.size() << " " << st_two.size() << " " << vst.size() << endl; cout << " stack st " << (st.empty() ? "" : "not ") << " empty" << endl; cout << " stack vst" << (vst.empty() ? "" :</pre>	6.st.swap(st_one); nio‘rniga st_one.swap(st); yozilsanimaho disasodirbo‘ladi.
	7. Navbatning dasturda ishlatimag an funksiyasini aniqlang.
	8. Dasturdagi izohga olingan dastur fragmenti olib tashlansa, dasturda qandayo‘zgarishlar bo‘ladi va natijasichi.

<pre> for (int h = 0; h < 7; h++) { priority_q.push(rand()% 100); } </pre>	
<pre> priority_q_one = priority_q; cout << "priority_q navbatdigi birinchi element " << priority_q.top() << endl; cout << " priority_q_one navbatdigi birinchi element " << priority_q_one.top() << endl; priority_q.pop(); cout << "priority_q navbatdigi birinchi element (o'chirishdan so'ng): " << priority_q.top() << endl; cout << "priority_q navbat bo'sh " <<(priority_q.empty())?" ":"emas") << endl; while </pre>	3. Dasturdanechtapriority_queueet o'plammavjud.
	4. Dasturdadeque<int> d(&Arr[0],&Arr[9]);funksiyasi qandayvazifanibajaradi.
	5. To'plamdanberilganqiymatniiz lashdasturfragmentiniyozing.

<pre> (!priority_q_one.empty()) { cout << "priority_q_one navbatdigi element : " << priority_q_one.top() << endl; priority_q_one.pop(); } cout << " priority_q_one navbat bo'sh " << (priority_q_one.empty() ?"": "emas") << endl; int Arr[]={1,78,89,23,51,49 ,100,18,50}; deque<int> d(&Arr[0],&Arr[9]); cout << "Joriy deque: " << endl; for (deque<int>::iterator it=d.begin();it!=d.end();it++) std::cout<<*it<<" "; cout << endl; stable_partition(d.begi n(),d.end(),mypred); cout << "Natija: " << endl; </pre>	6. Ustivornavbatshartkiritibikkita yanginavbatyaratuvchidasturtuzing. 7.Dasturda it qandaytip, nimauchunaynandeknavbatgafoydalanilgan. 8. operator= dasturningqaysiqismlaridafoydalanilganvanimauchun.

<pre> for (deque<int>::iterator it=d.begin();it!=d.end();it++) std::cout<<*it<<" "; cout << endl; system("pause"); return 0; } </pre>	
9. Dasturda jamibo‘lib, nechao‘zgartirish kiritildi. _____ 10. Shudasturning analogini yaratish sizga mustaqil vazifa dir.	

TO‘RTINCHI ASSISMENT TOPSHIRIG‘I	
	Konteynerlar bilan ishlash algoritmlariga oid berilgan quyidagi dastur bo‘yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan. 👉 Bunday dasturda giba ‘zio‘ zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include <vector> #include <algorithm> using namespace std; inline ostream& operator <<(ostream& out, const vector< unsigned >& obj) { cout << "< "; for(auto& p: obj) cout << p << " "; return out << ">"; } void pow2(unsigned& i) { i *= i; } int main(void) { const int examples = 4; for(int i = 0; i < examples; i++) { unsigned ai[] = { 1, 2, 3, 4 , 5, 6, 7, 8, 9 }, ni = sizeof(</pre>	1. Dasturda nechta global funksiyalar aratilgan.
	2. Dasturda xatolarni toping.
	3. for(auto& p: obj) fragmentni tushuntirib bering va anolini yozing.

<pre> ai) / sizeof(ai[0]); vector< unsigned > vi(ai, ai + ni); cout << vi; switch(i) { case 0: for_each(vi.begin(), vi.end(), pow2); cout << " => " << vi << endl; break; case 1: for_each(ai, ai + ni, pow2); cout << " => " << vector< unsigned >(ai, ai + ni) << endl; break; case 2: for(auto& i : ai) pow2(i); cout << " => " << vector< unsigned >(ai, ai + ni) << endl; break; case 3: for_each(vi.begin() + 2, vi.end() - 2, pow2); cout << " => " << vi << endl; break; } } </pre>	4. Dasturning qaysi fragmentlarini FIND() funksiyasi bilan almashtirish mumkin. 5. To'plamdan berilgan qiymatni izlash dastur fragmentini aniqlang va o'z izlash funksiyasini o'zgartiring.

<pre> system("pause"); return 0; } </pre>	
6. Dasturda jamiboʻlib, nechaoʻzgartirish kiritildi. _____ 7. Shudasturning analogini yaratish sizga mustaqil vazifa.	

BESHINCHI ASSISMENT TOPSHIRIG'I	
	Funktorlarning ko'lanilishiga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan. 👉 Bunday dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>// ConsoleApplication1.cpp : main project file. #include "stdafx.h" #include <iostream> #include <string> using namespace std; class myFunctorClass { private: int _x; char _ch; public: myFunctorClass (int x) : _x(x) {} myFunctorClass (char ch): _ch(ch){} int operator() (int y) { return _x + y; } int operator() () { return _x*_x; } void operator()(char ch, int a, int b) { switch (ch) { case '+' : cout <<</pre>	1. Dasturda nechta konstruktor funksiyalar yaratilgan.
	2. Dasturda xatolarni toping.
	3. operator() fragmentini tushuntirib bering va anoligini yozing.

<pre> a + b; break; case '-' : cout << a - b; break; case '*' : cout << a * b; break; case '/' : cout << ((b>0) ? to_string(a/b) : "maxraj 0"); break; case '%' : cout << ((b>0) ? to_string(a%b) : "maxraj 0"); break; case '^': { int ret = a; while(b-- > 1) ret *= a; cout << ret; } break; default: cout << "amal to'g'rimas !!!"; break; } } int operator()(int a, int b) { switch (_ch) { case '+' : return a + b; break; case '-' : return a - b; break; case '*' : return a * b; break; case '/' : return ((b>0) ? a/b : 0); break; case '%' : return ((b>0) ? a%b : 0); break; case '^': { </pre>	4. Massivning juft vatoqelem entlari yig'indisini idf sonini hisoblovchi funkto'ryozing. 5. To'plamdan berilgan qiymatni izlash funkto'rfunksiyasini zniyarating. 6. Taqqoslash uchun funkto'rfunksiyalarating. 7. Funkto'rfunksiyalarning qaysi uslubisizgayoqdi. Dasturni to'liq shu uslubga almashtiring.

```

        int ret = a;
        while( b-- > 1
) ret *= a;
        return ret;
        } break;
        default: {cout <<
"amal to'g'rimas !!!";
return 0; }
        break;
    }
}
};

int main()
{
    myFunctorClass add( 5
);
    cout << add( 6 );
    cout << endl;
    myFunctorClass F(5);
    F('+',3,4);
    cout << endl << F() <<
endl;

    cout <<
myFunctorClass('^')(2,10);
    system("pause");
    return 0;
}

```

8. Dasturda jamibo'lib, nechao'zgartirish kiritildi. _____
9. Shudasturning analogini yaratish sizga mustaqil vazifa dir.

2-BOB. DASTURLASH TILINING TAKOMILLASHTIRILGAN IMKONIYATLARI

2.1. Standartalgoritmlarvaiteratorlar.

 C++ tilining standartalgoritmlar kutubxonasi, iteratorlar, xotirani taqsimlashtirish talablari, dinamik xotira ajratish va foydalanishga asosida qidirish, saralash, taqqoslash funksiyalari, dasturlash da'vazlarini tushirishlari, iteratorlar, o'zgarma iteratorlar, teskari iteratorlar bilan ishlashga mo'ljallangan asosiy funksiyalar va ularning dasturlashdagi ahamiyati, xotirani taqsimlovchi sinflarni yaratish va talablari, turlitipli massivlar uchun dinamik xotira ajratish va foydalanish operatorlar uslublari, amallari, talablari, vazifalar va usullarini keltirilgan bo'lib, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan.

Bilimlarni mustahkamlash uchun 30
tanazariy savol va amaliy ko'nikma va malakalarni rivojlantirish uchun 3
ta assistent topshirig'ini va har assistentda 10(9,7) ta topshiriq, jami
26 ta topshiriq berilgan.

Yuqoridagi topshiriqlarni bajarish mavzu mustahkamlash uchun xizmat qiladi.

✍Kalit so'zlar. Algoritm, kutubxona, iterator, o'zgarma iterator, teskari iterator, qidirish, saralash, o'rinalmashtirish, shablon, sinf, find(), sort(), swap(), begin(), end(), rbegin(), rend(), crbegin(), crend(), xotira, pul, xotira bloki, new, delete, for_each(), funksional ob'ektlar.

✓Bilish shart bo'lgan tushunchalar. Tiplar tushunchasi, sinf va sinf ob'ekti, meros xo'rsinfi, to'plam, statik va dinamik massiv, element gamurojaat, funksiyaviy ko'rsatkich, konteyner, xesh, istisno holat, ma'lumotlarni kirish va chiqishi, dasturlashga oid dastlabki tushunchalar va C++ tilini qo'llab quvvatlovchi muhitda ishlashni bilish lozim.

🔑Bilimlasiz. Inspektorlar, mutatorlar, koordinatorlar, find() algoritmi, find_if() algoritmi, count() va count_if() algoritmlari, sort() algoritmi, <functional> kutubxonasi, for_each() algoritmi, algoritmlarning bajarilish tartibi, iterator amallari, iteratorlarning qo'shima amallari, xotirani taqsimlovchilar, new() operatori, delete() operatori, xotirani ishlashni tezlashtirish usullari, pul ob'ektlari, konteyner va uni xotirada qo'llanilishi, standart bo'yicha taqsimlovchilar usullarini o'rganish gizm mumkin.

REJA

1. Standartalgoritmlar.

2. Iteratorlarvaularningqo‘llanilishi.
3. Xotiranitaqsimlovchilarvaulargaqo‘yilgantalablar.
4. Standartbo‘yichataqsimlovchi.

KIRISH

Bugunkikundadasturlashjudasoddalashibbormoqda,
ba‘zibirmutaxassislardasturlashuchunmatematikakerakmasdebaytsa,
ba‘zilarimatematikadasturlashningasosidebaytadi.

Shuningnegizidanimayotibdi. Albattaalgoritmlar,
iteratorlarvaxotiranitaqsimlovchilarvaularningtalablariyotadi.

Algoritmlartayyobo‘lsa,
faqatbualgoritmlarinijoyjoyigaqo‘yishbujudaosondir.
Ammoalgoritmlarniyaratishvaunidasturlashjoriyqilisho‘tamushkulvamas
haqqatlimasaladir.

Shuninguchunbugungikundajudako‘plabalgoritmlarnisaqlovchistandartv
anostandartkutubxonalarmavjud.

Standartalgoritmlar. Yangidasturlashniboshlanuvchilarodatdasaral
ash,
qidirishyokiqatorelementlarinisanashkabinisbatanoddiyvazifalarnibajarish
huchunmaxsustakrorlanishgaasoslanganusullarniyozishuchunjudako‘pva
qtsarflashadi. Buusullaruldaxatoqilishqanchalikosonligijihatidanham,
umumiyishonchlilikvamavjudliknuqtainazaridanhammuammolibo‘lishi
mumkin, chunkibuusullartushunishqiyinbo‘lishimumkin.

Qidiruv,
sanashvasaralashdasturlashdajudakengtarqalganamallarbo‘lganligisababl
i, C++
standartkutubxonasidastlabbuvazifalarnibirnechaqatorkoddabajaradigan
kattafunksiyalarto‘plaminio‘zichigaolgan. Bundantashqari,
buxususiyatlaroldindansinovdano‘tgan,
samaralivaturlixilkonteynerlarniqo‘llab-quvvatlaydi.
Buxususiyatlarnihamparalelqo‘llab —
quvvatlashtezroqunibajarishuchunbixilmasalauchunbirnechaMP
(markaziyprotsessor) oqimlariniajratishqobiliyatiega.

Algoritmkutubxonasitomonidantaqdimetiladiganfunksiyalarodatda
chtoifadanbirigakiradi:

Inspektorlar (nazoratchilar) - konteynerdagima‘lumotlarni
(masalan, qidirishyokielementlarnisonihisoblashamallari)
ko‘rishuchunishlatiladi.

Mutatorlar

konteynerdagima'lumotlarni o'zgartirish uchun ishlatiladi (masalan, saralash yoki elementlarni qaytatib gasolish amallari).

Fasilitatorlar

(koordinatorlar)

elementlarining qiymatlarini sosidan natijahosil qilish uchun ishlatiladi (masalan,

qiymatlarni ko'paytiruvchi ob'ektlar yoki elementlarning qaysitartib juftliklarini datartiblanish hakerakligini aniqlovchi ob'ektlar).

Bu algoritm lar algoritm kutubxonasida

(`<algorithm>` sarlavha faylida)

joylashgan.

Oldingi biladigan va eng ko'p tarqalgan algoritm lar dan ba'zilarini imkoniyatlarini chuquroq ko'rib chiqamiz va bu algoritm lar ning barchasi iterator lar dan foydalanadi.

`find()`

algoritmi

berilgan qiymat bo'yicha elementlarni qidirish algoritmi.

Berilgan qiymatni birinchi topgunicha ishlaydi. Argumentsifatida 3 taparametni oladi.

ketma-ketlikda boshlang'ich element uchun iterator;

ketma-ketlikda oxirgi element uchun iterator;

qidirishning qiymati.

Qidirish natijada qidirilayotgan qiymatli elementga (agar topilsa) yoki iteratorning oxiriga (agar bunday element topilmasa)

ishora qiluvchi iterator qaytadi.

5.1-dastur. Find () funksiyasidan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <algorithm>
#include <array>
#include <iostream>

using namespace std;
int main(){
    array<int,10> arr = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19 };

    cout << "Qidirish va o'zgartirish uchun son kirit: ";
    int search;
```

```

int replace;
cin >> search >> replace;

// array<int,10>::iterator found;

auto found = find(arr.begin(), arr.end(),
search);

if (found == arr.end())
{
    cout << search << " topilmadi. " << endl;
}
else
{
    *found = replace;
}

for (int i : arr)
{
    cout << i << ' ';
}

cout << endl;
system("pause");
return 0;
}

```

5.1-dastur.Output	
Birinchisinov	
Qidirish va oʻzgartirish uchun son kirit: 19 21 1 3 5 7 9 11 13 15 17 21	
Ikkinchisinov	
Qidirish va oʻzgartirish uchun son kirit: 10 25 10 topilmadi. 1 3 5 7 9 11 13 15 17 19	

find_if() algoritmi - shartorqaliev elementni izlash. Judako phollardakonteynerdama vjud elementlar orasidama'lumbir qiymat saqlaydigan element bormidegan masala duch kelamiz. Masalan, konteynerdasi tri elementlar berilgan bo'lsa, biror bira tri qismi bormikan degan masala bo'lsin. Manashunday masalalarni yechishda find_if() algoritmi eng mos va qulay hisoblanadi. Bu algoritmi find() algoritmi bilan analoq sifatida ishlaydi ammo, qidirish uchun qiymat o'rniga funksiya ko'rsatgichlariga moslangan element qiymatida o'xshashlik bor bo'lgan ob'ekt haqiriladi. find_if() funksiyasi buning uchun har bir elementni chaqiradi va toshunday elementni topilmaguncha yoki konteynerdagi elementlartugaguncha.

Ma'lum bir berilgan satr bo'yicha, berilgan konteynerdasi satrni o'zida qismi satr qilib saqlovchi elementni borki y o'qligini aniqlovchi dasturni tuzish kerak bo'lsin.

5.2-dastur. find_if() funksiyasidan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <algorithm>
#include <array>
#include <iostream>
#include <string>

using namespace std;

bool containsNut(string str)
{
    return (str.find("oro") != string::npos);
}

int main()
{
    array<string, 4> arr = { "Toshkent", "Buxoro",
        "Samarqand", "Navoiy" };

    auto found = find_if(arr.begin(), arr.end(),
        containsNut);

    if (found == arr.end())
```



```

    {
        cout << " Topilmadi" << endl;
    }
    else
    {
        cout << " Topildi: " << *found << endl;
    }
    system("pause");
    return 0;
}

```

Agarodatdagistandardtartarzdagidekmuammonihalqilshganimizda, kamidaikkitakrorlanishkerakedi (birinchitakrorlanishelementlarnisaralashuchunvaikkinchisisatrqisminisolishtirishuchun). Standart C++ kutubxonasiningvazifalaribirnechtafragmentsatrlaridahamshundayqilishimkoniniberadi.

count() va count_if() algoritmlari – konteynerdanberilganelementlarinisanashvazifasinibajaradi. Berilganmezon (shart) bo'yichabarchaelementlarnitekshirib, sanaydi. Berilgankonteynerdanechtaelementberilgansatrqismiborliginisanashdasturinikeltiramiz.

5.3-dastur. count() va count_if() algoritmlaridanfoydalanish.

```

// Created by MBBahodir
#include "stdafx.h"
#include <algorithm>
#include <array>
#include <iostream>
#include <string>
#include <vector>
using namespace std;

bool containsNut(string str)
{
    return (str.find("daryo") != string::npos);
}

bool counter(string str){
    return true;
}

```

```

int main()
{
    array<string, 6> arr = { "Sirdaryo",
    "Amudaryo", "Qashadaryo", "Suxondaryo",
    "Zarafshon", "Qashadaryo" };

    auto countWithIf = count_if(arr.begin(),
    arr.end(), containsNut);
    int countofArr = count(arr.begin(), arr.end(),
    "Qashadaryo");
    cout << "daryo satr qismi borlarining soni: "
    << countWithIf << endl;
    cout << " elementlar soni: " << countofArr <<
    endl;

    system("pause");
    return 0;
}

```

5.3-dastur. Output

```

daryo satr qismi borlarining soni: 5
elementlar soni: 2

```

sort() algoritmi foydalanuvchining saralash funksiyasini qurish uchun ishlatiladi. Oldinlarisort() funksiyasibilan massivlarni saralagan bo'lishingiz mumkin, ammosort() funksiyabubilan chegaralanib qolmaydi. Bu funksiyayordamchibir funksiyaoladivabu argumentasosida foydalanuvchi o'zining saralash uslubini amalga oshirishim mumkin. Funksiya esataqqoslash uchun parametrik kitapametroladi. Agar birinchi element ikkinchisidan oldin bo'lishi kerak bo'lsa, ikkinchi paramet true qiymat qaytarishi kerak. Odatda sort() funksiyasio'z tartibida saralaydi. Berilgan massivni kamayish tartibida saralash dasturini tuzaylik.

5.4-dastur. sort() funksiyasidan foydalanish.

```

#include "stdafx.h"
#include <algorithm>
#include <array>

```

```

#include <iostream>

using namespace std;
bool greater(int a, int b)
{
    return (a > b);
}
bool d_greater(int a, int b)
{
    return (b > a);
}

int main()
{
    array<int,5> arr = { 28, 18, 16, 25, 25 },
    arr_one;

    arr_one = arr;

    sort(arr.begin(), arr.end(), greater);
    sort(arr_one.begin(), arr_one.end(),d_greater );
    for (int i : arr) {
        cout << i << ' ';
    }
    cout << endl;
    for (int i : arr_one ) {
        cout << i << ' ';
    }
    cout << endl;

    system("pause");
    return 0;
}

```

5.4-dastur. Output

```

28 25 25 18 16
16 18 25 25 28

```

Dasturdanko‘rinadiki,
o‘zimizsaralashalgoritminiboshidanyozishimizshartemasekan,
buninguchunbirqatorfragmentyetarlibo‘ldi. AmmoC++tili,
judako‘psaralashningkamayishtartibdafoydalanganligiuchungreater{} ti
plifunksiyasibo‘lishimumkin (<functional>kutubxonasidabor).
Bizo‘zimizninggreaterfunksiyasidanfoydalanishuchunsort(arr.begin(),ar
r.end(),greater);dasturfragmentiniyozdik,
ammokutubxonanikinichaqirishuchunsort(arr.begin(),arr.end(),std::great
er{}));kabifragmentyozishimizkerak. E‘tiborbering,
greater{} funksiyasifiguraliqavsgamuxtoj,
shuninguchununifunksiyadebatalmaydi,
ma‘lumottipidebyuritiladivaundanfoydalanishuchunnusxasiniyaratishi
mizkerak.

Figuraliqavslaranonimob’ektlarniyaratishuchunxizmatqiladi.

for_each() algortmi – konteynerningbarchaelementlaribo‘tish.
Bufor_each()

funksiyasikiruvchima‘lumotsifatidato‘plamniqabulqiladivafoydalanuvch
iningfunksiyasinijoriyto‘plamdagiharbirelementuchunqabulqiladi.

Buningqulayliktomonishundaki,

agarbiramalnitto‘plamningbarchaelementlaribo‘yichabajarishkerakbo‘lsa,
for_each() funksiyasiasqotadi.

Berilganmassivningbarchaelementlarinichiqarishuchun for_each()
funksiyasidanfoydalanishnidasturini keltiraylik.

5.5-dastur. for_each() funksiyasidanfoydalanish.

```
#include "stdafx.h"

#include <algorithm>
#include <array>
#include <iostream>

using namespace std;

void doubleNumber(int &i){
    i = i;
}

int main(){
    array<int,4> arr = { 1, 2, 3, 4 };
}
```

```
for_each(arr.begin(), arr.end(), doubleNumber);

for (int i : arr){
    cout << i << ' ';
}

cout << endl;
system("pause");
return 0;
}
```

Yoshdasturchilaruchunbutakrorlanishjarayoninioddiy operatorigaaniqparametrlarniqo‘yibishlatishmaquldir. funksiyasiningyutuqlarishandaniboratki, takrorlanishtanasidanqaytafoydalanishvaparellashtirish, kattaproektlardakattama’lumotlar, turlitiplariharxilma’lumotlartuzilmasibilanishlgandanengyaxshiisntrume ntbo‘libxizmatqiladi.	for for_each()

Algoritmlarning bajarilishtartibi

.Algoritmkutubxonasidaengko‘palgoritmlarma’lumbirijrotartibinikafolat
lamaydi.

Bunday algoritmlar uchun har qanday belgilangan funksiyalar bajarilish tartibi o'z zimmasiga o'lasligiga ishonch hosil qilish kerak, chunki bu funksiyalarni chaqirish tartibi ishlatiladigan kompilyatorga qarab farq qilishi mumkin. Quyidagi algoritmlar aniq bajariladi:

```
for_each()  
  copy()  
  copy_backward()  
  move()  
  move_backward()
```

Agar parametrsifatidabiror funksiyaberilmaganbo'lsa,
aniqlanmaganholato'tadi. Buninguchunharbiralgoritmgarr.begin()
va arr.end() aniqberilishikerak. AmmoC++20
standartidafaqatmassivberilsao'zianiqlanadi.

Algoritmkutubxonasingalgoritmlariro'yxati:

1	O‘zgartirilmaydiganketma-ketlikamallar:	
1.1	all_of	to‘plamdagibarchaelementlaruchunholatshartinitekshiris h
1.	any_of	to‘plamdagiharqandayelementuchunholatshartinibajaris

2		hinitekshirish
1. 3	none_of	to‘plamdagihechbirelementuchunholatshartgajavobbermasliginitekshirish
1. 4	for_each	funksiyanito‘plamgaqo‘llash
1. 5	find	to‘plamdaqiymatinitopish
1. 6	find_if	to‘plamdashartnibajaruvchielementnitopish
1. 7	find_if_not	to‘plamdashartnibajarmaydiganelementnitopish
1. 8	find_end	to‘plamdagioxirgielementnitopish
1. 9	find_first_of	to‘plamdagielementlardanbirinchisinitopish
1. 10	adjacent_find	to‘plamdatengqo‘shnielementlarnitopish
1. 11	count	to‘plamdagielementlarsonininihisoblash
1. 12	count_if	shartgajavobberadiganto‘plamdagielementlarsonininiqaytaradi
1. 13	mismatch	ikkifarqliintervallarnibirinchio‘rniniqaytaradi
1. 14	equal	to‘plamdagiIkkielementlartengliginitekshirish
1. 15	is_permutation	to‘plamdaalmashtirishboryokiyo‘qliginitekshirish
1. 16	search	ketmk-ketlikdanqidirish
1. 17	search_n	intervalndanqidirish

2	Ketma-ketlikamallarinio‘zgartirish:	
2.1	copy	to‘plamelementlardannusxaolish
2.2	copy_n	elementlarniko‘paytirish
2.3	copy_if	shartasosidadiapazonelementlarininusxalash
2.4	copy_backward	elementlarto‘plamidanOrqaganusxako‘chirish
2.5	move	elementlarqatoriniko‘chirish
2.6	move_backward	elementlarto‘plamniorqagako‘chirish
2.7	swap	ikkiob’ektningqiymatlarinialmashtirish
2.8	swap_ranges	ikkiintervalningqiymatlarinialmashtirish
2.9	iter_swap	ikkiiteratorlarningko‘rsatikichob’ektlariqiymatlarinialmashish
2.10	transform	to‘plamob’ektlariniaylantirish
2.11	replace	to‘plamningqiymatalmashtirish
2.12	replace_if	Qiymatlarnishartasosidato‘plamdaalmashtirish
2.13	replace_copy	almashtirishqiymatlarito‘plamninusxalash
2.14	replace_copy_if	almashtirishqiymatlarito‘plamnishartasosidanusxalash
2.15	fill	to‘plamniQiymatbilanto‘ldirish
2.16	fill_n	ketma-ketliginiQiymatbilanto‘ldirish

6		
2. 1 7	generate	funksiyabilanto‘plamuchunqiymatlarniyaratish
2. 1 8	generate_n	funksiyabilanketma-ketlikuchunqiymatlarnihosilqilish
2. 1 9	remove	to‘plamqiymatinio‘chirish
2. 2 0	remove_if	to‘plamma’lumotlarshartasisida o‘chirish
2. 2 1	remove_copy	qiymatnio‘chirish diapazonin inusxalash
2. 2 2	remove_copy_if	qiymatnio‘chirish diapazonin ishartasidan usxalash
2. 2 3	unique	to‘plamdagitakrorlanganelementlarnio‘chirish
2. 2 4	unique_copy	to‘plamdagitakrorlanganelementlarnin usxalash
2. 2 5	reverse	teskarito‘plam
2. 2 6	reverse_copy	teskarito‘plamninusxalash
2. 2 7	rotate	to‘plamdaelementlarningchapgaaylantirish
2. 2 8	rotate_copy	to‘plamdaelementlarningchapgaaylantirishninusxalash
2. 2	random_shuffle	to‘plamdagielementlarningtasodifiyqo‘yish

9		
2. 3 0	shuffle	to‘plamdagielementlarninggeneratoryordamidasodif iyqo‘yish
3	Bo‘laklar (qismlar) bilanamallari	
3. 1	is_partitioned	to‘plamningbo‘linganliginitekshirish
3. 2	partition	ikkigabo‘lishorlig‘i
3. 3	stable_partition	qonuniyatasosidaikkigabo‘lishorlig‘i
3. 4	partition_copy	birto‘plamniikkigabo‘lish
3. 5	partition_point	bo‘lishnuqtasiniolish
4	Saralashamallari:	
4. 1	sort	to‘plamma’lumotlarsaralash
4. 2	stable_sort	ekvivalentlartartibinisqaqlabqolganholdaelementlarnisara lash
4. 3	partial_sort	to‘plmdaelementlarniqismansaralash
4. 4	partial_sort_copy	to‘plmdaelementlarniqismansaralashvanusxalash
4. 5	is_sorted	to‘plamningtartiblanganliginitekshirish
4. 6	is_sorted_until	to‘plamdagijratilmaganelementnitopish
4. 7	nth_element	to‘plamdagielementnisaralash
5	Binarqidiruv (bo‘lingan / tartiblanganintervallarbilanishlash):	
5. 1	lower_bound	iteratorniquyichegarasiniqaytarish
5. 2	upper_bound	iteratorniyuqorichegarasiniqaytarishgaqaytarish
5. 3	equal_range	tengelementlardanto‘plamostioli
5.	binary_search	qiymatningtartiblanganketma-

4	arch	ketlikdamavjudliginitekshiring
6	Birlashtirish (tartiblanganintervallar bilan ishlash):	
6.1	merge	tartiblangan to'plamlarni birlashtirish
6.2	inplace_merge	ketma-kettartiblangan intervallarni birlashtirish
6.3	includes	tartiblangan to'plamlarni boshqatartiblangan to'plamni o'z ichiga olish yoki olmasligini tekshirish
6.4	set_union	ikkitartiblangan to'plamni birlashtirish
6.5	set_intersection	ikkitartiblangan intervallarning kesishishi
6.6	set_difference	ikkitartiblangan intervallar orasidagi farq
6.7	set_symmetric_difference	ikkitartiblangan intervallar orasidagi simmetrik farq
7	Elementlari yig'indisi amallari:	
7.1	push_heap	elementlari yig'indisini qatoriga surish
7.2	pop_heap	elementlari yig'indisidan elementni o'chirish
7.3	make_heap	elementlari yig'indisini yasash
7.4	sort_heap	elementlari yig'indisini tartiblash
7.5	is_heap	elementlari yig'indisini tekshirish
7.6	is_heap_until	elementlari yig'indisidan birinchisini topish
8	Kichik va katta amallari:	
8.1	min	eng kichik elementni qaytaradi
8.2	max	eng katta elementni qaytaradi
8.3	minmax	eng kichik va eng katta elementni qaytaradi
8.4	min_element	to'plamdagi eng kichik elementni qaytaradi

8.5	max_element	to'plamdagi eng katta elementni qaytaradi
8.6	minmax_element	to'plamdagi eng kichik va eng katta elementni qaytaradi
9	Boshqayordamchilar:	
9.1	lexicographical_compare	leksikografik jihatdan qiyoslash
9.2	next_permutation	to'plam aylantirish uchun keyingi indeks
9.3	prev_permutation	to'plam aylantirish uchun oldingi indeks

Buyerda 85 ta algoritmlar keltirilgan, ammobularning hammasi shablon funksiyalar bo'lganligi uchun C++ dasturlash versiyalarida farq qilib o'lishi mumkin. Shuningdek, eslab ko'ring, bualgoritmlarning ko'plaridan oldin mavzularimizda foydalanganmiz.

Iteratorlar va ularning qo'llanilishi. Iteratorlar konteynerlarning elementlariga murojaat qilish uchun foydalaniladi.

Iteratorlar bilan elementlar bilan ishlash juda qulay hisoblanadi. Iterator iterator tip bilan yoziladi.

Har qanday konteyner uchun iteratorlarning tiplari farq qiladi. Masalan, `list<int>` tipidagi konteyner uchun `list<int>::iterator` tipi, `vector<int>` tipidagi konteyner uchun esa `vector<int>::iterator` tipi ishlatiladi.

C++ tilidagi konteynerlardan iteratorlarni ajratib olish uchun `begin()` va `end()` funksiyalaridan foydalaniladi.

begin() funksiyasi konteynerdagi birinchi element ko'rsatuvchi iteratorni qaytaradi (agarkonteynerda elementlar bo'lsa).

end() funksiyasi konteynerdagi oxirgi elementdan keyingi pozitsiyaniko'rsatuvchi iteratorni (ya'ni konteyner oxirini) qaytaradi.

Agarkonteyner bo'sh bo'lsa,

begin() va **end()** funksiyalaribir xil qiymat qaytaradi.

Agar

begin() va **end()** funksiyalar o'zaro teng bo'lmasa, ularning orasidankamidabitta element bor.

Bu funksiyalarani qat'iylik konteyner uchun iterator qaytaradi.

Iterator yaratish gamisol:

```
std::vector<int> v = { 1, 2, 3, 4 };
std::vector<int>::iterator iter = v.begin();
```

Bu misolda tipiga mansub bo'lgan vektor tipidagi konteyner vektory yaratilgan. Konteyner doimiy qiymatlar bilan to'ldirilgan. **begin()**

funksiyasi (usuli) bilan vektor elementini olish uchun iterator keltirilgan. Bu iterator vektor konteynerining birinchi elementini ko'rsatadi.

Iterator mallari:

- ***iter** – iterator ko'rsatadigan elementni olish;

- **++iter** -

keyingi elementga murojjat qilish uchun iteratorni harakatlantirish

- **--iter**

oldingi elementga murojjat qilish uchun iteratorni harakatlantirish.

forward_list konteyner iteratorlaridekrement amalini qo'llab quvvatlamaydi.

- **iter1 == iter2** - ikki iterator teng,

agarularbixil iteratorlarni aniqlagan bo'lsa.

- **iter1 != iter2** ikki iterator teng emas,

agarularbixil iteratorlarni aniqlagan bo'lsa.

5.6-dastur. Iterator mallaridan foydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> myvector;
    for (int i = 0; i < 15; i++)
    {
        myvector.push_back(rand() % 100);
    }
    // vector<int>::iterator iter =
myvector.begin();
    auto iter = myvector.begin();
    while(iter!=myvector.end())
    {
        cout << *iter << " | ";
        ++iter;
    }
    cout << endl;
    system("pause");
    return 0;
}
```

Konteynerlar bilan ishlaganda, konteynerdagi elementlarni qo‘shish yoki o‘chirish hushbukonteyner uchun baʼr hajoriy iteratorlarni, shuningdek, uning elementlariga havola va ko‘rsatgichlarni bekor qilishiga olib kelish imk mkin. Iteratorlarni faqat elementlarni olish, balki ularni o‘zgartirish imkonini beradi.

5.7-dastur.

Iterator yordamida konteynerdagi elementlarni qiymatini o‘zgartirish.

```
#include "stdafx.h"

#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> myvector;
    for (int i = 0; i < 15; i++)
    {
        myvector.push_back(rand() % 10);
    }
    vector<int>::iterator iter = myvector.begin();
    //auto iter = myvektor.begin();
    while(iter!=myvector.end())
    {
        *iter = (*iter) * (*iter);
        ++iter;
    }

    for(iter = myvector.begin();
iter!=myvector.end(); ++iter)
    {
        cout << *iter << " | ";
    }
    cout << endl;
    system("pause");
    return 0;
}
```

5.7-dastur. Output

1	49	16	0	81	16	64	64	4	16	25	25	1	49	1

Dasturda while
takrorlanishoperatorikonteynerningelementlariniiteratorogaolib,
o'zini o'ziga ko'paytirib, yanashu iteratorga yozadi.
Shuning uchun krangakonteynerdagisonlarning kvadratlarini chiqadi.

O'zgarma iteratorlar. Agar konteynerda o'zgarma qiymatli elementl
arbo'lsa,
bu holda konteyner elementlariga murojaat qilish uchun o'zgarma iteratorlar
dan foydalanish sh kerak. Buning uchun `const_iterator` tipi ishlatiladi.
Bu iteratorlar faqat elementlarni sanash imkonini beradi.
O'zgartirish mumkin emas.

```
vector<int>::const_iterator iter ;  
  
    for(iter = myvector.begin();  
iter!=myvector.end(); ++iter)  
    {  
        cout << *iter << " | ";  
        /*iter = (*iter) * (*iter);  
    }  
}
```

O'zgarma iteratorning qiymatlarini `begin()` va `cend()`
funksiyalari ham ishlatiladi.

Agar iterator o'zgarma debolinmagan bo'lsaham,
bu funksiyalar uni o'zgarma qilib beradi.
o'zgartirish mumkin emas bo'ladi.

Shungdek,

5.8-dastur. O'zgarma iterator dan foydalanish.

```
// ConsoleApplication1.cpp : main project file.  
  
#include "stdafx.h"  
#include <iostream>  
#include <vector>  
using namespace std;  
  
int main(){  
    int myints[] = {16,2,77,29,28};  
    const vector<int> myvector (myints, myints +  
sizeof(myints) / sizeof(myints[0]) );  
    vector<int> myvector_one;  
    for (int i = 0; i < 15; i++)
```

```

        myvector_one.push_back(rand() % 10);

        vector<int>::const_iterator it;
        for (it = myvector.begin(); it !=
myvector.end(); ++it)
            cout << ' ' << *it;
        cout << endl;
        for (it = myvector_one.cbegin(); it !=
myvector_one.cend(); ++it)
            cout << ' ' << *it;
        cout << endl;
        system("pause");
        return 0;
    }

```

5.8-dastur.Output

```

16 2 77 29 28
1 7 4 0 9 4 8 8 2 4 5 5 1 7 1

```

Teskari iteratorlar. Bu iteratorlar konteynerning elementlarini teskari olishni amalga oshiradi. Buning uchun `reverse_iterator` tipi ishlatiladi. Bu tipdagi iteratorlarga konteynerning elementlarini olish uchun `rbegin()` va `rend()` funksiyalaridan foydalaniladi.

```

#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int myints[] = { 1, 2, 3, 4, 5 };
    vector<int> v (myints, myints+
sizeof(myints)/sizeof(myints[0]));
    for (vector<int>::reverse_iterator iter =
v.rbegin(); iter != v.rend(); ++iter)
        std::cout << *iter << " | ";

    cout << endl;
    system("pause");
}

```

```

    return 0;
}

```

Agarkonteynernio‘zgartirishdanhimoyayaqilishvateskarisikerakbo‘lsa, const_reverse_iterator tipidagiiteratorordanvauningcrbegin() va crend()funksiyalaridanfoydalanishmaqsadgamuvofiqdir.Yuqoridagidasturgaquyida gifragmentniyozishyetarlibo‘ladi.

```

for (vector<int>::const_reverse_iterator iter = v.crbegin(); iter !=
v.crend(); ++iter)
    std::cout << *iter << " | ";

```

Shuningdekiteratorlariqo‘shimaamallargahamega (list va forward_list konteynerlaridantashqari):

- **iter + n** - n tapozitsiya keyingielementniko‘rsatuvchiiterator niqaytaradi.
- **iter - n** - n tapozitsiya oldingielementniko‘rsatuvchiiterator niqaytaradi.
- **iter += n** -iterator ni n - chipozitsiyagao‘tkazadi
- **iter -= n** -iterator ni n - chipozitsiyagao‘tkazadi
- **iter1 - iter2** - iter1 va iter2 larorasidagipozitsiyalar soniniqaytaradi
- **[>], [>=], [<], [<=]** -taqqoslashamallari.

Agar iterator oxirigayaqinelementniko‘rsatsaboshqasidankatta.

5.9-dastur. Iteratorning qo‘shimcha amallaridan foydalanish.

```

#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int myints[] = { 1, 2, 3, 4, 5 };
    vector<int> v (myints, myints+
sizeof(myints)/sizeof(myints[0]));
    for (auto i = v.begin(); i < v.end(); i++)
    {
        cout << *i << " | ";
    }
    cout << endl;
    auto it1 = v.begin();
    auto it2 = it1 + 2;
    cout << "it1 + 2 => " << *it2 << endl;
}

```



```

    auto it3 = v.end() - 3;
    cout << "it1 - 3 => " << *it3 << endl;
    it2 += 2 ;
    cout << "it2 + 2 => " << *it2 << endl;
    it3 -= 2;
    cout << "it3 - 2 => " << *it3 << endl;
    cout << "it2 - it1 => " << it2 - it1 << endl;
    cout << "(it1 < it2) => " << (it1 < it2) <<
endl;
    cout << "(it1 >= it2) => " << !(it1 < it2) <<
endl;
    cout << "(it1 != it2) => " << !(it1 == it2) <<
endl;
    cout << "(it1 == it2) => " << (it1 == it2) <<
endl;
    cout << "(it1 != it2) => " << (it1 != it2) <<
endl;
    cout << endl;
    system("pause");
    return 0;
}

```

5.9-dastur.Output

```

1 | 2 | 3 | 4 | 5 |
it1 + 2 => 3
it1 - 3 => 3
it2 + 2 => 5
it3 - 2 => 1
it2 - it1 => 4
(it1 < it2) => 1
(it1 >= it2) => 0
(it1 != it2) => 1
(it1 == it2) => 0
(it1 != it2) => 1

```

Taqqoslashamallariuchunko‘chiribo‘tishiteratorlarivabazaviyiterat
orlaruchunba‘ziC++dasturlashdafarqlaribor.

5.1-jadval. Taqqoslashamallariyozishuslublari.

Ko‘chiribo‘tishiteratorlariuchun.	Ekvivalentbazaviyiteratorlariuchun.

move_ita == move_itb	basic_ita == basic_itb
move_ita != move_itb	!(basic_ita == basic_itb)
move_ita < move_itb	basic_ita < basic_itb
move_ita <= move_itb	!(basic_itb < basic_ita)
move_ita > move_itb	basic_itb < basic_ita
move_ita >= move_itb	!(basic_ita < basic_itb)

Xotirani taqsimlovchilar va ularga qo‘yilgan talablar. Dinamik xotirabirlanishlashda maxsus turlarni, fragmentlardan foydalanish bo‘lmasa, ko‘piga algoritmlardan foydalanish samarasiz bo‘lishi mumkin.

Bunga misol sifatida ikki holatni ko‘rib chiqamiz. `new` va `delete` operatorlarining qayta aniqlanib yuklanishida sintaktik konstruktorlarni kichik roq bo‘ladivada turlar lokalizatsiya qilish odatiy bo‘ladi.

Shuningdek protsessordagi amallarni tizimida ham qayta aniqlash sodir bo‘ladi.

Avvalo, xotirani ishlatishni tezlashtirishda aqlli allocator qanchalik foydasiz bo‘lgini aniqlash lozim.

Buning uchun oddiy test misollarini (C++ va S#) tillarida ko‘rib chiqaylik (buxotirabirlanishlash uchun yaxshi menejer hisoblanadivabo‘ektlarni avlodlarga ajratadi, turlar o‘lchamidagi ob‘ektlar uchun turlar pullardan (joylar) foydalanadi).

C++ dagidastur

```
class Node {
public:
    Node* next;
};
// ...
for (int i = 0; i < 10000000; i++) {
    Node* v = new Node();
}
```

S# dagidastur

```
class Node
{
    public Node next;
}
// ...
for (int l = 0; l < 10000000; l++)
```

```
{
    var v = new Node();
}
```

Dasturlarsharsimonvakumusuliniqo‘llabhamvaqtbo‘yichasolishtiril sa 10 barovar (62 ms va 650 ms) farqbo‘lmoqda. Shuningdek C#ishinitugutdi, ammoC++daesayaxshiginixotiraaajratilan, buhalio‘chirishkerak.

Pulob’ektlari. Pulob’ektlariuchunaniqechim - OSdankattaxotirablokiniolishvaunitengbloklilo‘lchamlargabo‘lishdir sizeof(node). Xotiraniajratishdabloknipuldanolinadivaunibo‘shatishdaunipulgaqaytaril adi. Pulnitashkilqilishningengosonusuli - bittaaloqastekni (stack) ishlatishdir.

Maqsaddasturdaminimalqaytaaniqlashlarbo‘lganiuchun, BlockAlloc sinfnimerosxo‘risifatida Node sinfinitanlashdir.

```
class Node : public BlockAlloc<Node>
```

Avvalo, OSyoki C-runtime dankattabloklaruchunbirpulolishkerak. Buni malloc vafreefunksiyalaribilantashkilqilishmumkin, lekinkattasamaradorligiuchun (ortiqchaabstraksiyabosqichinio‘tkazishuchun), Virtualloc/Virtualfreyefunksiyalaridanfoydalanishmaqsadgamuvofiq. Buvazifalar 4 karralibo‘lganbloklargaxotiraaajratadiva 64 karralibo‘lganbloklarjarayonmanziliuchunoraliqsaqlabturadi. Birvaqtningo‘zidacommit va reserveimkoniyatlariko‘rsatilganholdayanabirdarajadako‘tarilib, manziloralinqnirezervlashvabittachaqirishbilanxotirabloklariniajratamiz.

PagePoolsinfidasturfragmenti.

```
inline size_t align(size_t x, size_t a) { return
((x-1) | (a-1)) + 1; }
//#define align(x, a) (((x)-1) | ((a)-1)) + 1)

template<size_t PageSize = 65536>
class PagePool
{
public:
    void* GetPage() {
        void* page = VirtualAlloc(NULL,
        PageSize, MEM_COMMIT | MEM_RESERVE,
```

```

PAGE_READWRITE);
        pages.push_back(page);
        return page;
    }

    ~PagePool() {
        for (vector<void*>::iterator i =
pages.begin(); i != pages.end(); ++i) {
            VirtualFree(*i, 0,
MEM_RELEASE);
        }
    }
private:
vector<void*> pages;
};

```

Berilganoʻlchamgamos, pulbloklarni hosil qilish uchun BlockPool sinfini yaratamiz.

BlockPool sinfining dastur fragmentlari:

```

template<class T, size_t PageSize = 65536, size_t
Alignment = 8 /* sizeof(void*) */>
class BlockPool : PagePool<PageSize>
{
public:
    BlockPool() : head(NULL) {
        BlockSize = align(sizeof(T),
Alignment);
        count = PageSize / BlockSize;
    }

    void* AllocBlock() {
        // todo: lock(this)
        if (!head) FormatNewPage();
        void* tmp = head;
        head = *(void**)head;
        return tmp;
    }
}

```

```

        void FreeBlock(void* tmp) {
            // todo: lock(this)
            *(void**)tmp = head;
            head = tmp;
        }
private:
    void* head;
    size_t BlockSize;
    size_t count;

    void FormatNewPage() {
        void* tmp = GetPage();
        head = tmp;
        for(size_t i = 0; i < count-1; i++)
        {
            void* next = (char*)tmp +
BlockSize;

            *(void**)tmp = next;
            tmp = next;
        }

        *(void**)tmp = NULL;
    }
};

```

Potoklarorasi dagisinxronzatsiyanitashkil qilish uchun berilgan manzil **todo: lock(this)** izohga olib qo'yilgan. Buning uchun EnterCriticalSection yoki boost::mutex dan foydalanish ham mumkin.

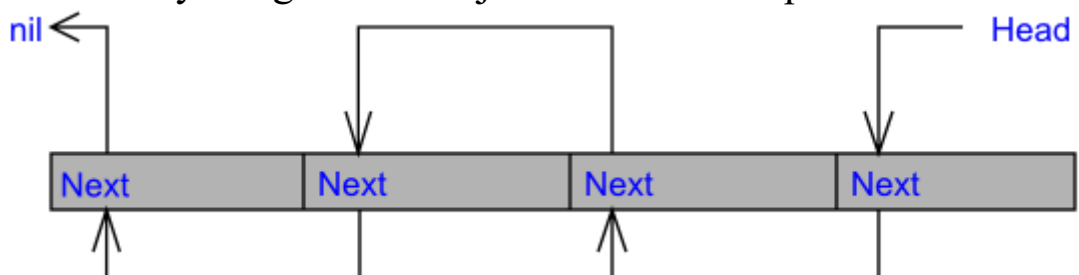
Agar biro narsa yozilgan bo'lsa, bloklarni majmuini formatlashda abstrakt FreeBlock pulga blok qo'shish shartlatilmaydi. O'ylab ko'ring.

```

for (size_t i = 0; i < PageSize; i += BlockSize)
    FreeBlock((char*)tmp+i);

```

FIFO tamoyilidagi bloklarni majmuini teskaribo'lib qoladi.



5.1-rasm. FIFO tamoyilidagi bloklarni majmuini.

Puldanketma-ketso‘ralganbirnechablokmanzillarkamayibketadi.
 Shuningdekprotsessororqagaqaytishniyoqtirmaydi,
 chunkiuningPrefetchamaliibuziladi
 zamonaviyprotsessorlaruchuntegishliemas).
 Agartakrorlanishbilanalmashtirilsachi?

(UPD:

```
for (size_t i = PageSize-(BlockSize-(PageSize%BlockSize)); i != 0; i -=
BlockSize) FreeBlock
```

Takrorlanishdaalmashtirishlarmanzillarbo‘yichaorqagabajariladi.
 Tayyorgalikko‘rilganidanso‘ng, sinfob‘ektniyozishmumkin:

```
template<class T>
class BlockAlloc{
public:
    static void* operator new(size_t s) {
        if (s != sizeof(T)) {
            return ::operator new(s);
        }
        return pool.AllocBlock();
    }
    static void operator delete(void* m, size_t s) {
        if (s != sizeof(T)) {
            ::operator delete(m);
        } else if (m != NULL) {
            pool.FreeBlock(m);
        }
    }
    static void* operator new(size_t, void* m) {
        return m;
    }
    static void operator delete(void*, void*) {
    }

private:
    static BlockPool<T> pool;
};
template<class T> BlockPool<T> BlockAlloc<T>::pool;
```

Sinfdaularishlagandanimauchunif (s != sizeof(T)) kerak.
 Tbazaviysinfdanmo‘rosxo‘rsinfyاراتilganvao‘chirilganbo‘lsa?
 degansavolbo‘lishimumkin.

Merosxo‘rlarodatdaoddiynew/delete operatorlaribilanishlatiladi, shuningdek BlockAlloc nihamjoylatirishmumkin. Buyo‘l bilan osonvaxavfsiz dastur dabi ornarsabuzibqo‘rqmasdan, pullarfoydalanishgaqaratilgansinflananiqlashmumkin. Birnechtamerosxo‘rlarhambubilanyaxshiishlaydi.

BlockAlloc danmo‘rosxo‘rsifatida Node niolamizvayanasinovo‘tkazamiz.

Sinovnatijasigaqarasak, 5 barovartezva 120 msdabajarilmoqda. Ammo, c#allokatorhalihamyaxshi. Mumkinuyerdaoddiybo‘lmaganbiraloloqaliro‘yxatdanfoydalanilgandir. Agar, new operatoridanso‘ngzudlikbilan delete operatoriishlatilsa, xotirakamroqishlatiladivakeshdama‘lumotlarhamkamayadivanatijani 62 msdaolishmumkin. Qiziqarlitomonishundaki, . NET CLR da GC nikutmasdanpulgamosbo‘shqolganlokalo‘zgaruvchinitezqaytaradi.

Konteynervauniquo‘llanilishi. Xotiradako‘phollardamerosxo‘rob’e ktlarinisaqlovchisinflanrvujudgakeladi.

Ularningxotiradano‘chirilishibazaviysinfbilanbog‘liqdir. Masalan, sinfbo‘lsinvatugunichidagimatndanolinganmo‘rosxo‘rlariNodevaAttribu tehamda(char*)qatorlardanto‘ldirilgan.

Yokipapkaniquaytao‘qishvayanao‘zgartirishda, birmartayuklanadifaylmenejerifayllarvakataloglarro‘yxati.

Yuqoridako‘rsatilgandek, new operatoriganisbattan delete operatorifoydalanishqiyinroqvamuammoliroqbo‘ladi.

Bazaviysinfob’ektibilanbog‘liqkattablokdamerosxo‘rsinfob’ektlariuchun xotiraaajratishkerak. Bazaviysinfob’ektiniyo‘qqilishpaytida, destruktorodatdagidek, merosxo‘rsinfuchunchaqiriladi, lekinxotiraniqaytarishingizshartemas — bukattablokdaozodqilinadi.

PointerBumpAllocator sinfyaratamiz. Businfkattablokdanturlio‘lchamdagiqismlarnikesibtashlashvaeskisitugag ach, yangikattabloknitanlashgaimkonberadi.

PointerBumpAllocatorsinfiningdasturfragmenti.

```
template<size_t PageSize = 65536, size_t Alignment
= 8 /* sizeof(void*) */>
class PointerBumpAllocator
{
public:
    PointerBumpAllocator() : free(0) { }

    void* AllocBlock(size_t block) {
```

```

        // todo: lock(this)
        block = align(block, Alignment);
        if (block > free) {
            free = align(block,
PageSize);

            head = GetPage(free);
        }
        void* tmp = head;
        head = (char*)head + block;
        free -= block;
        return tmp;
    }

    ~PointerBumpAllocator() {
        for (vector<void*>::iterator i =
pages.begin(); i != pages.end(); ++i) {
            VirtualFree(*i, 0,
MEM_RELEASE);
        }
    }

private:
    void* GetPage(size_t size) {
        void* page = VirtualAlloc(NULL,
size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
        pages.push_back(page);
        return page;
    }

    vector<void*> pages;
    void* head;
    size_t free;
};
typedef PointerBumpAllocator<> DefaultAllocator;

```

Nihoyat, new va delete operatorlaribilan childObject nituzamizvaberilgan allocator bilanmurojaatqilamiz:

```
template<class T, class A = DefaultAllocator>
```



```

struct ChildObject
{
    static void* operator new(size_t s, A& allocator) {
        return allocator.AllocBlock(s);
    }
    static void* operator new(size_t s, A* allocator) {
        return allocator->AllocBlock(s);
    }

    static void operator delete(void*, size_t) { }
    static void operator delete(void*, A*) { }
    static void operator delete(void*, A&) { }
private:
    static void* operator new(size_t s);
};

```

Child sinfigao‘zgaruvchilarniqo‘shishuchunbarchae‘lonlarni new operatoriorqaliamalgaoshirishkerakbo‘ladi. New operatoriquyidagichabo‘ladi.

```
new (... parametrlar... ) ChildObject (...konstruktorparametrlari... )
```

QulaylikuchunA&va A* uchun new operatorlarniqo‘llaymiz.

```
node = new(allocator) XmlNode(nodename);
```

Agarallokatorajratuvchisifatidaqo‘shilsa, ikkinchisidanfoydalanishqulayroq:

```
node = new(this) XmlNode(nodename);
```

Bundantushinarlibo‘ladiki, ortiqchabelgilardanqochishdaamallarnibo‘lishuchunko‘rsatkichvahavolal arkonvertatsiyabo‘ladi.

Ob’ektyaratishdaqaysi New delete operatoridanfoydalaniganligigaqaramasdan, operatoriyordamidao‘chirishamalgaoshirilmaydi, kompilyatorningo‘zistandart delete operatoridanfoydalanadi. Quyidagichasintaktikasosida:

```
delete node;
```

Agar ChildObject sinfiob’ektiyokiuningmerosxo‘riob’ektidanfoydalanayotganbo‘lsangiz, new istisnovaqtidaoperatorigamos delete orperatorichaqiriladi. Shuninguchungbuob’ektdanfoydalangandabirinchi size_t parametрни void*gao‘zgartirishlozim.

New operatorini private bo‘limigajoylashtirish, uniallokatorsizishlashagaruxsatbermaydi.

Standartbo‘yichataqsimlovchilar.

1. Tuzilmali o‘zgaruvchisi uchun dinamik xotira ajratish.

Tuzilmali o‘zgaruvchisi uchun xotira ajratish va bo‘shatish.

Quyidagi tavsifga ega bo‘lgan data tuzilmasi berilgan bo‘lsin:

```
struct Date
{
    int day;
    int month;
    int year;
};
```

struct

Datetipidagi o‘zgaruvchi uchun xotiradan joy ajratish va foydalanish uchun quyidagicha dastur fragmentini yozish kerak:

5.10-dastur. xotiradan joy ajratish va foydalanish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

// tuzilma, kun, oy, yil;
struct Date
{
    int day;
    int month;
    int year;
};

int main(){
    // new operatori bilan xotiradan joy ajratish
    struct Date * pd; // struct Date ga ko‘rsatkich

    try{
        pd = new struct Date; // xotiraga joy
    ajratish
    }
    catch (bad_alloc ba) {
        cout << "Xorita ajratilmadi" << endl;
```

```

        return -1;
    }

    // Agar xotira ajratilgan bo'lsa, unga
16.14.2020 qimat qilib beramiz.
    pd->day = 16;
    pd->month = 04;
    pd->year = 2020;

    cout << pd->day << ":" << pd->month << ":" <<
pd->year << endl;

    // pd o'zgaruvchisini xotiradan bo'shatish
    delete pd;
    system("pause");
    return 0;
}

```

2. Sinfob'ektiuchundinamikxotiraaajratish.

CDayWeek

sinfob'ektiuningko'rsatkichigaxotiraaajratishvafoydalanishnikeltiramiz

.

5.11-dastur. xotiraaajratishvafoydalanish.

```

// ConsoleApplication1.cpp : main project file.

#include "stdafx.h"
#include <iostream>
using namespace std;

// sinf, hafta kunlari
class CDayWeek{
    int d;
public:
    // konstruktor
    CDayWeek() { d = 1; }

    // funksiyalari
    int Get(void) { return d; }
    void Set(int nd) {

```

```

        if ((1<=nd)&&(nd<=7))
            d = nd;
    }
};

int main(){
    // new bilan xotiradan joy ajratish
    CDayWeek *p;

    try {
        // p ko'rsatkich uchun xotiradan joy
        ajratish
        p = new CDayWeek(); // CDayWeek sinf
        konstruktorini chaqirish
    }
    catch (bad_alloc ba) {
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }

    // qimat berish
    p->Set(3);
    cout << p->Get() << endl;
    // p ko'rsatkichdan xotirani bo'shatish
    delete p;
    system("pause");
    return 0;
}

```

3.

new

operatoriyordamidamassivlargaxotiradanjoyajratishvafoydalanish.new
operatorimassivuchunxotiraaajratishuchunfoydalanishmumkin.

Massivgaxotiraaajratishda new operatorningumumiyshakli:

```
ptrArray = newtype [size] ;
```

Bunda, *ptrArray* – xotiraaajratilishikerakbo'lganmassivnomi,
type – elementlarningtiplari, size –
ularbazaviyvaixtiyoriybo'lishimumkin, massiv o'lchami
(elementlar soni).

Delete operatori yordamida massivob'ektidan xotira nibo'shatish uchun quyidagi fragment yoziladi:

```
delete[] ptrArray;
```

4. Bazaviy tipliko'rsatkich massivigadina mikxotira ajratish va foydalanish. Quyida float tipidagi ko'rsatkich massivi uchun xotira ajratadi. So'ngra qator elementlari ixtiyoriy qiymatlar bilan to'ldiriladi. Shundan so'ng ajratilgan xotira delete [] operatoritomonidan bo'shatiladi.

```
float * ptrArray;  
ptrArray = new float[10];  
int d;  
for (int i = 0; i < 10; i++)  
    ptrArray[i] = i * 2 + 1;  
  
d = ptrArray[3];  
delete[] ptrArray;
```

5. Tizimlil o'zgaruvchilarmassivi uchun xotira ajratish va undan foydalanish. 3

tatalabatuzilmalarmassivi uchun xotira ajratish va bo'shatish ko'rsatilgan. shuningdek, tuzilmalarmassivining element maydonlariga murojaat qilish ham ko'rsatilgan.

5.12-dastur. Xotira ajratish va foydalanish.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
  
struct TStudent  
{  
    string name;  
    string numBook;  
    int course;  
    float rank;  
};  
  
int main(){  
    int n_students = 3;  
    TStudent * pS;
```

```

try{

    pS = new struct TStudent[n_students];
}
catch (bad_alloc ba){
    cout << "Xotira ajratilmadi " << endl;
    cout << ba.what() << endl;
    return -1;
}

pS->name = "Aliyev Akbar";
pS->numBook = "965874123";
pS->rank = 3.93;
pS->course = 1;

(pS + 1)->name = "Bahromov Batir";
(pS + 1)->numBook = "20930032";
(pS + 1)->rank = 4.98;
pS[1].course = 3;

pS[2].name = "Qobilov Usmon";
pS[2].numBook = "12128983";
pS[2].rank = 4.32;
pS[2].course = 2;

cout << pS->name.c_str() << endl;
cout << pS[1].rank << endl;
cout << (pS + 2)->numBook.c_str() << endl;

delete[] pS;
system("pause");
return 0;
}

```

6. Massivob'ektlari uchun xotira ajratish va undan foydalanish. New operatoridan foydalangan holda massivob'ektlari uchun xotira ajratishko'rsatiladi va delete operator bilan xotiradan massivob'ektlarini bo'shatishko'rsatiladi.

5.12-dastur. Xotiraajratishvafoydalanish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

class CMonth{
private:
    int month;

public:
    CMonth() { month = 1; }

    CMonth(int nmonth){
        if ((1 <= nmonth) && (nmonth <= 12))
            month = nmonth;
        else
            month = 1;
    }

    int Get(void) { return month; }
    void Set(int nmonth){
        if ((1 <= nmonth) && (nmonth <= 12))
            month = nmonth;
    }
};

int main(){
    int nMonths = 12;
    CMonth * pM;

    try {
        pM = new CMonth[12];
    }
    catch (bad_alloc ba)
    {
        cout << "Xotira ajratimadi: " << endl;
        cout << ba.what() << endl;
        return -1;
    }
}
```

```

    }

    for (int i = 1; i <= 12; i++)
        pM[i - 1].Set(i);

    cout << pM[3].Get() << endl;
    cout << pM[5].Get() << endl;

    delete[] pM;
    system("pause");
    return 0;
}

```

Yuqoridagi dastur fragmentida massiv obektlarichki o'zgaruvchi 1 qiymatiga initsializatsiya qilinadi, chunki bu qiymat Cmonth() parametrlarisiz konstruktordao'rnatiladi.

Bu konstruktormassiv uchun initsializator vazifasini bajaradi. Biroq, sinfboshqa konstruktorni amalga oshiradi, ya'ni konstruktor bilan 1 qiymatli parametryoki parametrik konstruktorni. C++ sintaktikigako'ra, massiv obektlariniparametrik konstruktor bilan ishgatushiribbo'lmaydi.

Shuning uchun, cmonth sinf parametrlarisiz konstruktor amalga oshirish kerak. Cmonth() parametrsiz konstruktorsinfdastur fragmentidan olib tashlansa, massiv obekt laruchun xotira ajratish mumkin bo'lmaydi.

Bitta ob'ektlaruchun xotira ajratish mumkin, massiv uchun mumkin emas.

Agar ma'lum birlar sinf massiv ob'ektlari uchun xotira ajratish kerak bo'lsa, bu sinf parametrlarisiz konstruktor ga ega bo'lishi vajabajarilish kerak.

7. Massivo'lchamini dinamik kattalashtirish (kichiklashtirish) uchun xotira ajratish va foydalanish. Tuzilmalar uchun xotira niqayta ajratish, tuzilmalarni ishgatushirish. masalan, Dayweek tuzilishi tip uchun xotira qayta ajratish jarayoni ko'rsatiladi.

Xotira dinamik ravishda ajratish va qayta ajratish bu usulning xotira ni statika jratishganisbatanasosiy afzalligini hisoblanadi.

Kerak bo'lganda va kerak bo'lganda dasturda xotira ajratishingiz mumkin.

Dayweek tuzilishi parametrlarisiz konstruktoryordamida amalga oshiriladi (yuqoridagidek).

5.13-dastur. Xotira ajratish va foydalanish.

```

#include "stdafx.h"
#include <iostream>
using namespace std;

```



```

struct DayWeek{
    int day;
    DayWeek() { day = 1; }
};

int main(){
    DayWeek * p;

    try {
        p = new DayWeek[5];
    }
    catch (bad_alloc ba){
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }
    for (int i = 0; i < 5; i++)
        p[i].day = i + 1;
    int d;
    d = (p + 1)->day;

    DayWeek * p2;

    try {
        p2 = new DayWeek[7];
    }
    catch (bad_alloc ba){
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }
    for (int i = 0; i < 5; i++)
        p2[i] = p[i];
    delete[] p;

    p = p2;
}

```

```

    for (int i = 5; i < 7; i++)
        p[i].day = i + 1;

    d = p[5].day;
    d = (p + 6)->day;

    delete[] p;

return 0;
}

```

Main() funkssiyasidaxotiraavval 5
 tatuzidmadaniboratmassivuchunajratiladi. Keyinbuxotira 7
 tuzilmalimassivuchunqaytaajratiladi (reallocated).
 Buninguchunqo'shimcha p2 ko'rsatikichishlatiladi.
 Qaytataqsimlashdaavval p2 (7 element) uchunxotiraajratiladi.
 So'ngrama'lumotlar p dan p2 gako'chiriladi. Bu
 ppko'rsatikichuchunajratilganedi, xotirabo'shatiladi (5 taelementdan).
 Keyingiqadamdapqiymatp2 gao'rnatiladi. Shundayqilib,
 harikkiko'rsatikichbirxilxotiramaydoniko'rsatadi.
 Massivbilanishlagandanso'ng, p massivuchunxotiraajratiladi.
 Buyuqoridakurrsatibo'tilgandasturlashdinamikxotiraniajratishvafoy
 dalanishningstandartshakllaribo'libhisoblanadi.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Standartalgoritmlarkutubxonasinimauchunyaratilgan.
2. Algoritm kutubxonasi tomonidan taqdim etiladigan funksiyalar odatdanechtatoifagabo'linadivaqaysilar?
3. Mutatorlarni mavqanday vazifani amalga oshiradi.
4. Berilgan qiymat bo'yicha elementlarni qidirish algoritmi qaysi vanechtaparametr qabul qiladi?
5. Konteynerdama vjud elementlar orasidama'lumbir qiymat saqlaydigan element qidirish uchun qaysi funktsiya ishlatiladi?

6. Konteynerdan berilgan elementlarini sanash vazifasi ni bajaruvchi algoritmlar nisanabbering?
7. Funksiya taqqoslash uchun parametrlar sifatida ikki ta parametroladi. Bu qaysi funksiya.
8. Ikki ob'ektning qiymatlarini almashtirish algoritmini ayting.
9. Tartiblangan to'plamlar ni boshqartirish uchun tartiblangan to'plamni o'z ichiga olish yoki o'lmasligini tekshirish algoritmini ayting.
10. Iteratorlar nima uchun foydalaniladi.
11. **begin()** va **end()** funksiyalaridan nima maqsadda foydalaniladi.
12. Agar **begin()** va **end()** funksiyalar o'zaro zarf bo'lmasa, ularning orasida kamida nechta element bor.
13. Keyingi elementga murojat qilish uchun iteratorni harakatlantirish amaliyati nima.
14. Agar konteynerda o'zgarishsiz elementlar bo'lsa, bu holda konteyner elementlariga murojat qilish uchun nima dan foydalanish sh kerak.
15. **cbegin()** va **cend()** funksiyalaridan nima maqsadda foydalaniladi.
16. Agar konteynerda o'zgaruvchi qiymatli elementlar bo'lsa, bu holda konteyner elementlariga murojat qilish o'zgarishsiz iteratorlardan foydalanish sh mumkinmi.
17. Iteratorlari qo'shimcha amallar qaysi konteynerlar uchun ishlatilmaydi.
18. Qaysi amal `iter1` va `iter2` lar orasida gap o'zgarishsizlikni ta'minlaydi.
19. `(move_it1 != move_it2)` va `!( move_it1 == move_it2)` munosabatlar bir xilmi, bir xil bo'lsa nima uchun.
20. Xotira ni taqsimlovchilar va ularga qo'yilgan talablarni nima uchun kerak.
21. Memoriya o'zgarishsizlikni ta'minlash uchun `new/delete` operatorlari bilan ishlatiladi, shuningdek `BlockAlloc` nima uchun joylatish kerak.
22. Agar, `new` operatoridan so'ngi xotira bilan `delete` operatori ishlatilsa, xotira kamroq ishlatiladi va qandayda gima'lumotlar ham kamayadi.
23. Kaysi operator dan `new` operator yoki `delete` operator foydalanish qiyinroq va muammoliroq, sababini tushuntiribbering.
24. `New` operator qaysi bo'limiga joylashtirilsa, unallokatsizlash agar xosat bermaydi.
25. Tuzilmali o'zgaruvchi uchun xotira ajratish yoki bo'shatish uchun qanday amallar bajarish kerak.
26. Sinf ob'ekti uchun dinamik xotira ajratish uchun qanday amallar bajarish kerak.

27. new
operatoriyordamidamassivlarga xotiradan joy ajratish va foydalanish uchun qanday amallar bajarish kerak.

28. Bazaviy tipliko'rsatkich massiviga dinamik xotira ajratish va foydalanish uchun qanday amallar bajarish kerak.

29. Ma'lum birlashtirilgan massivob'ektlari uchun xotira ajratish kerak bo'lsa, buni qanday parametrlilik konstruktorga egabo'lishi kerak.

30. Massivo'lchamini dinamik kattalashtirish (kichiklashtirish) uchun xotira ajratish va foydalanish uchun qanday amallar bajarish kerak.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	Standart algoritmlarga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlarni osida bajarang. 👉 Bunday dasturda gibab'zior'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include <algorithm> #include <array> #include <iostream> #include <utility> #include <vector> #include <string> using namespace std; bool myfunction (int i, int j) { return i<j; }</pre>	1. Dasturda nechta global funksiyalar yaratilgan. _____ _____ — _____ —
	2. Massivda berilgan massiv elementlarini baribir tekshiruvchi fragmentni o'zgartiring. _____ — _____ — _____ —

<pre> int main() { array<int,10> arr = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19 }; int container[] = {5,10,15,20,25,30,35, 40,45,50}; int continent[] = {40,30,20,10}; string myStr[] = {"air","water","fire" ,"earth"}; vector<string> foo (myStr, myStr + sizeof(myStr)/sizeof(myStr[0])); vector<string> bar (4); int myints[] = {10,20,30,30,20,10,10 ,20}; vector<int> v(myints,myints+8); cout << "Qidirish va o'zgartirish uchun son kirit: "; int search; int replace; cin >> search >> replace; auto found = find(arr.begin(), arr.end(), search); </pre>	3. Dasturning qaysi fragmentini izohlab qo'ysa, massivdaberilgan massiv elementlaribor deb chiqadi. 4. Massivdaberilgan massiv elementlaribor degan fragment qachon ishlamaydi. 5. Move amalini nechamarta bajarilgan. 6. Birinchi qaysi kutubxonaga 'luqli. 7. Ikkinchi qaysi kutubxonaga 'luqli.

<pre> if (found == arr.end()) { cout << search << " topilmadi. " << endl; } else { *found = replace; } for (int i : arr){ cout << i << ' '; container[i] = i; } cout << endl; sort (container,container+ 10); sort (continent,continent+ 4); if (includes(container,c ontainer+10,continent ,continent+4, myfunction)) cout << "ichma ich massiv !\n"; else cout << "ichma ich massiv emas !\n"; cout << endl; </pre>	8. Foovektorningelementlariekranga chiqarishfragmentinikiriting

<pre> move (foo.begin(), foo.begin()+4, bar.begin()); foo = move (bar); cout << endl; system("pause"); return 0; } </pre>	
9. Dasturda jamibo‘lib, nechao‘zgartirish kiritildi. _____ 10. Shu dasturning analogini yaratish sizga mustaqil vazifa.	

IKKINCHI ASSISMENT TOPSHIRIG‘I	
	Iteratorlar va ularning qo‘llanilishiga oid berilgan quyidagi dastur bo‘yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani. 👉 Bunday dasturda giba‘zio‘zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre> #include "stdafx.h" #include <iostream> #include <vector> using namespace std; int main() { vector<int> myvector; for (int i = 0; i < 15; i++) { </pre>	1. Dasturda <code>auto iter = myvector.begin();</code> uchun iterator yozing. _____ _____ _____ _____ _____

<pre> myvector.push_back(rand() % 100); } int myints[] = {16,2,77,29,28}; const vector<int> myvector_one (myints, myints + sizeof(myints) / sizeof(myints[0])); auto iter = myvector.begin(); while(iter!=myvector.end()) { cout << *iter << " "; ++iter; } /* vector<int> v (myints, myints+ sizeof(myints)/sizeof(myints[0])); for (auto i = v.begin(); i < v.end(); i++) { cout << *i << " "; } */ cout << endl; system("pause"); return 0; } </pre>	2.Dasturdavector<int> myvector;uchuno‘zgarmasiteratorya rating. 3. Dasturdagiortiqchakutubxonal arnio‘chiribtashlangvasabablariniayt ing. 4. Ikkitavektornio‘zarosolishtirin g. 5. Vektorlarniteskarichiqarishdast urfragmentlariniyozing. 6. Vector saralangvavektordagi 2,5,7 elementlarnipozitsiyasinianiqlang.

	7. Dasturdagi izohga olingan dastur fragmenti olib tashlansa, dasturda qandayo 'zgarishlar bo'ladiv natijasichi. _____ _____ _____ _____ _____ _____ _____
8. Dasturda jamibo'lib, nechao'zgartirish kiritildi. _____ 9. Shu dasturning analogini yaratish sizga mustaqil vazifa dir.	

UCHINCHI ASSISMENT TOPSHIRIG'I	
	Xotirani ajratish va foydalanishga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajaring. 👉 Bunda dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include <iostream> using namespace std; struct TStudent {</pre>	1. Dasturda nechta yangi tipdagi ob'ektlari yaratilgan. _____ _____ _____ _____ _____

<pre> string name; string numBook; int course; float rank; }; int main(){ int n_students = 3; TStudent * pS; try{ pS = new struct TStudent[n_students]; } catch (bad_alloc ba){ cout << "Xotira ajratilmadi " << endl; cout << ba.what() << endl; return -1; } pS->name = "Aliyev Akbar"; pS->numBook = "965874123"; pS->rank = 3.93; pS->course = 1; (pS + 1)->name = "Bahromov Batir"; (pS + 1)->numBook = "20930032"; (pS + 1)->rank = 4.98; pS[1].course = 3; </pre>	2. Dasturdagi barcha talabalar haqidagi ma'lumotni krangachiqaring. 3. Massivo'lchamini dinamik katta lashtirish uchun fragment yozing, masalan 5 talaba uchun. 4. Yangi tip uchun get() va set() funksiyalarini yarating. 5. Massivdan berilgan qiymatni izlash dastur fragmentini yozing.

<pre> pS[2].name = "Qobilov Usmon"; pS[2].numBook = "12128983"; pS[2].rank = 4.32; pS[2].course = 2; cout << pS- >name.c_str() << endl; cout << pS[1].rank << endl; cout << (pS + 2)- >numBook.c_str() << endl; delete[] pS; system("pause"); return 0; } </pre>	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

2.2. Sonlisinflarbilanishlash

 Sonli sinfvaularningfunksiyalariasosida complex, valarray, slice, gsilce, indirect_array, mask_array sinflarivafunksiyalari, intervalsonlisinfvafunksiyalari, sonlisinfelementlariniyaratish, foydalanishgaasosidataqqoslashfunksiyalari, dasturlashdao‘zlarinitutishlari, uslublari, amallari, talablari, vazifalarivausullarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

Bilimlarnimustahkamlashuchun 35
 tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 6
 taassimenttopshirig‘ivaharassimentda 7(6) tatopshiriq, jami
 49tatopshiriqberilgan.
 Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍Kalitso‘zlar. kutubxona, sinf, shablon, matematikamallari, kompleksson, to‘plam, to‘plamosti, to‘lamustidaamalar, saralash, intervalson.

✓Bilishshartbo‘lgantushunchalar. Tiptushunchasi, sinfvasinfob’ekti, merosxo‘rsinf, to‘plam, massiv, elementgamurojaat, funksiyavako‘rsatkich, ma’lumotlarnikirishvachiqishi, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo‘llabquvvatlovchimu hitdaishlashnibilishlozim.

☞Bilibolasiz. Sonlisinfvaularningfunksiyalari, komplekssonlarbilanishlashusullarivafunksiyalari, valarray soniob’ektibilanishlash, muo‘roso‘rsinfasosidato‘plamlarniustidaamallarnibajarish, surish, shartlarasosidayangito‘plamnihosilqilish, qismto‘plamlarniyaratish, intervalsonvauningfunksiyalarnio‘rganishingiz mumkin.

REJA

1. Complex sonlisinfvafunksiyalari
2. Vallaray sonlisinfvafunksiyalari
3. Slice va gsllice sonlisinflarivafunksiyalari
4. Qo‘shimchasonlisinflar

KIRISH

Hayotturmush tarzida foydalaniladigan sonlarto‘plamlardan boshqaso n larto‘plamivaularustidabajarilajiganamallarmavjud. Masalan, maktabdan bilamizki, komplekssonlar, oraliqsonlar, qolaversa, zamonaviy ilmiy sohalar daturli fazolardaturlichasonlarto‘plami ishlatiladi. Intervalsonlari, noravshanto‘plamlarshularkabi. Busonlarto‘plami hayotdagi qaysidirtizimni model lashtirish va unijarayonla riga oid masalalarni yechish gaxizmat qiladi. Bumasalarni ayniqsakompyuterdayechishkerak. Shuninguchunba‘zibirko‘p foydalaniladigan sonlarto‘plami uchun sinflarya ratilgan. Busonlarda ularning amallari, funksiyalarixususiyatlarikeltirilgan. Bundaysonlarga complex, vallaray, slice, gsllice kabisinflarvamual liftomonidan 2013 yildayaratilgan interval sinfkutubxonalarinimisolqilibolishmumkin.

Complex (kompleks) sonlisinfvafunksiyalari. C++ standartikutubxonasidagikomplekssonlarsinfiob’ektmodelidan foydalanishning yaxshinamunasidir.

Arifmetikamallarqaytaaniqlashtufaylibusinfob’ektlario‘rnatilganma’lum ottilaridan birigategishlikabi ishlatiladi. Bundantashqari,

an'anaviyarifmetiko'zgaruvchilarvakomplekssonlarbirvaqtningo'zidaixti yoriyamallardaishtiroketishimumkin. (Eslatibo'tamiz, buyerdakomplekssonlarmatematikasigaoidumumiysavollarbilanshug'ull anmayapmiz.

Matematikabo'yichakomplekssonlarnazariyasinio'qishkerak).

Businfniishlatishuchuncomplexsarlavhafaylqo'shilishikerak.

Kichikdasturfragmentigaqarang:

```
#include <complex>
// ...
complex <double> a;
complex <double> b;
// ...
complex< double > c = a * b + a / b;
```

Kompleksvaarifmatikamallaro'zarohamkorlikdabajarailadi.

```
complex< double > a;
complex< double > complex_obj = a + 3.14159;
```

Komplekstiplarniarifmetiktiplarbilanhamyaratishvaularuchun operator= niishlatishmumkin.

```
complex< double > complex_obj;
double dval = 3.14159;
complex_obj = dval;
// ....
int ival = 3;
complex_obj = ival;
```

Ammoteskarisi,

ya'nikomplekstipniarifmetiktipgatenglashtirishmumkinemas.

```
complex< double > complex_obj;
double dval = complex_obj;
```

Bundaarifmetiktipniakslantirishxatodeganxabarberadi.

Kompleksning 2 qismibo'ladihaqiqiyvamavhum. Haqiqiy (real) yokimavhum – qismini oddiyarifmetiktipgaqiymatqilibberishmumkin. Komplekssonlarsinfio'znavbatidarealvamavhumqismlarniqaytaruvchiikk itafunksiyagaega.

Bizsinfa'zolaridanfoydalanishuchunsintaktikiyordamidaularniolishimiz mumkin:

```
double re = complex_obj.real();
double im = complex_obj.imag();
```

Yokibusintaktikgaekvivaletsintaktik:

```
double re = real(complex_obj);
```

```
double im = imag(complex_obj);
```

Komplekssonsinfito‘rtmatematikamallar - operatorlariqo‘llab-quvvatlaydi. Bularga[+=], [-=], [*=]va[/=]operatorlarikiradi.Masalan,

```
complex< double > complex_obj;  
complex< double > second_complex_obj;  
complex_obj += second_complex_obj;
```

Komplekssonlarnikiritish/chiqarishhamqo‘llab-quvvatlanadi. Chiqishoperatorivergullarbilanajratilganrealvamavhumqismlarniqavslarichidachopetadi. Masalan, chiqishoperatorlarinibajarishnatijasi:

```
complex< double > complex0( 3.14159, -2.171  
);  
complex< double > complex1( complex0.real() );  
  
cout << complex0 << " " << complex1 << endl;
```

output

```
(3.14159,-2.171) (3.14159,0)
```

Kirishtishoperatorlariquyidagiformatlarbilanishlaydi:

Komplekssoningrealqisminikiritishoddiy kiritiladi, masalan, 525.25; mavhumqismiqavslarichidakiritiladi, masalan, (123.5), to‘liqkiritishuchunqavsichidarealvamavhumqismlarivergulbilanajratiladi, Masalan, (12.5, 2.5). 6.1-dasturgaqarang.

6.1-dastur. Koplekssonlarnikiritishvachiqarish.

```
#include "stdafx.h"  
#include <iostream>  
#include <complex>  
  
using namespace std;  
  
int main(){  
    complex< double > a, b, c;  
    cout << "Complex sinfi oid turli t=farmatli 3  
ta son kiritibg:" << endl;  
    cin >> a >> b >> c;  
    cout << "Complex sinfiga oid sonlar:" << endl;  
    cout << a << b << c << endl;  
    system("pause");  
}
```

```
Complex sinfiga oid turli formatli 3 ta son
kiritibg:
```

```
3.1415
```

```
(-0.25)
```

```
(2.71, -9.8)
```

```
Complex sinfiga oid sonlar:
```

```
(3.1415,0)(-0.25,0)(2.71,-9.8)
```

Ushbufunksiyalardantashqari,
kopmlekssonlarsinfi quyidagi funksiyalariga:

```
sqrt(), abs(), polar(), sin(), cos(), tan(), exp(),  
log(), log10() va pow().
```

Yozish vaqtida mavjud C++ standart kutubxona sio'ng operandli (masalan, qo'shishi +=) operatorlarkomplekssonibo'lmasamurakkabtayinlashamallarini qo'llab-quvvatlamaydi. Masalan, buyozuv garuxsat berilmaydi:

```
complex_obj += 1;
```

C++ standartiko'ra, bu operatorlar bo'lishi kerak, ishlab chiqaruvchilarko'pinchastandardlarni yozib tugatishga etib kelolmaydilar.

Bunday amallarni amalga oshirish uchun o'z operatorimizni aniqlashimiz mumkin.

Buyerda qo'shish operatorini amalga oshiruvchi funksiyani variantikeltirilgan.

```
inline complex<double>& operator+=(  
complex<double>&cval, double dval ){  
    return cval += complex<double>( dval );  
}
```

Bu misolyordamida, turiboshqamurakkabtayinlash uchunuchoperatorlarifunksiyalarini yozish muammo emas.

Quyidagi dastur uchun murakkabtayinlash uchunuchoperatorlarifunksiyalarini qo'shishga tekshirish uchun uni ishga tushirish.

```
int main() {  
    complex< double > cval ( 4.0, 1.0 );  
    cout << cval << endl;  
    cval += 1;
```

```

    cout << cval << endl;
    cval -= 1;
    cout << cval << endl;
    cval *= 2;
    cout << cval << endl;
    cout /= 2.;
    cout << cval << endl;
    system("pause");
}

```

C++

standartidakomplekssonuchuninkrementvadekrementamallariningbajarili shiko‘rsatilmagan. Biroq, ularningsemantikasijudaaniq, agar `cval += 1` niyozaolsak, bu `cval`ninghaqiqiyqismini 1 bilanoshirishnianglatadi, undainkrementamalijudaosonko‘rinadi. Murakkab<

double>tipiuchunbuamallarniamalgaoshirishvaquyidagidasturniishgatus hirish:

```

int main() {
    complex< double > cval( 4.0, 1.0 );
    cout << cval << endl;
    ++cval;
    cout << cval << endl;
}

```

Buyuqoridagilardantashqarikomplekssonlaribilashishlashsinfiningk o‘plab, matematikfunksiyalarmavjud. AmmoC++standartigaqarabayrimamallardan, funksiyalardanfoydalanibbo‘lmaydi. Shuninguchunfoydalanuvchiningo‘zmustaqilkutubxonayaratibishlashim aql.

Valarray

sonsinfivauningfunksiyalari.YaqinoradaC++damassivelementlaribilani shlashvasamaralisaqlashuchun Valarray konteyneripaydobo‘ldi. BuhammaC++standartlaridabo‘lmasligihammumkin.

-Businfmassivelementindekslaribo‘yichabirlashtrishoperatorlarivael elementlaromatematikamallarnibajaradi.

-Vektorbilansolishtirganda Valarray sinfmatematikamallarnibajarishdasamaraliroqhisoblanadi.

Valarray sinfningochiqfunksiyalari:

apply()

bu massiv elementlaribo'yichabirvaqtdao'zgarishlarnibajaradivayangimas
sivqaytaradi.

sum() – massivdagielementlaryig'indisiniqaytaradi.

6.2-dastur. apply () va sum () funksiyalaridanfoydalanish.

```
#include "stdafx.h"
#include<iostream>
#include<valarray>
using namespace std;
int main(){
    int myints[] = { 5, 25, 55, 85, 115 };
    valarray<int> varr (myints,
sizeof(myints)/sizeof(myints[0]));
    valarray<int> varr_one ;

    varr_one = varr.apply([](int x){return x=x+5;});

    cout << "Massivning yangi qiymatllar: ";
    for (int &x: varr_one) cout << '\t' << x;
    cout << endl;

    cout << "Oldingi massiv sum() => ";
    cout << varr.sum() << endl;
    cout << "Yangi massiv sum() => ";
    cout << varr_one.sum() << endl;
    system("pause");
    return 0;
}
```

6.2-dastur.output

Massivning yangi qiymatllar: 10 30 60 90 120

Oldingi massiv sum() => 285

Yangi massiv sum() => 310

Min()

va

max()funksiyalarimosravishdamassivengkichikvaengkattaelementlariqi
matlariniqaytaradi.

```
cout << "min() => ";
    cout << varr.min() << endl;
```

```
cout << "max() => ";  
cout << varr.max() << endl;
```

shift()

berilgansonqiymatbo'yichaindeksasosidamassivni suribyangimassivqaytaradi. Agarsonmusbatbo'lsachapgasuriladi, manfiybo'lsa, o'ngasuriladi.

cshift()

berilgansonqiymatbo'yichaindeksasosidamassivni aylanashaklidasuribyangimassivqaytaradi. Agarsonmusbatbo'lsachapgasuriladi, manfiybo'lsa, o'ngasuriladi.

6.3-dastur. shift() va cshift()danfoydalanish.

```
#include "stdafx.h"  
#include<iostream>  
#include<valarray>  
using namespace std;  
int main(){  
    int myints[] = { 5, 25, 55, 85, 115 };  
    valarray<int> varr (myints,  
sizeof(myints)/sizeof(myints[0]));  
    valarray<int> varr_one, varr_two;  
    cout << "Massivning eski qiymatllar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.shift(-2);  
    cout << "Massivning shift(-2) qiymatllar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.shift(2);  
    cout << "Massivning shift(2) qiymatllar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.cshift(2);  
    cout << "Massivning cshift(2) qiymatllar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;
```

```

    varr_one = varr.cshift(-2);
    cout << "Massivning cshift(-2) qiymatllar: ";
    for (int &x: varr_one) cout << '\t' << x;
    cout << endl;

    system("pause");
    return 0;
}

```

6.3-dastur.output

Massivning eski qiymatllar:

Massivning shift(-2) qiymatllar:	0	0	5	25	55
Massivning shift(2) qiymatllar:	55	85	115	0	0
Massivning cshift(2) qiymatllar:	55	85	115	5	25
Massivning cshift(-2) qiymatllar:	85	115	5	25	55

swap () - birmassivni ikkinchi birmassivga almashtiradi.

```
varr.swap(varr1);
```

Yuqoridagilardan tashqari quyidagi funksiyalar mavjud:

operator=	Massiv qiymatini o'zlashtirish
operator[]	Massiv indeksidagi elementiga murojaat
resize	Massiv elementlar soni (hajmi) niqayta aniqlash
size	Massiv elementlar soni (hajmi)

Valarrayning standart amallari

```

valarray operator+() const;
valarray operator-() const;
valarray operator~() const;
valarray<bool> operator!() const;

```

Valarray tiplari ustida amallar:

```

valarray& operator*= (const valarray& rhs);
valarray& operator/= (const valarray& rhs);
valarray& operator%= (const valarray& rhs);
valarray& operator+= (const valarray& rhs);
valarray& operator-= (const valarray& rhs);
valarray& operator^= (const valarray& rhs);
valarray& operator&= (const valarray& rhs);
valarray& operator|= (const valarray& rhs);
valarray& operator<<= (const valarray& rhs);
valarray& operator>>= (const valarray& rhs);

```

Valarraytipivaodatiyoddiytiplarustidaamallar:

```
valarray& operator*= (const T& val);  
valarray& operator/= (const T& val);  
valarray& operator%= (const T& val);  
valarray& operator+= (const T& val);  
valarray& operator-= (const T& val);  
valarray& operator^= (const T& val);  
valarray& operator&= (const T& val);  
valarray& operator|= (const T& val);  
valarray& operator<<= (const T& val);  
valarray& operator>>= (const T& val);
```

Yuqoridakeltirilganoperatorilar Valarray sinfiningfunksiyalarihisoblendi.

Valarray tiplaribilanbajariladiganamallarniqaytaaniqlashmumkin. Bundabirinchiamal Valarray tipi = Valarray tipi * Valarray tipi, ikkinchiamal Valarray tipi = oddiytip * Valarray tipi, uchinchitip Valarray tipi = Valarray tipi* oddiytipketmaketligiasosidaaniqlangan.

[*] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const T& val);
```

[*] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const T& val);
```

[/] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator/ (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator/ (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator/ (const  
valarray<T>& lhs, const T& val);
```

[%] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator% (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator% (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator% (const  
valarray<T>& lhs, const T& val);
```

[+] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator+ (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator+ (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator+ (const  
valarray<T>& lhs, const T& val);
```

[-] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator- (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator- (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator- (const  
valarray<T>& lhs, const T& val);
```

[^] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator^ (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator^ (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator^ (const  
valarray<T>& lhs, const T& val);
```

[&] – mantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator& (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator& (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator& (const  
valarray<T>& lhs, const T& val);
```

[|] – mantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator| (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator| (const T&
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator| (const
valarray<T>& lhs, const T& val);
```

[<<] – chapgasurishamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator<< (const
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator<< (const T&
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator<< (const
valarray<T>& lhs, const T& val);
```

[>>] – o'nggasurishamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator>> (const
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator>> (const T&
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator>> (const
valarray<T>& lhs, const T& val);
```

[&&] – tezkormantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator&& (const
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator&& (const
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator&& (const
valarray<T>& lhs, const T& val);
```

[||] – tezkormantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator|| (const
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator|| (const
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator|| (const
valarray<T>& lhs, const T& val);
```

[==] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator== (const
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator== (const
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator== (const
```

<code>valarray<T>& lhs, const T& val);</code>

[!=] – mulohazaamaliniqaytaaniqlash:

<code>template <class T> valarray<bool> operator!= (const valarray<T>& lhs, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator!= (const T& val, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator!= (const valarray<T>& lhs, const T& val);</code>

[<] – mulohazaamaliniqaytaaniqlash:

<code>template <class T> valarray<bool> operator< (const valarray<T>& lhs, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator< (const T& val, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator< (const valarray<T>& lhs, const T& val);</code>

[>] – mulohazaamaliniqaytaaniqlash:

<code>template <class T> valarray<bool> operator> (const valarray<T>& lhs, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator> (const T& val, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator> (const valarray<T>& lhs, const T& val);</code>

[<=] – mulohazaamaliniqaytaaniqlash:

<code>template <class T> valarray<bool> operator<= (const valarray<T>& lhs, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator<= (const T& val, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator<= (const valarray<T>& lhs, const T& val);</code>

[>=] – mulohazaamaliniqaytaaniqlash:

<code>template <class T> valarray<bool> operator>= (const valarray<T>& lhs, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator>= (const T& val, const valarray<T>& rhs);</code>

<code>template <class T> valarray<bool> operator>= (const valarray<T>& lhs, const T& val);</code>

Slicesinfi.Bu - xizmatchisinfbolib, valarray sinfningsmerosxo'ridirvabiro'lchovlito'plamostilarniyaratishdaishlatiladi. Agar valarray massivniikkio'lchovlimatritsasifatidaqarasak, undanbiro'lchovlivektornihosilqilishmumkin.

Businf slice_array tipdagiob'ektnitavsiflovchiparametrlarnisaqlaydi. Agarto'plamostibilvositayaratilgan valarray sinftipidabo'lsa, sinfob'ekti valarray ob'ektiuchunargumentsifatidako'rsatiladi. To'plamostidasaqlanadiganqiymatlar valarray tipinioladivaquyidagixususiyatlarnio'zichigaoladi:

valarraydagiboshlang'ichindeks.
massivningumumiyuzunligivaundagielementlarsoni;
valarray elementlarningketma-
ketindekslariorasidagiqadamyokimasofanianiqlash.

Agarqismto'plamyordamidaaniqlanganmassivdoimiy valarrayningkichikbo'lagibo'lsa, bumassivyangi valarraydir. Doimiy valarray tipidagito'plambirqismto'plamyordamidaaniqlanganbo'lsa, dastlabki valarray uchunmoshavolalarsemantikasidir. O'zgarmasbo'lgan valarray laruchunbaholashmexanizmivaqtvaxotirasarflaydi.

Qismto'plamasosidaaniqlanganbo'lsavafarqlansa, barchaelementlarigamurojaatbo'lsa, valarray tipidagimassivlaruchunamallar bajarilishikafoalatlanadi.

Slice sinfiuchun Slice konstruktoriishlatiladi. Bir-biridanbirxilmasofadabo'lganvabelgilanganelementbilanboshlanadigan, birnechaelementlardaniborat valarray birto'plamostisiniyaratadi.

Slice sinfiningfunksiyalariquyidagilar:
size() – valarray to'plamostiningelementlarisoninianiqlaydi.
start() – valarray to'plamostiningboshlang'ichindeksiniqaytaradi
stride() - valarray to'plamostiningelementlariorasidagimasofanitopadi.

6.3-dastur. Slicesinfidanfoydalanish.

```
#include "stdafx.h"

#include <valarray>
#include <iostream>
using namespace std;
int main( ){
    int i;
    size_t sizeVA, sizeVAR;
```



```

size_t startVAR, strideVAR;

valarray<int> va(28), vaResult;

for ( i = 0 ; i < 28 ; i += 1 )
    va [ i ] = (rand()%100)+1;

cout << "valarray elementlari: \n ( ";
    for ( i = 0 ; i < 28 ; i++ )
        cout << va[i] << '\t';
cout << ")." << endl;

sizeVA = va.size();
cout << "valarray elementlar soni: "
<< sizeVA << "." << endl << endl;
// slice konstruktori
slice vaSlice( 0, 6, 4);
vaResult = va[vaSlice];

cout << "valarray to'plam ostisi vaResult = "
<< "va[slice( 0, 6, 4)] =\n ( ";
    for ( i = 0 ; i < vaSlice.size() ; i++ )
        cout << vaResult [ i ] << " ";
cout << ")." << endl;

sizeVAR = vaSlice.size( );
cout << "to'plam ostining elementlar soni: "
<< sizeVAR << "." << endl;

startVAR = vaSlice.start( );
cout << "to'plam ostining birinchi element
indeksi: "
<< startVAR << "." << endl;

strideVAR = vaSlice.stride( );
cout << "to'plam ostining elementlari orasidagi
masofa: "

```

```
<< strideVAR << "." << endl;

    system("pause");
}
```

6.3-dastur. output

```
valarray elementlari:
( 42  68   35   1   70   25   79
59   63   65   6   46   82   28
62
92   96   43   28   37   92   5
3   54   93   83   22   17   ).
valarray elementlar soni: 28.

valarray to'plam ostisi vaResult = va[slice( 0, 6,
4)] =
( 42 70 63 82 96 92 ).
to'plam ostining elementlar soni: 6.
to'plam ostining birinchi element indeksi: 0.
to'plam ostining elementlari orasidagi masofa: 4.
```

Dasturgasinf gaberilganta'rifasosidauningkonstruktori 3
taparametrqabulqilarekan.

Birinchiparamertbelgilanganelementlarindeksi, ikkinchisielementlarsoni,
elementlarindeksiorasidagimasofa.

Gslice **sinf.** Xizmatchisinfbo'lib, valarray
ningko'po'lchamlito'plamostilaribilanishlashgamo'ljallangan. Agar
valarray massivdako'po'lchamlito'plamsifatidaqaralsa, Gslice
sinfundanko'po'lchamlivektornioliadi.

Sinf gslice_arraytipbilantavsiflanuvchiparamertlarnisaqlaydi, agar
alarray**<Type>tipidagiob'ektuchunargumentsifatida class gslice
ob'ektiberilganbo'lsa, Valarray
to'plamostisibilvositaqurilganhisoblanadi. **
to'plamostielementlaribo'lib, Valarray bazaviysinf danolinganva 3
taparamertniqabulqiladi:

- Boshlang'ich indeks;
 - Vektor uzunligi valarray<size_t>sinf;
 - Vektor qadami (elementlari orasidagi masofa) valarray<size_t>sinf;
- Vektorlarning uzunligi bir xil bo'lishi kerak.

Agart o'plam gslice bilan aniqlangan va valarray
ning to'plam osti bo'lsa, gslice yangi valarray hisoblanadi.
O'zgarmas bo'lgan Valarray
uchun baholash mexanizmi vaqt va xotira ni tejaydi.

boshlang'ich va oxirgi to'plamlar gslice bilan aniqlangan bo'lsa,
farqli, barcha indekslariga murojaat bo'lsa, Valarray
to'plam ustidagi barcha amallarga qo'llaniladi.

gslice sinfi uchun gslice konstruktori ishlatiladi.
Belgilangan element bilan boshlanadigan, bir nechta elementlardan iborat
valarray to'plam osti sinfi yaratadi.

gslice sinfining funksiyalari quyidagilar:

size() – valarray

to'plam osti umumiy massiv elementlarini aniqlaydi.

start() – valarray

to'plam osti umumiy boshlang'ich indeksini qaytaradi

stride() - valarray

to'plam osti umumiy elementlar orasidagi masofani topadi.

Gslice konstruktori:

```
gslice(
    size_t _StartIndex,
    const valarray<size_t>& _LenArray,
    const valarray<size_t>& _IncArray);
```

Parametrlari:

_StartIndex – to'plam ostidagi birinchi elementi indeks

_LenArray – har bir to'plam osti uchun massiv elementlar soni

_IncArray -

har bir to'plam osti uchun massiv elementlar orasidagi masofa (qadam);

Standart konstruktor boshlang'ich indeksini uzunlikdagi vektorlar va vektorlari uchun uzunlik va qadam nol qiymatini saqlaydi.

Ikkinchi konstruktor boshlang'ich indeks uchun _startindex saqlaydi,
qator uzunligi uchun _lenarray va qadam uchun _incarray
qiymatlarini saqlaydi.

gslice – bir nechta qismdani tashkil topgan valarray
uchun to'plam osti valarrayni aniqlaydi va hamma belgilangan birelementin
deksidan boshlanadi. gslice i slice ning farqlari birinchi bir nechta qism
valarray lar bilan ishlashdadir.

6.4-dastur. gslicesini foydalanish.

```
#include "stdafx.h"
#include <valarray>
```

```

#include <iostream>
using namespace std;
int main(){

    int i;

    valarray<int> va ( 20 ), vaResult;
    for ( i = 0 ; i < 20 ; i+=1 )
        va [ i ] = i+1;

    cout << "valarray massiv elementlari: " << endl
    << "(";
    for ( i = 0 ; i < 20 ; i++ )
        cout << '\t' << va [ i ];
    cout << " )" << endl;

    valarray<size_t> Len (2), Stride(2);
    Len [0] = 6;
    Len [1] = 4;
    Stride [0] = 2;
    Stride [1] = 4;

    gslice vaGSlice (0, Len, Stride );
    vaResult = va[vaGSlice];

    cout << " vaGSlice asosida vaResult
elementlari:" << endl
    << "va[vaGSlice] = (";

    for ( i = 0 ; i < 10 ; i++ )
        cout << '\t' << vaResult [ i ];
    cout << ")" << endl;

    const valarray <size_t> sizeGS = vaGSlice.size (
);

    cout << "vaResult elementlar soni:"
    << " vaGSlice.size ( ) = ( ";

```

```

        for ( i = 0 ; i < 2 ; i++ )
            cout << sizeGS[ i ] << '\t';
        cout << ")." << endl;

        size_t vaGSstart = vaGSlice.start ( );

        cout << "vaResult uchun birinchi element
indeksi: "
<< vaGSstart << "." << endl;

        const valarray <size_t> strideGS =
vaGSlice.stride ( );

        cout << "vaResult ychun qadamlar:"
<< " vaGSlice.stride ( ) = ( ";
            for ( i = 0 ; i < 2 ; i++ )
                cout << strideGS[ i ] << '\t';
        cout << ")." << endl;
        system("pause");
    }

```

6.4-dastur. Output

```

valarray massiv elementlari:
(      1      2      3      4      5      6
7      8      9     10     11     12     13
14
15     16     17     18     19     20 )
  vaGSlice asosida vaResult elementlari:
va[vaGSlice] = (      1      5      9     13
3      7     11     15     5      9)
vaResult elementlar soni: vaGSlice.size ( ) = ( 6
4      ).
vaResult uchun birinchi element indeksi: 0.
vaResult ychun qadamlar: vaGSlice.stride ( ) = ( 2
4      ).

```

Dasturtahlilibilanmutaqilshug‘ullaning.

Slice_array sinfi. Ichkiyordamchishablonsinfi hisoblanadi. Valarray qismlar bilan aniqlangano 'plamostimassivlar uchun amallarni qo'llab quvvatlaydi. Sinfsintaktiki quyidagicha:

```
template <class Type>
class slice_array : public slice {
public:
    typedef Type value_type;
    void operator=(const valarray<Type>& x) const;
    void operator=(const Type& x) const;
    void operator*=(const valarray<Type>& x) const;
    void operator/=(const valarray<Type>& x) const;
    void operator%=(const valarray<Type>& x) const;
    void operator+=(const valarray<Type>& x) const;
    void operator-=(const valarray<Type>& x) const;
    void operator^=(const valarray<Type>& x) const;
    void operator&=(const valarray<Type>& x) const;
    void operator|=(const valarray<Type>& x) const;
    void operator<<=(const valarray<Type>& x)
const;
    void operator>>=(const valarray<Type>& x)
const;
}
```

Businfob'ekti

valarray<Type> ob'ekti dagi elementlarni ketma-ketligi bilan yoziladigan slicesinfob'ekti bilan birgalikda valarray<Type> sinfiga havolasaqlaydigan ob'ektlarni tasniflaydi.

Shablon sinfi muayyan valarray amallari tomonidan bilvosita yaratilgan vada surdab vosita foydalanish mumkin emas. Ichkiyordamchishablonsinfi quyidagi sintaktik bilan ishlatiladi:

```
slice_array< Type> valarray< Type:: operator[] ( slice);
```

Slice_array<Type> ob'ekti faqat [SL] formatli SL slice valarray ifodasi ni yozish orqali yaratiladi. slice_array sinfining funksiyalari valarray< Type> uchun belgilangan mos funksiyalar bilan bir xil ishlatiladi, faqat tanlangan elementlarni ketma-ketligi ta'sir ko'rsatadi. Slice_array sinfi tomonidan nazorat natijasida konstruktor gauch parametrlar belgilanadi: massivning birinchi element indeks, elementlarni va elementlarni orasidagi masofa.

gslice_array sinfi. Ichkiyordamchishablonsinfi hisoblanadi. Valarray qismlar bilan aniqlangano‘plamostimassivlar uchun amallarni qo‘llab quvvatlaydi. Sinfi sintaktiki quyidagicha:

```
template <class Type>
class gslice_array : public gslice {
public:
    typedef Type value_type;
    void operator=(const valarray<Type>& x) const;
    void operator=(const Type& x) const;
    void operator*=(const valarray<Type>& x) const;
    void operator/=(const valarray<Type>& x) const;
    void operator%=(const valarray<Type>& x) const;
    void operator+=(const valarray<Type>& x) const;
    void operator-=(const valarray<Type>& x) const;
    void operator^=(const valarray<Type>& x) const;
    void operator&=(const valarray<Type>& x) const;
    void operator|=(const valarray<Type>& x) const;
    void operator<<=(const valarray<Type>& x)
const;
    void operator>>=(const valarray<Type>& x)
const;
}
```

Businfob’ekti - valarray<Type> ob’ekti dagi massiv elementlarni ketma ketligi bilan yoziladigan gslicesinfob’ekti bilan birgalikda valarray <Type> sinfiga havolasaqlaydigan ob’ektlarni tasniflaydi.

Shablon sinfi muayyan valarray amallari tomonidan bilvosita yaratilgan vada surdabe vosita foydalanishni mkinemas. Ichkiyordamchishablonsinfi quyidagi sintaksis bilan ishlatiladi:

```
gslice_array< Type> valarray< Type>:: operator[] ( const gslice& )
```

gslice_array <Type> ob’ekti faqat [GL] formatli va GL gslice valarray ifodasi ni yozish orqali yaratiladi. slice_array sinfining funksiyalari valarray<Type> uchun belgilangan mos funksiyalar bilan bir xil ishlatiladi, faqat tanlangan elementlarni ketma-ketligi ta’sir ko’rsatadi. Slice_array sinfi tomonidan nazorat natijasi da konstruktor ga uch parametrlar belgilanadi: massivning birinchi element indeks, elementlari soni va elementlari orasidagi masofa. Dastur fragmenti ga qarang.

```
const size_t lv[] = {2, 3};
const size_t dv[] = {7, 2};
const valarray<size_t> len(lv, 2), str(dv, 2);
```

indirect_array **sinfi.** Ichki yordamchishablonsinfi hisoblanadi. Bazaviy `valarray` sinfi elementlari asosida `ito'plam` o'lib, belgilangan `ito'plam` o'rtasida amallarni ta'minlash orqali `valarray` ob'ektlarining `ito'plam` ob'ektlar bo'lgan ob'ektlarni qo'llab-quvvatlaydigan yordamchisi `inf`.

Businfi ob'ekti - `valarray<Type>` ob'ekti dagi elementlarni ketma-ketligi bilan yoziladigan `valarray<size_t>` sinfi ob'ekti `xa` bilan birgalikda `valarray<Type>` sinfiga havola saqlaydigan ob'ektlarni tasniflaydi.

`Indirect_array<type>` ob'ekti faqat `va[xa]` shaklining ifodasini yozish orqali yaratiladi. Tanlangan elementlarni ketma-ketligi asosida `Indirect_array` sinfi funksiyalari `valarray` uchun belgilangan tegishli funksiyalar bir xil ishlaydi. Ketma-ketlik `X` dan iborat bo'ladi. VA ichidagi `XA[I]` indeksiga akslatirilganda `I` elementlarni o'lchamini bo'ladi.

6.5-dastur. `indirect_array` sinfi dan foydalanish.

```
#include "stdafx.h"
#include <valarray>
#include <iostream>

int main( )
{
    using namespace std;
    int i;

    valarray<int> va ( 10 );
    for ( i = 0 ; i < 10 ; i += 2 )
        va [ i ] = i;
    for ( i = 1 ; i < 10 ; i += 2 )
        va [ i ] = -1;

    cout << "Valarray maasiv elementlari: ( ";
    for ( i = 0 ; i < 10 ; i++ )
        cout << va [ i ] << '\t';
    cout << ")." << endl;

    valarray<size_t> indx(5);
    int j = 0;
    for (int i = 1; i < 10; i+=2)
```



```

    {
        indx [j] = i;
        //cout << indx[j] << '\t';
        j++;
    }
    cout << endl;
    va[indx] = 10;

    cout << "O'zgartirilgan valarray elementlari: (
";
    for (i = 0 ; i < 10 ; i++ )
        cout << va [ i ] << '\t';
    cout << ")." << endl;
    system("pause");
}

```

6.5-dastur.output

```

Valarray maasiv elementlari: ( 0      -1      2
-1      4      -1      6      -1      8      -1
).

O'zgartirilgan valarray elementlari: ( 0      10
2      10      4      10      6      10      8
10).

```

mask_array sinfi. Ichki yordamchi shablonsinfi hisoblanadi. Bazaviy valarray sinfi elementlari asosida `ito'plam` `ostibo'lib`, belgilangan `ito'plam` `ostiuchun mantiqiy amallarni ta'minlash` `orqali valarray ob'ektlarining` `ito'plam` `ostiob'ektlar bo'lgan ob'ektlarni` `niqo'llab-quvvatlaydigan yordamchi sinfi`.

`Businfob'ekti` - `valarray<Type>` ob'ektidagi elementlarning ketma-ketligi bilan yoziladigan `valarray <bool>` sinfi ob'ekti bilan birgalikda `valarray <Type>` sinfiga havola saqlaydigan ob'ektlarni ta'sniflaydi.

Ketma-ketlik `ba.size` elementlarga `ko'pmas`. `J` element faqat `ba[J]` gacha (true) o'rnatilgan `dagina` `fo'ldi`. Ketma-ketlikda element `ni` soni `ba` elementlarga `teng bo'ldi`. Agar `i` - `ba` da `eng kichik haqiqiy` `chin (true)` elementning `indeksi bo'lsa`, `u holda` `tanlangan ketma-ketlikda` `a[i]` ning qiymati `natolgan`.

6.6-dastur. `mask_array` sinfi sinfidan foydalanish.

```

#include "stdafx.h"
#include <valarray>
#include <iostream>
using namespace std;
int main( ){
    int i;
    valarray<int> va ( 10 );
    for ( i = 0 ; i < 10 ; i++ )
        va [ i ] = i*3;

    cout << "valarray elementlari: ( ";
    for ( i = 0 ; i < 10 ; i++ )
        cout << va [ i ] << '\t';
    cout << ")." << endl;

    va [(va > 20 || va < 10) ] = 10;
    cout << "o'zgartirikgan valarray elementlari: ( ";
    for ( i = 0 ; i < 10 ; i++ )
        cout << va [ i ] << '\t';
    cout << ")." << endl;
    system("pause");
}

```

6.6-dastur.Output

```

valarray elementlari:
( 0      3      6      9      12      15      18
21      24      27      ).
o'zgartirikgan valarray elementlari:
( 10     10     10     10     12     15     18
10     10     10     ).

```

In sinfi².
 Businfintervalmatematikaningamallarini bajarishgayo'naltirilganbo'lib,
 C++ningeskistandartiasosidakeltiriladivataqribiyhisoblashlaruchun yaratilgan. Sinfquyidagichaaniqlangan:

²Muallif MO'MINOV Bahodir Boltaevich tomonidan 2013 yillarda tadqiqot ishlari uchun yaratilgan. Barcha xoxlovchilar uchun foydalanish mumkin.

```

class In {
    private:
    public:
        double _a, a_;
        double m,n;
    In() {_a=a_=0;}
    In(double a, double b) {
        if (a>b)
        { a=a+b;
          b=a-b;
          a=a-b;
        }
        _a=a, a_=b;
    }

    //kiritish va chqarish...
    friend istream& operator>>(istream & S, In &
d) throw()
    {
        S>>d._a>>d.a_;
        return S;
    }

    friend ostream& operator << (ostream & S, In
&d) throw()
    { S<<"["<<d._a<<","<<d.a_<<"]\t";
      return S;}

    //  In a = 9;
    inline In & In::operator= (const double & a)
    {
        _a = a_ =a;
        return *this;
    }
    // - ishorasi
    friend In operator -(In &a) throw()
    {
        return In(-a.a_,-a._a);
    }
}

```

Sinfda 2 takonstruktorbo‘lib, birinchisiintervalson 0
gatenglashtiradi,

ikkinchisimosparametrlargaasoslanibintervalsonniyaratidi.

In kutubxonasi funksiyalari:

class In – intervalsinf.

intervalning _a, - quyichegarasi a_ - yuqorichegarasi;

In() {_a=a_=0;} Intervalningqiymatini 0 gatenglashtirish;

In(double a, double b) Intervalgaqiymatberish. Agar _a> a_ bo‘lsa,
qiymatlaro‘rni almashadi.

- 1) [> >] - intervalsonni o‘qish :: 1 2
- 2) [< <] - intervalsonni chiqarish, masalan: [a,b]
- 3) [=] - intervalsonga oddiysonni tenglashtirish, masalan: In a; a=1;
- 4) [-] - intervalsonni ishorasini almashtirish, masalan: In a; b=-a;
- 5) [+] - intervalqo‘shish, masalan: [a]+[b]; [a]+b; a+[b];
- 6) [-] - intervalayirish, masalan: [a]-[b]; [a]-b; a-[b];
- 7) [*] - intervalko‘paytirish, masalan: [a]*[b]; [a]*b; a*[b];
- 8) [/] - intervalbo‘lish, amallari [a]/[b]; [a]/b; a/[b];
- 9) [+=] - operator intervalqo‘shish [a] += [b] [a] += b
- 10) [-=] - operator intervalayirish [a] -= [b] [a] -= b
- 11) [*=] - operator intervalko‘paytirish [a] *= [b] [a] *= b
- 12) [/=] - operator intervalbo‘lish [a] /= [b] [a] /= b
- 13) [] - operator intervalyoki
- 14) [&] - operator intervalva
- 15) [|=] - operator intervalyoki
- 16) [&=] - operator intervalva
- 17) [==] - operator interval tenglik
- 18) [!=] - operator interval tengmas
- 19) [<=] - operator interval kichik yoki teng
- 20) [>=] - operator interval kattayoki teng
- 21) [<] - operator interval kichik
- 22) [>] - operator interval katta
- 23) [!] - operator interval 0 intervalga tegishligini aniqlaydi.
- 24) GetInf funksiyasi - intervalning kichik qiymati olish
- 25) GetSup funksiyasi - intervalning katta qiymati olish
- 26) SetInf funksiyasi - intervalning kichik qiymatini o‘zgartirish
- 27) SetSup funksiyasi - intervalning katta qiymatini o‘zgartirish
- 28) IsEmpty funksiyasi - intervalning xatoligi, masalan: [2,1] –
xato;
- 29) abs funksiyasi - interval moduli

- 30) Midfunksiyasi -intervalningmarkazi
- 31) Radfunksiyasi -intervalningradusi
- 32) Widfunksiyasi -intervalninguzunligi $2 * \text{Rad}$
- 33) MaxInfunksiyasi -intervalningkattasimodulda
- 34) MinInfunksiyasi -intervalningkichigimodulda
- 35) Singfunksiyasi -intervalningishorasi -1,0,1 ko‘rinishda

Indefinekutubxonasi

- 1) tab - "\t"dir.
- 2) NewLine - "\n"dir.
- 3) ZERO - 0.0000000qiymat
- 4) real - doubletipidir

ERRORkutubxonasi

- 1) DIV_BY_ZERO_ERROR
- 2) ERROR_INTERVAL_EMPTY_INTERVAL
- 3) ERROR_INTERVAL_SING_INTERVAL
- 4) ERROR_INTERVAL

INTMATH kutubxonasi

- 1) min funksiyasi– ikkihaqiqiysonningkichigi
- 2) max funksiyasi – ikkihaqiqiysonningkattasi
- 3) abs funksiyasi – haqiqiysonningmoduli

INMATH intervalmatematikakutubxonasidastandartqiymatlar.

- 1) #define E 2.7182818284590452354
- 2) #define LOG2E 1.4426950408889634074
- 3) #define LOG10E 0.43429448190325182765
- 4) #define LN2 0.69314718055994530942
- 5) #define LN10 2.30258509299404568402
- 6) #define PI 3.14159265358979323846
- 7) #define PI_2 1.57079632679489661923
- 8) #define PI_4 0.78539816339744830962
- 9) #define PI1_ 0.31830988618379067154
- 10) #define PI2_ 0.63661977236758134308
- 11) #define SQRTPI2 1.12837916709551257390
- 12) #define SQRT2 1.41421356237309504880
- 13) #define SQRT1_2 0.70710678118654752440
- 14) sin – sin(x);
- 15) cos – cos(x);
- 16) tan – tan(x);
- 17) sinh – sinh(x);

18) $\cosh - \cos(x)$;
 19) $\tanh - \tanh(x)$;
 20) $\operatorname{asin} - \operatorname{asin}(x)$;
 21) $\operatorname{acos} - \operatorname{acos}(x)$;
 22) $\operatorname{atan} - \operatorname{atan}(x)$;
 23) $\exp - \exp(x)$;
 24) $\log - \ln(a)$;
 25) $\log_{10} - \log(x)$;
 26) $\operatorname{pow}(\operatorname{In} \&a, \operatorname{int} n)$ funksiyasi – a intervalsonning n darajasi
 27) $\operatorname{pow}(\operatorname{In} \&a, \operatorname{double} n)$
 28) $\operatorname{pow}(\operatorname{const} \operatorname{In} \&b, \operatorname{In} \&a)$
 29) $\operatorname{pow}(\operatorname{const} \operatorname{double} b, \operatorname{In} \&a)$
 30) $\operatorname{pow}(\operatorname{In} \&b, \operatorname{In} \&a)$
 31) $\operatorname{sqrt}(\operatorname{In} \&a)$ funksiyasi intervalsondankvadratildizolish
 32) $\operatorname{sqrt}(\operatorname{In} \&a, \operatorname{double} n)$ funksiyasi intervalsondan n -chidarajaliildizolish
 33) ceil funksiyasi intervalsonni yaxlitlash
 34) floor funksiyasi interval sonning kattayaxlitlash
 35) round funksiyasi interval sonni kichikkeyaxlitlash
 INDEFMURAKKAB hosila olish uchun kutubxona
 class def_murakkab murakkab_hosilalar sinfi...

1) $\operatorname{div\_Def\_uv}$ funksiyasi - u/v ni hisoblash;
 2) $\operatorname{mult\_Def\_uv}$ funksiyasi - $u \cdot v$ ni hisoblash;
 3) $\operatorname{add\_Def\_uv}$ funksiyasi - $u+v$ ni hisoblash;
 4) $\operatorname{sub\_Def\_uv}$ funksiyasi - $u-v$ ni hisoblash;
 5) $\operatorname{pow\_Def\_u}$ funksiyasi - $f(x)$ ning d darajasining hosilasi;
 6) $\operatorname{f\_sqrt}$ funksiyasi - ildiz osti $f(x)$ ning hosilasi
 class def_fx x ning n darajasining hosilasi sinfi...
 1) $\operatorname{fx}(\operatorname{In} x)$ funksiyasi - x ning 1 darajasini qaytaradi;
 2) $\operatorname{fx}(\operatorname{In} x, \operatorname{int} n)$ funksiyasi - x ning n darajasini qaytaradi;
 3) $\operatorname{dfx}(\operatorname{In} x, \operatorname{int} n)$ funksiyasi - x ning n darajasini hosilasi;
 4) $\operatorname{dfx}(\operatorname{In} x, \operatorname{int} n, \operatorname{int} m)$ funksiyasi - x ning n darajasini m tartibli hosilasi;
 class def_func – matematik funksiyalardan hosila;
 1) $\operatorname{d\_exp}(\operatorname{In} x, \operatorname{int} m)$ funksiyasi – $\exp(x)$ ning m tartibli hosilasi;
 2) $\operatorname{d\_exp}(\operatorname{In} x, \operatorname{In} k, \operatorname{int} m)$ funksiyasi – $\exp(kx)$ ning m tartibli hosilasi;
 3) $\operatorname{const\_a}(\operatorname{In} b)$ funksiyasi – o'zgarmas interval sonning hosilasi;

- 4) `f_ax(In a, In x,int m)`funksiyasi- a^x ning m tartibli hosilasi;
- 5) `f_ax(In a, In x,int m, int k)` funksiyasi- a^{kx} ning m tartibli hosilasi;
- 6) `f_sin(In x,int m)` funksiyasi-sin ning m tartibli hosilasi;
- 7) `f_cos(In x,int m)`funksiyasi- cos ning m tartibli hosilasi;
- 8) `f_sqrt(float x, int m)`funksiyasi-sqrt ning m tartibli hosilasi;
- 9) `f_ln(In x, int m)` funksiyasi- $\ln(x)$ ning m tartibli hosilasi;
- 10) `f_log(float a, In x, int m)` funksiyasi- $\log(a,x)$ ning m tartibli hosilasi;
- 11) `f_tan(In x)` funksiyasi-tg(x) ning 1 tartibli hosilasi;
- 12) `f_ctan(In x)` funksiyasi-ctg(x) ning 1 tartibli hosilasi;
- 13) `f_acos(In x)` funksiyasi-arccos(x) ning 1 tartibli hosilasi;
- 14) `f_asin(In x)` funksiyasi-arcsin(x) ning 1 tartibli hosilasi;
- 15) `f_atan(In x)` funksiyasi-arctan(x) ning 1 tartibli hosilasi;
- 16) `f_actan(In x)` funksiyasi-arcctan(x) ning 1 tartibli hosilasi;
- 17) `f_cosh(In x)` funksiyasi-giperbolik cos(x) ning 1 tartibli hosilasi;
- 18) `f_sinh(In x)` funksiyasi-giperbolik sin(x) ning 1 tartibli hosilasi;
- 19) `f_tanh(In x)` funksiyasi-giperbolik tan(x) ning 1 tartibli hosilasi;
- 20) `f_ctanh(In x)` funksiyasi-giperbolik ctan(x) ning 1 tartibli hosilasi;

`class def_arg_func` argumentida $f(x)$ bo'lgan funksiyalar hosilasi sinf;

- 1) `d_exp(func f, func df, In x)` funksiyasi- $\exp(f(x))$ ning hosilasi;
- 2) `f_ax(In x, double a, func f, func df)` funksiyasi- $a^{f(x)}$ ning hosilasi;
- 3) `f_ax(In x, In a, func f, func df)` funksiyasi- $a^{f(x)}$ ning hosilasi;
- 4) `f_sin(In x, func f, func df)` funksiyasi-sin($f(x)$) ning hosilasi;
- 5) `f_cos(In x,func f, func df)` funksiyasi-cos($f(x)$) ning hosilasi;
- 6) `f_sqrt(In x, func f, func df)` funksiyasi-sqrt($f(x)$) ning hosilasi;
- 7) `f_ln(In x, func f, func df)` funksiyasi- $\ln(f(x))$ ning hosilasi;
- 8) `f_log(float a, In x, func f, func df)` funksiyasi- $\log(a,(f(x)))$ ning hosilasi;
- 9) `f_log(In a, In x, func f, func df)` funksiyasi- $\log(a,(f(x)))$ ning hosilasi;
- 10) `f_tan(In x, func f, func df)` funksiyasi-tan($f(x)$) ning hosilasi;
- 11) `f_ctan(In x, func f, func df)` funksiyasi-ctan($f(x)$) ning hosilasi;

- 12) $f\_acos(\ln x, func f, func df)$ funksiyasi- $acos(f(x))$ ning hosilasi;
- 13) $f\_asin(\ln x, func f, func df)$ funksiyasi- $asin(f(x))$ ning hosilasi;
- 14) $f\_atan(\ln x, func f, func df)$ funksiyasi- $atan(f(x))$ ning hosilasi;
- 15) $f\_actan(\ln x, func f, func df)$ funksiyasi- $actan(f(x))$ ning hosilasi;
- 16) $f\_cosh(\ln x, func f, func df)$ funksiyasi- $cosh(f(x))$ ning hosilasi;
- 17) $f\_sinh(\ln x, func f, func df)$ funksiyasi- $sinh(f(x))$ ning hosilasi;
- 18) $f\_tanh(\ln x, func f, func df)$ funksiyasi- $tanh(f(x))$ ning hosilasi;
- 19) $f\_ctanh(\ln x, func f, func df)$ funksiyasi- $ctanh(f(x))$ ning hosilasi;
- 20) $pow\_arg\_func(func f, func df, func dx, \ln x, int d)$ funksiyasi- $(f(x)^d)$ ning hosilasi;

- 21) $f\_msqrt(func f, func df, \ln x, float m)$ funksiyasi- $m\_sqrt(f(x))$ ($1/m$) ning hosilasi;

INGAUSSK_H – Integralni Gauss usulida hisoblash uchun koyfisentlar uchun kutubxona

- 1) P – ko‘p hadni hisoblash;
- 2) dP – ko‘p hadni hosilasini hisoblash
- 3) A – Koyfisentlarni hisoblash;
- 4) x0 – x uchun boshlang‘ish yaqinlashish;
- 5) xeps – x ni eps aniqlikda olish uchun;
- 6) convert_x – olingan x larni a,b oraliqqa convertlash;
- 7) convert_A - olingan A larni a,b oraliqqa convertlash;

INTEGRALIN_H interval integrallash uchun kutubxonaintegral_MTT1 – markaziy to‘g‘ri to‘rtburchak usuli;

- 1) integral_Trapfunksiyasi– trapetsiya usuli;
- 2) integral_Simsonfunksiyasi – Simson usuli;
- 3) integral_gauss1funksiyasi – Gauss usuli;
- 4) integral_gauss2funksiyasi - Gauss usuli ikki karrali

INGAUSS_H CHATSni gauss usuli bilan yechish uchun kutubxona
N = 3 uchun

- 1) PrintArray – massivni chiqarish;
- 2) PrintVector – vektorni chiqarish;
- 3) ScanArray – massivni kiritish;

- 4) ScanVectorB – vektorni kiritish;
- 5) SwapColumns – ustunlarni almashtirish;
- 6) SwapLines – qatorlarni almashtirish;
- 7) Destroy – quyi uchburchakni)ga aylantirish;
- 8) Solution – yechimni topish
- 9) F – yechimning aniqligini aniqlash;
- 10) GAUSS – birlashgan Gauss usulini qo‘llash;

6.7-dastur. Chiziqli algebraik tenglamalar sistemasini yechish dasturi.

```
#include <iostream.h>
#include <conio.h>
#include <e:\mylibs\interval.h>

int main(){
int n;
cout<<"n ni kirit:>>";
cin>>n;
In a[n][n],b[n];
//_____A ni kirit:_____
for(int i=0; i<n; i++)
    for(int j=0; j<n; j++)
        a[i][j]=KiritIn();
//_____b ni kirit:_____
for(int j=0; j<n; j++)
    b[j]=KiritIn();
//_____to'g'ri yo'l_____
In r[n][n];
for(int j=0; j<n-1; j++)
    for(int i=j+1; i<n; i++){
        r[i][j] =DivIn(a[i][j],a[j][j]);
        for(int k=j; k<n; k++)

a[i][k]=SubIn(a[i][k],MultIn(r[i][j],a[j][k]));
        b[i]=SubIn(b[i],MultIn(r[i][j],b[j])); }
//_____a[i][i]:_____
for(int j=0; j<n; j++){
    b[j] =DivIn(b[j],a[j][j]);
    for(int i=0; i<n; i++)
        r[i][j] =DivIn(a[i][j],a[i][i]); }
```

```

//_____a[][][]_____convert_
for(int j=0; j<n; j++)
    for(int i=0; i<n; i++)
        a[i][j] = r[i][j];
//_____ out _____
for(int i=0; i<n; i++) {
    for(int j=0; j<n; j++)
        { cout<<"["; chopIn(a[i][j]); cout<<"]\t"; }
        cout<<"= ["; chopIn(b[i]); cout<<"]\t";
        cout<<endl;}

In x[n],s;
for(int i=n-1; i>-1; i-- )
{   s._a=0; s.a_=0;
    for( int j=n-1; j>i; j--)
        s=AddIn(s,MultIn(a[i][j],x[j]));
        x[i]=DivIn(SubIn(b[i],s),a[i][i]);
    }

    for(int i=0; i<n; i++){
        cout<<"["; chopIn(x[i]); cout<<"]"<<endl;}

    getch();
return 0;
}

```

6.7-dastur.

Ikkikarraliintegralnigaussusulibilantaqribiyhisoblashdasturi.

```

#include <iostream.h>
#include <conio.h>
#include <math.h>
double P(double x, float n){
    if (n==0) return 1;
    if (n==1) return x;
    return (((2*n-1)/n)*x*P(x,n-1))-(((n-1)/n)*P(x,n-2));
}

```

```

double dP(double x, float n){
    return ((n/(1-x*x))*P(x,n-1))-(x*P(x,n));
}

double A(double x, float n){
    return 2./((dP(x,n)*dP(x,n))*(1-x*x));
}

double x0(int i, double n){
    return -cos(M_PI*((4*(i+1)-1)/(4*n+2))); }

double xeps(double x, double n, double eps){
    double temp=x-P(x,n)/dP(x,n);
    while (fabs(temp-x)>eps) {
        x=temp;
        temp=x-P(x,n)/dP(x,n);
    }
    return x;
}

double convert_x(double xx, float a, float b){
    return (0.5*(b+a))+(0.5*(b-a)*xx);
}

double convert_A(double AA, float a, float b){
    return ((b-a)/2)*AA;
}

int main(){
    float n,eps;
    freopen("InputData.in","r",stdin);
    cin>>n; cin>>eps;
    float a,b; cin>>a>>b;
    float x[(int)n];
    freopen("OutputData[a,b].out","w",stdout);
    cout<<"*** --- ---- n="<<n<<" ---- --- ***"<<endl;
    cout<<"*** -- a="<<a<<" b="<<b<<" -- ***"<<endl;
    cout<<"*** -- -- --- --- --- --- --- ***"<<endl;
}

```

```

    for(int i=0; i<n; i++) {
        x[i]=x0(i,n); cout<<i+1;
        x[i]=xeps(x[i],n,eps);
        printf(" :%.15f",convert_x(x[i],a,b));;
        printf(" :%.15f\n",convert_A(A(x[i],n),a,b));
    }
    //getch();
    return 0;
}

```

6.7-dastur. Output

```

----- 6 -----
eps=5
1.40656 3.17244
2.2895:2.2895
2.22385
6
0.882941
***____HISOBLANDI____***
*** -- n=5 uchun -- ***
1 :-0.90963196754455566000 :0.19411758457043865000
2 :-0.54064083099365234000 :0.47043172906368108000
3 :-0.000000000000000006123 :0.56888888888888889000
4 :0.54064083099365234000 :0.47043172906368108000
5 :0.90963196754455566000 :0.19411758457043865000

I=2.21087712506044950000

```

Ushbu insonni foydalanadigan dailmiy amaliy matematik va matematika-fizika tenglamalarini taqribiy kompyuterda hisoblashishlarini amalga oshirish uchun ishlatiladi.

Bukutubxona aynan A254-

Sloyi hasid o'irasida ishlab chiqilgan.

Loyihani asosiy vazifasi fizik jarayonlarni matematik model lashtirish va ularni sonli echimlarni olish uchun dasturiy ta'minotlarni ishlab chiqishga bag'ishlangan. Kutubxonada 10 dan ortiq mo'rosxo'rsinf, 200 dan ortiq funksiyalar, matematik fuksiyalarning modifikatsiyalangan funksiyalaribor.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Nima uchun sonlarning inflatsiyasi yaratiladi?
2. Qanday sonlarning inflatsiyasi bilasiz?

3. Sonlarning inflatsiyasida amallarni bajarish uchun funksiyalarni qaysi usul yordami da aniqlanadi?

4. Kompleks va arifmetik amallaro 'zaror hamkorlikda ishlaydimi?

5. Kompleks tipni arifmetik tipga tenglashtirish mumkinmi?

6. Haqiqiy (real) yok mavhum qismini oddiy arifmetik tipga qiymat qilib berish mumkinmi?

7. Kompleks sonning real qismi qaysi funksiyalar bilan aniqlanadi?

8. Kompleks sonning mavhum qismi qaysi funksiyalar bilan aniqlanadi?

9. `double re = complex_obj.real();` va `double re = real(complex_obj);` sintaktiklarning farqi bormi?

10. Kompleks sonning mavhum qismini qanday kiritiladi?

11. Kompleks sonni to'liq kiritish uchun qavs ichida real va mavhum qismlar qanday belgi bilan ajratiladi?

12. Kompleks sonni qanday operandlarni qo'llab quvvatlamaydi?

13. Valarray konteynerini nima uchun kerak?

14. Massiv element indekslarini to'liq yechib olish operatorlari va elementlari matematik amallarni hisoblash sinfi qanday?

15. **apply()** —
massiv elementlarini to'liq yechib olish vaqtida to'liq zgarishlarni bajaradigan maniqa yaratadi?

16. **cshift()** —
berilgan sonni qiymat to'liq yechib indeks asosida massivni qanday shaklida surib yotqizib beradi?

17. Valarray
tipi bilan bajariladigan amallarni qayta aniqlash mumkinmi?

18. Kaysi `inf` valarray
sinfining meros xo'ridir va biroz ichovlito'plam ostidagi yaratishda ishlatiladi?

19. `slice` sinfi qanday tipdagi ob'ektni tavsiyalar va chip parametrlarni saqlaydi.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	Komplekssonlarsinfiga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari sidi bajarang. 👉 Bundagi dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>// Created by // MBBahodir #include "stdafx.h" #include <complex> using namespace std; int main(){ complex< double > a, b, c; cout << "Complex sinfi oid turli t=farmatli 3 ta son kiritibg:" << endl; cin >> a >> b >> c; cout << "Complex sinfiga oid sonlar:" << endl; cout << a << b << c << endl; system("pause"); }</pre>	1. Dasturda xatosini toping. _____ _____ —
	2. Dasturda komplekssonlardan iborat massivni yarating. _____ — _____ — _____
	3. Komplekssonlarni yig'indisini hisoblovchi funksiyani yarating. _____ — _____ — _____ — _____ —

IKKINCHI ASSIMENT TOPSHIRIG'I	
	Valarray sonsinfivauning funksiyalariga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragme ntlari asosida bajarilgani. 👉 Bunday dasturda giba'zi o'zgartirishlarni topish orq ali topshiriqlar bosqichma bosqich amalga oshiriladi.
dastur	topshiriqlar
<pre>#include "stdafx.h" #include<iostream> #include<valarray> using namespace std; int main(){ int myints[] = { 5, 25, 55, 85, 115 }; valarray<int> varr (myints, sizeof(myints)/sizeof(m yints[0])); valarray<int> varr_one ; varr_one = varr.apply([](int x){return x=x+5;}); cout << "Massivning yangi qiymatlar: "; for (int &x: varr_one) cout << '\t' << x; cout << endl;</pre>	1. Dastur xatolarini toping.
	2. Dasturda varr_one = varr.apply([](int x){return x=x+5;}); ni mavzifani amalga oshi riradi.

<pre> cout << "Oldingi massiv sum() => "; cout << varr.sum() << endl; cout << "Yangi massiv sum() => "; cout << varr_one.sum() << endl; cout << "min() => "; cout << varr.min() << endl; cout << "max() => "; cout << varr.max() << endl; varr_one = varr.shift(2); cout << "Massivning shift(2) qiymatllar: "; for (int &x: varr_one) cout << '\t' << x; cout << endl; varr_one = varr.cshift(-2); cout << "Massivning cshift(-2) qiymatllar: "; for (int &x: varr_one) cout << '\t' << x; cout << endl; system("pause"); return 0; </pre>	3. Dasturdaberilganvalarrayningkichikvaengkattaelementlarifarqin itopuvchidasturfragmentiniyozing. 4. Ikkitamassivnio‘zarosolishtirinig, tengbo‘lsa 0, kattabo‘lsa 1, kichikbo‘lsa -1 qilibvalarray massivyarating. 5. Massivlarniteskarichiqarishdasturfragmentlariniyozing.

}	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

UChINChIASSIMENTTOPShIRIG‘I	
	Slice sinfigaoidberilganquyidagidasturbo‘yichaberil gantopshiriqlarkeraklifragmentlariasosidabajar ing. ☞ Bundadasturdagiba‘zio‘zgartirishlarnitopish orqalitopshiriqlarbosqichma — bosqichamalغانoshriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include <valarray> #include <iostream> using namespace std; int main(){ int i; size_t sizeVA, sizeVAR; size_t startVAR, strideVAR; valarray<int> va(28), vaResult; for (i = 0 ; i < 28 ; i += 1) va [i] =</pre>	1. Dasturdanechtayangitipdagiob’ ektlariyaratilgan. _____ _____ _____ _____
	2. Dasturdagibarchavaelementlari niekrangachiqaring. _____ _____ _____ _____ _____ _____ _____

<pre> (rand()%100)+1; sizeVA = va.size(); cout << "valarray elementlar soni: " << sizeVA << "." << endl << endl; // slice konstruktori slice vaSlice(0, 6, 4); vaResult = va[vaSlice]; cout << "valarray to'plam ostisi vaResult = " << "va[slice(0, 6, 4)] =\n ("; for (i = 0 ; i < vaSlice.size() ; i++) cout << vaResult [i] << " "; cout << ")." << endl; sizeVAR = vaSlice.size(); cout << "to'plam ostining elementlar soni: " << sizeVAR << "." << endl; startVAR = vaSlice.start(); cout << "to'plam ostining birinchi </pre>	3. Massivdan 2 indeksdan 4 taelementNqadambilanyangimassivc hiqarishmumkin. N ningqiymatlarinianiqlang. 4. vaSlice.size();ningtipiqanday? 5. Massivdanberilganqiymatniizl ash dasturfragmentiniyozing.

<pre> element indeksi: " << startVAR << "." << endl; strideVAR = vaSlice.stride(); cout << "to'plam ostining elementlari orasidagi masofa: " << strideVAR << "." << endl; system("pause"); } </pre>	
6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

TO'RTINCHI ASSISMENT TOPSHIRIG'I	
	gslice sinfigaoidberilganquyidagidasturbo'yichaberil gantopshiriqlarkeraklifragmentlariasosidabajar ing. 👉 Bundadasturdagiba'zio'zgartirishlarnitopish orqalitopshiriqlarbosqichma — bosqichamalغانoshiriladi.
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include <valarray> #include <iostream> using namespace std; int main(){ int i; </pre>	1. Dasturdanechtayangitipdagiob' ektlariyaratilgan. _____ _____ _____ _____ _____

<pre> valarray<int> va (20), vaResult; for (i = 0 ; i < 20 ; i+=1) va [i] = i+1; valarray<size_t> Len (2), Stride(2); Len [0] = 6; Len [1] = 4; Stride [0] = 2; Stride [1] = 4; gslice vaGSlice (0, Len, Stride); vaResult = va[vaGSlice]; cout << " vaGSlice asosida vaResult elementlari:" << endl << "va[vaGSlice] = ("; for (i = 0 ; i < 10 ; i++) cout << '\t' << vaResult [i]; cout << ")" << endl; const valarray <size_t> sizeGS = vaGSlice.size (); cout << "vaResult elementlar soni:" << " vaGSlice.size () = ("; for (i = 0 ; i < </pre>	2. Dasturdagi barcha va elementlari niekrangachiqaring. 3. Massivdan 2 indeksdan(4,6)ta element(N,M)qada mbilanyangimassivchiqarishmumkin . (N,M) ningqiymatlarinianiqlang. 4. vaSlice.size();ningtipiqandayvanatijasichi ? 5. Massivdanberilganqiymatniizl ash dasturfragmentiniyozing.

```

2 ; i++ )
        cout <<
sizeGS[ i ] << '\t';
        cout << ")." <<
endl;

        size_t vaGSstart =
vaGSlice.start ( );

        cout << "vaResult
uchun birinchi element
indeksi: "
<< vaGSstart << "." <<
endl;

        const valarray
<size_t> strideGS =
vaGSlice.stride ( );

        cout << "vaResult
ychun qadamlar:"
<< " vaGSlice.stride (
) = ( ";
        for ( i = 0 ; i <
2 ; i++ )
                cout <<
strideGS[ i ] << '\t';
        cout << ")." << endl;
        system("pause");
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____

7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

BEShINChIASSISMENTTOPShIRIG'I



indirect_arraysinfigaoidberilganquyidagidastu
rbo'yichaberilgantopshiriqlarkeraklifragmentl
ariasosidabajaring.

👉 Bundadasturdagiba'zio'zgartirishlarnitopish

	orqalipotshiriqlarbosqichma bosqichamalغانoshiriladi.	—
dastur	topshiriqlar	
<pre>// Created by MBBahodir #include "stdafx.h" #include <valarray> #include <iostream> int main() { using namespace std; int i; valarray<int> va (10); for (i = 0 ; i < 10 ; i += 2) va [i] = i; for (i = 1 ; i < 10 ; i += 2) va [i] = -1; valarray<size_t> indx(5); int j = 0; for (int i = 1; i < 10; i+=2) { indx [j] = i; //cout << indx[j] << '\t'; j++; } cout << endl; va[indx] = 10;</pre>	1. Dasturdanechtayangitipdagiob' ektlariyaratilgan.	— — — — —
	2. Dasturdagibarchavaelementlari niekrangachiqaring.	— — — — — — —
	3. Massivdan 2, 4 indeksli elementlarniqiymatini 100 o'zgartirishdasturinituzing.	— — — — — — —

<pre> cout << "O'zgartirilgan valarray elementlari: ("; for (i = 0 ; i < 10 ; i++) cout << va [i] << '\t'; cout << ")." << endl; system("pause"); } </pre>	4. va[indx] = 10;ningtipiqanday? _____ _____ _____
5. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____ 6. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

OLTINCHIASSISMENTTOPSHIRIG'I	
	mask_array sinfigaoidberilganquyidagidasturbo'yichaberil gantopshiriqlarkeraklifragmentlariasosidabajar ing. 👉 Bundadasturdagiba'zio'zgartirishlarnitopish orqalitopshiriqlarbosqichma bosqichamalganoshriladi.
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include <valarray> #include <iostream> using namespace std; int main(){ int i; valarray<int> va (10); for (i = 0 ; i < 10 ; i++) va [i] = i*3; va [(va > 20 va < </pre>	1. Dasturdanechtayangitipdagiob'ektlariyaratilgan. _____ _____ _____ _____
	2. Dasturdagibarchavaelementlari niekrangachiqaring. _____ _____ _____

<pre> 10)] = 10; cout << "o'zgartirigan valarray elementlari: ("; for (i = 0 ; i < 10 ; i++) cout << va [i] << '\t'; cout << ")." << endl; system("pause"); } </pre>	3. Massivdan 10 kattaelementlarianiqlovchidastu rinituzing. 4. va [(va > 20 va < 10)] = 10;ningtipiqandayvanimavanim avazifanibajaradi?
5. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____ 6. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

2.3. Sintaktik tahlil

 Sintaktik tahlil qiluvchi konstruktorlar, matnli-erkingrammatika, kompilyatorlaryaratuvchilarning muammolari, ifodalarning sintaktik tahlili, leksik va sintaktik tahlil bosqichlari, sintaktik tahlillovchisi nflari, past sathlargayo'naltirilgan analizatorlar, LL grammatika, LR – grammatika, rekursiv kamayish usuli, rekursiv kamayish usulidan foydalanish sharti, parsersinfi. FIRST to'plamning qurish algoritmi, chap rekursiv grammatikalar,

qiymatqaytarishasosidagirekursivkamayishusuli,
parserqurishvazifalarivausullarikeltirilganbo‘lib,
nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

bilimlarnimustahkamlashuchun 35
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 1
taassimenttopshirig‘ivaharassimentda 6 tatopshiriqberilgan.
Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍**Kalitso‘zlar.**Oqim, satr, iterator, sinf, shablon,
matematikamallar, fayl, EOF, rekursiya, iofda, qoida, grammatika.

✓**Bilishshartbo‘lgantushunchalar.**Tiptushunchasi,
sinfvasinfob‘ekti, oqim, kirishelementgamurojaat,
funksiyavako‘rsatkich, ma‘lumotlarnikirishvachiqishi,
dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo‘llabquvvatlovchimu
hitdaishlashnibilishlozim.

🔧**Bilibolasiz.** Sintaktiktahlillovchivazifasi, turligrammatikalar,
dasturlashtilini analizatoriningishlashtamoyili,
pastvayuqorisathlargayo‘naltirilganalgoritmalar, rekursivkamayishusuli,
analizatorlarniqurishdata‘rifvashartlarniqurishni, parsersinfi, FIRST
to‘planningqurishalgoritmi, chaprekursivgrammatikalar,
chaprekursivliknibartarafetishalgoritmi,
qiymatqaytarishasosidagirekursivkamayishusuli,
sintaktikanalizatorniyaratish, Forward va Input
iteratorlardanfoydalanish, Strin – Stdin oqimlardanfoydalanish,
sintaktiktahlillovchilarningbazaviyfunksiyalarinio‘rganishingiz mumkin.

REJA

- 1.Ifodalarningsintaktikhlili.
- 2.Parsersinfi.
- 3.Sintaktikanalizatorniyaratish.

KIRISH

Bugungikungakelibdasturlashtillariningrivojlanishigaularnisintakti
ktahlilqiluvchikonstruktorlarningko‘payib, ommalashganidir.
Harbirdasturlashtilito‘g‘ridasturtuzilishinibelgilovchiqoidalarmajmuasiy
ordamidatasvirlanadi.

Dasturlashtiliningintaktikkonstruksiyaalarinitasvirlashuchunengqulayfor
malusulmatnli-erkingrammatikadir (masalan, Backus-Naur
ningnormalshaklikengqo‘llaniladi).

Grammatikalartildanfoydalanadigandasturchilarsavashutiluchunkom
pilyatorlaryaratuvchilarningmuammolarinibirxildahalqilishgayordamber
adi:

-
Grammatikadasturlashtilininganiqvaosontushunadigansintaktikxususiyatlarinitalaydi.

- Ba'zigrammatikasinflaruchun, avtomatikravishdamanbadasturisintaktikto'g'riyokiyo'qliginianiqlaydiganamaraliparser (analizator, tahlillovchi) qurishmumkin.

- Aniqqurilgangrammatikadasturlashtiliganbadasturinito'g'riob'ektkodigatarjimaqilishvaxatolarnianiqlashuchunfoydali'lgantuzilmaniberishimumkin.

- Grammatikaasosidaishlabchiqilgankompilyatorlarjudaosonkengaytirilishimumkin (butilrivojlanishinatijasidapaydobo'lganyangikonstruksiyalarniqo'shishuchunaniqsafoydali).

Yanabirborta'kidlaymizki, matnli-erkingrammatikaldasturlashtiliningfaqatmatnlitarkibiyqismini, ya'nimuayyantiqurilishiqandayyozilganliginianiqlaydi. Dasturningsintaktikto'g'riliginianiqlashningyanabirmuhimqismi - dasturdatiplardanfoydalanishningto'g'riliginimatnli-erkingrammatikayordamidaaniqlabbo'lmaydi. Shuninguchun, agardasturgrammatikasito'liqdebishlansa, ubutunlaysintaktikjihatdanto'g'rideganiemas.

Ifodalarningsintaktiktahlili. Sintaktiktahlildegandajoriyqilingangrammatikagaleksemalarningma'lumketma-ketligihosilqilgantilgategishliekanliginibelgilovchijarayondir. Amalda, harqandaygrammatikadanparserqurishgizmmumkin, ammoamaldaishlatiladigangrammatikalar maxsusshaklgaega. Masalan, nuzunlikdagikiruvchisatruchunmurakkabligi $O(n^3)$ danoshmasligikerakbo'lganharqandaykontekst-erkingrammatikauchunanalizator (parser) qurishmumkin.

Lekin ko'phollardatezishlaydigananalizatorqurishimkoniniberadiganberilgandasturlashtiliasosidagrammatikaqurishmumkin.

Amaliyotdatilanalizatorlariodatdachiziqlimurakkablikkaega. Masalan, dasturdachapdano'nggaqarabbittaterminalbelgiga (leksiksinf) oldinga qarabko'rishorqalierishiladi.

Sintaktiktahlillovchikirishparametrlari - buleksemavajadvllarketma-ketligi, masalan, ichkitasvirlanganjadval, bujadvalsintaktiktahlillovchiuchunchiqishparametrlaridir.

Sintaktiktahlillovchichiqishparametrlari —
 jadvallarvatahlildaraxti hisoblanadi, masalan,
 kompilyatorningkeyingiko‘rishi uchunchiquvchibo‘lib hisoblangan identifi-
 katorlar jadvali va tiplari jadvali (masalan,
 tiplari nazoratni amalga oshiruvchiko‘rinish bo‘lishi mumkin).

Leksik va sintaktik tahlil bosqichlarining alohida qarashlarga ajratilishi
 shart emas. Odatda bu fazalar bir xil ko‘rinishda bir-
 bir bilan o‘zaro sirlashadi.

Bunday ko‘rinishning asosiy fazasi ajralish fazasi bo‘lib hisoblanadi va sintaktik
 analizator leksika analizatoriga harsafar boshqa terminal belgikera ko‘lganda
 murojaat qiladi.

Sintaktik tahlil ovlari.
 Juda ko‘pchilik tahlil ovlari quyidagi ikki sinflarning biriga tegishlidir

1. top-down algoritmlari (past sathlarga o‘ltirilgan);
2. bottom-up algoritmlari (yuqori sathlarga o‘ltirilgan);

Buterminlarning kelib chiqishi sintaktik daraxt tugunlarining yaratilishi
 ul bilan bog‘liq: ildizdan (grammatika aksiomalari) yuqori tugunlarga
 (terminal simvollar), yoki yuqori tugunlardan ildizgacha.

Past sathlarga o‘ltirilgan analizatorlar chiqishni qurish uchun gram-
 matika aksioma dan boshlab v terminal simvollar zanjirini bildiradi.

A analizatorlar quyidagi xususiyatlarga ega bo‘lgan LL

- natijalar bermaydigan analizatorlarga ega bo‘lishi mumkin;
- birinchi L harfi - kiruvchi zanjirni chapdan o‘ngga qarab o‘qilishini bildiradi (left-to-right scan);
- ikkinchi L harfi - zanjirning chap chiqishini bildiradi (*leftmost* derivation).

Past sathlarga o‘ltirilgan analizatorlarni yaratish juda qulay bo‘lib his-
 oblanadi, ularni qo‘ldan yaratish mumkin, masalan,
 rekursiv kamayish ul bilan. LL

grammatika bilan ishlash va o‘zgartirish murakkab emas. LL
 grammatika bo‘lmagan grammatikalarda analizatorlar uchun rekursiv kamay-
 ish usullaridan foydalanish mumkin.

Boshqa tomondan past sathlarga o‘ltirilgan analizatorlarga qaraganda
 yuqori sathlarga o‘ltirilgan analizatorlar ko‘proq ishlatiladi.

Chunki ular ko‘proq grammatikalarni tahlil qila oladi.

Shuning uchun bu analizatorlar uchun analizatorlarni yaratuvchi maxsus dastu-

rlarmavjud. yuqorisathlargayo‘naltirilgananalizatorlarbilan LR – grammatikabog‘liq. Bundagi L harfioldingidek, kiruvchizanjirnichapdano‘nga qarabo‘qilishinibildiradi (left-to-right scan), R harfiesa – zanjirningo‘ngchiqishiqurilayotganinibildiradi(*rightmost derivation*). Bugungikunda LR – grammatikagaasoslanganalizatorlarbilanjudako‘pdasturlashtillarifoydalanmoqda.

Rekursivkamayishusuli. Engoddiyvajudako‘pfoydalanilganpastsathlargayo‘naltirilgananalizatorlarniqurishusulirekursivkamayishusulidir (recursive descent method).

Rekursivkamayishusuliasosiytamoyillarinio‘rganishhuchunarifmetiki fodalarnibajarilishmasalasiniqaraymiz. Ulargabinaramallariqo‘shish (+), ayrish (-), ko‘pytirish (*), butunbo‘lish (/) vaqavslaramallarikirsin. Odatdagidek, ko‘paytirishvabo‘lishamallariningustivorligitengvaularningustivorligiqo‘shishvaayirishamallariningustivorligidankattabo‘lib, buamallariningustivorligihamtengdir. Qo‘shishturidagiamallarga (+) va (-) - amallarini, ko‘paytirishturidagiamallargaesa (*) va (/) - amallariqilibbelgilaymiz.

Qavslaramalistandaritartibinio‘zgartirishhuchunishlatiladi.

Vazifamizifodaqiymatinihisoblovchidasturyozishdaniborat.

Qaralayotganifodaniquyidagiko‘rinishdabo‘lsin

$$T_1 + T_2 + \dots + T_n$$

Bunda $T_i = F_{1i} * F_{2i} * \dots * F_{mi}$ ko‘rinishidagiifoda. Shuningdek, F_{ji} – busonyoqi qavsichidagiifoda.

Berilganifodanihisoblashjarayoninio‘ylabko‘ramiz, bundabirinchigalda F_{11} hisoblanadiva F_{11} dankeyinqaysiamalturganinianiqlanadi. Agar bu ko‘paytirishamalibo‘lsa, chap operandinibilib, o‘ng operandinianiqlaymizvaamal nibajaramiz. Shundayqilib, ko‘paytirishamali uchunchap operandinianiqlaymiz. $F_1 * F_2 * \dots * F_n$ amallarketmaketliginihisoblabbo‘lgandankeyin, keyingiamalsifatidabo‘lishamalinikelsa, bunihamyuqoridagijarayonbo‘yichahisoblaymiz.

Ifodalarni hisoblashdabarchahisoblashlarniquyidagisinflargaajratish mumkin:

1. Oddiyifoda, sonlarvaqavslarbilanberilganchiziqliifodalar, masalan, 28, (128-25+16);

2. Ifoda, ko'paytirish tipidagi amallar ibo'lgan ifoda, ko'paytirish vabo'lish amallarini kiradi, masalan, $2*15, 28*(15/3)+3$;
3. Ifoda, qo'shishtipidagi amallar ibo'lgan ifoda, qo'shish va ayirish amallarini kiradi, masalan, $5-3, (5*3)/10-2$;

Bular asosidagi hisoblash jarayonini tasavvur qilishimiz mumkin va quyidagi formulalar bilan ifodalaymiz.

Expression (Term (Factor ()));

Bunda Factor – oddiy ifodaning hisoblash protsedurasini ibo'lib, son, qavslarga egachiziqchi ifoda, Term – ko'paytirish amal tipidagi amallarni saqlovchi ifodaning qiymatini hisoblash protsedurasini, Expression – qo'shishtipidagi amallarni saqlovchi ifodaning qiymatini hisoblash protsedurasini.

Oddiy ifodaning hisoblash protsedurasini:

```
int Factor ()
{
    char ch = getChar();
    if (isDigit (ch)) return getValue(ch);
    if (ch == '(')
    {
        int result = Formula();
        if (getChar() == ')') return result;
        error ("kutilmagan belgi yoki amal");
        return 0;
    }
    return error ("kutilmagan belgi yoki amal");
}
```

Dastur ko'paytirish tipiga oid amallarning hisoblash uchun ifodaning tahlil qiladi.

Ko'paytirish tipiga oid amallarning hisoblash uchun ifodaning hisoblashning umumiy formulasini quyidagicha:

$$F_1 * F_2 * \dots * F_n$$

Aniq tushunishimiz keraki, hisoblash bajarilayotgan amallar faqat ko'paytirish yoki ibo'lish amal ibo'ladi. Bunday amallarning hisoblash uchun Term protsedurasini yaratish mumkin. Bu protseduraning parametrlari butun sonli qiymatlar bo'lishi kerak. Bunday chaqiruv operandlarini tanlashimiz va kerakli amalni bajarish uchun son yoki oddiy ifodaning iborati bo'lgan operandlarini aniqlashimiz lozim,

soʻngesaamal nibajarish mumkin.

Buni quyidagi dastur fragmenti asosida amalga oshirish mumkin.

```
int Term (int left)
{
    char ch = getChar(); int right;
    if (ch != '*' && ch != '/')
    {
        /* navbatdagi amal kutilmagan boʻlsa, uni
        qaytarishimiz kerak */
        returnChar(); /* kutilmagan belgini qaytarish */
        return left; /* kutilmagan qiymatni qaytarish */
    }
    /* Hammasi yaxshi boʻlsa, oʻng operandni aniqlash
    kerak */
    right = Factor();
    if (ch == '*')
    {
        return Term(left * right);
        /* bu hisoblashni bajarishni aniqlaydi */
    }
    if (right == 0) return error ("Nolga boʻlinish");
    return Term(left / right);
}
```

Dastur qoʻshishti piga oid amallarni hisoblash uchun ifodani tahlil qiladi. Qoʻshishti piga oid amallarni hisoblash uchun ifodani hisoblashning umumiy formulasini quyidagicha:

$$T_1 + T_2 + \dots + T_n$$

Aniqtushunishimiz keraki, hisoblash bajarilayotgan amallar faqat qoʻshish yoki ayirish amal boʻladi. Bunday amallarni hisoblash uchun Expression protsedurasini yaratish mumkin.

Bu protseduraning parametrlari butun sonli qiymatlar boʻlishi kerak.

Bunda chap operandni tanlashimiz va kerakli amal nibajarish uchun son yoki oddiy ifoda dan iborat oʻng operandni aniqlashimiz lozim, soʻngesaamal nibajarish mumkin.

Buni quyidagi dastur fragmenti asosida amalga oshirish mumkin.

```
int Expression (int left)
{
```



```

char ch = getChar (); int right;
if (ch != '+' && ch != '-')
{
    /* navbatdagi amal kutilmagan bo'lsa, uni
qaytarishmiz kerak*/
    returnChar();/*kutimagan belgini qaytarish */
    return left; /*kutimagan qimatni qaytarish */
}
/* Hammasi yaxshi bo'lsa, o'ng operandni aniqlah
kerak*/
right = Term (Factor());
if (ch == '+')
{
    return Expression(left + right);
/* bu hisoblashni bajarishni aniqlaydi */
}
return Expression(left - right);
}

```

Yuqoridagi protseduralar ishlatilishi uchun quyidagi usullarni mustaqil ishlatib chiqish kerak. Umuman olganda rekursiv kamayish usuli qiziq tirgan edi, o'ylaymizki, quyidagi fragmentlarni yaratib buning asosini hosil qilasiz.

1. Kirishoqimidan navbatdagi belgini kutuvchi, `getChar` parametrsiz usuli.

2. Kutilmagan belgini kirishoqimiga chiqarib berish `returnChar` usuli.

3. `isDigit` va `getValue` usullarini ham yozish shart emas. Agar parametrison bo'lsa, `isDigit` `true` qaytaradi. Ikkinchi usul esa parametr sifatida kirishoqimidan berilgan satrdan sonlarni ajratib olib qaytaradi.

4. `Error` usuli xatolar haqidagi xabarlar nishonlash uchun ishlatiladi.

Bu usullarni mustaqil hal qilish mumkin, ammo `.NET` standart sinflar usullari asosida barcha zaruri modifikatsiyalar yaratish mumkin.

Budasturlar fragmentini shakllantirishda qiymat qaytarish asosida rekursiv kamayish usulidan foydalanildi (*recursive descent with backtracking*). Qiymat qaytarmaslik asosida rekursiv kamayish usuli bilan chegaralanib qolmaganimizni ko'rib chiqamiz.

Rekursiv kamayish usulidan foydalanish sharti. Qiymat qaytarmaslik asosida rekursiv kamayish usulidan faqat quyidagi shart bajarilganda foydalanish mumkin.

Shart: har bir ifodaning birinchi belgisibuholatda qaysi ifodaga amal qilishini aniqlash uchun yetarli bo'lishi kerak.

Bushartni aniqlash uchun **FIRST**

to'plamini aniqlash orqali amalga oshirish mumkin.

Ta'rif:

KS

grammatika uchun terminal va terminal bo'lmagan belgilardan iborat

G

grammatika va w zanjirasosida **FIRST** $k(w)$

to'plamni quyidagicha aniqlaymiz:

$$\text{FIRST}_k(w) = \{x \mid w \Rightarrow^* xv, |x| = k \text{ yoki } w \Rightarrow^* x, |x| < k\},$$

k – natural son.

Boshqacha qilib aytganda, **FIRST** $k(w)$ to'plam w dan olingan terminal zanjirlarning uzunligi k bo'lgan barcha terminal prefikslardan iborat.

Universal algoritmi tildatiplarning kichik qismini yaratadigan grammatika ni ko'rib chiqaylik:

$\text{type} \rightarrow \text{imple}$

$\text{type} \rightarrow \wedge \text{id}$

$\text{type} \rightarrow \text{array}[\text{simple}] \text{ of type}$

$\text{simple} \rightarrow \text{integer}$

$\text{simple} \rightarrow \text{char}$

$\text{simple} \rightarrow \text{num} \dots \text{num}$

Bu grammatika uchun quyidagilarni aniqlash mumkin:

$$\text{FIRST}_1(\text{simple}) = \{\text{integer}, \text{char}, \text{num}\}$$

$$\text{FIRST}_1(\wedge \text{id}) = \{\wedge\}$$

$$\text{FIRST}_1(\text{array}[\text{simple}] \text{ of type}) = \{\text{array}\}$$

Agar w zanjir faqat terminallardan iborat bo'lsa, **FIRST** $k(w)$ - w zanjirda birinchi k belgilardir, aks holda $|w| \geq k$, yoki agar $|w| < k$ bo'lsa, w zanjirning o'zi.

Parsers infi.

FIRST

to'plamning qurish algoritmi. Birinchi navbatda belgilargrammatikasi uchun **FIRST** to'plamni aniqlaymiz.

1-qadam. Agar x terminal bo'lsa, **FIRST**(x)= x ;

2-qadam. $x \rightarrow \varepsilon$ qoida uchun **FIRST**(x) to'plamga ε ni qo'shamiz;

3-qadam. Agar x terminal emas va grammatika qoidasi $x \rightarrow y_1 y_2 \dots y_k$ bo'lsa, **FIRST**(x) ga terminalni qo'shamiz, agar qandaydir i uchun terminal **FIRST**(y_i) ga tegishli bo'lsava ε - **FIRST**(y_1), ..., **FIRST**(y_{i-1}) to'plamlarga tegishli bo'lsa, $y_1 y_2 \dots y_{i-1} \Rightarrow^* \varepsilon$, agar barcha $j=1, 2, \dots, k$ uchun ε - **FIRST**(y_j) to'plamga tegishli bo'lsa, ε ni **FIRST**(y) to'plamga qo'shamiz.

FIRST(w) to'plamni qurish algoritminishakllantiramiz.

Kirish: KS grammatika $G=(N, T, P, S)$ va w zanjirning terminal va terminal bo'lmagan belgilari.

Chiqish: FIRST(w).

Usul: FIRST (X₁) dan barcha bo'sh bo'lmagan belgilarni FIRST (X₁ X₂ ... X_k) ga qo'shamiz. Agar $\varepsilon \in \text{FIRST}(X_1)$ ga tegishli bo'lsa, FIRST (X₂) dan barcha bo'sh bo'lmagan belgilarni, shu bilan iteratsiya ni davom ettiramiz. Oxirida, Agar barcha j uchun FIRST (X_j) bo'sh belgiga ega bo'lsa, FIRST (X₁ X₂ ... X_k) to'plamga ε ni qo'shamiz.

Misol: Quyidagi qoidalar bilan berilgan grammatika qaraymiz:

$S \rightarrow BA$

$S \rightarrow +BA$

$A \rightarrow \varepsilon$

$B \rightarrow DC$

$C \rightarrow *DC$

$C \rightarrow \varepsilon$

$D \rightarrow (S)$

$D \rightarrow a$

Ushbu grammatika uchun FIRST to'plami quyidagicha aniqlanadi.

$\text{FIRST}(D) = \{ (, a \}$, $\text{FIRST}(C) = \{ *, \varepsilon \}$, $\text{FIRST}(B) = \text{FIRST}(D)$, $\text{FIRST}(A) = \{ +, \varepsilon \}$, $\text{FIRST}(S) = \{ (, a \}$.

LL(k)-grammatika ko'rib chiqamiz.

Ta'rif: $G = (VT, VN, P, S)$ grammatika LL(k)-grammatika deb atiladi, agar $\text{FIRST}_k(x) = \text{FIRST}_k(y)$ teng, $u = u_1$ bo'lsa va ikkita ixtiyoriy chapchiqishlar uchun ifoda o'rinli bo'lsa,

$S \Rightarrow^* wAv \Rightarrow wuv \Rightarrow^* wx$

$S \Rightarrow^* wAv \Rightarrow wu_1v \Rightarrow^* wy$

Av dan chiquvchi terminal va terminal bo'lmagan belgilar va k birinchi belgilardan (agar mavjud bo'lsa) iborat wAv joriy zanjir uchun mavjud bo'lsa, w bilan boshlanadigan va aytib o'tilgan terminallar bilan davom etadigan chiqishdagi qoida terminal zanjirini olish uchun aqaloq'lash mumkin bo'lgan kamida bitta qoida mavjud.

Qoidaga asoslangan grammatika ni qaymiz:

$S \rightarrow aAS$

$S \rightarrow b$

$A \rightarrow a$

$A \rightarrow bSA$

vaikkitachiqish

(1) $S \Rightarrow^* wSv \Rightarrow wuv \Rightarrow^* wx$

(2) $S \Rightarrow^* wSv \Rightarrow wu_1v \Rightarrow^* wy$.

Agarxvayzanjirlarabilanboshlansin. Buesachiqishda $S \rightarrow aAS$ qoidaishtiroketganinibildiradi. Haqiqattanhamu $= u1 = aAS$. Agarx va y zanjirlar b bilanboshlansin. Buesachiqishda $S \rightarrow b$ qoidaishtiroketganinibildiradi. Haqiqattanhamu $= u1 = b$. Bundayko‘rinadiki, ikkitachiqishamallario‘xshashvatengkuchlidir. Bundayesayuqoridagigrammatika LL(1) – grammatikaxususiyatlariniqo‘llabquvvatlaydi.

LL(1) –

grammatikagaasoslanganQiymatqaytarmaslikasosidagirekursivkamayish usuliuchuntahlilovchiqurishmumkin. Masalan,

(1) $S \rightarrow \text{if } S \text{ then } S \text{ else } S$

(2) $S \rightarrow \text{begin } S \text{ L}$

(3) $S \rightarrow \text{print } E$

(4) $L \rightarrow \text{end}$

(5) $L \rightarrow ; SL$

(6) $E \rightarrow \text{num} = \text{num}$

LL(1) –

grammatikagaasoslanganrekursivkamayishusuliuchunquyidagigrammatikaniqaraymiz:

(1) $S \rightarrow \text{if } E \text{ then } S \text{ else } S$

(2) $S \rightarrow \text{begin } S \text{ L}$

(3) $S \rightarrow \text{print } E$

(4) $L \rightarrow \text{end}$

(5) $L \rightarrow ; S \text{ L}$

(6) $E \rightarrow \text{num} = \text{num}$

Bugrammatikaniqo‘llabquvvatlvchitahlillovchinirekursivkamayishu suliasoslanibyaratishmumkin.

Buninguchunterminalbo‘lmagangrammatikauchunharbiriuchunbittadanp rotsedurayozishkerakbo‘ladi.

```
class SimpleParser {
    const int IF = 1;
    const int THEN = 2;
    const int ELSE = 3;
    const int BEGIN = 4;
    const int END = 5;
    const int PRINT = 6;
```

```

const int SEMICOLON = 7;
const int NUM = 8;
const int EQ = 9;

public static void nextStep(int lc){
    if (lexical_class == lc)
        lexical_class = getLC();
    else
        error();
}

public static void S(void){
    switch(getLC())
    {
        case IF:
            E(); nextStep(THEN); S();
nextStep(ELSE); S(); break;
        case BEGIN:
            S(); L(); break;
        case PRINT:
            E(); break;
        default:
            error(); break;
    }
}

public static void L(void){
    switch (lexical_class)
    {
        case END:
            getLC(); break;
        case SEMICOLON:
            getLC(); S(); L(); break;
        default:
            error(); break;
    }
}

```

```

public static void E(void){
    nextStep(NUM); nextStep(EQ); nextStep(NUM);
}

public static void main(void){
    lexical_class = getLC();
    S();
}
}

```

Chaprekursivgrammatikalar. LL (k) – xususiyatlarigrammatikauchunkuchlicheklovlar yuklaydi.

Natijadagrammatika LL(1) xususiyatigaegabo‘lishiuchunba‘zangrammatikanio‘zgartirishmumkin. Buniamalgaoshirishuchunhardoimhammuvaaffaqiyatlibo‘lmaydi, lekinbir LL(1) grammatikaolishimkoniyatigaegabo‘ldi, keyinrekursivkamayishusuliasosidaanalizatorqurishdafoydalanishingiz mumkin. Farazqilaylik, quyidagigrammatikaasosidatahlillovchiqurishkerakbo‘lsin.

$$E \rightarrow E + T | E - T | T$$

$$T \rightarrow T * F | T / F | F$$

$$F \rightarrow \text{num} | (E)$$

FIRST(T) to‘plamning terminallari

FIRST(E+T) to‘plamga tegishli bo‘lsin. Bu holda protseduralarning ketma-ket chaqirish bir qiyamatli aniqlab bilamaymiz, chunki kirishzanjiridagi larni tahlil qilishimiz kerak.

Muammo shundaki, terminal bo‘lmagan E ning o‘ng pozitsiyasida qoidaning o‘ng tomoni, chap qismi ham E dabo‘ladi. Bunday holatlarda terminal bo‘lmagan E – chaprekursiv debaytiladi.

Ta‘rif. Agar grammatikada $A \Rightarrow^* A_w$ chiqish mavjud bo‘lsa, A terminal bo‘lmagan KS grammatika G chaprekursiv debaytiladi.

Bittabo‘lsahamchaprekursiv qoidaga egabo‘lgan grammatika LL(1)-grammatika bo‘la olmaydi. Ikkinchi tomondan bilamizki, KS til bittabo‘lsahamchaprekursiv bo‘lmagan grammatika bilan aniqlanadi.

Chaprekursivlikni bartaraf etish algoritmi. To‘g‘ridan-to‘g‘ri chaprekursivlikni bartaraf etish algoritmini keltiramiz.

$G = (N, T, P, S)$ – KS grammatika va $A \rightarrow A\omega_1 | A\omega_2 | \dots | A\omega_n | v_1 | v_2 | \dots | v_m$ qoidalar P dagi barcha qoidalar ni ifodalasin, A ning chap qismi nisqalamaydi, chunki biror birzanjir v_i –

Aterminalbo‘lmaganbilanboshlanmaydi.

N

to‘plamgayanabirtaerminalbo‘lmagan

A qo‘shamizvaAdachaptomondagibo‘lganqoidalarnio‘zgartiramiz.

Quyidagiqoidalarniolamiz.

$$A \rightarrow v_1 | v_2 | \dots | v_m | v_1 A' | v_2 A' | \dots | v_m A' |$$

$$A' \rightarrow \omega_1 | \omega_2 | \dots | \omega_m | \omega_1 A' | \omega_2 A' | \dots | \omega_m A' |$$

Olingangrammatikaoldingisigaekvivaletkanliginiko‘rsatishmumkin

. Yuqoridakeltirilgangrammatikagabuakslantirishniqo‘yib, arifmetikamallarniyozi shuchun quyidagichagrammatikaniolamiz:

$$E \rightarrow T | TE'$$

$$E' \rightarrow +T | +TE'$$

$$T \rightarrow F | FT'$$

$$T' \rightarrow *F | *FT'$$

$$F \rightarrow (E) | \text{num}$$

TuzilgangrammatikaLL(1)

xususiyatlarigaegaekanliginiko‘rsatishmumkin.

Agarikkitaqoidahamterminalbo‘lmaganbixilbelgibilanbo‘lsa, aniqmuammonivaziyatgakeltiradi, masalan,

$$S \rightarrow \text{if } E \text{ then } S \text{ else } S$$

$$S \rightarrow \text{if } E \text{ then } S$$

Bundayhollardayanabirterminalbo‘lmaganqoidaqo‘shamiz, qaysikiyuqoridagivaziyatgaoydinlikkiritib, farqliqoidalarnikeltiribchiqaradivaquyidagiqoidaniolishimizmumkin:

$$S \rightarrow \text{if } E \text{ then } S S'$$

$$S' \rightarrow$$

$$S' \rightarrow \text{else } S$$

Olingangrammatikarekursivkamayishusulibilanamalgaoshirilishimumkin.

Qiymatqaytarishasosidagirekursivkamayishusuli. Shundayqilib, qiymatqaytarishasosidagirekursivkamayishusuliniqo‘llashuchungrammatikani FIRST to‘plamlarkesishmaydiganshaklgaaylantirishkerak. Bumurakkabvako‘pvaqttalabqiladiganjarayondir. Shuninguchun, amaliyotda, tez-tezishlatiladiganiborabor, qiymatqaytarishasosidagirekursivkamayishhodisadebataladi.

Buninguchunleksikanalizatoran‘anaviyusullarnito‘laligicha, scan va next

ana‘naviyusullarvanusxalashkonstruktorigaegabo‘lganob’ektsifatidanam oyonbo‘ladi.

Keyin,

noaniqlikyuzagakelishimumkinbo‘lganbarchaholatlarda,

tahlil qilishni boshlashdan oldin leksikanalizatorning hozirgi holatini aniqlaymiz (ya'ni, leksikanalizatorning nusxasini boshlaymiz) va matn ni tahlil qilishni davom ettirishga harakat qilamiz.

Butahlil muvaffaqiyatsiz bo'lsa, leksikanalizator holatini tiklaymiz va shunday qilib, keyingi grammatika variantini yordamida yanashu zanjirni tahlil qilishga harakat qilina divatugaguncha davom ettiriladi. Barcha tahlilimkoniyatlarimuvaffaqiyatsiz bo'lsa, xatolar haqida xabar beramiz.

Butahlil lovi chini bu usul qiyamat qaytarmaydigan rekursiv kamayish usuli ko'ra potentsialise kinikanligini ochiq-oydinko'ri y mumkin, lekin evaziga uning asl shakli da grammatika da qolishivada sturchi uchun hara katsaq lab qolish uchun bosh qarishosondir.

Xuddi shu sintaktik tahlil

bemolli kompilyatorida sintaktik tahlil uchun joriy qilingan. Masalan, quyidagi fragmentda tahlil paytida konstruktsiya ifoda operatori yoki izoh kan liginani qlovchi usul amalga oshiradi.

Ikki konstruktsiya nuqtasi bilan ajratilgan va ehtimol kvadrat qavslar ni o'z ichi ga olgan identifikatorlar ketma-ketligi bilan boshlanishi mumkin.

```
AST.Stmt stmt_or_decl ()
{
    Lexer saved = new Lexer (lexer);
    try {
        AST.Type t = type_opt ();
        if (lexer.Is (Token.Tag.Ident))
            return decl_tail (saved.Curr.coor, t);
    }
    catch (ParseFailed) {
        lexer = saved;
        AST.Expr expr = this.expr ();
        return new AST.Stmt.Expr (compiler,
            saved.Curr.coor | lexer.req
            (Token.Tag.Semicolon).coor, expr);
    }
}
```

try blokida unitavsi fde bhisoblab, konstruktsiya ajratishga harakat qilamiz. Bu muvaffaqiyatsiz bo'lsa, tahlil lovi chi istisno chaqiradivakeyin catch blokida tahlil boshlanadivashu konstruktsiya ifoda si fatida tahlil qilinadi.

Sintaktikanalizatorni yaratish. Odatda, mashinatilidagimatngaplar, gap-gaplarostidan, gapostiesa, o‘znavbatida, gaplarostidanvashukabibelgilardanhosilbo‘ladi. Masalan, XML hujjatdaboshlanishtegi, qiymat, yopishteglarimavjud. Boshlanishtegi [<] belgisi, tegnomivaqandayatributvaqiymatlari, [>] belgisidaniborat. Yopishtegiesa, [</]belgisi, tegnomi, [>]belgisibilantugaydi. Atributesanomi, [=], [“], belgilarto‘plamivayana [“] belgisidaniboratvatakrorlanishimumkin.

Qiymatesaqandaydirbelgilarto‘plamiyokiqandaydirelementlardan (gap, gaposti, ifoda) iborat. Tahlilnatijasidatahlildaraxtinihosilqilishmumkin.

Bundaytillarniharbirterminalbo‘lmagantilningma’lumbirjumlasigam oskeladiganBackus-Naurshakli (BNSh) yordamidatasvirlashqulay. Dasturlarniyozganda, odatdaularnifunksiyalargavafunksiyaostilargaajratamizvasietatiktahlillovchiyozmochibo‘lganimizuchun, harbirBNShterminalbo‘lmagansintatiktahlillovchimizningbittafunksiyasi gamoskelsinvaharbirbundayfunksiya:

- ushbujumlaniberilganpozitsiyadanajratishgaharakatqilish;
- buishnibajardimiyo‘qminatijasiniqaytarish;
- tahliltugaganiyokixatosodirbo‘lganpozitsiyaniqaytarish;
- tahlilnatijasidaolishkerakbo‘lganba‘ziqo‘shimchama’lumotlarniqaytarish;

tarish;

Masalan, BNShko‘rinishidaexpr ::= expr1 expr2 expr3 funksiyaniozimiz:

```
bool read_expr(char *& p, ....){
    if(!read_expr1(p, ....))
        return false;
    // read_expr1() fuksiyasi expr1 tahlil tugaga
    pozitsiyada p qo‘yadi
    // bu pozitsiyada p bilan hech qanday amal
    bajarilmaydi.
    if(!read_expr2(p, ....))
        return false;
    if(!read_expr3(p, ....))
        return false;
    return true;
}
```

Davomisifatida, BNShko‘rinishidaexpr ::= expr1|expr2|expr3 funksiyaniozimiz:

```

bool read_expr(const char *& p, ....){
    const char * l = p;
    if(read_expr1(p, ....))
        return true;
    // expr1 tahlil jarayonida p xato bo'lgan joyni
    ko'rsatadi
    p = l;
    if(read_expr2(p, ....))
        return true;
    p = l;
    if(read_expr3(p, ....))
        return true;
    return false;
}

```

Busintatiktahlillashqiymatqaytarishasosidagirekursivkamayishusuli gaasoslangan.

IxtiyoriymatnniBNShko'rinishidakeltirishmumkin. Masalan, XML tilidagiquyidagichayozishmumkin:

element ::= '<'identifier>'some_data'</'identifier>'

Agaridentifikatorlarimoskelsabuhaqiqattanhamto'g'rikeladi.

Bundayamallarnisintaktiktahlillovchigaqo'shishmuammoemas. Masalan, terminalfunksiyalarquyidagichabo'lishimumkin:

```

bool read_fix_char(const char *& it, char c){
    if(!*it)    return false; // satr oxiri
    if(*it!=c)   return false; // boshqa simvol
                // xato bo'lganda joyiga qolish
    bajariladi
    it++;       // keyingisiga o'tish
    return true;
}
bool read_fix_str(const char * & it, const ch_t *
s){
    while(*it)// satr tugamagancha
        if(!*s)
            return true;
        else if(*it!=*s)
            return false;
        // xato bo'lganda joyiga qolish
}

```

```

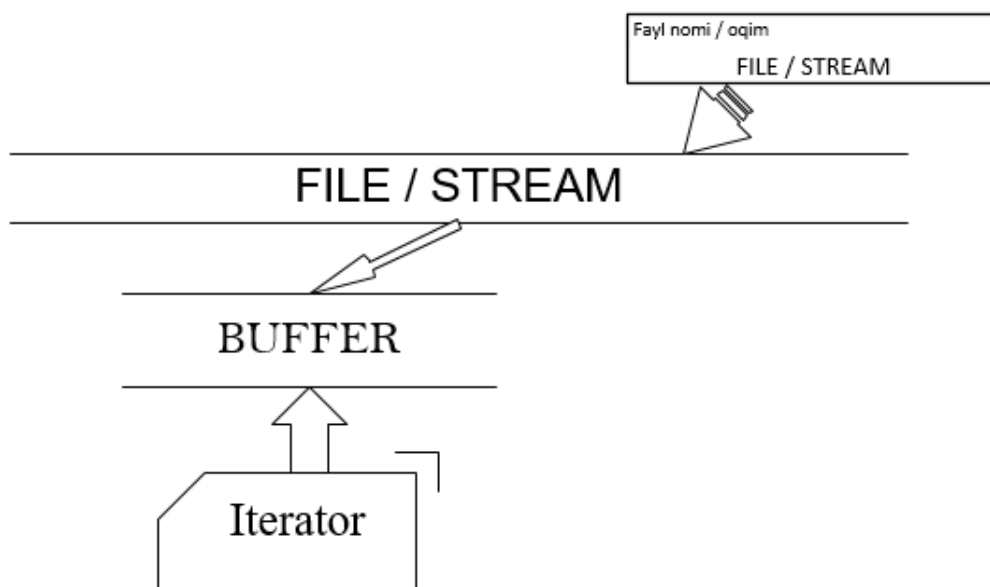
bajariladi
    else
        it++, C++;
    if(!*s) return true;
    return false;
}

```

Satrnisintaktiktahlilqilganimizdayuqoridagifunksiyalarningbarchatiplari uchun forward iteratorlaryetarliekanliginibiling.

forwarditeratorlarnitaqdimetadiganvafaylnibutunlayemas, faylo‘rnigakichikqismlargaxotiradasaqldigansinfoqimniyaratishniqaraymiz. Agarshablonlarasosidatahlillovchifunksiyaniiyaratishnixoxlasak, ularnisatrlarva/yokioqimlardanfoydalanishmumkin (base_parse.h kutubxonasiterminalfunksiyalariuchun, businfnidiskdagimavzugaooidpapkagaqarang). Birinchinavbatda unix «barchafayllarbor»: ideyalogiyasigaba‘zianiqliklarnikiritibolamiz: Shundayfayllarbo‘ladiki, ulardiskbo‘yichajoylashganvaularniixtiyoriypoziitsiyasidanbirnechamarta o‘qishmumkin(ularni random-access fayllardebaytiladi), shundayoqimlarbo‘ladiki, tarmoqdan, klaviaturadan, boshqailovalardanyo‘naltirilgan (bularni forward oqimlardebaytiladi). Bundayoqimlarnibirmartagao‘qishkerak. Shuninguchun, unisivabunisibilanishlashuchunbiryondashuvdafaylnio‘qiydigan, forwarditeratorlarniqo‘llabquvvatlaydigan, xotiradafaylnito‘liqemas, balkima’lumbirbo‘laginisaqldigansinf-oqimniyaratishmaqsadgamuvofiq.

Bundayoqimlar Input iteratorlaruchunanchaoldinyaratilgan, maqsadifaqatulara Input iteratorlar, buiteratorharakatlanishiuchunbittabuferbo‘ladi. Qachonki, buferoxirigakelgandabufergafaylnikeyingibo‘lagiyuklanadihamdaitera torbo‘shatilganbuferboshidanharakatlanishniboshlaydi (7.1-rasmgaqarang).



7.1-rasm. Input iteratorlardan foydalanish.

Forward iteratorlarning input iteratrlardan farqi ularni usxalash mumkinligidir.

Bunday masalada oqimlarda forward iteratorlardan foydalanish uchun buferlar ro'yxatini hosil qilish orqali echiladi.

Qandaydir iterator birinchi buferga murojaat bilan ro'yxatga yangi bufer qo'shiladi va fayldagi blok ma'lumotlar bilan to'ldiriladi.

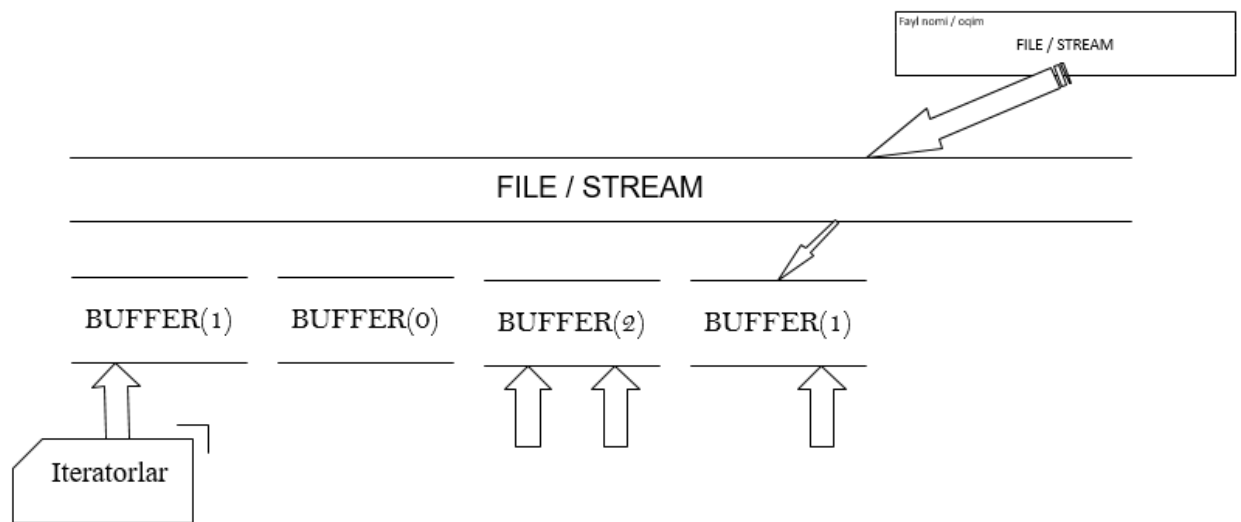
Ammo bu usulda fayl to'liq xotiraga yuklanadi, bu holat kerakmas.

Shunday qilamizki,

buferning chap tomonidagi joylashgan oxirgi chap iterator joylangan barcha buferlar o'chirilsin. Lekin,

unga joylashgan bufer ro'yxat bilan har bir iterator mos bo'lmaydi, lekin faqat ularning sonini mos bo'ladi (hisoblovchi iteratorlari).

Biror bir bufer 0 iterator qolganini va u engchapekanligini bilsa, u barcha undan o'ng buferdagi barcha qo'shni lari o'chiriladi. Chunki ular ham 0 iteratorga ega bo'ladi (7.2-rasmga qarang).



7.2-rasm. Forward iteratorlardan foydalanish.

Bunday kelib chiqadiki, har bir iterator quyidagilarni o'z ichiga olishi kerak:

- belgi ga ko'rsatkich (qaytanomlanishda qaytariladigan qiymat);
- bufer ga dagi belgilarning massivning oxiriga ko'rsatkich iterator harakatlanishiga qaratilgan hamalga oshiriladi);
- bufer o'yxatib o'yicha iteratorlar (bufer ma'lumotlariga murojjat uchun vaular orqali butun oqim ma'lumotlari murojjat).

Iterator bufer chegarasiga yetganda, ukeyingibuferlarro'yxatdaborligin tekshiradi, bo'lmasa - uniyuklaydi, oldingibufer iteratorlar sonini kamaytiradi va keyingisi sonini oshiradi. Agar oldingibuferda 0 iterator qolgan bo'lsa, u o'chiriladi. Agar iterator faylning oxiriga chakelgan bo'lsa, u faqat oldingibufer dancha qiriladi va "faylning oxiri" holatiga o'tadi. Bir iteratorning nusxasini olganda joriy buferda iteratorlarning soni bittaga ko'payadi, iteratorning destruktori ishlaganda bittaga kammayadi, va, yana 0 bufer bo'lsa qolgan iteratorlar, u va o'ngda undan keyingibuferlar, shuningdek 0 iteratoriborlar o'chiriladi. Amalga oshirish uchun diskda mavjud papka dagi forward_stream.h ga qarang.

Bunday iteratorlar anaviy iteratorlardan ularni ajratib turadigan ba'zi xususiyatlarga ega. Masalan, biroqimni (buferlar va ba'zi qo'shimcha ma'lumotlarro'yxatini saqlaydigan) o'chirish holdin (destruktortomonidan), barcha iteratorlar o'chirilgan bo'lishi kerak, ularni o'chirish vaqtida o'zlariga bog'liq bo'lmagan holda buferlarga o'z navbat

atidao‘chirilganliginianiqlashmurojaatqiladi. Agar begin()
 usulinichaqirish (birinchibuferniyaratish)
 natijasidabirinchiiteratornibirmartaolsakvaushupaytgachabirinchi buferall
 aqachono‘chirib tashlanganbo‘lsa, yana begin()
 usuliyordamidaiteratorniololmaymiz. Bundantashqarioqimning end()
 usuliyo‘q. Natijada, oqimdaichkiiteratoryaratishkerakbo‘ladi,
 ya‘nibarchaoqimlaryaratilayotgandayaratiladiganvamosyozuvlar iter()
 usuliyordamidaolinishimumkin. Bundantashqari,
 algoritmlarniishlatganda, iteratorlarantez-teznusxalashkerakemas,
 chunkixotirachapdano‘nggachabo‘lganiterator dabuferlarnisaqlaydi.
 Kattafayllaruchunbukattaxotiratalabqilishigaolibkeladi.

Yordamsifatidaharxilturdagibuferlarmavjud: oddiy
 (basic_simple_buffer), qator-ustunliiterator (basic_adressed_buffer).
 Bularyordamidahisoblashuchunturlikodlashlarorasidaqaytakodlashnibaja
 ruvchibuferqilishimumkin.

Shumunosabatbilanoqimbuferningtipibilanparametrlashtiriladi.

Buyruqlarsatriningoxirinullbelgibilanbelgilanadi.
 Bushunianglatadiki, uningoxirinitopishuchunsatrorqaliharakatqilsak,
 ko‘rsatgichniboshqako‘rsatgichbilansolishtirishimizshartemas
 (STLdaan‘anaviytarzdaamalgaoshiriladi), aksincha,
 ko‘rsatgichimiztomonidanko‘rsatilganqiymatnitekshirishkerak.

Fayllarbilanbog‘liqvaziyathamshunday.

Belgilikiritishdabelgilarasosidasonlarniolamizvaularkattaqiymatlarsatrig
 aegabo‘lganiuchunyanaharbirbelgini EOF() funksiyabilantaqqoslanadi.
 Realvaqtdakeladiganforwardoqimlaruchunfaylningoxiriholatioldindanno
 ma’lum.

Bumuanmonisatrliko‘rsatgichlaruchun atend(it)
 funksiyasinituzishorqaliuumulashtirishimumkin:

```
bool atend(const char * p)
{ return !*p; }
```

Agariteratorfaylningoxiriniko‘rsatsa, oqimbo‘yichaiteratorlarga true
 qiymatqaytaradi.

Foydalanuvchibilaninteraktivhamkorlikqilishuchun (stdin orqali)
 bloklibuferlashamalgaoshirilmaydi,
 chunkiblokdaodatdabirnechtasatrjoylashtiriladivabittasatr kiritilgandanso
 ‘ng, dasturfoydalanuvchidankirishnikutishnidavometadi,
 chunkiblokkalito‘lmagan.

Satrningoxirigachabo‘lganbelgilarbufernito‘ldiradivabuninguchunsatrlib
 uferlashkerak.

Ushbukutubxonadaoqimishgatushirilgandafayl tipinitanlabbuferlashturini tanlashingiz mumkin (basic_block_file_on_FILE yoki string_file_on_FILE).

strin – stdin orqalisatrlibuferlashuchunbuferningichkiiteratoridir. Interaktivbo‘lmaganfayllaruchunoqimiyaratishdabirinchi belgigaishoraqiluvchiiterator yaratiladi, ya‘nibirinchi bufer yuklanadi. Bunga strin uchun ruxsat berilmaydi, chunki dasturchi satr rejimidan kiritishni kutishdan oldin biror bir narsa amalga oshirish yoki ekranga chiqarigim mumkin.

Shu sababli birinchi bufer nito‘ldirishda satrlifayllari ishlatilganda belgil ibuyruq ‘\n‘dan foydalaniladi. Unio‘qish uchun, start_read_line (it) funksiyasi mavjud, shunda so‘ngi satrni tahlil qilish uchun dastur satrni kiritishni kutish rejimiga keladivakeyingi belgi ‘\n‘ tashqaridachiqishda iterator busatrniajratiboladi.

Agar bundan keyin yana foydalanuvchidan ma’lumotlarni kerak bo‘lsa, dasturchi yana ekranda biror narsa nichi qarishni xohlashi mumkin va uni olishdan oldin yana start_read_line(strin) dan foydalanish mumkin. Bunda quyidagicha takrorlanish hosil qilinadi:

```
while(true){
    cout << "Satr kirit:" << endl;
    start_read_line(strin); //satr kiritish
    analys(strin);
}
```

Albatta, bu fragment faqat qaytanomlash vaqtida buferni yuklashda iterator talab qilish mumkin, lekin bu qaytanomlash qachon qo‘shimcha tekshirishlar olib kelishivabuti zimni murakkablashtirishi mumkin.

7.1-dastur. STRIN Kutubxonasi.

```
#define ONE_SOURCE //strin.cpp
#include "../strin.h"

using namespace str;
using std::cout;
using std::endl;

#define ifnot(expr) if(!(expr))
#define r_if(expr) if((expr)==0)
```

```

#define r_while(expr)      while((expr)==0)
#define r_ifnot(expr)      if(expr)
#define r_whilenot(expr)   while(expr)

template <class it_t, class mes_t>
void dump(it_t tmp, mes_t mes){
    string s;
    read_fix_length(tmp,50,&s);
    std::cerr <<mes <<dump(s.c_str()) <<endl;
}

template<class it_t>
const char * read_sum(it_t & it, double * prez);

template<class it_t>
const char * read_expr(it_t & it, double * prez){
    const char * err;
    if(read_s_fix_char(it,'(')){
        r_ifnot(err = read_sum(it,prez))
            return err;
        ifnot(read_s_fix_char(it,')'))
            return "qavsni yoping";
        return 0;
    }
    int x;
    ifnot(read_dec(it,&x)){
        return "son";
    }
    *prez = x;
    return 0;
}

DEF_STRING(md,"*/")
template<class it_t>
const char * read_mul(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;

```



```

    r_ifnot(err = read_expr(it, prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it, make_span(md<ch_t>().
s), &zn))
            return 0;
        double x;
        r_ifnot(err = read_expr(it, &x))
            return err;
        if(zn == (ch_t) '*')    *prez *= x;
        else                    *prez /= x;
    }
    return 0;
}

DEF_STRING(pm, "+-")
template<class it_t>
const char * read_sum(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_mul(it, prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it, make_span(pm<ch_t>().
s), &zn))
            return 0;
        double x;
        r_ifnot(err = read_mul(it, &x))
            return err;
        if(zn == (ch_t) '+')    *prez += x;
        else                    *prez -= x;
    }
    return 0;
}

```

```

int main(){
    using namespace std;
    if(0)    {
        string str_expression_1 = "5+84/(51)";
        string str_expression_2 = "15 + (16 /6 *4)";
        string str_expression_3 = "7 + ( 4* 4)";
        string str_expression = str_expression_3;
        cout <<"hisoblash ifodasi:" <<endl
<<str_expression <<endl;
        const char * p= str_expression.c_str();
        const char * err;
        double rez;
        r_ifnot(err=read_sum(p,&rez)){
            int pos=p-str_expression.c_str();
            for(int i=1; i<pos; i++)
                cout <<' ';
            cout <<'^' <<endl;
            cout <<"Xatolik: " <<err <<endl;
            return -1;
        }
        cout << "natija: " << rez << endl;
        cout <<"=====" << endl;
    }

    if(1)    {
        try{
            while(!atend(strin)){
                cout <<"arifmetik ifoda kiriting va
'end' bilan tugating " <<endl;
                read_start_line(strin);
                const char * err;
                double rez=0;
                err=read_sum(strin,&rez);
                linecol lc=get_linecol(strin);
                read_until_str(strin,"end");
                r_if(err)
                    cout << "natija: " << rez <<

```

```

endl;
        else{
            cout <<" zanjir " <<lc <<"
xatolik: " <<err <<endl;
        }
    }
}
catch(const char * mes){
    cerr << "xatolik: " << mes << endl;
    return -1;
}
catch(string & mes){
    cerr << "xatolik: " << mes << endl;
    return -1;
}
catch(stream_exception & mes){
    cerr << "xatolik: " << mes.what() <<
endl;
    return -1;
}
catch(...){
    cerr << "noaniq xatolik" << endl;
}
}
return 0;
}

#ifndef STRIN_H
#define STRIN_H

#include <exception>
#include "base_parse.h"
#include "forward_stream.h"

#ifdef ONE_SOURCE
#    define STRIN_EXTERN
#else
#    define STRIN_EXTERN extern

```

```

#endif

namespace str{

    STRIN_EXTERN forward_adressed_stream STREAMin
#ifdef ONE_SOURCE
    (true, new file_on_FILE_stringbuf(stdin))
#endif
    ;
    STRIN_EXTERN forward_adressed_stream::iterator &
    strin = STREAMin.iter();
    inline int __set01(typename
    forward_adressed_stream::iterator & it){
        set_linecol(it,linecol(0,1));
        return 0;
    }
    STRIN_EXTERN int __unused_int
#ifdef ONE_SOURCE
    = __set01(strin)
#endif
    ;

    class str_error{
#define constructor str_error
    typedef str_error my_t;
    protected:
        const char * message;
    public:
        constructor():message(0){}
        constructor(const char * mes):message(mes){}
        constructor(int x):message(0){ if(x)throw "???"
int"; }
        constructor(bool x) :message(x?0:""){
        operator bool()const{ return message==0; }
        const char * what()const { return message; }
#undef constructor
    };
    DECLARE_AND_OR_default_error_operators(str_error)

```

```

DECLARE_AND_OR_error_operators_for_short_circuiting(s
tr_error)
str_error operator>>(base_parse_error from, str_error
to){
    if(from) return str_error(true);
    elsereturn to;
}

class strpos_error : public str_error{
#define constructor strpos_error
typedef strpos_error my_t;
typedef str_error base_t;
    linecol lc;
    friend strpos_error operator&&(strpos_error l,
strpos_error r);
    friend strpos_error operator||(strpos_error l,
strpos_error r);
public:
    linecol ignored_lc=linecol(0,0);
    bool ignored()const{return lc==ignored_lc; }
    constructor(){};
    constructor(const char * mes):base_t(mes){}
    constructor(int x) :base_t(x){}
    constructor(bool x) :base_t(x){}

    constructor(const linecol & l)
        :lc(l){};
    constructor(const char * mes, const linecol & l)
        :base_t(mes),lc(l){};
    constructor(bool mes, const linecol & l)
        :base_t(mes),lc(l){};

    linecol where()const
    { return lc; }
    my_t & set_what(const char * mes)
    { base_t::message=mes;return *this; }
    template<class it_t>
    my_t & set_ignored(const it_t & it)

```

```

        {   ignored_lc =linecol(it);return *this;   }
#undef constructor
};

template <class it_t>
strpos_error operator>>(str_error from, const it_t *
to){
    return strpos_error(from.what(),linecol(*to));
}
std::ostream & operator<<(std::ostream & str, const
strpos_error & e){
    return str<< "("<<"ignored:"<<e.ignored()<<" at
"<<e.ignored_lc<<)" "<<(bool)e<<"; "<<(e?"OK":e.what())
<<" at "<<e.where();
}
strpos_error operator&&(strpos_error l, strpos_error
r){
    r.set_ignored(l.ignored_lc);
    if(l)
        if(r)
            if(r.ignored())
                return l;
            else
                return r;    /*TT*/
        else
            return r;    /*TF*/
    else
        if(r)
            return l;    /*FT*/
        else
            if(r.ignored())
                return l;
            else
                return r;    /*FF*/
}
strpos_error operator||(strpos_error l, strpos_error
r){
    r.set_ignored(l.ignored_lc);

```

```

        if(l)
            if(r)
                if(r.ignored())
                    return l;
                else
                    return r;    /*TT*/
            else
                return l;    /*TF*/
        else
            if(r)
                return r;    /*FT*/
            else
                if(r.ignored())
                    return l;
                else
                    return r;    /*FF*/
    }
    DECLARE_AND_OR_error_operators_for_short_circuiting(s
    trpos_error)

} //namespace str
#endif //STRIN_H

```

Sintatiktahlillovchilarning bazaviy funksiyalari.

Harsafar terminal funksiyalarni yozish dasturchilarga noqulaylik keltirmasligi uchun «base_parse.h» sinfi datayyorlab qo'yilgan. Hozirda umumiy ko'rinishda (xaqiqiy sonlarning antashqari) tayyorlangan vakelajak dastur bo'yicha ko'rsatkiqlar va oqim bo'yicha iterat orlar uchun satrlifunksiyalar (strcmp, strstr, strchr, strspn, strcspn kabi) ishlaydigan qismini tuzish rejalashtirilgan. Shuningdek bu faylning oxiri haqida o'ylash kerak emas, faqat fragmentni to'g'ri yozsabo'ladi.

Quyidagi parsingning bazaviy funksiyalari qisqacha keltirilgan va ularni test parserni amalga oshirish paytida ularning foydalanish statistikasini aniq ravishda qiymatlarini fayl tarqatib berish uchun keltirilgan.

size_t	n
ch_t	c
ch_t *	s
func_obj	bool is(c) // bool isspace(c)

span	spn
bispan	bspn
func_obj	err pf(it*) // int read_spc(it*)
func_obj	err pf(it*, rez*)

Len – belgilarsoni, *pstr gaqo‘shilgan, satroxiriyokioxiriemasligianiqllovchistatistikavaqiymatlaririqaytaradi.

int read_until_eof	(it&)	.*\$	0	0	1	OK
int read_until_eof	(it&, pstr*)	.*\$	len	len		OK
int read_fix_length	(it&, n)	{n}	-1	0		OK
int read_fix_length	(it&, n, pstr*)	{n}	-(1+len)	0		2 OK
int read_fix_str	(it&, s)	str	-(1+len)	0 ili (1+len)	9	OK
int read_fix_char	(it&, c)	c	-1	0 yoki 1	11	OK
int read_charclass	(it&, is)	[]	-1	0 yoki 1		OK
int read_charclass	(it&, spn)	[]	-1	0 yoki 1		OK
int read_charclass	(it&, bspn)	[]	-1	0 yoki 1		OK
int read_charclass_s	(it&, is, pstr*)	[]	-1	0 yoki 1		OK
int read_charclass_s	(it&, spn, pstr*)	[]	-1	0 yoki 1		1 OK
int read_charclass_s	(it&, bspn, pstr*)	[]	-1	0 yoki 1		5 OK
int read_charclass_c	(it&, is, ch*)	[]	-1	0 yoki 1		OK
int read_charclass_c	(it&, spn, ch*)	[]	-1	0 yoki 1		1 OK
int read_charclass_c	(it&, bspn, ch*)	[]	-1	0 yoki 1		OK
int read_c	(it&, ch*)	.	-1	0	5	OK
int read_while_charclass	(it&, is)	[]*	-(1+len)	len		

OK				
int read_while_charclass	(it&, spn)	[]*	-(1+len)	len
2 OK				
int read_while_charclass	(it&, bspn)	[]*	-(1+len)	len
OK				
int read_while_charclass	(it&, is, pstr*)	[]*	-(1+len)	len
OK				
int read_while_charclass	(it&, spn, pstr*)	[]*	-(1+len)	len
OK				
int read_while_charclass	(it&, bspn, pstr*)	[]*	-(1+len)	len
1 OK				
int read_until_charclass	(it&, is)	.*[]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, spn)	.*[]<-	-(1+len)	len
1 OK				
int read_until_charclass	(it&, bspn)	.*[]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, is, pstr*)	.*[]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, spn, pstr*)	.*[]<-	-(1+len)	len
2 OK				
int read_until_charclass	(it&, bspn, pstr*)	.*[]<-	-(1+len)	len
OK				
int read_until_char	(it&, c)	.*c	-(1+len)	len
OK				
int read_until_char	(it&, c, pstr*)	.*c	-(1+len)	len
OK				

Oxirgibelginio‘qigandanso‘ng, iteratorundankeyinemas, balkiungaqaratilgan.

Funksiyanomidanfoydalanib, xatokodiniyokixatoxabariniqaytarishqulay, shuninguchunparserlarmatniyanadaaniqroqko‘rinadi, maxsusmakroslarkeltiramiz:

```
#define r_if(expr)          if((expr)==0)
#define r_while(expr)      while((expr)==0)
#define r_ifnot(expr)      if(expr)
#define r_whilenot(expr)   while(expr)
```

```

#define rm_if(expr)          if((expr)>=0)
#define rm_while(expr)      while((expr)>=0)
#define rm_ifnot(expr)      if((expr)<0)
#define rm_whilenot(expr)   while((expr)<0)
#define rp_if(expr)         if((expr)>0)
#define rp_while(expr)      while((expr)>0)
#define rp_ifnot(expr)      if((expr)<=0)
#define rp_whilenot(expr)   while((expr)<=0)

```

Buma'lumotlarnisaqlashva bool
tipigamosakslantirishbo'lsadaba'zisinfniqilishuchunjudamuhim.

Umuman, o'zgarmasbo'lganhavolalargaqarshiman,
funksiyasio'zgartirishkerakargumentlarko'rsatgichlaribilanemas,
balkibirko'rsatkichorqaliungao'tganbo'lishikerak

Ushbufunksiyalardanfoydalanishningikkitamisolkeltiramiz:

7.2-dastur. Arifmetikifodalarnihisoblash.

```

#define ONE_SOURCE //strin.cpp
#include "../strin.h"

using namespace str;
using std::cout;
using std::endl;

#define ifnot(expr)          if(!(expr))
#define r_if(expr)          if((expr)==0)
#define r_while(expr)       while((expr)==0)
#define r_ifnot(expr)       if(expr)
#define r_whilenot(expr)    while(expr)

template <class it_t, class mes_t>
void dump(it_t tmp, mes_t mes){
    string s;
    read_fix_length(tmp,50,&s);
    std::cerr <<mes <<dump(s.c_str()) <<endl;
}

template<class it_t>
const char * read_sum(it_t & it, double * prez);

```

```

template<class it_t>
const char * read_expr(it_t & it, double * prez){
    const char * err;
    if(read_s_fix_char(it,'(')){
        r_ifnot(err = read_sum(it,prez))
            return err;
        ifnot(read_s_fix_char(it,')'))
            return "qavsni yoping";
        return 0;
    }
    int x;
    ifnot(read_dec(it,&x)){
        return "son";
    }
    *prez = x;
    return 0;
}

```

```

DEF_STRING(md,"*/")
template<class it_t>
const char * read_mul(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_expr(it,prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it,make_span(md<ch_t>().
s),&zn))
            return 0;
        double x;
        r_ifnot(err=read_expr(it,&x))
            return err;
        if(zn==(ch_t) '*')    *prez *= x;
        else                  *prez /= x;
    }
}

```

```

    }
    return 0;
}

DEF_STRING(pm,"+-")
template<class it_t>
const char * read_sum(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_mul(it,prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it,make_span(pm<ch_t>().
s),&zn))
            return 0;
        double x;
        r_ifnot(err = read_mul(it,&x))
            return err;
        if(zn==(ch_t)'+')    *prez += x;
        else                *prez -= x;
    }
    return 0;
}

int main()
{
    using namespace std;
    if(0)
    {
        string str_expression_1 = "5+84/(51)";
        string str_expression_2 = "15 + (16 /6 *4)";
        string str_expression_3 = "7 + ( 4* 4)";
        string str_expression = str_expression_3;
        cout <<"hisoblash ifodasi:" <<endl
<<str_expression <<endl;
        const char * p= str_expression.c_str();
    }
}

```

```

        const char * err;
        double rez;
        r_ifnot(err=read_sum(p,&rez)){
            int pos=p-str_expression.c_str();
            for(int i=1; i<pos; i++)
                cout <<' ';
            cout <<'^' <<endl;
            cout <<"Xatolik: " <<err <<endl;
            return -1;
        }
        cout << "natija: " << rez << endl;
        cout <<"===== " << endl;
    }

    if(1)
    {
        try{
            while(!atend(strin)){
                cout <<"arifmetik ifoda kiriting va
'end' bilan tugating " <<endl;
                read_start_line(strin);
                const char * err;
                double rez=0;
                err=read_sum(strin,&rez);
                linecol lc=get_linecol(strin);
                read_until_str(strin,"end");
                r_if(err)
                    cout << "natija: " << rez <<
endl;
                else{
                    cout <<" zanjir " <<lc <<"
xatolik: " <<err <<endl;
                }
            }
        }
        catch(const char * mes){
            cerr << "xatolik: " << mes << endl;
            return -1;
        }
    }

```

```

    }
    catch(string & mes){
        cerr << "xatolik: " << mes << endl;
        return -1;
    }
    catch(stream_exception & mes){
        cerr << "xatolik: " << mes.what() <<
endl;
        return -1;
    }
    catch(...){
        cerr << "noaniq [atolik" << endl;
    }
}

return 0;
}

```

7.2-dastur. Output

```

--> runcpp -f calc.cpp
option -f
runcpp: g++ calc.cpp (options = -std=gnu++11 -Wall
-Wno-parentheses)
runcpp: compile messages lines 0
arifmetik ifoda kiriting va 'end' bilan tugating
1
end
natija: 1
arifmetik ifoda kiriting va 'end' bilan tugating
1+1
end
natija: 2
arifmetik ifoda kiriting va 'end' bilan tugating
1+1+1+1+7*7*(3)
end
zanjir 5:1 xatolik: son
arifmetik ifoda kiriting va 'end' bilan tugating

```

```

1end
natija: 1
arifmetik ifoda kiriting va 'end' bilan tugating
7*(3)
end
zanjir 8:1 xatolik: son

arifmetik ifoda kiriting va 'end' bilan tugating
1+(2)
end
natija: 3
arifmetik ifoda kiriting va 'end' bilan tugating
1+e end
zanjir 12:3 xatolik: son
arifmetik ifoda kiriting va 'end' bilan tugating
(7)*3end
natija: 7
arifmetik ifoda kiriting va 'end' bilan tugating
7*(3)end
zanjir 2:1 xatolik: son

arifmetik ifoda kiriting va 'end' bilan tugating
(5)+(5)end
natija: 10

```

Bugungi kundasturchilar uchun maxsus sintaktik tahlillovchilari yaratilishko'phollardashartemas. Chunki bumashaqqatli vako'pvaqttalabqiladi. Umumanolganda qandaydirdasturiyta'minotlarni, axborot tizimlarining lingvistikta'minoti yaratilsa, shuta'minot uchun analizatori yaratish kerak bo'lishi mumkin. Dunyoda dasturlashtirishlabchi qaruvchilar busohada juda oldilab ketgan. Shuning uchun sintaktik tahlillovchilar bo'yicha nazariy tushunchaga egabo'li shyetarlidi. Ammo ayrim matematik yangi sonli inflar uchun ishlabchiqish mumkin.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Dasturlashtilining sintaktik konstruksiyalarini tasvirlash uchun eng qulay formalusul qaysi usul?
2. Dasturlarning sintaktik to'g'riligini aniqlashning yanabir muhim qismi - dastur datiplardan foydalanishning to'g'riligini qaysi grammatika yordamida aniqlab bo'lmaydi?
3. Sintaktik tahlil deganda nimatushuniladi?
4. n uzunlikdagi kiruvchi satr uchun murakkabligi qanchadan oshmasligi kerak bo'lgan har qanday kontekst-erkin grammatika uchun analizator (parser) qurish mumkin?
5. Sintaktik tahlillovchi kirish parametrlarini madani borat?
6. Sintaktik tahlillovchi chiqish parametrlarini maldan borat?
7. Leksik va sintaktik tahlil bosqichlarining alohida qarashlarga ajratilish shartmi?
8. Sintaktik tahlillovchi sinflarini nechta va qaysi sinflar algoritmlariga ta'luqli?
9. Past sathlarga to'g'raltirilgan analizatorlar chiqishni qurish uchun grammatika nima dan boshlab va nima bilan tugaydi.
10. LL grammatika bo'lmagan grammatikalarda analizatorlar uchun qaysi usullarda foydalanish mumkin.
11. Past sathlarga to'g'raltirilgan analizatorlarga qaraganda yuqori sathlarga to'g'raltirilgan analizatorlar qanchalik ishlatiladi.
12. Yuqori sathlarga to'g'raltirilgan analizatorlar bilan qaysi grammatika bilan bog'liq.
13. Bugungi kunda qaysi grammatikaga asoslangan analizatorlar bilan juda ko'p dasturlashtirilar foydalanmoqda.
14. Eng oddiy va juda ko'p foydalanilgan past sathlarga to'g'raltirilgan analizatorlarni qurish usulini ayting?
15. Ifodalarni hisoblashda barcha hisoblashlarni qandaysinflarga ajratish mumkin.
16. Chap operandni tanlashimiz va kerakli amalni bajarish uchun son yoki oddiy ifoda dan borat o'ng operandni aniqlashimiz lozim, so'ngesani manibajarish mumkin.
17. Qo'shishtipiga oid amallarni hisoblash uchun ifoda ni hisoblashning umumiy formulasi qanday?
18. Qiymat qaytarmaslik asosida girekursiv kamayish usulidan faqat qaysi hart bajarilganda foydalanish mumkin.
19. KS grammatika uchun terminal va terminal bo'lmagan belgilardan borat G

- grammatikava w zanjirasosida $FIRST_k(w)$ to'plamniqandayaniqlaymiz?
20. $FIRST_k(w)$ to'plam w danolinganterminalzanjirlarninguzunligi k bo'lganbarcha terminalnimalaridaniborat.
 21. Agar w zanjirfaqatterminallardaniboratbo'lsa, $FIRST_k(w)$ - w zanjirdabirinchi k belgilardir, aksholda $|w| \geq k$, yokiagar $|w| < k$ bo'lsa, nimabo'ladi.
 22. Agarxteminalemasvagrammatikaqoidasix $\rightarrow y_1 y_2 \dots y_k$ bo'lsa, $FIRST(X)$ gaqandayterminalniqo'shamiz.
 23. Agar $FIRST_k(x) = FIRST_k(y)$ teng, ... o'rinli bo'lsa $G = (V_T, V_N, P, S)$ grammatikaLL(k)-grammatikadebaytiladi.
 24. LL(k) – xususityalarigrammatikauchunqandaycheklovlaryuklaydi.
 25. Agargrammatikada $A \Rightarrow^* Aw$ chiqishmavjudbo'lsa, Aterminalbo'lmagan KS grammatikanimadebaytiladi.
 26. Qiymatqaytarishasosidagirekursivkamayishusuliniqo'llashuchungrammatikani FIRST to'plamlarqandayshaklgaaylantirishkerak.
 27. Leksikanalizatoran'anaviyusullarnito'laligicha, scan va next ana'anaviyusullarvanusxalashkonstruktorigaegabo'lgannimasifatidanamoyonbo'ladi?
 28. Mashinatilidagimatnnimalardaniborat?
 29. Harbirterminalbo'lmagantilningma'lumbirjumlasigamoskeladiganqaysishaklyordamidatasvirlashqulay?
 30. Qaysiko'rinishida $expr ::= expr1|expr2|expr3$ funksiyaniyozamiz?
 31. Forward iteratorlarnimauunkkerak?
 32. Input iteratorlarnimauchunkkerak?
 33. Input iteratorlardanfoydalanishnitushuntiribbering?
 34. Forward iteratorlardanfoydalanishnitushuntiribbering?
 35. Forward iteratorlarning input iteratrlardanfarqinimada?



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	Sintaktikanalizatorni yaratishga oid berilgan quyida gidasturbo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajar. 👉 Bunda dasturda giba'zi o'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.

dastur	topshiriqlar
<pre>// Created by MBBahodir class SimpleParser { const int IF = 1; const int THEN = 2; const int ELSE = 3; const int BEGIN = 4; const int END =; const int PRINT = 6; const int SEMICOLON = 7; const int NUM = 8; const int EQ = 9; public static void nextStep(int lc){ if (lexical_class == lc) lexical_class = getLC(); else error(); } public static void S(void){ switch(getLC()) { case IF:</pre>	1. Sinfxatosinitoping. _____ _____
	2. Sinfdanob'ektolibdasturdafoydala ning. _____ _____ _____
	3. Faqatqo'shishamalinitahlilqiluvch ivaifodanihisoblovchidasturtuzing. _____ _____ _____ _____ _____ _____
	4. Faqatayrishamalinitahlilqiluvchivaif odanihisoblovchidasturtuzing. _____ _____ _____ _____ _____ _____ _____

<pre> E(); nextStep(THEN); S(); nextStep(ELSE); S(); break; case BEGIN: S(); L(); break; case PRINT: E(); break; default: error(); break; } } public static void L(void){ switch (lexical_class) { case END: getLC(); break; case SEMICOLON: getLC(); S(); L(); break; default: error(); break; } } public static void E(void){ nextStep(NUM); </pre>	5. Qo‘shishvaayrishamallarinitahlilqiluvchivaifodanihisoblovchidasturtuzing. 6. Ko‘paytirishamalnitahlilqiluvchivaifodanihisoblovchidasturtuzing.

<pre> nextStep(EQ); nextStep(NUM); } public static void main(void){ lexical_class = getLC(); S(); } } </pre>	
7. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 8. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

2.4. Murakkabsaralashalgoritmlari. Amaliy dasturlash

 Murakkab saralashalgoritmlarini, ularningmurakkabligi, rekursivfoydalanishvatakomillashtirish, ishlatishg‘oyalarivasaralashalgoritmlarining 16 tavarientiningturlidasturfragmentlari, saralashalgoritmlarnitaqqoslash, turlisondagimassivlar, turliqiymatlardao‘zlarinitutishinisolishtirishnatijalari, kattasonlarbilashishlash, mashinaxatoliginianiqlash, foydalanuvchiningarifmetikamallarnibajaruvchifunksiyalarniyaratishbo‘yichag‘oyalar, kattasonlisinflarniishlatish, mavjudkattasonlarbilanishlovchilar, kattasonlaribilashishlovchikutubxonalariniyaratishvanamunalarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan. bilimlarnimustahkamlashuchun 35 tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 4 taassimenttopshirig‘ivaharassimentda 29 tatopshiriqberilgan. Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

 **Kalitso‘zlar.** Saralash, massiv, tip, xotira, sinfkutubxona, massiv, satrlimassiv.

 **Bilishshartbo‘lgantushunchalar.** Tiptushunchasi, sinfvasinfob‘ekti, oqim, kirishelementgamurojaat, funksiyavako‘rsatkich, ma’lumotlarnikirishvachiqishi, arifmetikamallar, kutubxonayaratishvadasturlashgaoiddastlabkitushunchalarhamdaC++tili niqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

❖Bilibolasiz.

Murakkabsaralashalgoritmlardan:
Havosharchakabisaralash, Shakersaralash, Taroqsimon caralash,
Qo'shishorqalisaralash, Shell caralash, Iearxiksaralash, Navsaralash,
Tanlashorqalisaralash, Piramidakabisaralash, Tezkorsaralash,
Birlashtirishorqalisaralash, Blokorqalisaralash, Bitlisaralash (LSD),
Bitlisaralash (MSD), Bitoniksaralash, Timsortsaralash,
bualgoritmlarnitaqqoslashvamurakkabligini,
Tiplarvaularningxususiyatlari, Kattasonlarchegarasinitekshirish,
Tiplargabog'liqkattasonlardanfoydalanishvaxatoligi,
Kattasonlarningqo'shish, ayirish,
ko'pytirishg'oyalarivaalgoritmlarinituzishni, Bignumber kutubxonasi,
sinfivafunksiyalari, xususiyatlariyaratish, OpenSSL, Libgcrypt, GMP,
Boost.Multiprecision, wide integer tiplari,
BigIntkutubxonasiniyaratishuslubivabazaviyfunksiyalarinio'rganishingiz
mumkin.

REJA

1. Murakkabsaralashalgoritmlari.
2. Saralashalgoritmlaritaqqoslash.
3. Judahamkattasonlarbilanishlash.
4. Large sinfiningqo'llanilishi.
5. Integer sinfining qo'llanilishi

KIRISH

Turlima'lumotlartuzilmalaridaniboratto'plamlarnisaralashalgoritml
arinibilasiz.

Ammoularniko'psonliqiymatlario'zinitutishiqandaydegansavolqiziqtirad
imi?

Nafaqatoddiysaralashbalkishundaymurakkabsaralashalgoritmlarihamma
vjud.

Ko'pelementlimassivlarkattasonlarbilashishlashgahamfoydalanishmumk
in.

Shuningdektiplarningo'zaromodifikatsiyasibilanhamuzunsonlarbilanishl
anadi.

Saralashalgoritmlarivamurakkabligi.Avvalo,
algoritmlarnitadqiqqilishga,
ularniimkonqadartezishlashiuchunoptimallashtirishmuhimdir.
Buustidaishlayotganda,
turlisaralashuchunsamaraliusullarnio'ylabtopishimkoniyatigaegabo'lishi
ngizmumkin.

Ko'pjihtdan,
 barchasarlalshalgoritmlarnio'rganishvaularnisinashkerak.
 Agardasturlashningo'zhaqidagapiradiganbo'lsak,
 ba'zankutilmaganqiyinchiliklarpaydobo'lishimumkin (C++
 optimizatorijudayaxshi). Biroq,
 qaysitestlarnivaqandaymiqdordaalgaoshirilishikerakliginihalqilishqiyi
 nemas. Menko'rsataolmaydiganyagonanarsa-budeyarli 150
 GBvazngaegaqiyamatlarnisaralashbo'ldi.

Asosiysarlalshalgoritmlaritavsifivaularniamalgaoshirishusullari.
 Saralashniqisqachavaaniqta'riflashgavamurakkabliginibelgilashgaharaka
 tqilaman.
 Murakkabma'lumotlartuzilmalarnifoydalanishda(daraxtsarlalshkabi)
 odatdaxotirakattamiqdordasarflanadivaengyomonholatdaboshqaxilfaqaty
 ordamchiqatoryaratishkerakbo'ladi. Barqarorlik (stabillik)
 saralashtushunchasiham mavjud. Demak,
 elementlarningnisbiytartibitengbo'lgandao'zgarmaydi.

Havosharchakabisarlalash (Bubble sort). Massivdachapdano'nggaqarabamal bajariladi.
 Agarjoriyelementkeyingisidankatabo'lsa, ularnialmashtiramiz.
 Bunimassivtartiblanmagunchabajaramiz. E'tiborbering,
 birinchiiterasiyadankeyinengkattaelementmassivningoxirida,
 to'g'rijoylashaganbo'ladi.
 Ikkitaiterasiyadankeyinikkiinchiengekattaelementto'g'rijoylashaganbo'ladi
 vahokazo. Ravshanki, n
 taiterasiyadanko'pbo'lmagandankeyinmassivtartiblanganbo'ladi.
 Shundayqilib, engyomonvao'rtachaholatdamurakkabligi $O(n^2)$,
 engyaxshiholatdaesa $O(n)$ bo'ladi.

8.1-dastur. Saralashniamalgaoshirish.

```
void bubblesort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    bool b = true;
    while (b) {
        b = false;
        for (int* i = l; i + 1 < r; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
    }
}
```

```

    }
    r--;
}
}

```

Shakersaralash (Shaker sort, bundantashqari,
 aralashgansaralashvakoktelsaralashsifatidanilgan).
 Havosharchasaralashoxiridakichikelementlarjoylashtirilgantestlardaseki
 nishlaydi (ba‘zan "tashboqa" debataladi).
 Buelementalgoritmniningharbirqadamidachaptomongaafaqatbittapozitsiyag
 aharakatqiladi. Shuninguchunnafaqatchapdano‘ngga,
 balkio‘ngdanchapgahamo‘tadi.
 Massivningtartiblanmaganqisminianiqlashuchunikkita begin va end
 ko‘rsatikichniqo‘llabquvvatlaydi. Keyingiiterasiyada
 endgaetganimizdaundanbiriniayiramizvao‘ngdanchapgako‘chamiz.
 Xuddishunday,
 begingaetganimizdaungabiriniqo‘shamizvachapdano‘nggako‘chamiz .
 Algoritmhavosharchasaralashbilanbixilmurakkabligibor,
 lekinrealvaqtdayaxshiishlaydi.

8.2-dastur. Saralashniamalgaoshirish.

```

void shakersort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    bool b = true;
    int* beg = l - 1;
    int* end = r - 1;
    while (b) {
        b = false;
        beg++;
        for (int* i = beg; i < end; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
        if (!b) break;
        end--;
        for (int* i = end; i > beg; i--) {
            if (*i < *(i - 1)) {
                swap(*i, *(i - 1));
            }
        }
    }
}

```

```

        b = true;
    }
}
}
}

```

Taroqsimon **caralash(Comb sort).** Shakersaralashningyanabiro‘zgartirilgani. "Toshbaqalar" danqutulishuchunmasofadaturganelementlarniqaytatashkilqilamiz. Keling, unituzatamizvachapdano‘nggaboramiz, bumasofadaturganelementlarnitaqqoslaymiz, agarkerakbo‘lsa, ularniqaytatartibgasolaylik. Shubhasiz, bu "toshbaqalar"nitezqatorboshigaolishimkoniniberadi. Dastlabqatoruzunligigatengmasofaniolib, so‘ngraunitaxminan 1.247 gatengbo‘lganbirorkoeffitsientgabo‘lishmaqbuldir. Masofabirbirigatengbo‘lgandashakersaralashbajariladi. Engyaxshisi, murakkabligi $O(n \log n)$, engyomoni, $uO(n^2)$. O‘rtachamurakkabliginimauchunjudaaniqmas, amalda $O(n \log n)$ kabiko‘rinadi.

8.3-dastur. Saralashniamalgaoshirish.

```

void combsort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    double k = 1.2473309;
    int step = sz - 1;
    while (step > 1) {
        for (int* i = l; i + step < r; i++) {
            if (*i > *(i + step))
                swap(*i, *(i + step));
        }
        step /= k;
    }
    bool b = true;
    while (b) {
        b = false;
        for (int* i = l; i + 1 < r; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
    }
}

```



```

    }
}
}

```

Qo'shishorqalisaralash (Insertion sort). Algoritmyakunlangandankeyinjavobnio'zichigaolganbiryangimassiv yaratiladi.

Javobmassivdagielementlarhardoim tartiblanib turishi uchun manba massiv idanelementlarni birlama-bir kiritamiz.

Murakkabligi $O(n^2)$ va eng yaxshi $O(n)$ bo'ladi.

Boshqa algoritmlarga nisbatan joriy qilish yaxshiroq (yangi massiv yaratish va albatta unimanidir qo'yish qiyin):

joriy massivning ba'zi birlamchi elementlari tartiblashtirilganiga ishonch hosil qilish va modomiki ular noto'g'ri tartibda ekan uni kiritish o'rniga joriy elementni avvalgisibilano'zgartiramiz.

8.4-dastur. Saralashni amalga oshirish.

```

void insertionsort(int* l, int* r) {
    for (int *i = l + 1; i < r; i++) {
        int* j = i;
        while (j > l && *(j - 1) > *j) {
            swap(*(j - 1), *j);
            j--;
        }
    }
}

```

Shell

caralash

(Shellsort). Saralashdabir xil fikrdan foydalanishnio'ylab ko'ring, taroqsimonsaralashvaqo'shishorqalisaralash uchun uniqo'llaymiz.

Masofat uchun chasinikiritaylik vatanlaylik.

So'ngi massiv elementlaris inflarga ajratiladi,

shuningga o'zgar masofaning bir nechta elementlarini kiradi.

Har bir sinfni qo'shib, tartiblash yo'libilantartiblaylik.

Taroqsimonsaralashdan farqli o'laroq, optimal masofalar to'plamini qimas.

Turlitaxminlarga ega bo'lgan bir nechta ketma-ketliklar mavjud bo'lsin.

Shellsaralashdabir inchi element ketma-ketligi massiv uzunligiga teng,

har bir keyingi element ketma-ketligi oldingisining yarimo'lchamiga tengdir.

Hibbard ketma-ketligi $2n-1$ uchun murakkabligi eng yomon holat $O(n^2)$ dir,

Sedgwick ketma-ketligi $O(n^{4/3})$ uchun (ifodao'zgaruvchan xususiyatli) va

Pratt (ikkivauchdarajali elementlar) ketma-ketligi $O(n \log^2 n)$

murakkabligi eng yomon holat $O(n^{1.5})$ dir.

Bubarchaketmaketliklarfaqatmassivhajmigaqadabhisoblabvakichikuchun ko'proqishlatishkerak

(aksholdaufaqtqo'shimchalaribilantartiblashtiriladibo'ladi).

Qo'shimchatadqiqotqildimva $s_i = a * s_{i-1} + k * s_{i-1}$ shaklidaturliketmaketliklar (buqismanempirik siurketma – ketlikdailhomlanib,

elementlarkichiksoniuchunengyaxshimasofaketmaketliklardanbiri)

sinovdano'tkazdim. Engyaxshiketmaketliklara = 3, k =1/3,a = 4, k = 1/4 vaa = 5, k = -1/5 koeffitsentlaribilanmaqsadgaerishishmumkin.

8.5-dastur. Saralashniamalgaoshirish.

1- variant
<pre> void shellsort(int* l, int* r) { int sz = r - l; int step = sz / 2; while (step >= 1) { for (int *i = l + step; i < r; i++) { int *j = i; int *diff = j - step; while (diff >= l && *diff > *j) { swap(*diff, *j); j = diff; diff = j - step; } } step /= 2; } } </pre>
2- variant
<pre> void shellsorthib(int* l, int* r) { int sz = r - l; if (sz <= 1) return; int step = 1; while (step < sz) step <<= 1; step >>= 1; step--; while (step >= 1) { for (int *i = l + step; i < r; i++) { int *j = i; </pre>

```

        int *diff = j - step;
        while (diff >= 1 && *diff > *j) {
            swap(*diff, *j);
            j = diff;
            diff = j - step;
        }
    }
    step /= 2;
}
}

```

3- variant

```

int steps[100];
void shellsortsedgwick(int* l, int* r) {
    int sz = r - l;
    steps[0] = 1;
    int q = 1;
    while (steps[q - 1] * 3 < sz) {
        if (q % 2 == 0)
            steps[q] = 9 * (1 << q) - 9 * (1 << (q
/ 2)) + 1;
        else
            steps[q] = 8 * (1 << q) - 6 * (1 << ((q
+ 1) / 2)) + 1;
        q++;
    }
    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= 1 && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}

```

```
}
```

4- variant

```
void shellsortpratt(int* l, int* r) {  
    int sz = r - l;  
    steps[0] = 1;  
    int cur = 1, q = 1;  
    for (int i = 1; i < sz; i++) {  
        int cur = 1 << i;  
        if (cur > sz / 2) break;  
        for (int j = 1; j < sz; j++) {  
            cur *= 3;  
            if (cur > sz / 2) break;  
            steps[q++] = cur;  
        }  
    }  
    insertionsort(steps, steps + q);  
    q--;  
    for (; q >= 0; q--) {  
        int step = steps[q];  
        for (int *i = l + step; i < r; i++) {  
            int *j = i;  
            int *diff = j - step;  
            while (diff >= l && *diff > *j) {  
                swap(*diff, *j);  
                j = diff;  
                diff = j - step;  
            }  
        }  
    }  
}
```

5- variant

```
void myshell1(int* l, int* r) {  
    int sz = r - l, q = 1;  
    steps[0] = 1;  
    while (steps[q - 1] < sz) {  
        int s = steps[q - 1];  
        steps[q++] = s * 4 + s / 4;  
    }  
}
```

```

    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= l && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}

```

6- variant

```

void myshell2(int* l, int* r) {
    int sz = r - l, q = 1;
    steps[0] = 1;
    while (steps[q - 1] < sz) {
        int s = steps[q - 1];
        steps[q++] = s * 3 + s / 3;
    }
    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= l && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}

```

7- variant

```

void myshell3(int* l, int* r) {

```

```

int sz = r - 1, q = 1;
steps[0] = 1;
while (steps[q - 1] < sz) {
    int s = steps[q - 1];
    steps[q++] = s * 4 - s / 5;
}
q--;
for (; q >= 0; q--) {
    int step = steps[q];
    for (int *i = 1 + step; i < r; i++) {
        int *j = i;
        int *diff = j - step;
        while (diff >= 1 && *diff > *j) {
            swap(*diff, *j);
            j = diff;
            diff = j - step;
        }
    }
}
}

```

Iearxihsaralash

(Tree

sort). Binarqidiruviearxiyasigaelementlarnikiritamiz.

Barchaelementlarkiritilgandanso‘ng,

iearxiyadantartiblanganmassivolishkifoya.

Agarsizqizil-

qorakabimuvozanatliiearxiyadanfoydalansangiz,

murakkabligiengyomon, o‘rtachavaengyaxshiholatlarda $O(n \log n)$.

Amalgaoshirishda multiset konteynerfoydalanadi.

8.6-dastur. Saralashniamalgaoshirish.

```

void treesort(int* l, int* r) {
    multiset<int> m;
    for (int *i = l; i < r; i++)
        m.insert(*i);
    for (int q : m)
        *l = q, l++;
}

```

Navsaralash

(Gnome

sort). Algoritmqo‘shishorqalisaralashgao‘xshash.

Ko‘rsatgichnijoriyelementgako‘rsatamiz,

agaruavvalgisidankattabo'lsayokiubirinchibo'lsako'rsatgichnito'g'rihola
tgako'chiramiz,
aksholdajoriyvaoldingielementlarnio'zgartiramizvachaptomongaharakatl
antiramiz.

8.7-dastur. Saralashniamalgaoshirish.

```
void gnomesort(int* l, int* r) {
    int *i = l;
    while (i < r) {
        if (i == l || *(i - 1) <= *i) i++;
        else swap(*(i - 1), *i), i--;
    }
}
```

Tanlashorqalisaralash

(Selection

sort).Navbatdagiiterasiyabo'yichamassivdagiminimumnijoriyelementda
nkeyintopamizvakerakbo'lsaubilano'zgartiramiz. Shundayqilib, i-
iterasiyadankeyinbirinchi i elementlaroz'joylaridaqoladi.
murakkabligiengyaxshi, o'rtachavaengyomonholatlarda $O(n^2)$.
Busaralashniikkiyo'lbilanamalgaoshirilishimumkinliginiunutmang -
minimalvauningindekstutib, yokishunchakiularnoto'g'ritartibdabo'lsa,
ko'ribchiqilayotganvajoriyelemento'rinalmashtirishkerak.
Birinchiusulbiroztezroq.

8.8-dastur. Saralashniamalgaoshirish.

```
void selectionsort(int* l, int* r) {
    for (int *i = l; i < r; i++) {
        int minz = *i, *ind = i;
        for (int *j = i + 1; j < r; j++) {
            if (*j < minz) minz = *j, ind = j;
        }
        swap(*i, *ind);
    }
}
```

Piramidakabisaralash

(**Heapsort**).Tanlashorqalisaralashg'oyasinirivojlantirilganvarianti.

Piramidashaklidagima'lumotstrukturasidanfoydalanaylik.

Elementlarniqo'shishvaminimuminichi qarishorqali $O(\log n)$, $O(1)$
minimuminiolishimkoniniberadi. Shundayqilib, $O(n \log n)$
murakkabligiengyomon, o'rtachavaengyaxshiholatlarda. C++ da
priority_queue konteynerimavjudbo'lsa-da,

bukonteynerjudasekinbo‘lganiuchunPiramidagayo‘natirilganyangito‘pla
msinfniamalgaoshirdim.

8.9-dastur. Saralashniamalgaoshirish.

Heap sinfi

```
template <class T>
class heap {
private :
    vector<T> h;
    void SiftUp(int a) {
        while (a) {
            int p = (a - 1) / 2;
            if (h[p] > h[a]) swap(h[p], h[a]);
            else break;
            a--; a /= 2;
        }
    }
    void SiftDown(int a) {
        while (2 * a + 1 < n) {
            int l = 2 * a + 1, r = 2 * a + 2;
            if (r == n) {
                if (h[l] < h[a]) swap(h[l], h[a]);
                break;
            }
            else if (h[l] <= h[r]) {
                if (h[l] < h[a]) {
                    swap(h[l], h[a]);
                    a = l;
                }
                else break;
            }
            else if (h[r] < h[a]) {
                swap(h[r], h[a]);
                a = r;
            }
            else break;
        }
    }
public:
```



```

heap():n(0){}
int n;
int size() {
    return n;
}
int top() {
    return h[0];
}
bool empty() {
    return n == 0;
}
void push(T a) {
    h.push_back(a);
    SiftUp(n);
    n++;
}
void pop() {
    n--;
    swap(h[n], h[0]);
    h.pop_back();
    SiftDown(0);
}
void clear() {
    h.clear();
    n = 0;
}
T operator [] (int a) {
    return h[a];
}
};

```

Heap cinfiaosidasaralash

```

void heapsort(int* l, int* r) {
    heap<int> h;

    for (int *i = l; i < r; i++) h.push(*i);
    for (int *i = l; i < r; i++) {
        *i = h.top();
    }
}

```

```

        h.pop();
    }
}

```

Tezkorsaralash (quicksort). Baʼzimosyozuvlarelementinitanlaylik. Shundansoʻngchapdankichikboʻlganbarchaelementlarni, kattaboʻlganlariniesaoʻngtomongaharakatlantiramiz. Takrorlanuvchiqismlarningharbiridanrekursivchaqirishmumkin. Natijadatartiblanganmassivgaegaboʻlamiz, chunkiharbirkattaelementdanoldinmassivdankichikboʻlganharbirelement joylashtirilgan. Murakkabligi: $O(n \log n)$ oʻrtachavaengyaxshiholda, $O(n^2)$ engyomonholda. Malumotelementinotoʻgʻritanlangandaengyomonreytinggaerishiladi. Bualgoritmniamalgaoshirishbutunlaystandarthisoblanadi, birvaqtningoʻzidachapgavaoʻnggaborib, elementlarnijufttopish, bundachapelementmosyozuvlaritanlangandankattaekanliginivaoʻngkichi k, ularalmashtiriladi. Softezkortartiblashdantashqari, elementlarsonikamboʻlgandatartiblashnijoylashtirishga, taqqoslashdahamsaralashishtiroketdi. Qoʻshishorqalisaralashvazifauchunmosboʻlganengyaxshisaralashdirvado imiysinovorqalitanlanadi.

8.10-dastur. Saralashniamalgaoshirish.

Birinchivariant
<pre> void quicksort(int* l, int* r) { if (r - l <= 1) return; int z = *(l + (r - l) / 2); int* ll = l, *rr = r - 1; while (ll <= rr) { while (*ll < z) ll++; while (*rr > z) rr--; if (ll <= rr) { swap(*ll, *rr); ll++; rr--; } } if (l < rr) quicksort(l, rr + 1); if (ll < r) quicksort(ll, r); } </pre>
Ikkinchivariant

```

void quickinssort(int* l, int* r) {
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int z = *(l + (r - l) / 2);
    int* ll = l, *rr = r - 1;
    while (ll <= rr) {
        while (*ll < z) ll++;
        while (*rr > z) rr--;
        if (ll <= rr) {
            swap(*ll, *rr);
            ll++;
            rr--;
        }
    }
    if (l < rr) quickinssort(l, rr + 1);
    if (ll < r) quickinssort(ll, r);
}

```

Birlashtirish orqali saralash (Merge sort). Bo‘lish-boshqarish paradigmasi asosida tartiblash. Massivni ikkigabo‘lamiz, qismlarni rekursiv ravishda tartiblang va keyin birlashtirish protsedurasini bajaring. Birinchi qismning joriy elementiga, ikkinchi qismning joriy elementiga ikki ta ko‘rsatkichni qo‘llang. Bu ikki elementdan minimumni tanlab, javobga joylashtiring va minimumga mos keladigan ko‘rsatkichni siljiting. Birlashtirish $O(n)$, barcha $\log n$ darajalari uchun ishlaydi, shuning uchun murakkabligi $O(n \log n)$. Bu oldindan vaqtinchalik masiv yaratish va vazifa argument sifatida o‘tishis amari bo‘ladi. Busaralash rekursiv, hamda tezlashning uchun elementlar soni kam bo‘lgan kvadratiko‘tish mumkin.

8.11-dastur. Saralashni amalga oshirish.

Asosiy funksiya

```

void merge(int* l, int* m, int* r, int* temp) {
    int *cl = l, *cr = m, cur = 0;
    while (cl < m && cr < r) {
        if (*cl < *cr) temp[cur++] = *cl, cl++;
    }
}

```

```

        else temp[cur++] = *cr, cr++;
    }
    while (cl < m) temp[cur++] = *cl, cl++;
    while (cr < r) temp[cur++] = *cr, cr++;
    cur = 0;
    for (int* i = l; i < r; i++)
        *i = temp[cur++];
}

```

Birinchivarinat

```

void _mergesort(int* l, int* r, int* temp) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    _mergesort(l, m, temp);
    _mergesort(m, r, temp);
    merge(l, m, r, temp);
}

void mergesort(int* l, int* r) {
    int* temp = new int[r - l];
    _mergesort(l, r, temp);
    delete temp;
}

```

Ikkinchivarinat

```

void _mergeinssort(int* l, int* r, int* temp) {
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int *m = l + (r - l) / 2;
    _mergeinssort(l, m, temp);
    _mergeinssort(m, r, temp);
    merge(l, m, r, temp);
}

void mergeinssort(int* l, int* r) {
    int* temp = new int[r - l];
    _mergeinssort(l, r, temp);
    delete temp;
}

```

Sanashorqalisaralash (Counting sort). r - l hajmlimassiv yaratilgan, massivning l - minimalva r maksimalelementi hisoblanadi. Shunday qilib, massiv orqali barcha elementlarni hisoblash.

Endi qiymatlar massividan o'tib, har bir sonni nechta marta bo'lganini yozishingiz mumkin. murakkabligi $O(n + r)$ bo'ladi. Bualgoritmlar qaytar qaytilish uchun o'zgartirishingiz mumkin: buning uchun, keyingi soni joyini aniqlash kerak bo'lib, vachapdano'ngga original massiv orqali borish, to'g'ri joyda elementni qo'yib, va o'rnini birga oshirish lozim. Bularni sinovdan o'tkazilmagan, testlari yetarli darajada kattaroq amallar uchun mavjud, chunki yangi massiv yaratish ancha xotiratalab qiladi. Biroq, bualgoritmlar masalaga qarab foydali bo'ladi.

Bloklar qalisaralash (Bucket sort). Bualgoritmlar savatvach o'nta saralash sifatida tanilgan algoritmdir. Massivning l minimalva r maksimalelementi bo'lsin. Elementlarni bloklarga ajratamiz, birinchi blokda l dan $l + k$ gacha, ikkinchisigal $l + k$ dan $l + 2k$ gacha va hokazo elementlarni o'z ichiga oladi, buyurda $k = (r - l) / n$, n bloklar soni. Umuman olganda, agar bloklar soni ikkigateng bo'lsa, bualgoritmlar tez korsaralashga aylanadi. Ushbu bualgoritmlarning murakkabligi aniqlanmas, ma'lumotlariga murojaat vaqt va bloklar soniga bog'liq. Muvaffaqiyatli ma'lumotlar bo'yicha ish vaqt ichi qiziqlik ekanligi aytilib o'tiladi. Ushbu bualgoritmlar amalga oshirish eng qiyin vazifalardan biri ekanligini isbotlagan. Buning uchun hali yangi massivlarni yaratish, ularni rekursiv saralash va ularni bir-biriga birlashtirish orqali amalga oshirishingiz mumkin. Biroq, buyondashuv juda sekin va men qoniqmadim.

Samarali amalga oshirishda bir nechta oylardan foydalaniladi:

1) yangi massivlarni yaratmaymiz. Buning uchun sana o'rqalisaralash texnikasidan foydalanamiz: har bir blokda elementlarni sonini, prefiks summalarini va shu tariqa har bir elementning massivdagi o'rnini sanaymiz.

2) Bo'sh bloklardan boshlamaymiz. Bo'sh bo'lmagan bloklarning indekslarini alohida massivda kiritish va ularni boshlash orqali saralash.

3) Massiv tartiblanganligini tekshiramiz. Agar hali ham minimal va maksimal topish uchun biroz tishqilish kerak, chunki, bu ishlayotgan vaqti yomonlashmaydi, lekin elementlar original massiv bir xil tartibda yangi bloklari joylashtirilgan, chunki u qisman tartiblashtirilgan ma'lumotlarni saralashni tezlashtirish uchun algoritmi beradi.

4) Algoritmi juda noqulay bo'lgani uchun, u juda kam sonli elementlar bilan juda samarasiz.

Joylashtirishga o'tish tartibida ishni taxminan 10 marta gachatezlashtiradi.

Bu faqat qancha bloklarni tanlashni tushunishga qolmoqda. Randomize testlarda quyidagi ballarni olishga muvaffaq bo'lgan: 1500 elementlari uchun 107 bloklar va 3000 uchun 108 ta. Murakkablik formulasini topolmadik, ish vaqti bir necha marta yomonlashadi.

8.12-dastur. Saralashni amalga oshirish.

```
void _newbucketsort(int* l, int* r, int* temp) {
    if (r - l <= 64) {
        insertionsort(l, r);
        return;
    }
    int minz = *l, maxz = *r;
    bool is_sorted = true;
    for (int *i = l + 1; i < r; i++) {
        minz = min(minz, *i);
        maxz = max(maxz, *i);
        if (*i < *(i - 1)) is_sorted = false;
    }
    if (is_sorted) return;
    int diff = maxz - minz + 1;
    int numbuckets;
    if (r - l <= 1e7) numbuckets = 1500;
    else numbuckets = 3000;
    int range = (diff + numbuckets - 1) /
numbuckets;
    int* cnt = new int[numbuckets + 1];
    for (int i = 0; i <= numbuckets; i++)
        cnt[i] = 0;
    int cur = 0;
    for (int* i = l; i < r; i++) {
```

```

        temp[cur++] = *i;
        int ind = (*i - minz) / range;
        cnt[ind + 1]++;
    }
    int sz = 0;
    for (int i = 1; i <= numbuckets; i++)
        if (cnt[i]) sz++;
    int* run = new int[sz];
    cur = 0;
    for (int i = 1; i <= numbuckets; i++)
        if (cnt[i]) run[cur++] = i - 1;
    for (int i = 1; i <= numbuckets; i++)
        cnt[i] += cnt[i - 1];
    cur = 0;
    for (int *i = 1; i < r; i++) {
        int ind = (temp[cur] - minz) / range;
        *(1 + cnt[ind]) = temp[cur];
        cur++;
        cnt[ind]++;
    }
    for (int i = 0; i < sz; i++) {
        int r = run[i];
        if (r != 0) _newbucketsort(1 + cnt[r - 1],
1 + cnt[r], temp);
        else _newbucketsort(1, 1 + cnt[r], temp);
    }
    delete run;
    delete cnt;
}

void newbucketsort(int* l, int* r) {
    int *temp = new int[r - 1];
    _newbucketsort(l, r, temp);
    delete temp;
}

```

Bitlisaralash(Radix sort).

Shuningdek,

raqamlisaralashsifatidanilganalgoritmdir.

Busaralashningikkiversiyasimavjudbo‘lib,

ular,

qandaydir sonlarsistemasida (masalan, binar – ikkilik) sonniifodalashdan foydalanish fikridan boshqa umumiylikka ega emas.

Birinchi usuli. LSD (least significant digit, kichik muhim son). Har bir sonni binar shaklda ifodalaylik.

Algoritmning har bir qadamida sonlarni shunday tartiblaymizki, ular k doimiy bo'lgan birinchi k bitlar bo'yicha tartiblangin.

Buta'rifdani shunday xulosakilib chiqadiki,

har bir qadamda elementlarni yangi k

bitlar bo'yicha uzluksiz tartiblash kifoyadir.

Shumaqsadda,

sanash orqali saralashideldir (agarmuvaffaqiyatlisini doimiy tanlasangiz $2k$ xotiravavaqt ko'p kerak emas).

Murakkabligi $O(n)$,

agar sonlar fiksi rang katta likdada e'faraz qilsak va aksholda ikki sonni taqqo slash vaqt birligibo'yicha bajariladideb hisoblash mumkin bo'lmaydi.

Amalga oshirish juda oddiy.

8.13-dastur. Saralashni amalga oshirish.

```
int digit(int n, int k, int N, int M) {
    return (n >> (N * k) & (M - 1));
}
void _radixsort(int* l, int* r, int N) {
    int k = (32 + N - 1) / N;
    int M = 1 << N;
    int sz = r - l;
    int* b = new int[sz];
    int* c = new int[M];
    for (int i = 0; i < k; i++) {
        for (int j = 0; j < M; j++)
            c[j] = 0;
        for (int* j = l; j < r; j++)
            c[digit(*j, i, N, M)]++;
        for (int j = 1; j < M; j++)
            c[j] += c[j - 1];
        for (int* j = r - 1; j >= l; j--)
            b[--c[digit(*j, i, N, M)]] = *j;
        int cur = 0;
        for (int* j = l; j < r; j++)
            *j = b[cur++];
    }
    delete b;
    delete c;
}
```



```

}
void radixsort(int* l, int* r) {
    _radixsort(l, r, 8);
}

```

Ikkinchisuli. MSD (most significant digit, engmuhimraqam). Aslidaqandaydirbloknisaralashalgoritmigao‘xshaydi. Shublokdateng k bitlisonlarbo‘ladi. Murakkabligi LSD versiyasidagidek. Amalgaoshirishbloksaralashjudao‘xshaydi, lekinoddiy. Bu LSD versiyasiniamalgaoshirishdagibelgilanganraqamlivazifasidanfoydalanadi .

8.14-dastur. Saralashniamalgaoshirish.

```

void _radixsortmsd(int* l, int* r, int N, int d,
int* temp) {
    if (d == -1) return;
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int M = 1 << N;
    int* cnt = new int[M + 1];
    for (int i = 0; i <= M; i++)
        cnt[i] = 0;
    int cur = 0;
    for (int* i = l; i < r; i++) {
        temp[cur++] = *i;
        cnt[digit(*i, d, N, M) + 1]++;
    }
    int sz = 0;
    for (int i = 1; i <= M; i++)
        if (cnt[i]) sz++;
    int* run = new int[sz];
    cur = 0;
    for (int i = 1; i <= M; i++)
        if (cnt[i]) run[cur++] = i - 1;
    for (int i = 1; i <= M; i++)
        cnt[i] += cnt[i - 1];
    cur = 0;
    for (int *i = l; i < r; i++) {

```

```

        int ind = digit(temp[cur], d, N, M);
        *(l + cnt[ind]) = temp[cur];
        cur++;
        cnt[ind]++;
    }
    for (int i = 0; i < sz; i++) {
        int r = run[i];
        if (r != 0) _radixsortmsd(l + cnt[r - 1], l
+ cnt[r], N, d - 1, temp);
        else _radixsortmsd(l, l + cnt[r], N, d - 1,
temp);
    }
    delete run;
    delete cnt;
}
void radixsortmsd(int* l, int* r) {
    int* temp = new int[r - l];
    _radixsortmsd(l, r, 8, 3, temp);
    delete temp;
}

```

Bitoniksaralash

(Bitonic

sort).Massivelementlariniparalelsaralashalgoritmibo‘lib, tarmoqlardafoydalaniladi. Bualgoritmningg‘oyasishundaniboratki, massivBitonketma – ketlikka-avvalortib, keyinkamayibboradiganketma-ketlikkaaylanadi. Buniquyidagichasamaralitartiblashtirishmumkin: massivniikkiqismgaajratiladi, ikkitamassivyaratiladi, barchaelementlarniikkiqismdanharbirtegishlielementlarbirinchimassivga minimalvaikkinchisigamaksimaltengqo‘shiladi. IkkitaBitonketma-ketlikolinishi, ularningharbirinibirxilusuldarekursivravishdatartiblashmumkin, so‘ngraikkitamassivnibirlashtirishmumkin (birinchisiningharqaysielementiikkiinchisiningharqaysielementidankamy okitengbo‘lganligiuchun). MassiviniBitonketma-ketlikkaaylantirishuchunquyidagilarnibajarishlozim: agarmassivikkielementdaniboratbo‘lsa, unishunchakibajarishmumkin, aksholdamassivniikkigaajratib, algorithmnibo‘lakmassivlardanrekursivchaqiriladi, keyinbirinchiqisminitartibbilan, ikkinchisiniteskaritartibdasaralab, bir-birigabirlashtiriladi. Shubhasiz, natijadabitonalketma-ketlikniolanadi.

Murakkabligi $O(n \log^2 n)$, chunki Biton ketma-ketligini qurishda $O(n \log n)$ uchun ishlaydigan tartiblashlardan foydalandik va darajalarning umumiy soni $\log n$ edi. Bundan tashqari, massiv hajmi k ning darajasiga k arralib o'lishi kerak, shuningdek, elementlar bilan to'ldirish kerak bo'lishi mumkin.

8.15-dastur. Saralashni amalga oshirish.

```
void bitseqsort(int* l, int* r, bool inv) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    for (int *i = l, *j = m; i < m && j < r; i++, j++) {
        if (inv ^ (*i > *j)) swap(*i, *j);
    }
    bitseqsort(l, m, inv);
    bitseqsort(m, r, inv);
}

void makebitonic(int* l, int* r) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    makebitonic(l, m);
    bitseqsort(l, m, 0);
    makebitonic(m, r);
    bitseqsort(m, r, 1);
}

void bitonicsort(int* l, int* r) {
    int n = 1;
    int inf = *max(l, r) + 1;
    while (n < r - l) n *= 2;
    int* a = new int[n];
    int cur = 0;
    for (int *i = l; i < r; i++)
        a[cur++] = *i;
    while (cur < n) a[cur++] = inf;
    makebitonic(a, a + n);
    bitseqsort(a, a + n, 0);
    cur = 0;
    for (int *i = l; i < r; i++)
        *i = a[cur++];
}
```

```

    delete a;
}

```

Timsortsaralash (Timsort). Bugibridsaralashhisoblanadi, qo‘shishorqalisaralashvabirlashtirishorqalisaralashalgoritmlaridanfoydalanib yaratilgan.

Massivelementlarinibirnechakichikmassivlargaajratamizvaundagielementlarnitartiblanganholdamassivostigakengaytiramiz.

Tartiblanganmassivlarustidasamaraliishlashidanfoydalanib, massivostilarnijoylashtiribtartiblashyo‘liblantartiblaymiz.

Keyinchalikmassivostilarnitaxminantengkattalikdaolib, birilashtirish (merge) saralashdagikabibirlashtiramiz (aksholdasaralashvaqtikvadratgayaqinlashadi).

Buninguchunmassivostilarniinvariantsaqlab, stekkasaqlashqulay — tepadanuzoqlashgansarikattalashibboradivauchinchimassivostiningo‘lchamiularningkattaliklariyg‘indisidankattayokitengbo‘lgandaginayuqoriqismidamassivostilarnibirlashtiradi. murakkabligi $O(n)$

engyaxshiholatdava $O(n \log n)$ o‘rtachavaengyomonholatda.

Amalgaoshirisharzimasvaungaqt‘iyishonchyo‘q, lekinujudayaxshisaralashvaqtiko‘rsatdi.

8.16-dastur. Saralashniamalgaoshirish.

```

void _timsort(int* l, int* r, int* temp) {
    int sz = r - l;
    if (sz <= 64) {
        insertionsort(l, r);
        return;
    }
    int minrun = sz, f = 0;
    while (minrun >= 64) {
        f |= minrun & 1;
        minrun >>= 1;
    }
    minrun += f;
    int* cur = l;
    stack<pair<int, int*>> s;
    while (cur < r) {
        int* c1 = cur;
        while (c1 < r - 1 && *c1 <= *(c1 + 1))
            c1++;
        int* c2 = cur;

```

```

        while (c2 < r - 1 && *c2 >= *(c2 + 1))
c2++;
        if (c1 >= c2) {
            c1 = max(c1, cur + minrun - 1);
            c1 = min(c1, r - 1);
            insertionsort(cur, c1 + 1);
            s.push(make_pair(c1 - cur + 1, cur));
            cur = c1 + 1;
        }
        else {
            c2 = max(c2, cur + minrun - 1);
            c2 = min(c2, r - 1);
            reverse(cur, c2 + 1);
            insertionsort(cur, c2 + 1);
            s.push(make_pair(c2 - cur + 1, cur));
            cur = c2 + 1;
        }
        while (s.size() >= 3) {
            pair<int, int*> x = s.top();
            s.pop();
            pair<int, int*> y = s.top();
            s.pop();
            pair<int, int*> z = s.top();
            s.pop();
            if (z.first >= x.first + y.first &&
y.first >= x.first) {
                s.push(z);
                s.push(y);
                s.push(x);
                break;
            }
            else if (z.first >= x.first + y.first)
{
                merge(y.second, x.second, x.second
+ x.first, temp);
                s.push(z);
                s.push(make_pair(x.first + y.first,
y.second));

```

```

        }
        else {
            merge(z.second, y.second, y.second
+ y.first, temp);
            s.push(make_pair( z.first +
y.first, z.second ));
            s.push(x);
        }
    }
    while (s.size() != 1) {
        pair<int, int*> x = s.top();
        s.pop();
        pair<int, int*> y = s.top();
        s.pop();
        if (x.second < y.second) swap(x, y);
        merge(y.second, x.second, x.second +
x.first, temp);
        s.push(make_pair( y.first + x.first,
y.second ));
    }
}

void timsort(int* l, int* r) {
    int* temp = new int[r - l];
    _timsort(l, r, temp);
    delete temp;
}

```

Yuqorida 16
tasarlalshalgoritmlarni hosil qilish vatayyodasturiy fragmentlarikeltirildi.

Saralashalgoritmlaritaqqoslash. Buning uchun barcha yaratilgan saralashalgoritmlarini bir faylga joylashtiramiz.

Bularni taqqoslash uchun elementlarni o'z navbat bilan oshirib boramiz va natijalarini sonli, grafiklik keltiramiz.

8.1-jadval. Saralashalgoritmlar ro'yxati.

№	Saralashalgoritmning nomi	funksiyasi
1	Havosharchakabisaralash	void bubblesort(int* l, int* r)
2	Shakersaralash	void shakersort(int* l,

		int* r)
3	Taroqsimon caralash	void combsort(int* l, int* r)
4	Qo'shishorqalisaralash	void insertionsort(int* l, int* r)
5	Shell caralash	void shellsort(int* l, int* r) void shellsorthib(int* l, int* r) void shellsortpratt(int* l, int* r) void myshell_one(int* l, int* r) void myshell_two(int* l, int* r) void myshell_three(int* l, int* r)
6	Iearxiksaralash	void treesort(int* l, int* r)
7	Navsaralash	void gnomesort(int* l, int* r)
8	Tanlashorqalisaralash	void selectionsort(int* l, int* r)
9	Piramidakabisaralash	void heapsort(int* l, int* r)
10	Tezkorsaralash	void quicksort(int* l, int* r) void quickinssort(int* l, int* r)
11	Birlashtirishorqalisaralash	void mergesort(int* l, int* r) void mergeinssort(int* l, int* r)
12	Blokorqalisaralash	void newbucketsort(int* l, int* r)
13	Bitlisaralash (LSD)	void radixsort(int* l, int* r)
14	Bitlisaralash (MSD)	void radixsortmsd(int* l,

		int* r)
15	Bitoniksaralash	void bitonicsort(int* l, int* r)
16	Timsortsaralash	void timsort(int* l, int* r)

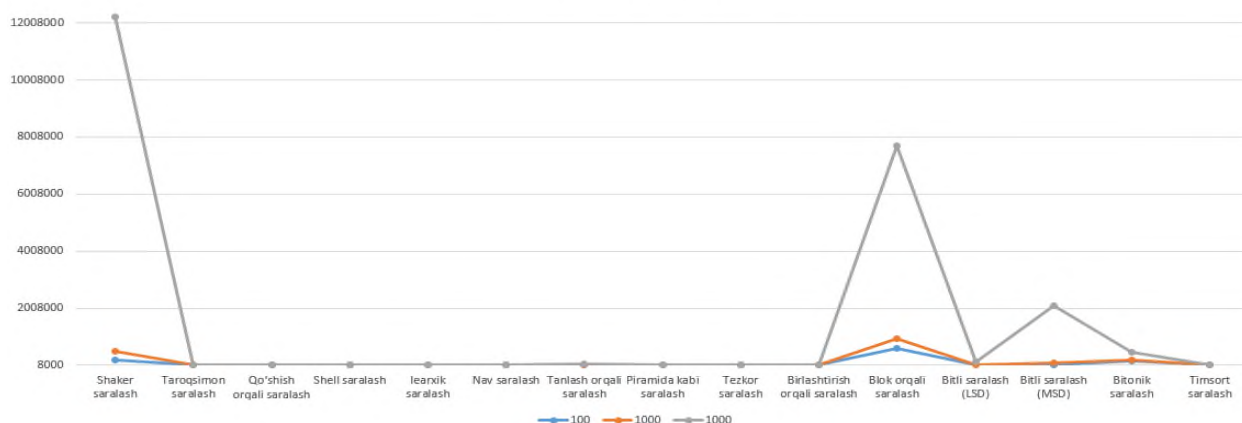
Bualgoritmlarni vaqt bo'yicha taqqoslash uchun tasodifiy sonlar bilan massivni to'ldiramiz, tasodifiy sonlarni hamguruhlarga ajratamiz, xonalar bo'yicha, massiv elementlar sonini ham 100 tadan 100000 tagacha tanlab olamiz.

Birinchitest. Massiv elementlar soni 100, 1000, 10000 ta tasodifiy sonlarning eng kattasi 1000:

Elementlar soni: 100 Random eng katta soni: 1000 bubblesort(a,b) => 189949 shakersort(a,b) => 9912 combsort(a,b) => 10017 insertionsort(a,b) => 9962 shellsort(a,b) => 9996 shellsort Hib(a,b) => 9983 shellsort Pratt(a,b) => 10001 myshell_one(a,b) => 10013 myshell_two(a,b) => 10000 myshell_three(a,b) => 9983 treesort(a,b) => 600071	Elementlar soni: 1000 Random eng katta soni: 1000 bubblesort(a,b) => 509917 shakersort(a,b) => 8523 combsort(a,b) => 9949 insertionsort(a,b) => 12012 shellsort(a,b) => 12536 shellsort Hib(a,b) => 9983 shellsort Pratt(a,b) => 10039 myshell_one(a,b) => 10026 myshell_two(a,b) => 9951 myshell_three(a,b) => 10056 treesort(a,b) => 940049	Elementlar soni: 10000 Random eng katta soni: 1000 bubblesort(a,b) => 12229978 shakersort(a,b) => 10000 combsort(a,b) => 19984 insertionsort(a,b) => 25365 shellsort(a,b) => 20009 shellsort Hib(a,b) => 10026 shellsort Pratt(a,b) => 40028 myshell_one(a,b) => 10044 myshell_two(a,b) => 9996 myshell_three(a,b) => 20044 treesort(a,b) => 7690211

gnomesort(a,b) =>6510	gnomesort(a,b) => 10044	gnomesort(a,b) =>120041
selectionsort(a,b) => 29963	selectionsort(a,b) => 79987	selectionsort(a,b) => 2090445
heapsort(a,b) => 160055	heapsort(a,b) => 190026	heapsort(a,b) => 460513
quicksort(a,b) => 10013	quicksort(a,b) => 10001	quicksort(a,b) => 10462
quickinssort(a,b)) => 9979	quickinssort(a,b)) => 9992	quickinssort(a,b)) =>10020
mergesort(a,b) => 10120	mergesort(a,b) => 20005	mergesort(a,b) => 29924
mergeinssort(a,b)) => 9975	mergeinssort(a,b)) => 10004	mergeinssort(a,b)) => 9979
newbucketsort(a,b) => 20014	newbucketsort(a,b) => 19967	newbucketsort(a,b) => 19959
radixsort(a,b) => 10060	radixsort(a,b) => 30057	radixsort(a,b) => 19980
radixsortmsd(a,b)) => 19992	radixsortmsd(a,b)) => 20031	radixsortmsd(a,b)) => 20249
bitonicsort(a,b) => 29988	bitonicsort(a,b) => 69974	bitonicsort(a,b) => 220502
timsort(a,b) => 149973	timsort(a,b) => 190021	timsort(a,b) => 140443

Saralash algoritmlarini taqqoslash
(massivdagi eng katta son 1000)



8.1-rasm. Saralashalgoritmlarinitaqqoslash.

UshburasmdakeltirilgantaqqoslashnatijarigaShaker, bitli blokorqali, (MSD),

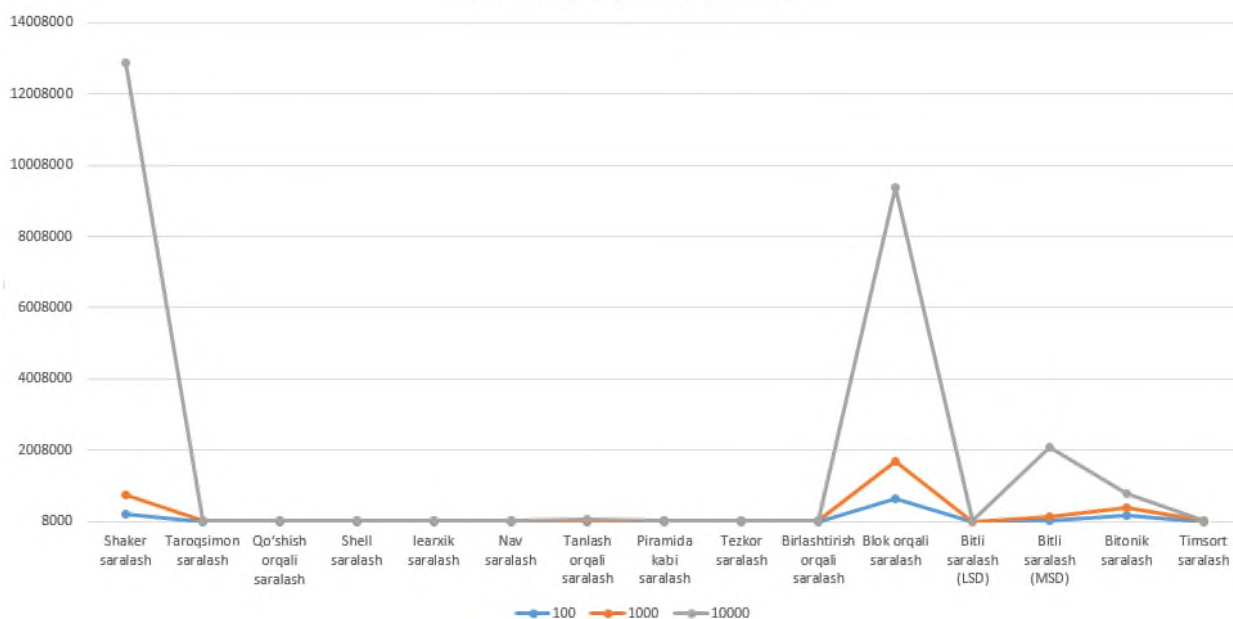
bitoniksaralashalgoritmlariningelementlarningsoninio‘zgarishbilansarala shuchunsarflanadiganvaqtnio‘zgarishinikuzatishimizmumkin.

Ikkinchitest.Massivelementlarisoni 100, 1000, 10000 tavatasodifiysonlarningengkattasi 10000:

Elementlar soni:100 Random eng katta soni:10000	Elementlar soni:1000 Random eng katta soni:10000	Elementlar soni:10000 Random eng katta soni:10000
bubblesort(a,b) => 230011	bubblesort(a,b) => 509982	bubblesort(a,b) => 12139952
shakersort(a,b) => 9889	shakersort(a,b) => 9956	shakersort(a,b) => 10056
combsort(a,b) => 10018	combsort(a,b) => 10009	combsort(a,b) => 20001
insertionsort(a, b) => 9562	insertionsort(a, b) => 9986	insertionsort(a, b) => 10009
shellsort(a,b) => 10018	shellsort(a,b) => 9992	shellsort(a,b) => 20001
shellsorthib(a,b) => 10000	shellsorthib(a,b) => 10021	shellsorthib(a,b) => 20018
shellsortpratt(a ,b) => 8452	shellsortpratt(a ,b) => 9992	shellsortpratt(a ,b) => 40006
myshell_one(a,b) => 10022	myshell_one(a,b) => 10025	myshell_one(a,b) => 10017
myshell_two(a,b) => 9988	myshell_two(a,b) => 10360	myshell_two(a,b) => 10005
myshell_three(a, b) => 12001	myshell_three(a, b) => 15201	myshell_three(a, b) => 20018
treesort(a,b) => 660039	treesort(a,b) => 1030011	treesort(a,b) => 7700494
gnomesort(a,b) => 7563	gnomesort(a,b) => 8624	gnomesort(a,b) => 11254
selectionsort(a, b) => 29997	selectionsort(a, b) => 99992	selectionsort(a, b) => 1950542
heapsort(a,b) => 170016	heapsort(a,b) => 210079	heapsort(a,b) => 400520
quicksort(a,b)	quicksort(a,b)	quicksort(a,b)

=> 9562 quickinssort(a,b)) => 9996 mergesort(a,b) => 10069 mergeinssort(a,b)) => 8996 newbucketsort(a,b) => 20060 radixsort(a,b) => 10005 radixsortmsd(a,b)) => 10013 bitonicsort(a,b) => 30014 timsort(a,b) => 179974	=> 10009 quickinssort(a,b)) => 10005 mergesort(a,b) => 10000 mergeinssort(a,b)) => 9996 newbucketsort(a,b) => 20040 radixsort(a,b) => 10018 radixsortmsd(a,b)) => 9954 bitonicsort(a,b) => 59934 timsort(a,b) => 150012	=> 10210 quickinssort(a,b)) => 10004 mergesort(a,b) => 30318 mergeinssort(a,b)) => 20502 newbucketsort(a,b) => 19954 radixsort(a,b) => 30075 radixsortmsd(a,b)) => 29680 bitonicsort(a,b) => 229981 timsort(a,b) => 149973

Saralash algoritmlarini taqqoslash
(massivdagi eng katta son 10000)



8.2-rasm. Saralash algoritmlarini taqqoslash.

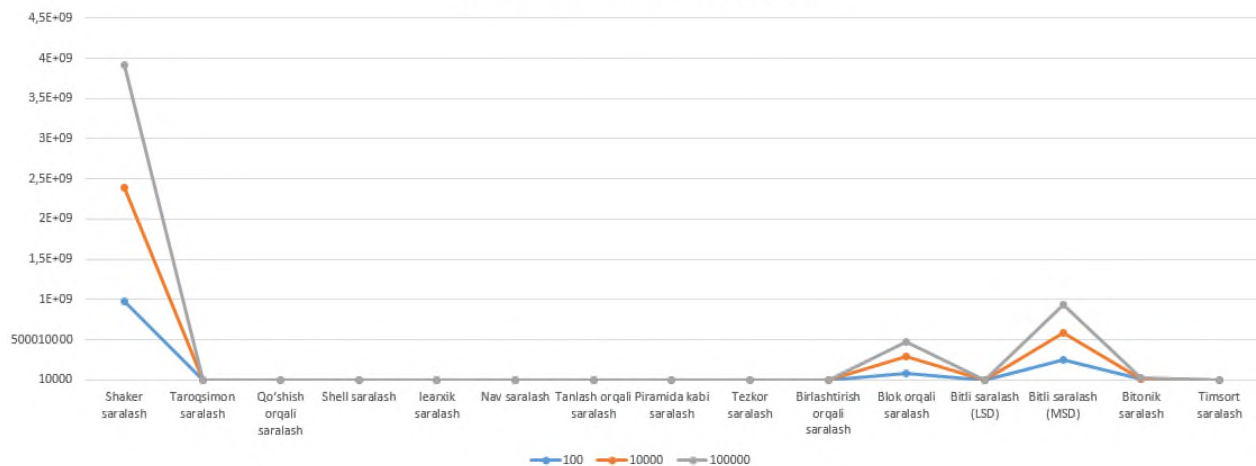
Ushbu rasmidagi keltirilgan taqqoslash natijalariga Shaker, blok orqali, bitli (MSD), bitonik saralash algoritmlarining elementlarning sonini o'zgarish bilan saralash uchun sarflanadigan vaqtning o'zgarishini kuzatishimiz mumkin.

Uchinchi test. Massiv elementlar soni 100000
tavatasodifiy sonlarning eng kattasi 100, 10000, 100000:

Elementlar soni:100000 Random eng katta soni:100 bubblesort(a,b) => 977222383 shakersort(a,b) => 10035 combsort(a,b) => 190531 insertionsort(a, b) => 10514 shellsort(a,b) => 110523 shellsorthib(a,b) => 119988 shellsortpratt(a ,b) => 439987 myshell_one(a,b) => 59994 myshell_two(a,b) => 69969 myshell_three(a, b) => 60028 treesort(a,b) => 81039998 gnomesort(a,b) => 10299 selectionsort(a, b) => 249475206 heapsort(a,b) => 7190050 quicksort(a,b) => 340165 quicksort(a,b) => 209933 mergesort(a,b)	Elementlar soni:100000 Random eng katta soni:10000 bubblesort(a,b) => 1413069355 shakersort(a,b) => 20103 combsort(a,b) => 589990 insertionsort(a, b) => 30142 shellsort(a,b) => 340080 shellsorthib(a,b) => 299988 shellsortpratt(a ,b) => 1020010 myshell_one(a,b) => 150166 myshell_two(a,b) => 159905 myshell_three(a, b) => 140058 treesort(a,b) => 212009993 gnomesort(a,b) => 20014 selectionsort(a, b) => 338029149 heapsort(a,b) => 6460020 quicksort(a,b) => 229943 quicksort(a,b) => 110074 mergesort(a,b)	Elementlar soni:100000 Random eng katta soni:100000 bubblesort(a,b) => 1522791244 shakersort(a,b) => 10022 combsort(a,b) => 420097 insertionsort(a, b) => 20130 shellsort(a,b) => 240105 shellsorthib(a,b) => 239879 shellsortpratt(a ,b) => 950139 myshell_one(a,b) => 140276 myshell_two(a,b) => 150114 myshell_three(a, b) => 139913 treesort(a,b) => 173539873 gnomesort(a,b) => 29920 selectionsort(a, b) => 346130116 heapsort(a,b) => 7080178 quicksort(a,b) => 200168 quicksort(a,b) => 110095 mergesort(a,b)

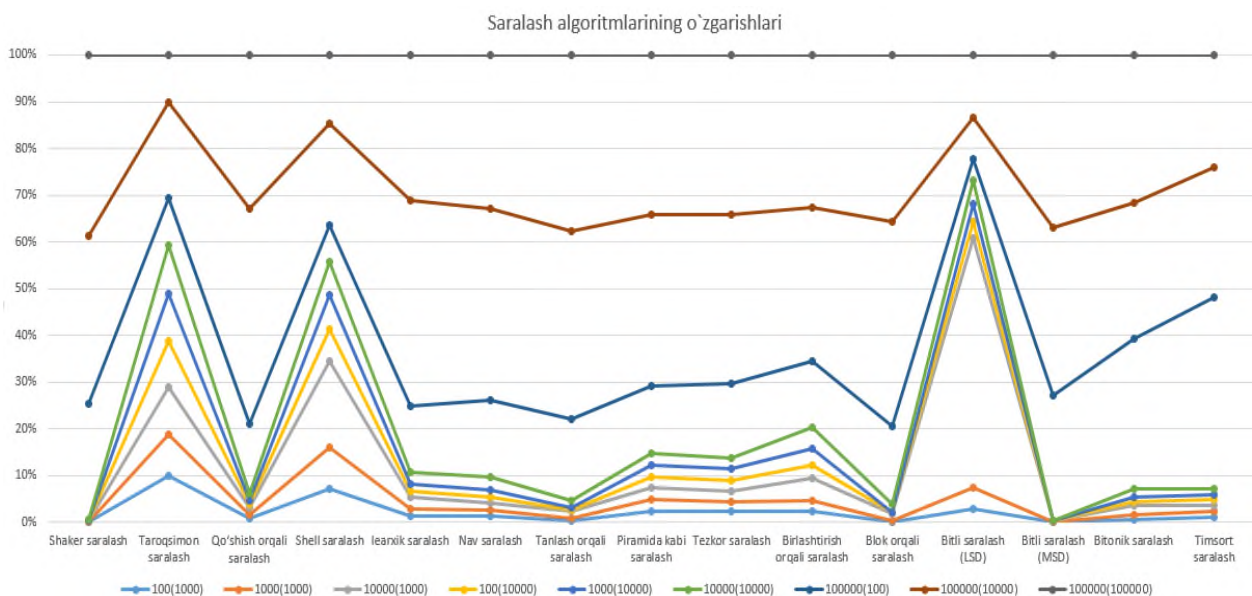
=> 439970 mergeinssort(a,b)) => 259845 newbucketsort(a, b) => 119831 radixsort(a,b) => 349931 radixsortmsd(a,b) => 320020 bitonicsort(a,b) => 4389857 timsort(a,b) => 469881	=> 520012 mergeinssort(a,b) => 340191 newbucketsort(a, b) => 110005 radixsort(a,b) => 400045 radixsortmsd(a,b) => 429952 bitonicsort(a,b) => 5839897 timsort(a,b) => 450090	=> 500015 mergeinssort(a,b) => 289851 newbucketsort(a, b) => 150140 radixsort(a,b) => 349846 radixsortmsd(a,b) => 570139 bitonicsort(a,b) => 5079891 timsort(a,b) => 360411

Saralash algoritmlarini taqqoslash
(massiv elementlar soni 100000)



8.3-rasm. Saralashalgoritmlarinitaqqoslash.

UshburasmdakeltirilgantaqqoslashnatijarigaShaker, blokorqali, bitli (MSD) saralashalgoritmlariningelementlarningqiymatinio‘zgartirgandasaralashu chunsarflanadiganvaqtnio‘zgarishinikuzatishimizmumkin.



8.4-rasm. Saralashalgoritmlarini o'zgartirish.

Ushbu tahlil grafikda algoritmlarning massiv elementlar va qiymatlarini o'zgartirganimizda o'zlarini qanday tutishlarifoizmi qandoridan o'zaroni batlari keltirilgan. Masalan, tanlash orqali saralash o'zini juda yaxshi tutganligini ko'rishimiz mumkin, ya'ni o'zgarishlarda ham uning uchun sarflanadigan vaqt uzluksiz bo'lgan.

Bu algoritmlarning o'rganishdan maqsad, ular asosida murakkablik kichik, ishlash vaqtini kambo'lgan saralash algoritmlarini yaratish, masalaning mohiyatidan kelib chiqqan holda qachon va qanday qilib foydalanishni belgilab olishimiz mumkin.

Judaham kattasonlar bilan ishlash. Kattasonlar bilan ishlash uchun asosiy tip va uning xotiradan necha bayt egallashini bilish muhim hisoblanadi. Tiplarni yaxshi bilish kerak.

Ammo C++ tilining rivojlanib ketgan variantlaridagi tiplarni mavjud. Tiplardan foydalanishga birmisol keltiramiz. Masalan, byte tipi bo'lsin, xotiradan 1 bayt egallaydigan. Bu tipga $a=200$, $b=100$ bo'lsan, a 44 bo'ladi. Qanday? Buning uchun avvalo byte ning diapozonini, ya'ni $[0, 255]$ bilish shart. Agar tipning eng yuqori qiymati a va b bo'lsa, a 1 ta 0 ni ketmaket qo'yilsa 300 chiqib keladi $(255 + 44 + 1(0)) = 300$. Har qanday tipning diapozonini doirasida hisoblashlarni bajaradi.

Avvalo C++ standartlariga mos tiplarni keltiramiz.

8.2-jadval. Tiplar va ularning xususiyatlari.

No	Tip nomi	bayt	Ikkinchi nomi	Diapozoni
1	int	4	signed	$[-2\ 147\ 483\ 648, 2\ 147\ 483\ 647]$

Nº	Tipnomi	bayt	Ikkinchinomi	Diapozoni
2	Unsigned int	4	Unsigned	[0, 4 294 967 295]
3	__int8	1	char	[-128, 127]
4	unsigned __int8	1	Unsigned char	[0, 255]
5	__int16	2	Short	[-32 768, 32 767]
6	unsigned __int16	2	UnsignedShort	[0, 65 535]
7	__int32	4		[-2 147 483 648, 2 147 483 647]
8	unsigned __int32	4		[0, 4 294 967 295]
9	__int64	8	Long	[-9 223 372 036 854 775 808, 9 223 372 036 854 775 807]
10	unsigned __int64	8		[0, 18 446 744 073 709 551 615]
11	bool	1		[false] yoki [true]
12	char	1		[-128,127]
13	char	1		[-128 , 127]
14	unsigned char	1		[0, 255]
15	short	2		[-32 768, 32 767]
16	unsigned short	2	unsigned short int	[0, 65 535]
17	long	4	long int	[-2 147 483 648, 2 147 483 647]
18	unsigned long	4	unsigned long int	[0, 4 294 967 295]
19	long long	8	__int64	[-9 223 372 036 854 775 808, 9 223 372 036 854 775 807]
20	unsigned long long	8	__int64	[0, 18 446 744 073 709 551 615]
21	enum	null		
22	float	4		[3,4E +/- 38 (7 belgi)]
23	double	8		[1,7E +/- 308 (15 belgi)]
24	long double	8		[1,7E +/- 308 (15 belgi)]
25	wchar_t	2	__wchar_t	[0, 65 535]

Texnikaining rivojlanishi dasturlashtillarida keng yondashuvlarni amalga oshirishga sabab bo'lmog'da. Masalan, tiplarning diapazonini bilish uchun ikki darajasigacha raliqilib tuzib chiqmoqdalar. Bujuda oson. Buyuqoridagi jadvaldan ham ko'rish mumkin.

Tiplarning ichida eng kattasi bu long tipidir, uning ikki kilanganlari long long eng kattasi bo'lishi mumkin, ammo bulardan foydalanish uchun __int64 tipi ham yaratilgan ki bulardan foydalanish qulaydir. Yuqoridagi jadvaldan bu ko'rishimiz ham, o'zimiz dasturiyozib tekshirishimiz ham mumkin. Odatda dasturchilar quyidagicha fragmentlardan foydalanishadi:

```
#define LL long long
#define uLL unsigned long long
//...
uLL b = 0;
```

Ammo bunday kattasonlar bilan ishlashni C++ standartining o'ziga qo'llab quvvatlamaydi. Bunga ikki tamisol keltiramiz.

8.17-dastur. Kattasonlarche garasini tekshirish.

```
int main()
{
    cout << ULLONG_MAX << endl;
    __int64 a, b = 0; // 20 xona
    cin >> a >> b;
    if ((a+b) > ULLONG_MAX)
        cout << "Chegaradan chiqdi" << endl;
    else
        cout << "Natija:" << a+b << endl;

    system("pause");
    return 0;
}
```

Agar dasturga 20 xona lisonni (< ULLONG_MAX) kiritsangiz xato ishlaydi. Nima sababdan, sanoq sistemaning xatosini ko'rish mumkin. Shuningdek, ikkinchi 64 darajasine chagateng? Albatta ULLONG_MAX gatengdir. Endi yanabir gibrid tip kiritamiz, long double vabuo r qali 2^{64} darajasini hisoblaymiz.

```
#define LD long double
// ..
cout << ULLONG_MAX << endl;
```



```
LD f = powl(2,64);
//cout << f << endl;
printf("%.0Lf\n",f);
```

Dasturfragmentini ishlatib, natijaning oxirgi 4 taxonasi ga e'tibor bering.

Tiplar kombinatsiyalarini yordamida kattasonlar xatolibo'lganligi uchun amaliy matematikaning turlimasallarini echganda aniq nechtaxonani aniqligi oldin dan belgilab olinadi. Masalan, quyidagi dasturga qarang.

8.18-dastur. Tiplarga bog'liq kattasonlardan foydalanish

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <math.h>
using namespace std;
int main(){
    double n=0,a=0;
    cin>>a>>n;
    vector<double>arr;
    arr.push_back(pow(a,n));
    for(int i=0; i<arr.size(); i++){
        printf("%.0Lf\n", arr[i]);
    }
    system("pause");
    return 0;
}
```

8.18-dastur. Output

```
1- test:{
son kirit: 2
darajasini kirit:50
1125899906842624
}
2- test:{
son kirit: 2
darajasini kirit:64
18446744073709552000
}
3- test:{
son kirit: 5
```

[illegible]

}

Dasturixtiyoriy sonningixtiyoriy darajasini hisoblash uchun mo'ljallangan, ammo hammasining oxirgi raqamlariga e'tibor bersangiz, 0
lar bilan to'ldirilgan.

Bu esayana bir bor hisoblash dasanoq sistemasining xatoligini ko'rsatadi.

Xulosashundan iborat bo'ladiki, 20
xonaga chaya xshixisoblash amalga oshirish mumkin,
ammo bu xonadan boshlab xatoliklarni nazorat qilib bo'lmaydi.

Kattasonlar bilan ishlaganda,
yondashuvni boshqacha amalga oshirish mumkin,
masalan oqim dankiruvchiraqamlarni massiv elementlar deb qarab uni ustida
mustaqil arifmetik amallarni bajarish mumkin.

Har bir arifmetik amalni oldindan
C++ da amalga oshirish dastur fragmentini yozib, Avvalo,
alohid ko'rib chiqish mumkin.
cheksiz uzoq raqamni faqat dinamik massiv sifatida ifodalash mumkinligini
shunishingiz kerak.

Lekin raqamlar dinamik massiv sifatida taqdim etilsa ham,
ba'zicheklovlar bo'ladi. Masalan, bunday sonning (massivning)
uzunligi kompyuter xotirasini qandor bilan chegaralangan bo'ladi.

Shuni ham tushunish kerakki,
qo'shish va ayirish amallaridan foydalanishdan natija operandlarga qaraganda
kompyuter xotirasida ko'proq joy egallaydi.

Kattasonlarning qo'shish. Arifmetikada ishlatiladigan arifmetik qo's
hish amalini ko'rib chiqamiz.

Ushbu oddiy arifmetik amal uchun algoritmi hayratlanarli darajada oddiydir,
dastur fragmentiga qarang:

```
// birinchi g'oya
if (size_a > size_b)
    length = size_a + 1;
else
    length = size_b + 1;
// ikkinchi g'oya
for (int ix = 0; ix < length; ix++)
{
    b[ix] += a[ix];
    b[ix + 1] += (b[ix] / 10);
    b[ix] %= 10;
}
```

```
// uchinchi g'oya
if (b[length - 1] == 0)
length--;
```

Dasturfragmentida qo'shadigan raqamlar n taxminan a va b massivlarida yoziladi. Raqamlar uzunligi size_a va size_b o'zgaruvchilarda saqlanadi, lekin har qanday bosh qo'zg'aruvchilardan foydalanishingiz mumkin. Birinchi g'oyadan imma uchun birif shart operatoribor, nima uchun ubuyerda? – degan savoltug'ililadi.

Bu fragment blokida yig'indini natijasida olinadigan sonning maksimal uzunligini aniqlaymiz. Axir, qo'shiladigan raqamlar turli uzunlikda, birikattava ikkinchi siki kichikroq va har bir raqam gamoskelishi uchun xotira ajratishimiz kerak.

Algoritm, matematik adarslarida o'rgatilgan usul bo'yicha amalga oshiriladi: birinchi oxiridaboshlab, eng kichik razriyaddan raqam kiritish, natijada summani ajratish va darhol keyingiraqamga qo'shish va o'ngabo'ling andabutunqismini olish.

Natijasida sonining birinchi razriyadivakeyingira zryaduchun qo'shiladigan son bo'ladi, albatta, bor bo'lsa. Asosiynarsasoni b massivga saqlanadivau oxiridachi qishilozim.

Bunda massivni razryadlarini tenglashtirish kerakligini unutmang.

Yuqoridagi dastur fragmentiga o'zg'oyalarimizni qo'shib dasturni keltirami z:

8.19-dastur. Katta sonlarni qo'shish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;
int main(){
    string a, b, a1, b1;
    cin >> a >> b;
    int leng = 0;
    // birinchi g'oya
    if (a.length() > b.length())
        leng = a.length()+1;
    else
        leng = b.length()+1;
```

```

// dasturchining g'oyasi
int k = 0;
while(k < leng){
    if( k < leng - a.length() ) a1 += '0';
    if( k < leng - b.length() ) b1 += '0';
    k++;
}
a1 += a;
b1 += b;
// ikkinchi g'oya
int temp = 0;
for (int ix = leng-1; ix >= 0 ; ix--)
{
    int temp_a = a1[ix] - 48;
    int temp_b = b1[ix] - 48; temp_b = temp_b
+ temp;
    temp = 0;
    temp_b += temp_a;
    temp += (temp_b / 10);
    temp_b %= 10;
    a1[ix] = (char)temp_a + 48;
    b1[ix] = (char)temp_b + 48;
}
// uchinchi g'oya
if(b1[0] == '0') {
    b1.erase(0,1);
}
cout << b1 << endl;
system("pause");
return 0;
}

```

8.19-dastur. Output

```

a=984654313156311315315464876546487946456465431311132
15318653186316632158632183218693168316853541867931484
3164131654315313315315365315315331563131831853
b=546464546464654654654555464546546546546546546546
54644654654646431543186464564879876489794648994648979

```

4897979874897979879797987489648364584364854684
natija=1531118859620965969970020341093034493003011977
85767869963307840963063701818647783573044806648190862
5804638062111529213293195113352804963696147496686537

Dasturnitahlilqilingvaamaldasinabko‘ring.

Kattasonlarniayirish. Ikkinchiengko‘pishlatiladiganarifmetikamal
- ayirishdir.

Endikattasonlarniayirishalgoritmlariniyozishnio‘rganishgaqaraymiz.

Sonlarimiz a va b massivlardasaqlanadidebfarazqilamiz,
mosravishda m va n busonlarninguzunliklaribo‘lsin.
Raqamlarnirazryadlarinitenglashtirishniunutmang (yuqoridagidek).
Albatta, qaysisonkattaekanliginibilsak, vazifasoddalashadi.
Lekinbunibilmasligimizmumkin.

Keyinavvalsonlarningqaysibirikattaekanliginitopishimizkerak.

Olinganraqamningbelgisinianiqlashuchunbukerak.

Boshqachaqilibaytganda, agarbirinchirraqamikkinchisidankambo‘lsa,
javobishorali (-) belgisiniko‘rsatadi. Shundayqilib,
algoritmningbirinchiqisminiyoziishniboshlaymiz,
ya‘nikattasonnianiqlaymiz. Algoritmshundayko‘rinadi:

```
int k = 3; // agar k == 3 sonlar bir xil
length = size_a;
if (size_a > size_b){
    length = size_a;
    k = 1; // agar k == 1 birinchi son
    ikkinchisidan uzun
}
else
    if (size_b > size_a) {
        length = size_b;
        k = 2; // agar k == 2 ikkinchi son
        birinchisidan uzun
    }
    else // agar ular teng bo‘lsa, kattasini
    aniqlash lozim
        for (int ix = 0; ix < length;) //
        razryad bo‘yicha tekshirish
        {
            if (a[ix] > b[ix]) // birinchi
            sonni razryadi katta bo‘lsa
```

```

        {
            k = 1; // birinchi son
ikkinchisidan katta
            break; // chiqish
        }

        if(b[ix] > a[ix]) // ikkinchi sonni
razryadi katta bo'lsa
        {
            k = 2; // ikkinchi son
birinchisidan katta
            break; // chiqish
        }
    }

```

Endiyozganlarimnitushuntiraman. Avval k o'zgaruvchi 3 gao'rnatilganiniko'rishmumkin. Algoritmningbuqismida k o'zgaruvchitekshirishnatijasihisoblanadi. Agarsonlartengbo'lsa, k=3 gatengbo'libqoladi, agarbirinchisiikkinchisidankattabo'lsa, k= 1 qiymatini, ikkinchisibirinchisidankattabo'lsa, k= 2 qiymatinioladi. Length o'zgaruvchikattasonuzunliginingqiymatinioladi. Endibualgoritmningsamaradorliginiasoslashgao'tamiz.

Sonlarnitaqqoslashikkibosqichdaamalgaoshadi. Birinchidan, raqamlaruzunliklarinitaqqoslaymiz: qaysiraqamuzunroqligi, sonlاربirxiluzunlikdabo'lsa, uholdataqqoslashgao'tamiz. Sonlarnitartibbilantaqqoslashniengkattarazryadlardanboshlaymiz, shuninguchunsonningkattasinianiqlaymiz.

Buesabirinchiqismningmohiyativamurakkabligibildiradi.

Endialgoritmnigikkinchiqismi — ayirmagao'tamizvauquyidagicha:

```

int difference (int *x, int *y, int *z, int
length){
    for (int ix = 0; ix < (length - 1); ix++){
        if (ix < (length - 1)){
            x[ix + 1]--;
            z[ix] += 10 + x[ix];
        } else
            z[ix] += x[ix];
        z[ix] -= y[ix];
        if (z[ix] / 10 > 0){
            z[ix + 1]++;

```

```

        z[ix] %= 10;
    }
}
return 0;
}

```

Ayirmaning o'zi uchun funksiyani yozish qulay, chunki unda ikki hol uchun ikki ta algoritmni yozish shart emas, birinchi son ikkinchisidan kattabo'lganda va aksincha massiv kattason, y massiv kichikson va z massiv natijahisoblanadi. Algoritm juda oddiy: har bir raqam uchun eng yuqori raqamdan ayirishni hisobga olgan holda 10 ni qo'shamiz. Buraqamlar ayirmasini soddalashtirish uchun -1 amalga oshiriladi.

Bu amalfaqat massivning oxirgi raqariyadibo'lsa amalga oshiriladi (soni birinchi uchun). Sonlarni ayirganda son o'ng z massivida gison hosil bo'ladi. Javob z massiviga teskari usulida yoziladi. Protse durani quyidagicha qirish kerak:

```

if (k == 1) difference(a,b,c, length); // birinchi
son ikkinchisidan katta
if (k == 2) difference(b,a,c, length); // ikkinchi
son birinchisidan katta

```

Endi javob shuteskaritartibda massiv dasaqlanadi.

Bu esa kattasonlarni ayirishni amalga oshirishni hal qilib beradi.

Kattasonlarni ko'paytirish. Bu algoritm masalalarni yechishda oldin gi ikki tasiga qaraganda keng tarqalgan.

Bevosita algoritmning o'zi gao'taylik.

E'tiboringizga algoritmni taqdim etaman:

```

length = size_a + size_b + 1;
for (int ix = 0; ix < size_a; ix++)
    for (int jx = 0; jx < size_b; jx++)
        c[ix + jx + 1] += a[ix] * b[jx];
for (int ix = 0; ix < length; ix++){
    c[ix + 1] += c[ix] / 10;
    c[ix] %= 10;
}
while (c[length] != 0)
    length++;

```

Bu algoritmni quyidagicha ko'rib chiqamiz va qanday ishlashini aniqlash uchun qushunishga harakat qilaylik. Avvaliga a va b massivlari bir xil teskarishaklda ikki ta son bor. Sonlari uzunliklari size_a va

size_b o'zgaruvchilardasaqlanadi. length
o'zgaruvchinatijaviysonuzunliginianiqlaydi.
Uberilgansonlaruzunliklariningyig'indisigayokibuyig'indidanbittagakatt
atengbo'ladi.
Lekinolingansonninganiquzunliginibilmaganimizuchunuzunroquzunlikn
i, ya'niikkinchivariantniolamiz. Endiqiyinbo'lmaganhisob-
kitoblardanso'ng, raqamlarniko'paytiraylik.
Maktabdao'rganganizmizbo'yichaularniko'paytiramiz. Buninguchun,
ikkitakrorlanishishlatamiz, biri size_a uchun, vaboshqa size_b uchun.
Shutufaylimassivdagisonniyozishdamassivningharbiryacheykasidaolinga
nsonningbirraqaminiolamiz.
Oxirgitakrorlanishnatijasidasonininganiquzunliginitopishuchunzarurbo'l
gansonitaxminuzunligihaqiqiyqismidankattabo'lishimumkin.
Javobsmassivdasaqlanadi.
Dasturlashtilidaamalgaoshirilgandaunitushunishosonroq.
Judakattaraqamlarniko'paytirishimkerak, algoritmiyordamida,
lekino'zgaruvchilartiplarinimoslashtiraolmaydi.
Bucheklovniqandaybartarafetishmumkinliginitipgaqarabtanlabolishmum
kin.

```
long int Multi(int A, int B) {
    string Astring = to_string(A);
    string Bstring = to_string(B);

    unsigned long int a, b, c, d, sizeA =
Astring.size(), sizeB = Bstring.size();

    if (sizeA == 1 || sizeB == 1) return A * B;

    unsigned long int n = max(sizeA, sizeB);

    a = A / static_cast<long int>(pow(10, n / 2));
    b = A % static_cast<long int>(pow(10, n / 2));
    c = B / static_cast<long int>(pow(10, n / 2));
    d = B % static_cast<long int>(pow(10, n / 2));

    unsigned long int p1 = Multi(a, c);
    unsigned long int p2 = Multi(b, d);
    unsigned long int p3 = Multi(a, d) + Multi(b,
c);
```

```

        return pow(10, n) * p1 + pow(10, n / 2) * p3 +
p2;
}

```

Buhechkingasiremas, butunsonlarmaksimalvaminimalqiymatlaribo'yichachegasigaega, turliarxitekturalardaturlisha. Masalan, int tipidagibutunsonuchununingqiymatlariorali*g'i* -2147483646 dan 2147483647 gacha. Harbiryo'nalishda 2 milliardbutunsonbo'libtuyuladi, lekinhaqiqiykriptografiyayokimashinalio'qitish, ehtimolliknazariyasiyokizamonaviymatematikagakirganingizdanson'ng, bujudakichikekanliginitushunasiz. Buholatlarda, kattaraqamlarbilanishlashsinflaridanfoydalanishmumkin.

Large Integer
sinfiningqo'llanilishi. Ko'psonlihisoblashlarstandartma'lumotturlarining cheklanishidantashqarigachiqish, cheksizkattasonlarbilanishlash, hajmifaqatmashinaninghisoblashkuchibilancheklanadi.

Bungaqandayerishishmumkin?

Engmantiqiyyo'limassivyozihsivastandarttipraqamlarizchilmassivuchun maxsustarzdaaylantirish.

Misoluchun, 123456789123456789
raqaminiqandaysaqlashingizmumkin, u int tipigamoskelmaydi. Uniqo'yishuchun, int tipidagimassivolishmumkin, arr[0] = 123456, arr[1] = 789123, arr[2] = 456789, bundayyozihsnimaxsususullaridebbo'ladi. Shuningdek, to'g'rikiritilsa, qo'shish, ayirish, ko'paytirishvabo'lishamallariniyozihsimumkin.

Endimaxsus Bignumber
sinfiniamalgaoshirishniboshlaymizvaqo'shish, ayirishuchun. Businfbilanishlashdaizohlarbilandasturfragmentinito'liqkeltirilgan. Dasturfragmentigaqarabizohlarnio'qishingizmumkin.

Kattasonniinitsializasiyaqilish. Quyidagichakiritishishlarisifatida Bignumber(string str) sinfkonstruktorimavjud. Satrningoxiridanboshlabjoriyuzunlikdagi substringlartanlanadi, raqamlargaaylantiriladivaqismlarvektorigaberiladi.

Satrdamanfiyishorabor-

yo'qligigaqarabsinfningsboshqaxususiysohasiinitsializatsiyaqilinadi.

Ba'zannormallashtirishdaziddiyatlarbo'lmasligiuchunsinfningsbarchaob'ektlariuchunbirxilbo'lganstatikdoimiydonishlatiladi.

Kattasonlarniqo'shish. + operatoriniqaytayuklaymiz, buesao'znavbatidaikkita kattasonniqo'shadigan `_plus()` usulinichaqiradi. Bundantashqari, harikkiraqamlarqismlarini, massivbirxilhajmliqilishkerak. Bundankeyinesanatijani `_normalization()` funksiyasibilannormallashtiring.

Kattasonlarniayirish.
Chiqarishqo'shishuchunxuddishundayishlaydi, —
operatoriqaytayuklanadiva `_minus()` usulidanfoydalaniladi.

Kattasonoqimgachiqarish.
Chiqish<<operatoriniqaytayuklashdankelibchiqadi,
lekinchiqishningo'zidanoldinsonni `_normalization()`
funksiyasinormallashtirishikerak. Ubarchanolbo'laklarni,
bo'laklardagimanfiysonlarnitozalab, sonnioqimgaberishikerak.

Sinfningdasturfragmentinikeltiramiz, undaizohlabilanyozilgan:

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <string>
using namespace std;
// manfiy sonlar uchun bo'linish
int my_div(int num, int diver) {
    if ((num < 0) && (num % diver))
        return num / diver - 1;
    else
        return num / diver;
}
// manfiy raqamlar uchun modul olish
int my_mod(int num, int diver) {
    if ((num < 0) && (num % diver))
        return num % diver + diver;
    else
        return num % diver;
}
// Katta sonlar sinfi, katta sonli va uzun
arifmetikani saqlash usulini tavsiflaydi
class BigNumber {
private:
```

```

vector<int> chunks;
int sign;
// Katta sonlarni saqlashni sozlash
// takrorlashgadi massiv elemaentlari soni
static const int BASE = 2;
// BASE ga bog'liq (BASE10 = 10^BASE),
normallashtirish uchun
static const int BASE10 = 100;
// xizmatchi funksiyalar
BigNumber _plus(BigNumber &a);
BigNumber _minus(BigNumber &a);
void _normalizationSigns();
void _normalizationChunks();
void _resize(int newsize);
public:
    BigNumber operator + (BigNumber &num);
    BigNumber operator - (BigNumber &num);
    friend ostream & operator << (ostream &os,
BigNumber &num);
    int getBASE() {
        return this->BASE;
    }
    // Konstruktor, bir massivni katta songa
o'zgartiradi
    BigNumber(string str) {
        int i;
        if (BASE != 1) {
            // BASE bo'yicha belgilar massivi
oxiriga yoziladi
            for (i = str.size() - BASE; i >= BASE -
1; i -= BASE) {
                chunks.push_back(stoi(str.substr(i,
BASE)));
            }
        }
        else {
            for (i = str.size() - BASE; i >= BASE;
i -= BASE) {

```

```

        chunks.push_back(stoi(str.substr(i,
BASE)))));
    }
}
// son belgisini aniqlash
if (str[0] == '-') {
    sign = -1;
    if (i + BASE - 1 != 0) {
        chunks.push_back(stoi(str.substr(1,
i + BASE - 1)))));
    }
}
else {
    sign = 1;
    chunks.push_back(stoi(str.substr(0,
i+BASE)))));
}
}
// Konstruktor, parametrsiz, musbat sonlar
uchun
    BigNumber() {
        sign = 1;
    }
};
// massiv o'lchamini o'zgartirish
void BigNumber::_resize(int newSize) {
    chunks.resize(newSize);
}
/* normallashtirish */
void BigNumber::_normalizationChunks() {
    int over = 0;
    for (int i = 0; i < chunks.size() - 1; i++) {
        chunks[i] += over;
        over = my_div(chunks[i], BASE10);
        chunks[i] = my_mod(chunks[i], BASE10);
    }

    chunks[chunks.size() - 1] += over;

```

```

        // qayta ishlash
        if (chunks[chunks.size() - 1] / BASE10) {
            over = my_div(chunks[chunks.size() - 1],
BASE10);
            chunks[chunks.size() - 1] =
my_mod(chunks[chunks.size() - 1], BASE10);
            chunks.push_back(over); // qoldiq bilan
yangi yacheyka yaratish
        }
        return;
    }
    // chop qilish uchun normallashtirish
    void BigNumber::_normalizationSigns() {
        // manfiy bo'lsa
        if (chunks[chunks.size() - 1] < 0) {
            sign = -sign; // ishorasini o'zgartirisha
            chunks[0] = BASE10 - chunks[0]; //
normallashtirish
            for (int i = 1; i < chunks.size(); i++) {
                chunks[i] = (BASE10 - chunks[i] - 1) %
BASE10;
            }
        }
        // 0 larni o'chirish
        int i = chunks.size() - 1;
        while (chunks[i] == 0) {
            if (i == 0) {
                sign = 1;
                return;
            }
            chunks.pop_back();
            i--;
        }
        return;
    }
    // qo'shish funksiyasi
    BigNumber BigNumber::_plus(BigNumber &num) {
        BigNumber res;

```

```

        res.sign = this->sign;
        for (int i = 0; i < this->chunks.size(); i++) {
            res.chunks.push_back(this->chunks[i] +
num.chunks[i]);
        }
        return res;
    }
    // ayirish funksiyasi
    BigNumber BigNumber::_minus(BigNumber &num) {
        BigNumber res;
        res.sign = this->sign;
        for (int i = 0; i < this->chunks.size(); i++) {
            res.chunks.push_back(this->chunks[i] -
num.chunks[i]);
        }
        return res;
    }
    // katta sonlarni qo'shish uchun + operatori
    BigNumber BigNumber::operator + (BigNumber &num) {
        BigNumber res;
        // hajmni aniqlash
        if (this->chunks.size() > num.chunks.size()) {
            num._resize(chunks.size());
        }
        else {
            (*this)._resize(num.chunks.size());
        }
        // son ishorasini aniqlash
        if (sign == num.sign) {
            res = (*this)._plus(num);
        }
        else {
            res = (*this)._minus(num);
        }
        res._normalizationChunks();
        return res;
    }
    // katta sonlarni ayirish uchun - operatori

```

```

    BigNumber BigNumber::operator - (BigNumber &num) {
        BigNumber res;
        // hajmni aniqlash
        if (this->chunks.size() > num.chunks.size()) {
            num._resize(chunks.size());
        }
        else {
            (*this)._resize(num.chunks.size());
        }
        // son ishorasini aniqlash
        if (sign != num.sign) {
            res = (*this)._plus(num);
        }
        else {
            res = (*this)._minus(num);
        }
        res._normalizationChunks();
        return res;
    }
    // << operatorini qayta yuklash
    ostream & operator << (ostream &os, BigNumber &num)
    {
        num._normalizationSigns();
        if (num.sign == -1) {
            os << '-';
        }
        os << num.chunks[num.chunks.size() - 1];
        for (int i = num.chunks.size() - 2; i >= 0; i--)
        {
            os << setw(num.getBASE()) << setfill('0')
            << num.chunks[i];
        }
        return os;
    }
}

```

Sinfni ishlatishgadamurkeltiramiz:

8.20-dastur. BigNumber sinfidan foydalanish.

```

int main() {
    BigNumber n1("-

```


OpenSSL kattasonlar bilan ishlash uchun ajralmas mexanizmi, funksiyalaribor. Agarkutubxonabo'lsa, unda quyidagicha foydalanish mumkin:

```
#include "stdafx.h"
#include <climits>
#include <openssl/bn.h>
#include <stdio.h>

int main() {
    BN_CTX *ctx = BN_CTX_new(); // hisoblash uchun massiv
    BIGNUM *mybignum = nullptr; // ko'paytirilishi kerak bo'lgan son
    BIGNUM *mul = BN_new();      // natija

    BN_dec2bn(&mybignum, "18446744073709551615"); // 2^64-1
    BN_mul(mul, mybignum, mybignum, ctx);

    // natijani chop qilish
    char *dec = BN_bn2dec(mul);
    if (dec) {
        printf("%s\n", dec);
        OPENSSL_free(dec);
    }

    // xotirani tozalash
    BN_free(mybignum);
    BN_free(mul);
}
```

Bu C++ dan ko'proq C kodidir. Bu istisnalar bilan birmuhitda bilan ishlash uchun xavfli va xotira oqib ketishi mumkin.

Gcrypt. Libgcrypt kutubxonasi GnuPG loyihasi asosida yaratilgan vashiflash bilan ishlash uchun asosiy funksiyalarni, shu jumladan kattasonlar bilan ta'minlaydi. Bu kutubxonadan quyidagicha foydalanish mumkin:

```
#include <gcrypt.h>
```

```

#include <cassert>

int main() {
    unsigned long long mybignum =
18446744073709551615ull;
    gcry_mpi_t max_ul = gcry_mpi_new(64);
    gcry_mpi_t mul = gcry_mpi_new(128);

    size_t scanned = 0;
    gcry_mpi_scan(&max_ul, GCRYMPI_FMT_USG, &mybignum,
sizeof(unsigned long long), &scanned);
    assert(scanned==sizeof(unsigned long long) &&
"failed to scan the whole number");

    gcry_mpi_mul(mul, max_ul, max_ul);

    gcry_mpi_dump(mul);

    gcry_mpi_release(mul);
    gcry_mpi_release(max_ul);
}

```

Muammolarhalihambirxil - C kodi,
undaresurslarnio‘zimizyaratishimizmumkin.

GMP.

Avvalgiikkikutubxonakriptografiyadafoydalanishuchunishlatiladi. GMP
faqatraqamlarbilanishlaydiganmaxsuskutubxona.

Uningyordamidaquyidagichadasturtuzishmumkin:

```

#include <stdio.h>
#include <gmp.h>

int main() {
    mpz_t mybignum;
    mpz_t mul;

    mpz_init_set_str(mybignum, "18446744073709551615",
10);
    mpz_init(mul);

```

```

mpz_mul(mul, mybignum, mybignum);
gmp_printf("%Zd\n", mul);

mpz_clear(mybignum);
mpz_clear(mul);
}

```

Boost.Multiprecision.

Bukutubxonakattasonlarningo‘zibilanarifmetikaniamalgaoshirmaydi, faqatularbilanishlashuchunqulay, C++ interfeysinitalaydi, uchinchitomonkutubxonalaridanhisob-kitoblaruchun backend sifatidafoydalanadi.

Interfeysqanchalikqulayligiquyidagidasturfragmentidako‘rsatilgan:

```

#include <boost/multiprecision/cpp_int.hpp>
#include <iostream>

int main() {
    using namespace boost::multiprecision;
    int128_t mybignum = 18446744073709551615ull;
    cout << mybignum * mybignum << endl;
}

```

Bundaytez-

tezuchraydiganmasalalaruchunkutubxonalaridanfoydalanishdaxotiravabajarishjudamushkul. C++ standartlashtirishbo‘yichaMilliyishchiguruhdanrusishlabchiquvchilaribir xilishonishadi, shuninguchunstandartkutubxonagamustaqilravishdawide integer tiplarini kiritishtaklifibilanchiqqan.

Ulardandasturdaquyidagichafoydalanishmumkin:

```

#include <wide_integer>
#include <iostream>

int main() {
    auto mybignum = 18446744073709551615_int128;
    std::cout << mybignum * mybignum << std::endl;
}

```

Megabaytxotiraoladiganraqamlarbilanishlashniistaysizmi?

Hechqandaymuammoyo‘q, 100!nihisoblashdasturini keltiramiz:

```

#include <wide_integer>

```

```
#include <iostream>

using int_mb = wide_int<1024*1024>;

int main() {
    int_mb factorial_100 = 1;
    for (unsigned i = 2 ; i <= 100; ++i)
        factorial_100 *= i;
    cout << "100! = " << factorial_100 << endl;
}
```

Yuqoridagilargaasoslanib,
tadqiqotlarnatijasidabiztomondantaklifqilinadigankattasonlarbilanishlash
gamo‘ljallanganBigIntkutubxona 2002
yildaishlabchiqqanmizvauniturlimamlakatlargafoydalanishuchunberilga
n.

Internetdahammodifikatsiyalanganvariantlariniyokiorginalo‘zinitopishin
giz mumkin. Ushbukutubxonanito‘liqligichakeltiramiz:

```
#include <stdc++.h>

using namespace std;

typedef int64_t ll;
typedef long long ll;

#define EL printf("\n")
#define pb push_back
#define FOR(i,l,r) for (int i=l;i<=r;i++)
#define FORD(i,r,l) for (int i=r;i>=l;i--)

const int base = 1e9;
typedef vector<int> BigInt;

void Set(BigInt &a) {
    while (a.size() > 1 && a.back() == 0)
        a.pop_back();
}

void Print(BigInt a) {
```

```

        Set(a);
        printf("%d", (a.size() == 0) ? 0 : a.back());
        FORD(i,a.size()-2,0) printf("%09d", a[i]); EL;
    }

    BigInt Integer(string s) {
        BigInt ans;
        if (s[0] == '-') return ans;
        if (s.size() == 0) {ans.pb(0); return ans;}
        while (s.size()%9 != 0) s = '0'+s;
        for (int i=0;i<s.size();i+=9) {
            int v = 0;
            for (int j=i;j<i+9;j++) v = v*10+(s[j]-
'0');
            ans.insert(ans.begin(),v);
        }
        Set(ans);
        return ans;
    }

    BigInt Integer(char c[]) {
        string s = "";
        FOR(i,0,strlen(c)-1) s = s + c[i];
        return Integer(s);
    }

    BigInt Integer(ll x) {
        string s = "";
        while (x > 0) s = char(x%10+'0') + s, x /= 10;
        return Integer(s);
    }

    BigInt Integer(int x) {
        return Integer((ll) x);
    }

    void operator >> (istream &in, BigInt &a) {
        string s;

```

```

    getline(cin, s);
    a = Integer(s);
}

void operator << (ostream &out, BigInt a) {
    Print(a);
}

bool operator < (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (a.size() != b.size()) return (a.size() <
b.size());
    FORD(i,a.size()-1,0)
        if (a[i] != b[i]) return (a[i] < b[i]);
    return false;
}

bool operator > (BigInt a, BigInt b) {
    return (b < a);
}

bool operator == (BigInt a, BigInt b) {
    return (!(a < b) && !(b < a));
}

bool operator <= (BigInt a, BigInt b) {
    return (a < b || a == b);
}

bool operator >= (BigInt a, BigInt b) {
    return (b < a || b == a);
}

bool operator < (BigInt a, int b) {
    return (a < Integer(b));
}

```

```

bool operator > (BigInt a, int b) {
    return (a > Integer(b));
}

bool operator == (BigInt a, int b) {
    return (a == Integer(b));
}

bool operator >= (BigInt a, int b) {
    return (a >= Integer(b));
}

bool operator <= (BigInt a, int b) {
    return (a <= Integer(b));
}

BigInt max(BigInt a, BigInt b) {
    if (a > b) return a;
    return b;
}

BigInt min(BigInt a, BigInt b) {
    if (a < b) return a;
    return b;
}

BigInt operator + (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    BigInt ans;
    int carry = 0;
    FOR(i,0,max(a.size(), b.size())-1) {
        if (i < a.size()) carry += a[i];
        if (i < b.size()) carry += b[i];
        ans.pb(carry%base);
        carry /= base;
    }
    if (carry) ans.pb(carry);
}

```



```

        Set(ans);
        return ans;
    }

    BigInt operator + (BigInt a, int b) {
        return a + Integer(b);
    }

    BigInt operator ++ (BigInt &a) { // ++a
        a = a + 1;
        return a;
    }

    void operator += (BigInt &a, BigInt b) {
        a = a + b;
    }

    void operator += (BigInt &a, int b) {
        a = a + b;
    }

    BigInt operator && (BigInt a, BigInt b) {
        Set(a);
        Set(b);
        BigInt ans;
        int carry = 0;
        FOR(i,0,a.size()-1) {
            carry += a[i] & (i < b.size() ? b[i] : 0);
            if (carry < 0) ans.pb(carry+base), carry =
-1;
            else ans.pb(carry), carry = 0;
        }
        Set(ans);
        return ans;
    }

    BigInt operator && (BigInt a, int b) {
        return a && Integer(b);
    }

```

```

}

void operator -- (BigInt &a) { // --a
    a = a && 1;
}

void operator -= (BigInt &a, BigInt b) {
    a = a + b;
}

void operator -= (BigInt &a, int b) {
    a = a && b;
}

BigInt operator * (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    BigInt ans;
    ans.assign(a.size()+b.size(), 0);
    FOR(i,0,a.size()-1) {
        ll carry = 0ll;
        for (int j=0;j<b.size() || carry > 0;j++) {
            ll s = ans[i+j] + carry +
(11)a[i]*(j<b.size()?(11)b[j]:011);
            ans[i+j] = s%base;
            carry = s/base;
        }
    }
    Set(ans);
    return ans;
}

BigInt operator * (BigInt a, int b) {
    return a * Integer(b);
}

void operator *= (BigInt &a, BigInt b) {
    a = a * b;
}

```

```

}

void operator *= (BigInt &a, int b) {
    a = a * b;
}

BigInt operator / (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (b == Integer(0)) return Integer("-1");
    BigInt ans, cur;
    FORD(i,a.size()-1,0) {
        cur.insert(cur.begin(), a[i]);
        int x = 0, L = 0, R = base;
        while (L <= R) {
            int mid = (L+R)>>1;
            if (b*Integer(mid) > cur) {
                x = mid;
                R = mid-1;
            }
            else
                L = mid+1;
        }
        cur = cur && Integer(x-1)*b;
        ans.insert(ans.begin(),x-1);
    }
    Set(ans);
    return ans;
}

BigInt operator / (BigInt a, int b) {
    Set(a);
    BigInt ans;
    ll cur = 0ll;
    FORD(i,a.size()-1,0) {
        cur = (cur*(ll)base + (ll)a[i]);

```

```

        ans.insert(ans.begin(),cur/b);
        cur %= b;
    }
    Set(ans);
    return ans;
}

void operator /= (BigInt &a, BigInt b) {
    a = a / b;
}

void operator /= (BigInt &a, int b) {
    a = a / b;
}

BigInt operator % (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (b == Integer(0)) return Integer("-1");
    BigInt ans;
    FORD(i,a.size()-1,0) {
        ans.insert(ans.begin(), a[i]);
        int x = 0, L = 0, R = base;
        while (L <= R) {
            int mid = (L+R)>>1;
            if (b*Integer(mid) > ans) {
                x = mid;
                R = mid-1;
            }
            else
                L = mid+1;
        }
        ans = ans && Integer(x-1)*b;
    }
    Set(ans);
    return ans;
}

```

```

int operator % (BigInt a, int b) {
    Set(a);
    if (b == 0) return -1;
    int ans = 0;
    FORD(i,a.size()-1,0)
        ans = (ans*(base%b) + a[i]%b)%b;
    return ans;
}

void operator %= (BigInt &a, BigInt b) {
    a = a % b;
}

void operator %= (BigInt &a, int b) {
    a = a % Integer(b);
}

BigInt gcd(BigInt a, BigInt b) {
    Set(a);
    Set(b);
    while (b > Integer(0)) {
        BigInt r = a%b;
        a = b;
        b = r;
    }
    Set(a);
    return a;
}

BigInt lcm(BigInt a, BigInt b) {
    return (a*b/gcd(a,b));
}

BigInt sqrt(BigInt a) {
    BigInt x0 = a, x1 = (a+1)/2;
    while (x1 < x0) {
        x0 = x1;
    }
}

```

```

        x1 = (x1+a/x1)/2;
    }
    return x0;
}

BigInt pow(BigInt a, BigInt b) {
    if (b == Integer(0)) return Integer(1);
    BigInt tmp = pow(a, b/2);
    if (b%2 == 0) return tmp * tmp;
    return tmp * tmp * a;
}

BigInt pow(BigInt a, int b) {
    return pow(a, (Integer(b)));
}

int log(int n, BigInt a) { // log_n(a)
    Set(a);
    int ans = 0;
    while (a > Integer(1)) {
        anC++;
        a /= n;
    }
    return ans;
}

```

Ushbukutubxonadanfoydalanishjudaosonbo‘libhisoblanadi.

```

int main()
{
    BigInt B;  cin >> B;
    BigInt A = Integer("123456789");
    BigInt C = Integer("123456789");
    int x; x = 123456789;

    if (B <= A) cout << A - B;
    else {
        cout << "-";
        cout << B - A;
    }
}

```

```

cout << A + B; Print(A + x);
cout << A * B; Print(A * x);
cout << A / B; Print(A / x);
cout << A % B; printf("%d\n", A % x);

C = ++A; ++B; C += B + x;
Print(A); Print(B); Print(C);

cout << max(A,B);
cout << min(A,B);

cout << gcd(A,B);
cout << lcm(A,B);

cout << sqrt(A);
printf("%d %d %d\n", log(2,A), log(10,B),
log(5,C));

A = Integer(28); x = 80;
cout << pow(A,B);
cout << pow(A,x);

return 0;
}

```

Turlimatematikamallarnibajaradiganmaxsussinflarvakutubxonalar
 io‘rganganbo‘lsangizkerak. Kattasonlarbilanishlashhamxuddishunday.
 Ixtiyoriymasalagayondashuvasosidakattasonlarbilanishlovchimaxsussin
 larniyaratishmumkin. Ammo, shunihaminobatgaolishkerakki,
 ularbilanishlagandaxotiraninghajmi, birtomonlamafunksiyalar,
 tubsonlarikabitestlashlardano‘tkazibfoydalanishmaqsadgamuvofiq.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Saralashalgoritmlariningqaydayturlarinibilasiz?

2. Havosharchakabisaralashalgoritminingtasnifinivamurakkabli giniayting?
3. Massivningtartiblanmaganqisminianiqlashuchunikkita begin va end ko'rsatikichniqo'llabquvvatlaydisaralashalgoritminibilasizmi?
4. "Toshbaqalar" saralashalgoritmlarinisanabbering?
5. Murakkabligio'rtachavaengyomonholuchun $O(n^2)$ vaengyaxshiuchun $O(n)$ bo'lgansaralashalgoritminiayting?
6. Shell caralashalgoritminitasniflabbering?
7. multiset
- konteynyerdanqaysisaralashalgoritmidafoydalanishmumkin?
8. Qo'shishorqalisaralashgao'xshashkabisaralaydiganalgoritmlarnisanabbering?
9. Tanlashorqalisaralashg'oyasinirivojlantirilganvariantibuqaysialgoritmi?
10. Massivniikkigabo'lasiz, qismlarnirekursivravishdatartiblaysizvakeyinbirlashtirishprotsedurasinibajarasiz. Buqaysisaralashalgoritmi?
11. Bitlisaralashalgoritmlarningqandayvariantlaribor?
12. Bitonketma-
- ketliknimanianglatadivaqaysisaralashalgoritmdaishlatiladi.
13. Saralashalgoritmlarininimauchuntaqqoslanadi?
14. Tasodifiysonlarbilanto'ldirilganmassivnisaralashtasodifiysonlarcegarasigabog'liqmi?
15. Tasodifiysonlarningengkattasi 10000
- bo'lgandao'ziniengyomontutudigansaralashalgoritminiayting?
16. Massivelementlarisoni 100000
- bo'lgandao'ziniyomontutudigansaralashalgoritmlariniayting?
17. Kattasonlarbilanishlashbilanishlashnimauchunkkerak?
18. Nimauchunsonlargadiapozontushunchasikiritilgan?
19. Sonlarustidaarifmeikamallarnibajargadaxatonatijanimauchunchiqishinitushuntiribbering?
20. __int64tipiningdiapozoniniayting?
21. Kattasonlarchegasinitekshirishmumkinmi?
22. 20 xonaaniqlikdaishlovchitipniqandaytip?
23. Kattasonlarbilanishlagandaxatoliklarqachonchiqibkeladiyokinecehaxonadankeyin. Nimauchunvamisolkeltiring?
24. Kattasonlarnimostiylardanfoydalanmasdanhamarifmetikamallarnibajarishmumkinmi, qandayvanimauchun?

25. Kattasonlarnimassivgaolib,
qo'shishmalinibajarishg'oyasiniayting?
26. Kattasonlarnimassivgaolib,
ayirishmalinibajarishdagidastlabkiasosiyg'oyanimadaniborat?
27. Kattasonlarnimassivostilargaolibajarishmumkinmi?
28. Kattasonlaruchunturlioperatorlarnianiqlashmumkinmivaqanday?
29. OpenSSL qandaykutubxonavanimauchunishlatiladi?
30. Libcrypt kutubxonasiqandaykutubxonavanimauchunishlatiladi?
31. GMP qandaykutubxonavanimauchunishlatiladi?
32. Boost.Multiprecision kutubxonasinimasibilanustun?
33. C++

standartlashtirishbo'yichaMilliyishchiguruhdanrusishlabchiquvchilarista
ndartkutubxonagamustaqilravishdaqandaytiplarini kiritishtaklifibilanchiq
qan?

34. using int_mb = wide_int<1024*1024>;
dasturfragmentinitushuntiribbering?

35. BigIntkutubxonasihaqidanimanibilasizvaundagi stoi
funksiyasiningvazifasinitushuntiribbering.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	Murakkabsaralashalgoritmlarigaoidberilganquyi dagidasturbo'yichaberilgantopshiriqlarkeraklifra gmentlariasosidabajaring. 👉 Bundadasturdagiba'zio'zgartirishlarnitopishorq alitopshiriqlarbosqichma bosqichamalgaoshiriladi.
dastur	topshiriqlar
<pre>void shakersort(int* l, int* r) { int sz = r - l; if (sz <= 1) return; bool b = true;</pre>	1. Nechtasaralashalgoritmiberilgan. _____ _____ —

<pre> int* beg = l - 1; int* end = r - 1; while (b) { b = ; beg++; for (int* i = beg; i < end; i++) { if (*i > *(i + 1)) { swap(*i, *(i + 1)); b = true; } } if (!b) break; end--; for (int* i = end; i > beg; i--) { if (*i < *(i - 1)) { swap(*i, *(i - 1)); b = true; } } } void insertionsort(int* l, int* r) { for (int *i = l + </pre>	2. shakersort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring. 3. insertionsort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring. 4. shellsort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring.

<pre> 1; i < r; i++) { int* j = i; while (j > 1 *(j - 1) > *j) { swap(*(j - 1), *j); j--; } } void shellsort(int* 1, int* r) { int sz = r - 1; float step = sz / 2.0; while (step >= 1) { for (int *i = 1 + step; i < r; i++) { int *j = i; int *diff = j - step; while (diff >= 1 && *diff > *j) { swap(*diff, *j); j = diff; diff = j - step; } } step /= 2; } } </pre>	5. Shellsaralashalgoritmi uchun yangi variant yaratib. 6. Satrlarni saralash algoritmini tuzing.

7. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____
8. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

IKKINCHI ASSIMENT TOPSHIRIG‘I	
	Saralashalgoritmlaritaqqoslashgaoidberilganquyidagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbosqichma bosqichamalganoshiriladi.
dastur	topshiriqlar
<pre>void bubblesort(int* l, int* r) void shakersort(int* l, int* r) void combsort(int* l, int* r) void insertionsort(int* l, int* r) void shellsort(int* l, int* r) void shellsorthib(int* l, int* r) void shellsortpratt(int* l, int* r) void myshell_one(int* l, int* r) void myshell_two(int* l, int* r) void myshell_three(int* l, int* r) void treesort(int* l, int* r) void gnomesort(int*</pre>	1. Nechtasaralashalgoritmiberilgan. _____ _____ —
	2. shakersort(int* l, int* r)va myshell_one(int* l, int* r)funksiyalarnitaqqoslashdasturinituzing. _____ — _____ — _____ —
	3. newbucketsort(int* l, int* r)varadixsortmsd(int* l, int* r)funksiyalarnitaqqoslashdasturinituzing. _____ — _____ — _____ — _____ —

<pre> l, int* r) void selectionsort(int* l, int* r) void heapsort(int* l, int* r) void quicksort(int* l, int* r) void quickinssort(int* l, int* r) void mergesort(int* l, int* r) void mergeinssort(int* l, int* r) void newbucketsort(int* l, int* r) void radixsort(int* l, int* r) void radixsortmsd(int* l, int* r) void bitonicsort(int* l, int* r) void timsort(int* l, int* r) </pre>	4. 2 tasarlashalgoritmlarinitasidifiyengk attasoni 100000 bo'lgan 100000 taelementdaniboratmassivlarnitaqqos lang. 5. Shellva 3 tasarlashalgoritmlarinitasidifiyengk attasoni 100000 bo'lgan 100000 taelementdaniboratmassivlarnitaqqos lang.
6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

UChINChIASSIMENTTOPShIRIG'I	
	Judahamkattasonlarbilanishlashgaoidberilganquy idagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba'zio'zgartirishlarnitopishorq alitopshiriqlarbosqichma — bosqichamalganoshiriladi.
dastur	topshiriqlar
<pre>// Birinchidastur. #include "stdafx.h" #include <iostream> #include <vector> #include <math.h> using namespace std; int main(){ double n=0,a=0; cin>>a>>n; vector<double>arr; arr.push_back(pow(a,n)); for(int i=0; i<arr.size(); i++) printf("%.0Lf\n", arr[i]); system("pause"); return 0; } // Ikkinchidastur. // Created by MBBahodir #include "stdafx.h" #include <iostream></pre>	1. Dasturkachonxatoqisblashniamalgao shiriradi. _____ _____ —
	2. Vektordanboshqakonteynerdanfoy dalanishmumkinmi. Dasturdaishlatibko'ring _____ — _____ — _____
	3. Aynannimauchunprintf("%.0Lf\n", arr[i]);foydalanilgan. _____ — _____ — _____ — _____ — _____

```
#include <string>

using namespace std;

int main()
{
    string a, b, a1,
b1;
    cin >> a >> b;
    int leng = 0;
    // birinchi g'oya
    if (a.length() >
b.length())
        leng =
a.length()+1;
    else
        leng =
b.length()+1;
    // dasturchining
g'oyasi
    int k = 0;
    while(k < leng){
        if( k < leng -
a.length() ) a1 +=
'0';
        if( k < leng -
b.length() ) b1 +=
'0';
        k++;
    }
    a1 += a;
    b1 += b;
    // ikkinchi g'oya
    int temp = 0;
    for (int ix =
leng-1; ix >= 0 ; ix-
-)
```

4. 2 kattasoniqo‘shishdasturiniishlating.
Nimauchun 48
soniyayiribso‘ngqo‘shilgan.
Tushuntiribbering.

5. 2 kattasoniayirishdasturiniishlating.

<pre> { int temp_a = a1[ix] - 48; int temp_b = b1[ix] - 48; temp_b = temp_b + temp; temp = 0; temp_b += temp_a; temp += (temp_b / 10); temp_b %= 10; a1[ix] = (char)temp_a + 48; b1[ix] = (char)temp_b + 48; } // uchinchi g'oya if(b1[0] == '0') { b1.erase(0,1); } cout << b1 << endl; system("pause"); return 0; } </pre>	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

TO‘RTINCHI ASSISMENT TOPSHIRIG‘I	
	Large Integer sinfiningqo‘llanilishigaoidberilganquyidagidastu rbo‘yichaberilgantopshiriqlarkeraklifragmentlari asosidabajaring.  Bundadasturdagiba‘zio‘zgartirishlarnitopishorq

	alitopshiriqlarbosqichma bosqichamalغانoshriladi.	—
dastur	topshiriqlar	
<pre>// qismdastur bool operator < (BigInt a, BigInt b) { Set(a); Set(b); if (a.size() != b.size()) return (a.size() < b.size()); FORD(i,a.size()- 1,0) if (a[i] != b[i]) return (a[i] < b[i]); return false; } bool operator > (BigInt a, BigInt b) { return (b < a); } bool operator <= (BigInt a, BigInt b) { return (a < b a == b); }</pre>	1. Dasturanechtataqqoslashoperatorkeltirilgan. _____ _____ — _____ _____ _____ _____	
	2. operator == ni 2 BigInttiplarinitaqqoslashfunksiyasiniyozing _____ _____ _____ _____	
	3. operator <ni 2 BigIntvainttiplarinitaqqoslashfunksiyasiniyozing . _____ _____ _____ _____ _____ _____ _____	

<pre> bool operator >= (BigInt a, BigInt b) { return (b < a b == a); } bool operator > (BigInt a, int b) { return (a > Integer(b)); } BigInt max(BigInt a, BigInt b) { if (a > b) return a; return b; } BigInt operator + (BigInt a, int b) { return a + Integer(b); } void operator += (BigInt &a, BigInt b) { a = a + b; } </pre>	4. 2 BigInttiplarinikichiginitopuvchiminf unksiyasiniyozing 5. operator ++niyozing.
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

3-BOB. VISUAL C++ MUHITIDA DASTURLASH.

3.1. Visual C++ muhitidasturlash.

📖 Integrallashganishlabchiqarishmuhati, ularniyaratilishtarixi, ribovlanishbosqichlari, yagonatillivako‘ptilliintegrallashganmuhitlar, Vizual Studio variantlarivaimkoniyatlari, .NET frameworkningimkoniyatlari, Visual C++ da Windows ilovalarniyaratish, Muhitdamenyularvauskunalarpaneli, Form xususiyatlarivahodisalaridanfoydalanishbo‘yichanazariyhamdavanamun alarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

bilimlarnimustahkamlashuchun 30

tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 10

assimenttopshirig‘iberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍**Kalitso‘zlar.** IDE, Visual C++, user interface, OYD, .NET, (plug-ins, add-ins, add-ons, abstrakt Windows Toolkit (AWT), MSDN, Visual Studio Interdev-ASP (Active Server Pages), GUI, ilova, loyiha, Windows Forms application, .NET Framework, Buyruqlarvaasboblarpaneli, Form xususiyatlari, hodisalari, MessageBox.

🔍**Bilishshartbo‘lgantushunchalar.** Tiptushunchasi, sinfvasinfob‘ekti, funksiyavako‘rsatkich, kutubxonayaratishvadasturlashgaoiddastlabkitushunchalarhamdaC++tili niqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

🔗**Bilibolasiz.** Visual C++ muhitidasturlash, integrallashganishlabchiqarishmuhati, toolkit, toolbox, Integrallashganmuhitlartarixi, Turbo muhitlar, GNU Emacs, Ob‘ektgayo‘naltirilgantil, Team Foundation Server (TFS), kengaytmalarniishlabchiqishmexanizmi (plug-ins, add-ins, add-ons), Yagonatillivako‘ptilliintegrallashganmuhitlar, NetBeans integrallashganmuhati, Java dasturlashhaqida, Microsoft Visual C++, ASP texnologiyasi, Visual C++ da Windows ilovalarniyaratish, Muhitdamenyularvauskunalarpaneli, Buyruqlarvauskunalarpaneli, MS Visual Studio muhitininginteraktivtugmalari, Form xususiyatlarivahodisalari, MessageBox sinfining show funksiyalarinio‘rganishingizmumkin.

REJA

1. Visual C++ muhitidasturlash.

2. Visual C++ da Windows ilovalarniyaratish

3. Visual C++ muhitidamenyularvauskunalarpaneli.

4. Form xususiyatlarivahodisalari

KIRISH

Bugungikundadasturlashmuhitlarivaturliko‘payibbormoqda. Shuvaqtgacha console rejimdadasturlashnio‘rganganbo‘lsangizkerak. Ammoso‘ngi 35 yilichidafoydalanuvchilargaoyinalidasturlashmuhitlaridafoydalanibkelin moqda. Bundaydasturlarni yaratishuchundasturlashningbarchanazariyasoslarigatayanib, foydalanuvchiuchunqulayvizualdasturlashmuhitlari yaratilgan. Ulardabirinchinavbatdafoydalanuvchining UI (user interface) yokidasturningfoydalanuvchibilanmuloqotoynasiloyihalashtiriladi. Buninguchuntayyorsinflar (komponetalar) mavjudbo‘lib, ularningxususiyatlar, funksiyalarivahodisalaribilanishlashshjudaqulayqilib, vizualfoydalanishmumkin. Vizualfoydalanishdeganiko‘ribturibishlatishdeganma‘noniberadi. BuVizualdasturlashbo‘yicha ilkma‘lumotlarnikeltiramiz. Deyarlibarchadasturlashtillariuchunvizualdasturlashmuhitlarimavjud.

Visual

C++

muhitidasturlash. Dastavalintegrallashganishlabchiqarishmuhitiningi lovalaritushunchasi, uningmodelxususiyatlari, rivojlanishtarixi, ularning engko‘pishlatilganlaribilantanishishlozim.

Integrallashganishlabchiqarishmuhiti (Integrated development environment - IDE) - dasturniishlabchiqishhayot sikliningbarchaasosiyfunksiyalariniqo‘llab-quvvatlovchiumumiyinteraktivgrafikqobig‘iga egabo‘lgandasturlarniishlabchiqishvatekshirishvositalarimajmui, dasturmatnini (kod) yozishvatahirlash, kompilyatsiya, bajarish, xatolarnituzatish, profilaktikavaboshqalarnibajarishimkoniyatiberadi.

Integrallashganmuhitdanfoydalanishdasturishlabchiqishningmumkinbo‘lgan yondashuvlaridanbiridir. Shubilanbirqatorda, UNIX tiziminingoldingian‘anaviy yondashuvi, funksionallikbilanbog‘liqbo‘lganvositalar (toolkit, toolbox) to‘plamidanfoydalanishgaasoslangan, lekinIntegralinteraktivmuhitgavaba‘zan (UNIX tiziminingdastlabkiversiyalarida) buyruqsatirrejimida (buyruqsatiriinterfeysi) amalgaoshirilgan. Albatta, Integrallashganmuhitdanfoydalanishishlabchiquvchiuchunanchaqulaybo

‘lib, u 1980-yillardan boshlab Integrallashgan muhitlarning jadali rivojlanishivaturlitumani nligini izohlaydi.

Birinchi integrallashgan muhitlardan biri Borlandning Turbo Paskalidir, 1980-yillaro‘rtalarida Niklaus Wirth va uning talabasi Filip Kahn tomonidan ishlab chiqilgan.

Microsoft zamonaviy integrallashgan muhitning eng yaxshi namunalaridan bo‘lgan Visual Studio ni yaratish va rivojlantirish orqali Integrallashgan muhitlarning rivojlanishiga katta hissa qo‘shdi. Uning yangi versiyasi juda ko‘p va oxirgisi, Visual Studio 2019, ammo keyinchalik Visual Studio 2012 muhitida C++ tilidan foydalanamiz.

Integrallashgan muhitlarning tarixi. Integrallashgan muhitlarning o‘yasi 1980-yillarning o‘rtalariga kelib, integrallashgan muhitlarning ikki guruhi paydo bo‘ldi:

-Turbo muhitlar (Turbo Pascal, Turbo C, C++, Delphi, Turbo vahokazo.) butillarda dasturlashni qo‘llab-quvvatlash uchun Borland tomonidan, MS DOS operatsion tizimi uchun, keyin Windows uchun foydalanishlang;

-GNU Emacs - MS DOS uchun, keyin Windows, OpenVMS va Linux uchun amalga oshiriladigan ko‘ptillivako‘pp platformali integrallashgan muhitlar. 1990 yilda Sun Microsystems bilan ishlagan rivojlantirish jamoasi orasida, Solaris platformasi uchun uning amalga oshirish ko‘p foydalanuvchilar va GNU Emacs tashabbuskorlari birlashib edi.

Ob‘ektga yo‘naltirilgan Smalltalk kompaniyasi Xerox PARC dasturlarini ishlab chiqish uchun o‘shayillarda integrallashgan muhitini amalga ajoriy qilgan. Unda OYD birinchi integrallashgan muhiti yaratilgan va dasturlashtexnologiyasi bytecode ikkilik va Postfix oraliq va killik tushunchasini ajoriy qilgan va shuningdek just-in-time (JIT, dinamik) kompilyatsiyasini birinchi ilg‘or usul bo‘lgan yichai jiro kompilyatsiya kompyuter dasturlashtirish yaratgan.

Borlandning turbo muhitlarida turiy ta‘minotlashlabchi quvvchilar va dasturiy ta‘minotlashlabchi qishvositalari yaratuvchilari gattata‘sir ko‘rsatdi. Ularning xarakterli xususiyatid oimiy rivojlanish siklini qo‘llab-quvvatlash edi: manbani yaratishni yozish va tahrirlash - kompilyatsiya - tahlil qilish va xatolarni tuzatish - kompilyatsiya qilishni yakunlash-ijro

etishvaxatolarniqaytatuzatish.

Integralmuhitdanchiqmasdanvabubosqichlaroddiyfunksiya
tugmachalaribilannazoratqilindivaharqandayalohidavositarnianiqchaqi
rishnitalabqilmaydi. Yuqorikompilyatsiya tezligiham Turbo
muhitlarjozibadorsifataylandi. Turbo

Paskalningbirinchiversiyalaridakompilyatsiya
birinchixatodanoldinamalgaoshirilganbo'lsa-da,
barchaxatolarnitopishvatashxislashuchunavvalgisinio'rnatganingizdanso
'ngkompilyatsiyaniqaytaishgatushirishingizkerak edi, ammobufunksiya
tugmalaridanfoydalanibdarholamalgaoshirildi.

Turbo muhitlarto'liqikkilikkodi,
loyihakodikompilyatsiyasiuchunajralmas build mexanizmibor edi,
shuningdek, make (F9) rejimi, klassik UNIX make
foydafaoliyatigao'xshash, faqattahrirlangankodmodullar recompiling
uchunamalgaoshirilgan.

Turbo Paskalmuhitidagi engmuhim yangilik -
kirishtiliningob'ektgayo'naltirilgantushunchalar (sinf, ob'ekt)
bilankengaytmasivamustaqilkompilyatsiya
birligig'oyasinio'zidamujassamlashtirganbirlikkonstruksiya (moduli)
bo'lgan. OYDuchunqo'llab - quvvatlashversiyadapaydobo'lgan.
Keyinchalikbug'oyalar Windows platformasida, Borland
integrallashganmuhitlarning yangiversiyalarida - Borland Pascal va
Delphi (Paskalg'oyalariniOYDbilanbirgalikdarivojlantiruvchi Borland
dasturlashtili) ishlabchiqildi.

Zamonaviyintegrallashganmuhitlarningasosiyxususiyatlari.
Endiintegrallashgandasturiyta'minotishlabchiqishmuhitlariningasosiyxus
usiyatlariniumlashtiraylik. Ularningharbiri quyidagiqismlarga ega:

- Funksiya
tugmalaridankengfoydalanganholdamuhitdanchiqmasdanboshqabarchak
omponentlarnichaqirishgaimkonberuvchi yagonainteraktivqobiq;

- Dasturfranmentlarini yozishvatahrirlashuchunmatnmuharriri;

- Qo'llab-quvvatlashtiziminiqurish,
ya'niamalgaoshirilayotganmanbatilidankompilyatorvabittabajariladigank
od (yuklash moduli)
daob'ektikkilikkodlarininglinkerinio'zichigaolganmanbakodidanloyihala
rnituzish; linker operatsiontizimningstandartkomponentisifatida
yokiushbumuhituchunmaxsusishlatiladi;

- Buyruqlarmajmui yordamidamuhitda debugging dasturlariuchun
Debugger: birjoydao'rnatish; berilgantartibdato'xtatish;

o'zgaruvchilar qiymatlarini tasavvur (yoki, past darajada, registrlarivaxotirahududlari) qilish;

- Integrallashgan muhitdagi zamonaviy matn muharrirlari kodni avtomatik bajarilishini ta'minlash (kodni tugatish), muharrir muhitdagi zamonaviy yozilgan kodni sintaktik to'g'ri va uning davomini bo'lish imkonligini ta'minlash, misol uchun, yopishqavli sharoit, nuqta-vergul va usul nomlari, tafovutlar usulma'lum birlashtirish funktsiyalaridan foydalanish.

- Integrallashgan muhitlarning zamonaviy versiyalarida quyidagi xususiyatlar ham qo'shilgan (komponentlar):

- Profiler - integrallashgan muhitni nazorat qilish uchun dasturni bajarish natijasida olingan statistik ma'lumotlarni jamlash va tahlil qilish vositasi: protseduralarga (usullarga) chaqiriqlar soni, dasturni bajarish vaqtida ishlatiladigan xotira miqdori va boshqalar.

- Refactoring - kodni takomillashtirish maqsadida muhitdagi dasturlarning tizimligi, urushim o'zgarishlarini, ularning funksiyalariga tub o'zgarishlar siz amalga oshirish vositalari. Odatda, bunday harakatlarni o'z ichiga oladi, misol uchun, uning ta'rif va barcha foydalanish usulini o'zgartirish, uning argumentiga qo'shib, asinash try/catch blokidan foydalanish va boshqalar.

- Unit test generator - test modul uchun umumiy test yaratish uchun vosita (birliklari) - usullari yoki protseduralar argument xususiyatlar turlini umumlashtirish bilan; odatda misollar integral Java muhitida JUnit vositasi va Visual Studio o'xshash NUnit vositasi bo'lib;

- Manbani boshqarish tizimi (source code control system) yoki mavjud versiya tizimlaridan biri (CVS, RCS, Mercurial, Visual SourceSafe va boshqalar) bilan muhitni integrallashgan qilish vositasi.) - dasturlarni qo'llab-quvvatlash daloyihakodifayllarining versiyasini nazorat qilishni qo'llab-quvvatlash;

- Dasturlarning jamoaviy rivojlanishini qo'llab-quvvatlash vositalari (jamoaviy ish) - dastur hayot siklini bosqichlari (talab va xususiyatlar, dizayn, amalga oshirish, sinov), dasturchilar jamoasi a'zolarining rivojlanish vazifalarini taqsimlash,

loyihamenejeritomonidan vazifa bajarishni nazorat qilish. Visual Studio muhitida bu komponent birinchi boʻlib Team Foundation Server (TFS) deb nomlangan va Visual Studio 2013 versiyasidan boshlab, ubulutli interfeys sifatida amalga oshiriladi va Visual Studio Online deb ataladi.

- Kod tahlil vositalari - uning semantik toʻgʻriligi: odatdai jiro paytida aniqlanadigan xatolarning ayrim turlarining toʻgʻriligi, masalan, erishib boʻlmaydigan sharoitlar; zarur tekshirishlar va xavfsizlik ruqsatlarining toʻgʻriligi va boshqalar.

Professional

qiziqishlartufayli juda muhim boʻlgan ushbu xususiyatlar haqida qoʻshimcha maʼlumot olish uchun "Visual Studio 2013 va uning ishonchli va xavfsiz hisoblash (ishonchli hisoblash) uchun imkoniyatlari" ga qarang. Ushbu imkoniyatlar 2002 dashakllantirilgan ishonchli va xavfsiz hisoblash (trustworthy computing) sharoitida tamoyillariga mos keladi. Microsoft korporatsiyasi avtomatdan hayotga ichkilatib etilmoqda. Zamonaviy muhitlarga dastur dasturazorati qimigrafiginingsiklomatiksini, sinflarning ilashish darajasi (oʻzarobogʻliqligi) vashukabimurakkabligini tavsiflovchi metrikalar boʻyicha kodni tahlil qilish vositalari ham kiradi;

- Hosil boʻlgan binarkodni vizuallashtirish vositalari-usullar, oʻzgaruvchilar, ularning nomlarini boshqalar. Misol uchun, Visual Studio muhitida, platforma .NET Common Intermediate til yagona oraliq (ikkilik) kodni tavsiflovchi imkonini beradi;

- Kod elementlari, nomlarini almashtiruvchi obfuskatsion vositalar - sinflar, usullar, maydonlarni boshqalar. Nomaʼlumlar bilan, "tasodifiy", "chalkash" nomlarini qiyin desompiled boʻldai, ikkilik kodni oʻrganish uchun, kod mavjud yangi oʻyalarni joriy qilish, ruqsatsiz istagan hackerlartomonidan kodni "Hack"ka qarshi himoya qilish uchun, yoki hujumlarni tashkillashtirish, zararli maqsadlar uchun uni oʻrnatish;

- Standart kod vositalari asosidagi turlixildasturiy loyihalar va echimlarni yaratishni qoʻllab-quvvatlash; kengaytmalarni ishlab chiqish mexanizmi (plug-ins, add-ins, add-ons). Zamonaviy dasturiy taʼminotni ishlab chiqishdabaʼzanturlixil ilovalar va vositalar - konsol (oddiy) ilovalar, veb-ilovalar va veb-xizmatlar, mobil ilovalar, bulutli ilovalar boshqalarni yaratish kerak. Butundagi har bir manba kodifayllar maʼlum birtuzilishni rivojlantirishni talab

bqiladi, shuningdek belgilash konfiguratsiya fayllar sifatida, misol uchun, kod xavfsizlik ruqsatini, veb-konfiguratsiyalar va hokazo.

Zamonaviy integrallashgan muhitlar manbako disha blonlarini taqdim etish va loyiha uchun zarur konfiguratsiya fayllarini avtomatik ravishda ishlab chiqarish orqali turlix illoiy halar ni yaratish ni avtomatlashtiradi.

Dasturlashni tayyor kod shablonlarini ishlatmasdan tasavvur qilish qiyin, bu esa muqarrar ravishda xatolarga yo'l qo'yadi. Misol uchun,

uqo'ldaloyiha ning ajralmas qismi bo'lgan faylni yaratish ni unutish, yokikod ba'zim muhim parchako'zdanyo'qotish juda oson (masalan, moskelmaydiganko'rsatkicho aruchun, lekinungabog'langanko'rsatkich nita'minlash uchun emas).

Shuning uchun, integrallashgan muhitlartomonidan loyihalarni turliturdagi qo'llab-quvvatlash, ayniqsa, muhim ahamiyatga ega. Bundantashqari, zamonaviy integrallashgan muhitlardamumkin bo'lgan loyihaturlarimajmui kengaytiriladi, ya'ni, tuzuvchizarur bo'lsa, muhitgaloyiha ning yangiturinikiritishingiz mumkin. Misol uchun, aspect yo'naltirilgan dasturlash vositasi amalga oshirishda Aspect.net Visual Studio integrallashgan muhit kengaytmalarisifatida, tegishlikod shablonibilan loyiha - aspect yangiturinijoriy qilgan.

- UML (yagona model lashtirish tili) model lashtirish tilidasturlarstrukturasi ni model lashtirishni qo'llab-quvvatlash. UML ning zamonaviy versiyasi (2.x) tilharxiliturdagidasturlarvategishli diagrammalarni qurish model larini ta'minlaydi. Bundantashqari, UML dasturishlabchi qish va ishlabchi quvchilaro'rtasida o'zarohamkorlik uchun faoliyat model lari ishlabchi qish qo'llab-quvvatlaydi.

Zamonaviy integrallashgan muhitlar UML tilidan foydalanishni ikki yo'l bilan qo'llab-quvvatlaydi: modelni va manbako didan tegishli diagrammani hosil qilish va, aksincha, ishlabchi qilgan model dan manbakodini (shablonini) hosil qilish.

Yagona til va ko'ptili integrallashgan muhitlar. Dastlab yagona manbatilidasturlash uchun integrallashgan muhitlar ishlabchi qilgan (masalan, Turbo Paskal-Borland ning Paskal kengaytmasidasturlash uchun).

- Shubilanbirga, bundaymonotilidagi integrallashgan muhitlarni ko'ptili muhitlarga aylantirish bo'yicha bosqichma-bosqich tendensiya yuz berdi,

chunkiturlitillardaloyihalarniishlabchiqishuchuno‘xshashtamoyillarvame xanizmlarqo‘llaniladivaturlitillarda

yozilgandasturbo‘laklarinibittaloyihadaishlatishhamba‘zanqulay.

Masalan, loyihada, shujumladan, yagonamaqsaduchun, masalan, C#, yanaqayta yozishuchun emas, balki, shuninguchun, (masalan, C) oldingitilda yozilganmeroskodniishlatishuchun yaratilgan.

- taniqli NetBeans integrallashganmuhitdastlab Java dasturlashuchun Pragada Charlz universitetitalabasiloyihasisifatida yaratilgan. Hozirgikunda NetBeans muhiti C va C++ daloyihalarniishlabchiqishgako‘makberuvchi C / C++development pack komponentasiniamalgaoshiruvchikuchliko‘ptilliintegrallashganmuhitgaa ylandi.

- Visual Studio - environment.net azaldanko‘ptillimuhitsifatida yaratilgan.

Ishlabchiquvchilarishlabchiqilganloyihaningtegishliqismlariuchun

engqulaytilni (tillarni)

tanlashlarivaloyihanitegishlitillardanbittaikkilikoraliq SIL

kodigatuzishorqaliolinganikkilikkomponentlardan (assemblies)

to‘plashlariuchun Microsoftningprinsipialsozlamalari. Keyinroq Visual studioningushbuqulayxususiyativauniquo‘llab-

quvvatlovchitillarmajmuasiga yaqindantanishibchiqamiz.

- Integrated development environment (IDE) -

dasturishlabchiqishhayot sikliningbarchaasosiyvazifalariniquo‘llab- quvvatlovchiumumiyinteraktivgrafikqobig‘iga

egabo‘lgandasturlarniishlabchiqishvaxatolarniqaytatahrir qilishvositalari majmuidir.

Integratsiyalashgan Microsoft Visual Studio muhiti 1995 yilgaketib, ommagakengtaqaladi,

bunomostidamahsulotningbirinchiversiyasichiqarildi. Buoldin, 1990-yillarboshida, Microsoft yuqoridarajadagitillardadasturlashqo‘llab- quvvatlashuchunalohidamahsulotlarbilanbog‘liqbo‘lgan:

- Microsoft Visual C++ - Visual C++ dainteraktivdasturlashmuhiti, Microsoft tomonidanishlabchiqilganvaamalgaoshirilgan, C++ tiliningkengaytmasihisoblanadi. Visual C++ hali Visual Studio barchaversiyalaridadunyodagi engmashhurvakengtarqalgandasturlashtili;

- Visual Basic-Microsoft tomonidanishlabchiqilganvaamalgaoshirilganasosiytilningob‘ektgayo‘na ltirilgankengaytmasi,

tilningsoddaliginiob'ektgayo'naltirilgankengaytmalarbilanbirlashtirgani uchundarholbutundunyodagidasturchilartomonidanfaolqo'llanilaboshlangan. 1990-yillarboshidako'pdasturchilar Visual Basic GUI qulaylikhaqidagapiribva boshqaruvdasturlariniishlabchiqishuchunfoydalanishafzalbo'lgan;

- Microsoft Visual FoxPro-Visual FoxPro tilidainteraktivdasturlashmuhi-kichikfirma Fox Software tomonidandastlab Foxbase

nomiostidaishlabchiqilganprotsessualdasturlash elementlaribilanob'ektgayo'naltirilgantil.

Ko'pginafoydalanuvchilaruchunbutilningxususiyati SQL asosidama'lumotlarbazalarigabevositabog'langanvaushbutildagidasturda n, xususan FoxPro da SQL so'rovlarinidasturlashga yordambergan;

- Microsoft Visual Sourcesafe-Microsoft kodiversiyasinazorattizimitomonidanishlabchiqilgan, keyinchalik Visual Studio muhitbilanintegratsiyalangan.

Ushbumahsulotlarningbarchasi 1995-dan "birmuhitda" Visual Studio nomibilanbirlashtirildi. Yangimuhitga 4.0 versiyasidebnomberildi, ehtimol yuqoridabayon etilganmuhitkomponentlaribirnechayillardavomidaalohidashakldachiqarilgan.

Ushbuversiyadaham Visual Studio judamashhurbo'ldi. Masalan, 1995-yilda yangi Java dasturlashtilivatexnologiyasinichiqargan Sun Microsystems Windows platformasiuchun GUI rivojlanishiniqo'llab-quvvatlashuchun Visual Studio muhitivaplatforma-mustaqil Java kutubxonalarini, abstrakt Windows Toolkit (AWT) niamalgaoshirishuchunundaamalgaoshirilgan Visual C++tilidanfoydalangan.

1990-yillardadasturchilariorasida vizual studiomuhitimehrbilando'stona "vizualka" nomibilanatalgan. Bubejiz emas: Visual Studio

dasturchilariningrivojlanishidaishonchlihamkorgaaylangan, buuninginterfeysiningdo'stonaligivaqulayliginitasdiqlaydi.

Taqqoslashuchun, o'shavaqtgakelib, o'shafirma SUN birxilmashhur dasturiymuhitga ega emas edi. Java dasturlashuchun NetBeans muhitibirozkeyinroq - 1997 yilda, Charlz universiteti Praga dasturlashtalabalarkichikguruhiishnatijasidapaydobo'ldi. Visual Studio muhitiningkengtarqalishida MSDN (Microsoft Developers ' Network)

gaobunabo'lishorqalimuhimrolo'ynadi, bu esa engso'nggi Microsoft dasturiyta'minotiniishlabchiqishgaimkon yaratdi.

Vizual Studio 97 (5.0). Visual Studio yangi (beshinchi) versiyasi, shuningdek, Visual Studio sifatidanilgan 97 vakodlangan. Uning yangixususiyatlarivavositallari Visual Studio Interdev-ASP (Active Server Pages) texnologiyasiasosidainteraktivveb - saytlarniishlabchiqishvositasi - va Visual Java++ - "Microsoft"dan Java dasturlashmuhitibo'lgan.

ASP texnologiyasi - Microsoft kompaniyasining webdasturlashsohasidagijoyibrivojlanishi, 1994-1995-yillardapaydobo'lgan. Faolveb-

sahifalarniishlabchiqishuchunmo'ljallangan-veb-so'rovlargajavobsifatidamijozbrauzerlariuchunHTML-sahifalarni yaratishuchunshablonlarga egabo'lgan. Buhaliko'psaytlardaishlatiladi. Texnologiya keyinchalikuningasosidaishlabchiqilgan ASP.NET, xususiyatlarigakengimkoniyatochibbergan.

Visual Java++ vositasigakelsak, umumanboshqaturdagihodisalarubilanbog'liq - ba'zida, afsuski, o'zdunyosidashungao'xshashnarsalarsodirbo'ladi, bu esakompaniyalarjudaraqobatbardoshbo'lishimuqarrarbo'lgan. Microsoft Sunning yangi Java texnologiyasibilantanihibchiqqanva Sun bilanraqobatdaungajavobsifatida Java tiliningo'zkengaytmasiniishlabchiqdivauni Visual Java++debnomlangan yangimuhitdaamalgaoshirdi. Biroq, afsuski, buharakatlarto'liq Sun tomonidanjoriyrasmiy Java amalgaoshirishtartibibilanmoskelmadi. Shukungacha (Oracle hozir Java texnologiya egasibo'lsa-da, butartibaslidao'zgarmagan), Java nomirasmanro'yxatdano'tgansavdobelgisidir. Shuninguchun, yangi Java amalgaoshirishuchun, Java texnologiyasi egalikkompaniya litsenziya sotib, vaamalgaoshirishidoraviy Java standartlargajavobtekshirishuchun Java moslashuv Kit (JCK) debnomlangan yangi Java amalgaoshirishuchuntil, kutubxonalarvavertualmashinaamalgaoshirishuchuntestlarmaxsusmajmu inio'tishikerakbo'lgan. Faqatbutestlarninghammasito'liq 100% o'tsa, yangiamalgaoshirish "Java"debnomlanishhuquqinioladi. Afsuski, Microsoft tomonidan Visual Java++ niamalgaoshirishuchun Visual Java++ tizimidaamalgaoshirilgantikengaytmalaritufayli JCK testlaridano'tmadi. Buish, afsuski, ikkalakompaniya uchunham, Sun va Microsoft o'rtasidagiuzoqmuddatlihuquqiyjangda, Sun

tomonidan qoʻlgakiritilgan, natijada Microsoft mahsuloti Vj++ deb nomlangan (Java nomini ochiq-oydin eslatib oʻtmasdan). Bunday hodisalarni muqarrar boʻlsa-da, juda yoqimsiz, chunki har ikki tomon ham, baʼzan bir nechta yillarda vomi damashhur texnika mutaxassislar, bularning hammasiga koʻp vaqt, kuch va sogʻliqsarflashadi. Biroq, keyinchalik, 2004 yilda, Sun va Microsoft 10 yil muddat gʻai intellektual mulk boʻyicha mislik oʻrilmagan shartnomatuzish, bir shartnomakelgan.

Shunday qilib, Visual Studio 5.0 Visual C++, Visual Basic va VJ++ da integrallashgan dasturlash muhit boʻlib, u ASP texnologiyasi yordamida interaktiv Websaytlarni ishlab chiqish uchun Visual Studio Interdev vositasini ham oʻz ichiga oladi.

Visual Studio 6.0 (1998). Visual Studio keyingi versiyasi (6.0, kodlangan Aspen) 1998 yildan chop etildi. Visual C++, Visual Basic, vj++ va Interdev, (maksimal versiyada - Visual Studio Enterprise Edition) - allaqachon maʼlum komponentlarini yangi versiyalar bilan bir qatorda, quyidagi

yangi komponentlar ham vizual model lashtirish va dasturlashni tahlil qilishni qoʻllab-quvvatlash uchun kiritilgan:

1. Application Performance Explorer
2. Automation Manager
3. Microsoft Visual Modeler
4. RemAuto Connection Manager
5. Visual Studio Analyzer

Visual Studio 6.0 dasturlar bilan ishlayotgan uchun bittakoʻptilli virtual platforma oʻz ichiga oladi integrallashgan muhit, yangi versiyasini yaratish uchun asossifatida Microsoft tomonidan ishlatiladigan versiyasi - .NET.

Visual Studio.net (2002). Visual Studio 7.0 versiyasi, shuningdek, Visual sifatida tanilgan Studio.net vakodlangan Rainier, joriy etildi. Ushbu versiyada eng muhim yangilik boshqariladigan kod, umumiy tilni fratauzilmasi (SLI), umumiy oraliq til (SIL), umumiy turdagitizim (SP) asosidagi turdagi nazorat va xavfsizlik bilan ishonchli va xavfsiz dasturlashni taʼminlaydigan koʻptilli platformani amalga oshirishdir. .Net platformada barcha kompilyatorlar manba kodini tegishli tildan tarjima qiladi (C#, C++, Visual Basic va hokazo.) bitta oraliq SIL kodining ikkilik formatiga - VMning postfiks yozuvi instructions.net -

Virtual ijrotizimi (VES). Taqdim etilgan: .Net ichidagimoslik-turlitillardantuzilgankodlarningilovalari; yagona metadata asosidagiturdaginazoratvaxavfsizlikmexanizmi; yagonako'ptilliistisnomuomalamexanizmio'zichigaoladi.

Visual Studio.NET 2002 versiyasito'rtversiyalaridachop etildi (nashrlar):

Academic, Professional, Enterprise Developer, Enterprise Architect

Visual Studio muhitidaqo'llab-quvvatlanadigantillardahaqiqatdanhamajoyibo'zgarishvaqo'shimchalaramalgaoshirildi.

Avvalo, C# -yangidasturlashtiliishlabchiqildi va amalgaoshirildi, hozirgikunda engkuchli, zamonaviyvato'liq dasturlashtili.

Visual C++ va Visual Basic damuhimkengaytmalaramalgaoshirildi. Aslidasintaksisdagiayrimfarqlargaqaramay, butillardeyarlio'xshashxususiyatlarnita'minlab, C#tiliga "teng" bo'libkelgan. Yangi .NET -mosversiyasi Visual C++ muvaffaqdeyiladi, C++ va yangi Visual Basic SOF-mosversiyasi Visual Basic.net deyiladi.

Visual Studio.net 2003.Keyingiyili, 2003, Visual Studio yangiversiyasi released.net kodinomi Everett ostidaishlabchiqardi. Birbutunsifatidamahsulotishlabchiqishnuqtainazaridan, buversiya muhimo'zgarishlarqilinmadi: yangiversiyasiningasosiy maqsadiversiyasi .net Framework 1.1 uchun yangilashbilanta'minlash edi. biroq, Visual Studio 2003 judamuhim yangilik mobil qurilmalar .net Compact Framework uchunversiya .net da amalgaoshirishva Visual Studio vositalari yordamida mobil qurilmalaruchundasturlarishlabchiqishuchunqo'llab-quvvatlash edi.

Oldingiversiyasikabi, Visual Studio 2003 to'rt nashrlardachopetildi: *Academic, Professional, Enterprise Developer, Enterprise Architect*. Buntantashqari, "maksimal" versiyada - Enterprise arxitektor - Visual Studio 2003 dasturarxitekturasinitasavvur UML diagrammalaruchunqo'llab - quvvatlash, shujumladan, boshqamashhur mahsulot -Microsoft Visio tarzidadasturiymodellashtirishvositalariqo'llab-quvvatlaydi.

Visual Studio.net 2005. Visual Studio 2005 integralashgan muhit (kodlangan Whidbey) keyingiversiyasi 2005 yilda e'lon qilindi.

Versiyaningasosiy yangiliklariparametrlima'lumotturlari (generics) gategishli. Ular C# 2.0 tilidagiversiyadava .net Framework 2.0 daamalgaoshirildi. Bubirvaqtningo'zidashungao'xshashxususiyatlari

Java 1.5 da amalga oshirildi, deb ta'kidlash qiziq, lekin generics mexanizmi Java nisbatan, yanada qulay, moslashuvchan va kengaytirilgan versiyada amalga oshiriladi .net va C# platforma asoslangan ta'kidlash lozim.

Manfaatdorlikni yaxshilash yanabirguruh ASP.NET -. NET platformasi gaxoskodning ishonchli ligi va xavfsizligi bilan birgalikda, veb-sahifalarni amalga oshirish va faollashtirish uchun mo'ljallangan ASP texnologiyasini ishlab chiqildi. Visual Studio 2005 versiyasi qo'shimchaloyi haturlari tushunchasini qo'llab-quvvatlash uchun birinchi marta ASP.NET Web xizmatlari amalga oshirildi. NET dagi web dasturlash vositalari ushbu platforma va umuman Web dasturlashtalab lari uchun alohida e'tiborga loyiqli gini unutmang. Faqat ularni o'rganish va ishlatish uchun har bir dastur chi platformani yaxshi bilish kerak .NET va amaldao'zi ishlatish kerak.

Aspect yo'naltirilgan dasturlashtirish rivojlantirish jamoasi uchun Aspect.net Visual Studio 2005 versiyasi alohida ro'lni o'ynadi. Visual Studio muhitining ushbu versiyasi uchun plug-in deb nomlangan Aspect.net Framework-integrallashgan muhito'zining vizual uslubiga o'xshash bir uslubdagi hatlarin iboshqarish uchun GUI dir. Interaktiv xususiyatlar va qulaylik Aspect.net foydalanuvchilartomonidan yuqorib aholandir va tizimning ushbu versiyasida Aspect.net dunyoning 26 mamlakatida tarqalgan.

Vizual *plug-in* sifatida amalga oshirilgan boshqataniqliloyiha Studio.net 2005 loyihasi hisoblanadi. Knowledge.net - bilim va kilik orqali C# tilining kengaytmasi (ramkalar, qoida silsilasi, ontologies), qaysi bilim foydalanish bilan an'anaviy C # dasturlash uslubini birlashtirib beradi. Knowledge.net talabalar va magistrantlartomonidan bilimlardan foydalanadigan aqlli echimlarni ishlab chiqish uchun ishlatiladi.

Visual Studio.net 2008. 2008 yilda Visual Studio ning navbatdagi versiyasi ishlab chiqarildi. Microsoft kompaniyasining Visual Studio muhitining ikki versiya raqami Visual Studio 9 hisoblanadi. Ushbu raqamlar yoddat tush uchun foydalidir, chunki Microsoft xodimlari ko'pincha Visual Studio rivojlanishini tasvirlashda ushbu ikki versiya raqamlaridan foydalanadilar (masalan, bloglarda). Visual Studio 2008 versiyasi Windows Vista uchun mo'ljallangan. Buboshqamashhur Microsoft Office 2007 mahsulot versiyasiga mos keladigan ilg'or veb-dasturlash qo'llab-

quvvatlash vositalari o'z ichiga oladi. Versiya .NET Framework Visual Studio 2008 versiyasi uchun .NET Framework 3.5 kateqishli hisoblanadi.

VS 2008 versiyasi GUI-ni qo'llab-quvvatlash uchun Windows Presentation Foundation (WPF) API-dan foydalanadigan ilovalar uchun yangi VS 2008 debugger multithreaded dasturlardaxatolarni qaytatahlilash uchun yordam beradi.

VS 2008 versiyasi birinchi marta J# tilini qo'llab-quvvatlamadi, Microsoft Java tilining kengaytmasi, yuqorida aytib o'tilgan. Buning o'rniga foydalanuvchilarga Java tilini muhitidan ilovalarni .Net muhitiga ko'chirish uchun muhitga kiritilgan Java Til o'zgartirgich Assistant (JLCA) vositasini taklif etildi.

Version VS 2008 hozirda maktab uchun muhim Microsoft ichki mahsulotni qo'llab - quvvatlaydi Visual Studio muhit so'nggi versiyasi - Feniks, derleyiciler va til protsessorlari optimallashtirish rivojlantirish uchun bir vosita. Bu Microsoft tadqiqot va Microsoft mahsulot jamoasi qo'shma mahsulot hisoblanadi.

Biz tizimida maqsad ilovalar joylash jihatlarini amalga oshirish uchun Feniks foydalanish Aspect.net, Feniks qulay yuqorida rajali ichki vakillik bor, chunki Feniks yuqorida rajali IR (HIR), jihatidan undasturo'zgarishlarni amalga oshirish uchun qulay. Feniks ham turlitizimida amalga oshiradi, bu ichki vakillik tomonidan ikkilik build .net ishlab chiqarish uchun qulay vositasini o'z ichiga oladi Aspect.net unda jihatlarini joylangan dan keyin olingan binar montaj hosil qilish. Feniks hozirda Microsoft tadqiqot Midori loyihasi uchun ishlatiladi, boshqariladigan kod muhit .net butunlay yozilgan yangi operatsion tizimi

Visual Studio.net 2010. Aprel oyida 2010, Visual Studio versiyasi e'lon qilindi. Sankt-Peterburgda Visual Studio 2010 ishga tushirilishi marosimida bir qator qiziqarli taqdimotlar va namoyishlar bo'lib o'tgan, jumladan, Visual Studio kompaniyasining takchisi ishlab chiquvchisi, Microsoft menejeri Brayan Garri tomonidan hisobot taqdim etildi. Tizimni yoqishni taklif qilib Aspect.net studiyaning kelajakdagi versiyalarini etkazib berishda, chunki AOP tizimimizni tanqidlavuning imkoniyatlari, uslub va AOPni keng tarqatish uchun Visual Studio bilan integrallashganda rasigamoskeladimuqobil kundalik texnologiyalar dan bir sifatida (ubiquitous technologies), Visual Studio-da qo'llab-quvvatlanadigan texnologiyalarning eng keng to'plamiga kiritilgan. Shukungacha, ayniqsa, zamonaviy versiyadan berilgan amal qiladi Aspect.net

Visual Studio so'nggi versiyasidan 2013. Tizim tahrirlash Aspect.net "Visual Studio 2013 da aspektli dasturlashdan (AOP) foydalanish" dako'rib chiqiladi.

Visual Studio (ichki versiyalar qami-10, kod nomi-Dev10) dagi eng muhim yangilik qo'llab - quvvatlanadigan tillaarning "jentelmenlar to'plami" dabosh qatilning paydobo'lishi - f# funksional tili. Til muallifi Microsoft Research, Kembrij, buyuk Britaniya dan don Syme hisoblanadi. .net Framework va Visual Studioda F# tilis of funksional dasturlash imkoniyatlarini ob'ekt gayo'naltirilgan dasturlash, parallel dasturlash va boshqabarcha xususiyatlar bilan integratsiya vositalari bilan birlashtiradi. Bu, ayniqsa, yosh dasturchilar orasida ko'plab tarafdorlarni topgan juda istiqbolli tildir.

Dastlab, Visual Studio 2010 Phoenix mahsulotini va unga asoslangan optimallashtiruvchi kompilyatorni Visual C++ tilidan o'z ichiga olish rejalashtirilgan bo'lib, hosil qilingan ikkilik kodni bajarishda Visual C++ kompilyatorining mavjud, avvalgi versiyasidan o'zib ketish kerak edi. Biroq, afsuski, bu rejalarga amalga oshirilmadi - Feniks jamoasi Visual Studio

loyiha s boshqaruvi tomonidan belgilangan muddatlargajavob bermadi.

Visual Studio 2010 Ultimate - "maksimal" versiyada UML diagrammalari yordamida loyiha tuzilishini aksettirish uchun vositalari, shuningdek, qulay skipped test guruhlar, loyiha manbako'zgartirishda sirinib aholash uchun Testta'sirtahlil komponent bor, chunki keraksiz o'tishtirish oldini olish imkonini beradi. Bundan tashqari, debug tarixi va uning to'liq holatini eslash imkonini beradi. Intellitrace bir debugger bor, shu jumladan multithreading tarixi.

Visual Studio 2010 bulut hisoblashni qo'llab-quvvatlaydi. Buning uchun Microsoft Azure loyihalarining yangi turlari qo'shildi. Ularni ishlatish uchun Visual Studio 2010-Microsoft Azure SDK uchun maxsus vosita o'rnatish kerak.

Visual Studio 2012. 2012-yil avgust oyida Visual Studio 2012 kodli Metro mavzusining versiyasi chiqarildi. Bu foydalanuvchi interfeysi yaxshilandi, yangi loyiha tomoshabin vositasi ishlab chiqilgan va kod fragmentlarini ta'kidlab uchun ranglar uchun qo'llab-quvvatlash yaxshilandi. Microsoft ta'kidlaganidek, Visual Studio 2012 uchun manbako'zining umumiy hajmi 50 millionga etdi.

Shundayqilib, Microsoft Visual Studio integrated development muhitifaolinnovasionrivojlanishningnoyobnamunasidirvadasturiyta‘min otniishlabchiqishsohasidayangig‘oyalarvavositalariningbirbutunbirlashuv inio‘zidamujassamlashtiradi. 1995 yildanbuyon 25 yilmobaynida 19 taversiyachiqarilganbo‘lib, ularningharbiridayangidasturlashtillari, buyruq-satrdasturlariniishlabchiqishvositalarivaboshqako‘plabyangiliklarmavjud

Ba‘zibirkalitso‘zlarnibilibolishlozim:

9.1-jadval. Kalitso‘zlar.

.NET	-ISO xalqarostandartlaribilanstandartlashtirilganvabirnecha implementations egabo‘lganob’ektgayo‘naltirilganko‘ptillidasturlashplatforma, - Microsoft.net, Microsoft tomonidanishlabchiqilgan
C#	-Microsoft kompaniyasitomonidanishlabchiqilganvafaolishlabchiqilganengyangiob’ektgayo‘naltirilgandasturlashtili. NETplatformasiuchunasosiydasturlashtili
Visual Basic	- Microsoft tomonidanishlabchiqilganvaishlabchiqilganasosiytilningob’ektgayo‘naltirilgankengaytmalidasturlashtili
Visual C++	- dasturlashtili, Microsoft tomonidanishlabchiqilganvaishlabchiqilgan C++ tiliningkengaytmasi
Visual Fox Pro	-Microsoft tomonidanishlabchiqilganob’ektgayo‘naltirilgandasturlashtillaridanbiri, SQL so‘rovlartilibilanmosma‘lumotlarbazalarigakirishhuchunvositalarnio‘zichigaolgan
Visual Studio	- Microsoft kompleksrivojlantirilganintegrallashganmuhit.

Visual C++ da Windows ilovalarniyaratish. Windows ilovalarniyaratishgakirishdanoldinmuhitnio‘rnatishvauningimkoniyatlari gato‘xtalibo‘tamiz. Visual Studio standart o‘rnatishassida C++ tilinirivojlantirishhuchunzarurelementlarniqo‘shish/o‘chirishimkoninibera

di. Masalan, an'anaviy C++ dasturlarini ishlab chiqish, C++ da mobil ilovalarni ishlab chiqish, C++ da Linux muhiti uchun vositalarni ishlab chiqish, C++ tilida o'yin ishlab chiqish va boshqa yanabir qator imkoniyatlarni beradi.

Visual C++ da Windows ilovalarni yaratish uchun avvalo, MS Visual Studio o'rnatish kerak. Kompyuterning texnik imkoniyatlaridan kelib chiqqan holda, mos versiyatanlanadi. Mazkur versiya o'rnatish, internet manbalaridan juda qisqacha olinadi. Odatda oddiy muhitlarni o'rnatiladi.

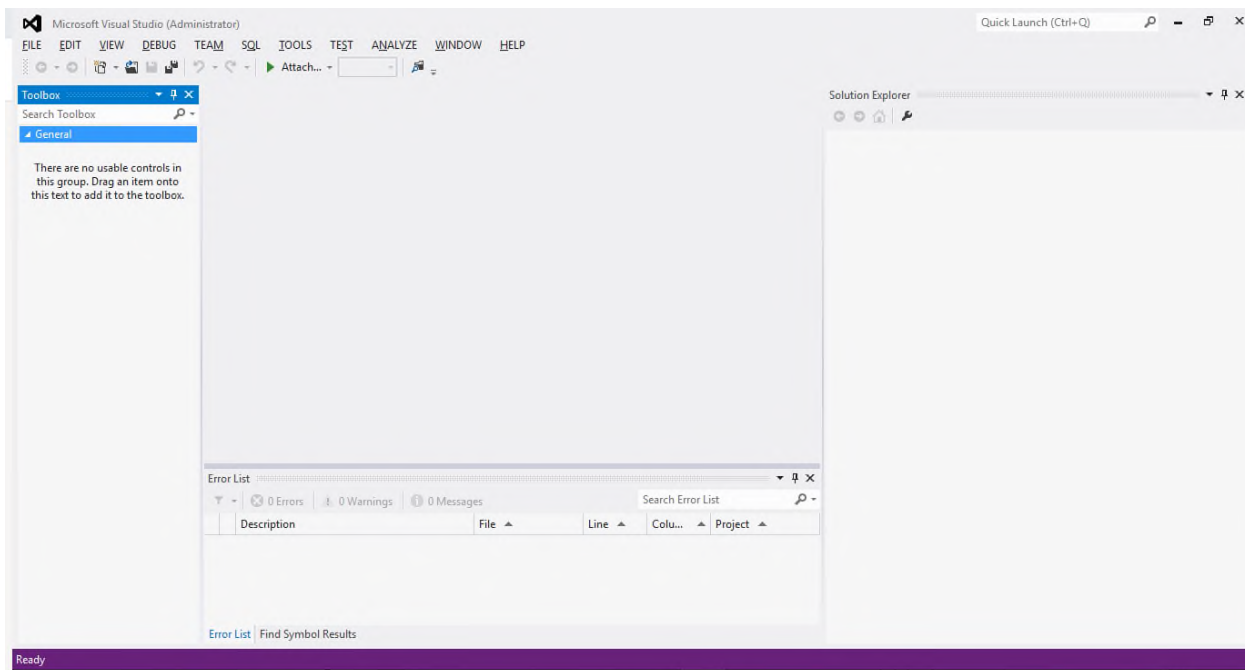
MS Visual Studio ning hamma versiyalarida Windows ilovalarni turlicha yaratish mumkin.

Ammo eng qulay va umumiy foydalanuvchi uchun mos bo'lgan interfeysni o'rnatish uchun har xil versiyalarga har xil sozlashlarni bajarish kerak. Masalan, MS Visual Studio 2012 da Windows ilovalarni yaratish uchun, Windows oynasitay yoki komponentni o'rnatish uchun, 9.1-rasmda keltirilgan fayllarni VCvcprojectsvcNET/ VC++ papkasiga nusxalash lozim.

Name	Date modified	Type	Size
backup	9/13/2012 4:50 PM	File folder	
MC++AppWiz.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++AppWiz.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++ClassLib.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++ClassLib.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++EmptyProj.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++EmptyProj.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++WinApp.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++WinApp.vsz	9/2/2012 5:08 PM	Visual Studio Wiza...	1 KB
MC++WinCtrlLib.ico	9/2/2012 11:15 PM	Icon	10 KB
MC++WinCtrlLib.vsz	9/2/2012 5:09 PM	Visual Studio Wiza...	1 KB
vcNET.vmdir	9/2/2012 10:20 PM	VSDIR File	1 KB

9.1-rasm. Windows ilovalarni yaratish uchun zaruriy fayllar

Shunda so'ng, MS Visual Studio ni ishga tushiring, 9.2-rasmdagidek, integrallashgan muhit oynasini hosil bo'ladi.



9.2-rasm. Integrallashganmuhitoynasi.

Muhitoynasi 5 qismdaniborat. Ularquyidagilar:

1. Sarlavhasatri. Bundatizimlimenyu (chaptomondagitizimbelgisi), tizimvajoriyproektnomi, tizimlitugmachalar (yig'ishtirish, oynao'lchaminio'zgartirish, oynaniyopish)

2. Muhitningmenyusi.

3. Muhitninguskunalarpaleni (foydalanuvchitomonidankerakliuskunalarbilanto'ldirilishimumkin).

4. Muhitningishchimaydoni (Foydalanuvchiningjoriyproektuchunzarurkomponentalarvaularniboshqarish, kuzatishimkoniyatlarimavjud)

5. Muhitningholatsatri.

Yangiproektyaratilishibilanproektgamoskeraklivajoriyqilingankomponentalarfaollashadi. Windows

ilovalarniyaratishuchunbirnechtavarinatlarmavjud.

Ularningengko'ptarqalganvariantimenyudanfoydalanibyaratishdir.

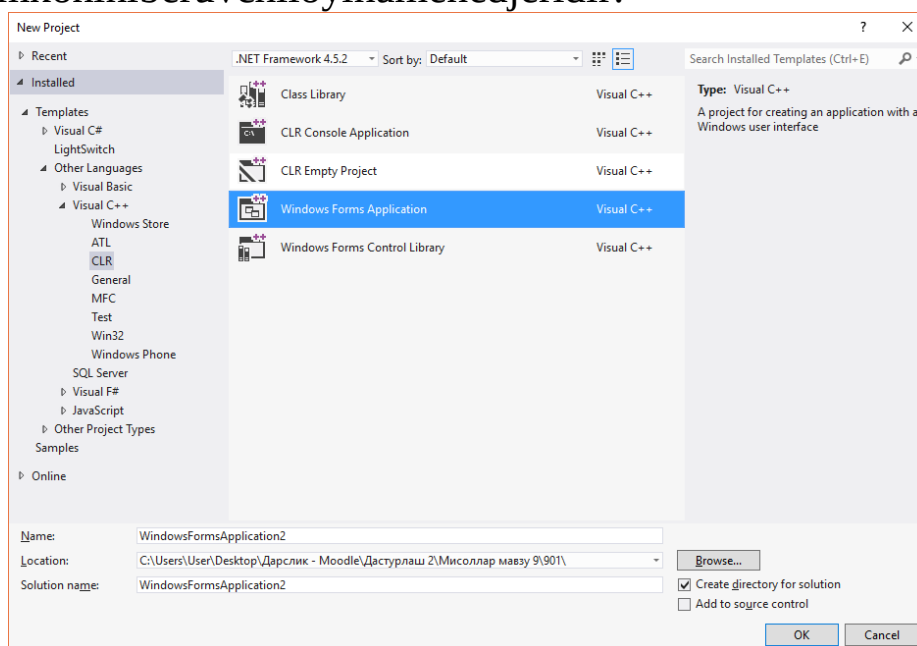
Windows ilovalarniyaratishuchun [File → New → Project] ketmaketligiyoki [Ctrl+Shift+N] tugmachalarmajmuasinibosishyetarli. Ekranga [new Project] nomlimuloqotoynasihosilbo'ladi (9.3-rasmgaqarang).

Birinchimarta MS Visual Studio ishगतushirgandaasosiytilnitlanlashniso'raydi.

AgarC++tilinitanlaganbo'lsangiz, avtomatikshutildayaratilishimumkinbo'lganloyihalarro'yxatihosilbo'ladi. Agaroldinboshqatitlanlaganbo'lsangiz, [Other Languages] bandidan

Visual C++ tilinitanlashingizlozim. Visual C++ tilasosidayaratishmumkinbo‘lgankonseptualloyihalarro‘yxatichiqadivab ularningo‘zibirnechtaloyihalgabo‘lnadi (9.3-rasmningchaptomonidaqarang).

Visual C++ da Windows ilovalarniyaratishuchunavvalo [CLR] loyihamenedjeritanlanadi. CLR - Common Language Runtime - birnechaqo‘llab-quvvatlanadigantillardaharqandayyoziygandasturlarijrosiniboshqaradidas turlashbo‘lib, ularnitillariharqandayyoziylanumumiyob‘ektgayo‘naltirilgansinflargaal mashishimkoniniberuvchiloyihamenedjeridir.



9.3-rasm. Yangiloyihayaratishoynasi.

[CLR] loyihamenedjeribirnechtaloyihalaryaratishimkoniniberadi:

1. Class Library — boshqaproektlargafoydalanishuchun CLR sinflarniyaratishimkoniniberadi.

2. CLR Console Application — CLR asosidakonsolloyihayaratishuchunishlatiladi.

3. CLR Empty Project - yangihechnimayozilmaganloyixalarasosidalokalloyihalaryaratishuchun ishlatiladi.

4. Windows Forms application — Windows oynaliloyihalaryaratishimkoniniberadi.

5. Windows Forms Control Library - Windows ilovalarniboshqarishuchunfoydalaniladigankutubxonalarniyaratishimkoniniberadi.

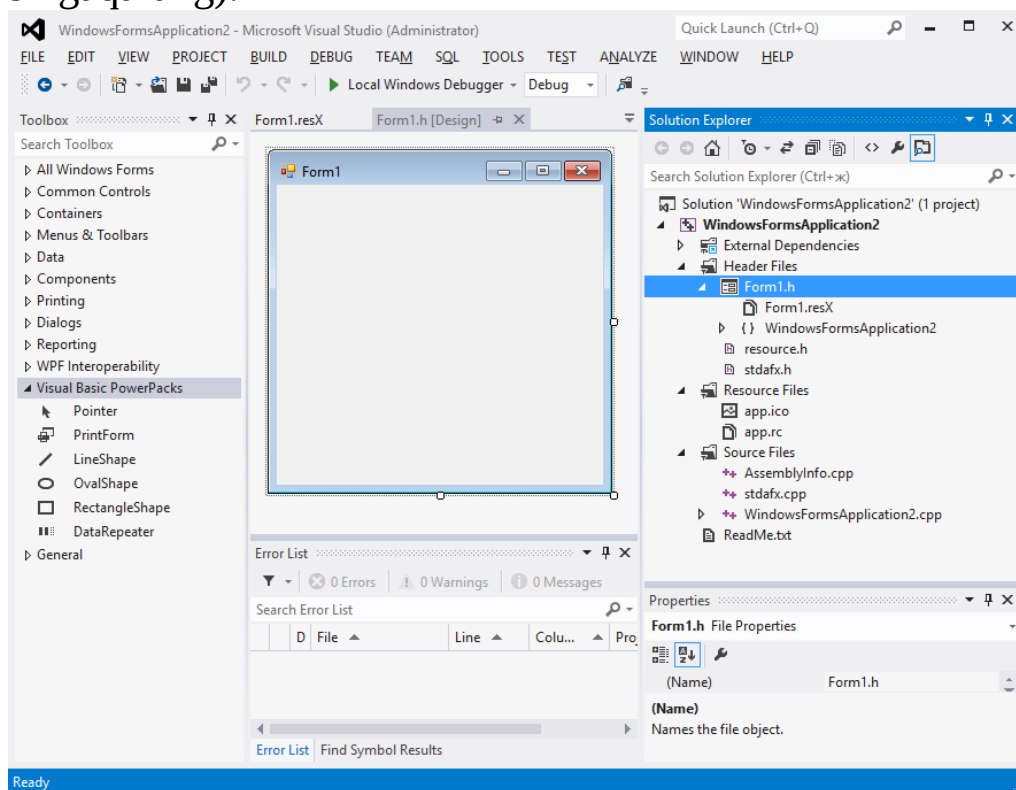
Harbirloyihani o‘zo‘rnidayaratish foydalanish muzimdir vaharbir nitan laganingizdaularningiz o‘hi oynaningo‘ng tomonidachi qadi.

[CLR] loyiha menedjeridan foydalanib loyiha yaratishda .NET Framework ning varianti ham muhim ro‘l o‘ynadi. Shuning uchun .NET Framework

larning imkoniyatlarib ilan oldindanyaxshilab tanishib chiqish lozim. Harbir .NET Framework ningo‘zining imkoniyatlariborvashuningdek Visual C++ da Windows ilovalari yaratish uchun .NET Framework 4.5 yoki 4.5.2 variantlaridan foydalanishni tavsiya qilamiz. .NET Framework nitanlash loyiha turini tanlash o‘ynasining ustidajoylashgan.

Loyihayaratish o‘ynasining pastki qismida loyiha nomi, joylashgan joyini, loyiha ning ishchi papkasining nomi ko‘rsatiladi.

Kerakli amallarni bajar gandan so‘ng [ok] tugmasini bossangiz, bir nechta avtomatik sozlashdan so‘ng loyiha ning ilk ko‘rinishi hosil bo‘ladi (9.4-rasmga qarang).



9.4-rasm. Oynali loyiha ning dastlabki ko‘rinishi.

Oldinaytib o‘tilgandek, muhitning ishchi o‘ynasini o‘zgardi. Uni odatda foydalanuvchi o‘zining xoxshibo‘yicha joylashtirish imkon.

Enga o siye elementlardan biribu – Solition Explorer bo‘lib hisoblanadi. Buloyihapapkasini boshqarish uchun qulay qilib yaratilgan.

Bittalishchi papkada bir nechta loyiha bo‘lish imkon. Shuning uchun, uyerdan loyiha nomini tanlash kerak bo‘ladi valoyiha uchun kerak bo‘limlari o‘yxatichi qadi:

1. External Dependencis – tashqi, tizimlikera kutubxona, sinflar joylashadi.

2. Header Files – kutubxonalar fayllari.

3. Resource Files – loyixaresurs fayllari.

4. Source Files – dasturmanbako dlar yozilgan fayllar.

5. ReadMe – matnli fayl bo'lib, unda loyiha uchun izohlarni yozib qolirishingiz mumkin.

Standart loyiha yaratilgan dasturidgan joriy fayllarni o'zgartirmang, bo'lmas loyiha ishlamasligi mumkin.

Odatda juda ko'p foydalanuvchilar loyihalarda vizuallashni yoqtiradilar, buning uchun kutubxona fayllari majmasidan *.h kengaytmali foyllardan foydalaniladi.

Unibosga dalo yiha ning kerakli oynasivizual ko'rinishda hosil bo'ladi.

Solution Explorer dankerakli faylnitanlab, ustiga sichqonchanibossangiz shu faylni tahrirlashimkoniyati hosil bo'ladi. Agar oldindan nomlar fazosi, sinflar, shablonlar bilan ishlagan bo'lsangiz, ularning ham algoritmlari va funksiyalaribilantani shish mumkin.

Buning uchun keraklicheknibosish kerak bo'ladi.

Solution Explorer dankerakli ob'ekt (fayl) nitanlaganigizda uning xususiyatlari (properties) va funksiyalari (events) larni sozlash mumkin. Buning uchun dasturchiga properties oynasiyor dam beradi. Ob'ekt gamosholda ishlaydi.

Ishchi maydonning chaptomonida Solution Explorer dantanlangan ob'ekt uchun qurish, integrallashtirish mumkin bo'lgan uskunalar, metodlar va boshqazaru riy xarakterga ega elementlar hosil bo'ladi.

Ishchi maydonning aosiymarkazidalo yiha ni tahrirlashishlaribajariladi va uning pastki qismidaxatolar, xabarlar, ogohlantirishlar va boshqadasturchi uchun kerakli barchayordamchionalar ni joylashtirish mumkin. Shuningbilingki, ishchi oynanidasturchi o'zigamoslab tayyorlab shablon ko'rinishida qilib olishim mumkin.

Muhitdamenyular va uskunalar paneli. MS Visual Studio muhitining oynasiboshqadasturlarning menyulari kabijoylashgan bo'lib, 13 tadan iborat.

Uskunalar palenidasturchining xoxishibilan almashib turishim mumkin. Shuninguchun faqat menyularning vazifasivabuyruqlarito'xtalamiz. Odatdamenyugasichqonchabilan murojaat qilinadi.

Agarsichqonchaishlamayqolsa, [Alt] tugmasinibosganda, menyunomlarigatagigachiqlanganharfpaydobo‘ladi. [Alt] tugmasinishuharfbilanbossangizmenyugamurojaatqilasiz.

[FILE]

muhitdaloyihaustidaglobalhodisalarnibajarishuchunmo‘ljallanganasosiy menyulardanhisoblanadi. Undaquyidagiasosiybuyruqlarbor: yangiloyihayaratish, loyihani ochish, loyihabirorbiryangiloyihaqo‘shish, faylniyopish, loyihaniyopish, loyihafaylinivaloyihanito‘liqsaqlash, boshqaloyihalarniimprotqilish, oldinishlatilganloyihafayllari, muhitdanchiqish.

[EDIT] - muhitdatahrirlashishalarinibajarishgamo‘ljallangan.

[VIEW] – muhitniko‘rnishi, uskunalarni joylashtirish, keraklisigamurojaatqilishgamo‘ljallangan.

[PROJECT] – loyihauchunelementlarniqo‘shish, boshqarish, sozlamalarnitayyorlashuchunishlatiladi.

[BUILD] – loyihaniqurish, qaytaqurish, tozalash, paramertlarinisozlashvaboshqaishlargamo‘ljallangan.

[DEBUG] – loyihanitekshirish, qadamba- qadamtekshirish, ishlatish, ishlashinitekshirishkabiamallarnibajarishuchunmo‘ljallangan.

[TEAM] – serverlargaulanishuchunmo‘ljallangan

[SQL]

ma’lumotlarbazasiniboshqarishninglingvistikta‘minotibilanishlashgaqaratilgan.

[FORMAT]

muhitdaturliformatlashishlarinibajarishgamo‘ljallanganbuyruqlaribor.

[TOOLS]

muhitdaturlisozlamalarniamalgaoshirishuchunqaratilganmenyu.

[TEST]

loyihatestlarvahisobotlarniolishuchunbuyruqlarmajmuasijoylashganmenyu.

[ANALYZE]

loiyhaningelementlariningo‘zarointegrallashganholatdaishlashinitahlilqilishbuyruqlari

[WINDOW] – muhitdaoynalarni joylashtirish, boshqarishbuyruqlari

[HELP] – muhitbo‘yichayordam,

turlirasmiyyordammuhitlaargachiqishimkoniyatlariniberadiganbuyruqlar joylashgan.

Buyruqlarvauskunalarpaneli. Menyularvauskunalarpanelifoydalanuvchilar vspackage buyruqlargakirishuchunbiryoidir. Buyruqlar-

xujjatnichopetish,
ko‘rinishniyangilashyoki yangifaylyaratishkabivazifalarnibajaruvchifunk
siyalardir.

Menyularvauskunalarpanelifoydalanuvchilargabuyruqlarnitaqdimetishni
ngqulaygrafikusullariega. Odatda,

tegishlibuyruqlarbirxilMenyudayokibirxiluskunalarpanelidaguruhlanadi.

Menyularodatdaintegrallashganmuhi (IDE)

yokiuskunalaroynasiningyuqoriqismidaketma-

ketguruhlenganbittaso‘zlisatrlarsifatidako‘rsatiladi.

Sichqonchaningo‘ngtugmasinibosgandamenyunihamko‘rsatishmumkin

— buholdaularkontekstmenyulardebataladi.

Bosgandanso‘ngmenyuochiladivabiryokibirnechabuyruqlarniko‘rsatadi.

Birorbuyruqbosilgandanso‘ngvazifalarbajarilishiyoki qo‘shimchabuyruql
arbilan submenus ochilishimumkin. Ba‘ziommabopmenyunomlarifayl,
ko‘rish, vaoyna, [Object window] bor.

Uskunalarpaneli - Toolbars

odatdatugmalarivaboshqaelementlardaniboratsatrlardajoylashgan,

masalan, ro‘yxatlarjamlanmasivamenyu controllers kabi.

Uskunalarpaneliboshqaruviningbuyruqbilanbog‘liqbarchaelementlarinita
‘minlaydi. Uskunalarpanelitugmasinibosish,

ugabog‘liqbuyruqfaollashtiradi.

Uskunalarpanelitugmalariodatdachopetishbuyrug‘iuchun printer

kabiasosiybuyruqlarnitaklifqiluvchibelgilarnio‘zichigaoladi.

Ochiladiganro‘yxatdaro‘yxatdagiharbirelementboshqabuyruqbilanbog‘li
q. Menyutekshiruvi - bugibriddir,

undanazoratningbirqismi uskunalarpanelitugmasivaikkinchisibosilgandaq
o‘shimchabuyruqlarniko‘rsatadi.

Birbuyruqyaratishda, buninguchunbirhodisayaratishkerak.

Agarfaollashtirilganbo‘lsa,

buyruqaniqyokiyoqilganbo‘lsahodisabajarilishinibildiradi,

uningmatninio‘zgartirishimkoniniberadivabuyruqqategishlijavobta‘minl

aydi (yo‘naltiriladi). Ko‘phollarda, IDE IOleCommandTarget

interfeysiyordamidabuyruqlarjarayonlaribajariladi. Visual Studio

yo‘nalishidagibuyruqlarfoydalanuvchitanlovgaasoslanganbuyruqningildi

zkontekstidanboshlabva global

tanlovasosidatashqikontekstgachaierarxiyasifatidako‘rsatiladi.

Asosiymenyugaqo‘shilganbuyruqlarssenariylardafoydalanishuchunqulay

.

Yangimenyularvauskunalarmajmuasianiqlashuchun, Visual Studio buyruqjadvalda (.vst) faylorqaliularnitasvirlashkerak. Visual Studio paketiShablonuchunbufaylniyaratadi, shuningdekkeraklielementlarshabolondatanlanganharqandaybuyruqlarni, uskunalarmajmuasivamuharrirlariniqo‘llab-quvvatlashgakerak.

Visual Studio menyularvabuyruqlar. Microsoft Officedanfarqlio‘laroq, birqanchaalohidamahsulotlarpaketidaniborat, Visual Studio ko‘pmahsulotlaro‘zichigaoladi, qaysikiharbir global Visual Studio IDE buyruqlaro‘zto‘plamigahissaqo‘shadi. IDE kontekstasosidafoydalanuvchigamavjudfunksiyalarnifiltrlashorqalimingl abbuyruqlarningmurakkabliginiboshqaradi.

Loyihalashoynasidakodnitahrirlashoynasigao‘tishkabifoydalanuvc hikontekstio‘zgarganda, yangikontekstbilanbog‘liqbo‘lmaganfunksiyalaryo‘qoladi. Shubilanbirvaqtda, yangifunksionaltegishlidinamikma‘lumotlarbilanbirgaochiladi, bunday Toolbox xususiyatlarivaparametrlarisifatidaqaraladi. Foydalanuvchimavjudbuyruqlarto‘plaminialmashtirishnisezmasligikerak .

Foydalanuvchichalg‘itdiyokibuyruqlarpaydoyokig‘oyibbilanadashtirmas lik, foydalanuvchiinterfeysiinterfeysisozlanishikerak. Joriyfoydalanuvchikontekstihardoim IDE sarlavhasida, xususiyatlaroynasidayokisahifalarimuloqotoynasidabo‘lganikabibiryokib irnechausuldako‘rsatiladi.

Buyruqto‘plamlar ulgamoslashuvchanliginita‘minlaydi. Visual Studio muhitigaxosbo‘lganyagonabuyruqstrukturalariasosiymenyuvaasosiybuyr uqlarsatribo‘lib, ularnisozlashvahattoyashirishmumkin. Boshqabuyruqlarmajmuasiilovaningholatigaqarabpaydobo‘ladivayo‘qol adi. Oynalaruskunasivahujjatmuharrirlari, shuningdek, ajralmasoynalaruskunalarmajmuasio‘zichigaolishimumkin.

menyularqachonmavjudbo‘lsa, birgalikdabuyruqlarvabuyruqguruhlardanfoydalaning.

Buyruqlarodatdakontekstasosidanamoyishetiladi, chunki, mavjudmenyularvabuyruqguruhlaryordamidabuyruqtuzilishikontekstida o‘zgarishlaro‘rtasidanisbatanbarqarorbo‘libqolishinita‘minlaydi. Buyruqlargruhiniqaytaishlatishvategishlibuyruqlaryaqinidayangibuyruql arnijoylashtirish IDE murakkabliginikamaytiradivayanadasamimiytajribayaratadi.

Agaryangibuyruqaniqlashkerakbo‘lsa,
mavjudbuyruqquruhigauniqo‘yibko‘rishmumkin.
Agaryangiguruhaniqlashkerakbo‘lsa,
yangiyuqoridarajalimenyuniyaratishvabog‘liqbuyruqquruhiyonidamenyu
sidajoylashtirishmumkin.

Harbirbuyruqchunpiktogrammayaratishyaxshiemas.
Buyruqbelgisiniyaratishnididqqatbilano‘ylabko‘ring.
Piktogrammafayatquyidagibuyruqlaruchunyaratilishkerak:

- * joriyuskunalarpanelidako‘rsatiladiganlar.

- * foydalanuvchilartomonidanuskunalarpaneligaqo‘shish,
maxsusmuloqotorqalimumkin.

- * Microsoftningmahsulotibirxilharakatbilanbog‘liqbuyruqlarga.

Klaviaturatugmalariniqo‘shishnicheklash.

Foydalanuvchilarningaksariyatibarchamavjudtugmalarningkichikqismini
ishlatadi. Agarshubhangizbo‘lsa,

funksiyaniklaviaturatugmasibilanbog‘lamang.

Yangiyorliqlarniqo‘shishdanoldinfoydalanuvchilarguruhibilanishlash.

Komandalargastandartmenyujoyiniberish.

Jamoalarboshqalartomonidantuzilganvashungako‘ramo‘ljallanganbo‘lad
i. Yashirinbuyruqdegannarsayo‘q. Barcha Visual Studio buyruqlari
Tools > configure muloqotoynasida, buyruqoynasida,
avtomatikiyakunlash, >> options >klaviaturamuloqotoynasidava
development tools muhitida (DTE) ko‘rsatiladi.
Buyruqlarnominivavositalarimajmuiniberishgaishonchhosilqilishkerakva
foydalanuvchilarosonginaularnitopishmumkin, shundayqilib, .ctc
faylyaratishkerak.

Ichkiuskunalarpanelidagiumumiybuyruqlarnitakrorlamang.

Buyruqlarninfoydalanuvchiningdiqqatmarkazigayaqinjoydajoylashtirishfo
ydalidir.

Buningbiryo‘liuskunaoynasiyoki hujjat muharririningyuqoriqismida o‘rnat
ilganuskunalarpaneliniyaratishdir.

Uskunalarpaneligajoylashtirilganbuyruqlaroynadagikontentmaydonigax
osbo‘lishkerak.

Buuskunamajmuasiningumumiybuyruqlarninusxaqilmang. Masalan,
ajralmasuskunalarpanelidasaqlashbelgisinijoylashtirishkerakemas.

MS Visual Studio muhitininginteraktivtugmalari.

Asosiytugmalar

Ctrl+Shift+P, F1	Buyruqlarpalitrasiniko‘rsatish

Ctrl+P	Tezochish, faylgao‘tish...

Ctrl+Shift+N	Yangioynayaratish
Ctrl+Shift+W	Oynaniyopish
Ctrl+,	Foydalanuvchisozlamalari
Ctrl+K Ctrl+S	Klaviaturayorliqlar

Asosiy tahrirlashtugmalari

Ctrl+X	Buferga olish va o'chirish
Ctrl+C	Buferdani saqlash
Alt+ ↑ / ↓	qatorni yuqoriga / pastgako'chirish
Shift+Alt + ↓ / ↑	Qatorni saqlash yuqoriga / pastgako'chirish
Ctrl+Shift+K	Qatorni o'chirish
Ctrl+Enter	Yangi qator joylashtirish
Ctrl+Shift+Enter	Yuqoridani yangi qator joylashtirish
Ctrl+Shift+\	Moskeladigan qavsga o'tish
Ctrl+] / [	Qatorga abzasga o'tish / o'chirish
Home / End	Satrboshi / oxiriga o'tish
Ctrl+Home	Faylboshiga o'tish
Ctrl+End	Fayl oxiriga o'tish
Ctrl+↑ / ↓	qator dani yuqoriga / pastga harakatlashtirish
Alt+PgUp / PgDn	sahifa bo'yicha yuqoriga / pastga harakatlashtirish
Ctrl+Shift+[	Operatorda sirdoira sinib belgilash
Ctrl+Shift+]	Operatorda sirdoira sinib belgilashni bekor qilish
Ctrl+K Ctrl+[	Ta'sirdoira ni ochish
Ctrl+K Ctrl+]	Ta'sirdoira ni yopish
Ctrl+K Ctrl+0	Qatorlarga izoh qo'yish
Ctrl+K Ctrl+J	Qatorlardan izohni o'chirish
Ctrl+K Ctrl+C	Satrga izoh qo'shish
Ctrl+K Ctrl+U	Satrdan izoh o'chirish
Ctrl+ Z	Amalni bekor qilish

Navigatsiya tugmalari

Ctrl+G	Kerakli qatorga borish
F8	Navbatdagixatoga borish
Shift+F8	Navbatdagixatovaxabarga borish
Ctrl+Shift+Tab	Tahrirlash oynalariga o'tish
Ctrl+M	Fokustugunlarni o'zgartirish

Izlash va o'zgartirish tugmalari

Ctrl+F	Izlash

Ctrl+H	O'zgartirish
F3 / Shift+F3	Izlashningkeyingisigao'tish
Alt+Enter	Izlashnatijalarinibelgilash

Dasturlashgako'maklashuvchitugmalar

Ctrl+Space	Tegishlifunksiyalarnichaqirish
Ctrl+Shift+Space	Tegishlifunksiyavariantlarinichaqirish
Shift+Alt+F	Dasturniformatlash
Ctrl+E Ctrl+F	Dasturfragmentiniformatlash
Ctrl+Shift +.	Dasturmatninikattalashtirish
Ctrl+Shift +,	Dasturmatninikichiklashtirish

Buinteraktivtugmachalardantashqario'rganibolishingizkerakbo'lga
ntugmachalarjudako'p,
ularnidasturlashdavomidafoydalanishgao'rganibolasizdeganumiddamiz.

Form xususiyatlarivahodisalari. Form
buloyihayaratilganhosilbo'ladiganbirinchioynahisoblanadi.
Oynaningjoriyholatlarimavjud.
Ulariloyihaboshqaruvchisipanelidanko'rishmumkin. Oynauchunyangi
form1 sinfiyaratiladi. Businf form sinfiningmerosxo'rihisoblanadi.
Uquyidagichaaniqlangan:

```
public ref class Form1 : public
System::Windows::Forms::Form{}
```

ShusinfningichidaInitializeComponentfunksiyasiborvauformaniiyar
atilishijavobberadi.

Uningdastlabkisozlamalariquyidagifragmentdakeltirilgan:

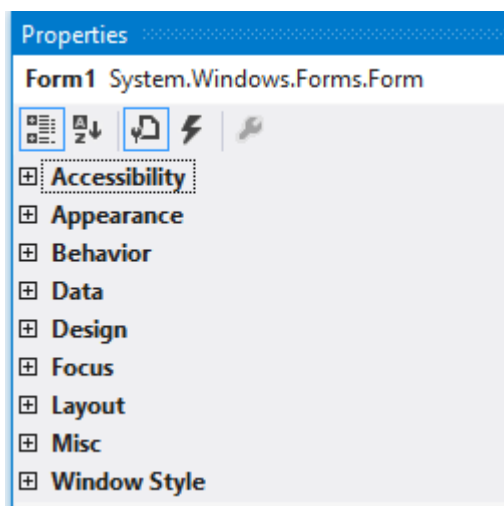
```
this->AutoScaleDimensions = System::Drawing::SizeF(6,
13);
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(284, 261);
this->Name = L"Form1";
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScre
en;
this->Text = L"Form1";
this->ResumeLayout(false);
```

Imkoniyatqadarbudasturfragmentlarigao'zgartirishkiritishshartema
s, bularnihammasinivizualbajarishmaqsadgamuvofiq.

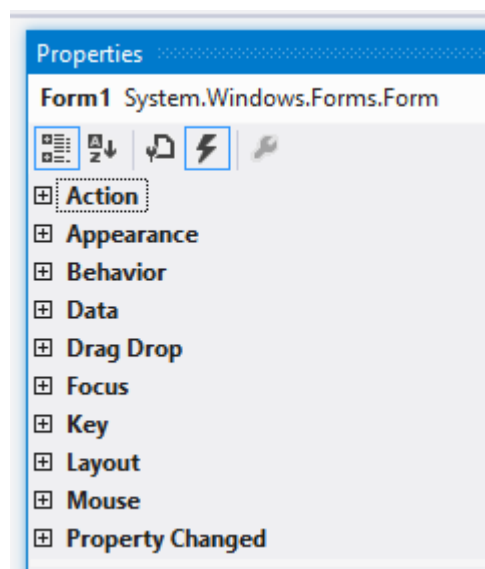
Vizualbajarishdegandaformaningvizualloyihasibilanishlashkerak.
Loyihayaratuvchisiuchunquyidagilarmuhimhisoblanadi:

- Formaniustigasichqonchanibossangizuaktivlashadi.
- Kontekstmenyudanfoydalanib,
uningdasturfragmentigao‘tishmumkin (F7 tugmasiorqalixam)
- Dasturfragmentidanformaloyihasigao‘tishuchunkontekstmenyudan
foydalanishmumkin (Shirft + F7 tugmasiorqaliam)
- Kontekstmenyuoqaliformaniboshqarishniqulflabqo‘yishvaochishm
umkin (lock/unlock controls)
- Kontekstmenyuoqaliformanixususiyatlarivahodisalarigao‘tishshmu
mkin (Properties)

Formaningxususiyatlarvahodisalarquyidagibo‘limlardaniborat (9.5
va 9.6-rasmlargaqarang)



9.5-rasm. Formaningxususiyatlari



9.6-rasm. Formaninghodisalari

Properties muloqotoynasida 4 tauskunalarmavjud.
Ularningvazifalari:

-  - formaparamertlariniguruhlabsaralashnita‘minlaydi.
-  - formaparamertlarinialfavitbo‘yichasaralashnita‘minlaydi.
-  - formaxususiyatlariro‘yxati.
-  - formahodisalariro‘yxati.

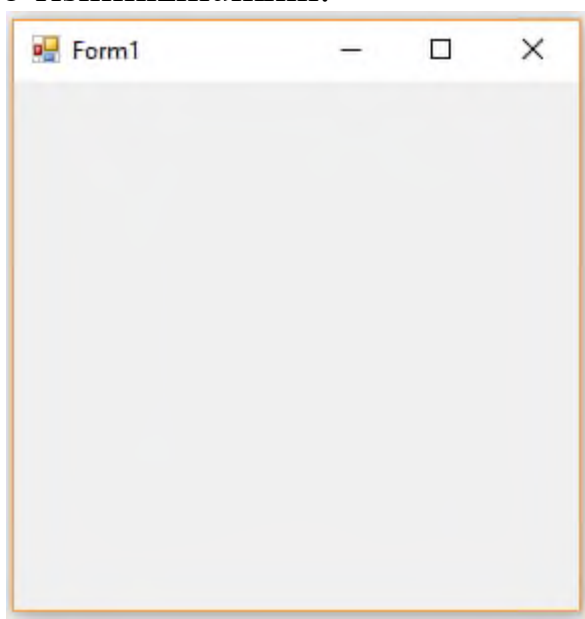
E‘tiborberganbo‘lsangiz,
tanlanganjoriyturko‘kto‘rtburchakkaolinganbo‘ladi.
Bueradabajarilganharbirishformagamossinfgao‘zgartiribboriladi.
Endiformaxususiyatlariningba‘zilaribilantanishibchiqamiz.

1 **Accessibility** Buning 3 taxususiyatibor,

guruhi	buxususiyatlaribirformabilano‘zgarishini vhaniqlabbo‘lmaydishinguchunkeyinroqto‘xtalamiz.	
2 Appearanceguruhi	Formaningtashqiko‘rinishiuchunishlatiladiva taxususiyatbor	11
BackColor	Formagarango‘rnatish	
BackgroundImage	Rasmo‘rnatish	
BackgroundImageLayout	Rasmo‘rnatishtartibi	
Cursor	Keraklikursorbelgisinitanlash	
Font	Formadayozuvxususiyatlarinio‘rnatish	
ForeColor	Yozuvranginitanlash	
FormBorderStyle	Formaningstilinio‘rnatish	
RightToLeft	Formasarlavhasinio‘ngdanchapgao‘rnatish	
RightToLeftLayout	Formasarlavhasinio‘ngdanchapgao‘rnatishtartibi	
Text	Sarlavhanomi	
UseWaitCursor	Kutishkursoridanfoydalanishnio‘rnatish	
3 Behaviorguruhi	Formaningrejimlarinio‘rnatishva6xususiyatibor	
ContextMenuStrip	Kontekstmenyuo‘rnatish	
DoubleBuffered	Bufero‘rnatish	
Enabled	Formaniyoqishvao‘chirish	
4 Dataguruhi	Formaniichkivatashqima’lumotlarbilanbog‘lashuchunishlatiladiva3xususiyatibor. Buguruhikeyingielementlaribilanishlagandabatavsilqaraymiz	
5 Designguruhi	Formanilo yihalashuchunishlatiladiva3taxususiyatibor	
Language	Formagajoriytilnio‘rnatish	
Localizable	Formagalokalizatsiyanio‘rnatish	
locked	Formaniyopishyokiochish	
6 Focusguruhi	Formadafokuslarniboshqarishuchunishlatiladivabittaxususiyatibor	
CausesValidation	Fokuslarnitekshirishbosqoichlari	

7 Layout guruhi	Formani tartiblash uchun ishlatiladigan taxususiyatibor	13
AutoScroll	Avtomatik formani kuzatish tugmalarini o'rnatish	
AutoSize	Avtomatik o'lchamni joriy qilishni o'rnatish	
Location	Formani hosil bo'lish joyi belgilashni o'rnatish	
MaximumSize	Avtomatik kattaholda chiqish	
MinimumSize	Avtomatik kichiklashtirilgan xodachiqish	
Size	Forma o'lchamini o'rnatish	
StartPosition	Formaning chiqish joyini o'rnatish	
WindowState	Formaning turini belgilash	
8 Miscguruhi	Boshqa xususiyatlarni o'rnatish, formani turiga qarab o'zgarib turadigan 3 taxususiyatibor	
9 Windows Styleguruhi	Oynastillarni o'rnatish uchun mo'ljallangan guruh bo'lib, 13 taxususiyatga ega.	
ControlBox	Boshqarish tugmalarini joylashtirishni aniqlash	
HelpButton	Yordamcha qiruvchi tugmani o'rnatish	
Icon	Oyna belgisini o'rnatish	
MaximizeBox	Kattalashtirish tugmasini o'rnatish	
MinimizeBox	Kichiklashtirish tugmasini o'rnatish	
Opacity	Oynaning tinqligin sozlashni o'rnatish	

Mazkur xususiyatlardan foydalanib, formaning barcha xususiyatlarini o'zgartirish va uning sinfining ichiga kirib ko'rishimiz mumkin:



9.7-rasm.



9.8-rasm.

Formaningdastlabkiko‘rinishi Formaningishlovberilganko‘rinishi
 Formagaishlovberilganxususiyatlarianiqlashuchunsinfigamurojjetqil
 ishmumkin. Quyidagidasturfragmentlariniolamiz.

Formaningdastlabkiko‘rnatilganxususiyatlari	Formaningishlovberilganxususiyatlari	+/-
<code>this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);</code>	<code>this->AutoScaleDimensions = System::Drawing::SizeF(11, 22);</code>	+
<code>this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;</code>	<code>this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;</code>	-
	<code>this->BackColor = System::Drawing::SystemColors::ActiveCaption;</code>	+
<code>this->ClientSize = System::Drawing::Size(284, 261);</code>	<code>this->ClientSize = System::Drawing::Size(230, 257);</code>	+
	<code>this->Cursor = System::Windows::Forms::Cursors::Hand;</code>	+
	<code>this->Font = (gcnew System::Drawing::Font(L "Times New Roman", 14.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, static_cast<System::Byte>(204)));</code>	+
	<code>this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::Fixed</code>	+

	3D;	
	this->HelpButton = true;	+
	this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources->GetObject(L"\$this.Icon"))));	+
	this->Margin = System::Windows::Forms::Padding(6, 6, 6, 6);	+
	this->MaximizeBox = false;	+
	this->MinimizeBox = false;	+
this->Name = L"Form1";	this->Name = L"Form1";	-
this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;	this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;	-
this->Text = L"Form1";	this->Text = L"Birinchi oynam";	+
this->ResumeLayout(false);	this->ResumeLayout(false);	-

Birinchinavbatdamuhitningo‘ziushbuxususiyatlarnio‘rnatishnivatarb ilashninazoratqiladi. Formadaquyidagio‘zgarishlaramalgaoshirilgan: formaningchiqishmaydoni, rangi, joriyo‘lchamlari, oynakursori, oynaningyozuv turi, ko‘rinishi, yordamchitugmao‘rnatilgan, ikonkasio‘zgartirilshan, holatio‘zgartirilgan, kattalashtirishvakichiklashtirish tugmalariolibtashlangan, oynaningsarlavhasio‘zgargan.

Bularniamalgaoshirishuchun
munutvaqtketadivahammasivizualbajariladi.
Bundaoldinqilingano‘zgarishlarochirib,
oxirgio‘zgarishlarqoshibboriladi.

2-3

Formaning hodisalaribilantani shibchiqamiz. O'z hodisanimi?

Hodisa bu bir vazifa bajarilgan jarayon bo'lishi mumkin, masalan, formani yaratish uchun 3 ta hodisa bo'lishi mumkin, formayaratilishidan oldin, formayaratilayotgan vaqt, formayaratilib bo'lingandan keyin.

Ularning ham 9.6- rasmda keltirilgan 10 ta guruhi bor. Ularning ba'zilariga izoh berib o'tamiz

- 1 **Action guruhi** Global hodisalar guruhi bo'lib, 8 ta hodisadan iborat
 - Click Formaga tugma bosilganda
 - DoubleClick Formaga tugma 2 marta bosilganda
 - MouseCapture Sichqonchaning holati o'zgarganda
 - Changed
 - MouseClicked Sichqoncha bosilganda
 - MouseDoubleClick Sichqoncha ikki marta bosilganda
 - ResizeBegin Formaga o'lcham o'zgarishi boshlanganda
 - ResizeEnd Formaga o'lcham o'zgarishi tugaganda
- 2 **Appearance guruhi** Formaning ichki hodisalarini uchun ishlatiladigan bitta xususiyat, ya'ni ubi bilan keyingimavzulartanishamiz. t
- 3 **Behavior guruhi** Formaning holatlarini uchun hodisalarini o'z ichiga oladigan 15 ta hodisadan iborat
 - ControlAdded Formaga biror boshqarish qurilganda
 - ControlRemoved Formadan biror boshqaruvo'chirilganda
 - FormClosed Forma yopilganda
 - FormClosing Forma yopilishidan oldi holatda
 - HelpButtonClicked Yordam tugmasi bosilganda
 - InputLanguageChanged Tilo'zgartirilganda
 - InputLanguageChanging Tilo'zgartirilayotganda
 - Load Forma yuklanayotganda
 - Shown Forma chiqirilganda
- 4 **Data guruhi** Ichki va tashqi ma'lumotlar bilan ishlash hodisalarini bo'lib, 2 ta hodisadan iborat, bu xususiyatlarga ma'lumotlar bazasini ishlaga datush unarlibo'ladi
- 5 **Drag** Formada harakatlanish hodisalarini bo'lib, 6

	dropguruhi	tahodisadaniborat	
	DragDrop	Birorbirharakatbo'lganda	
	DragEnter	Intertugmasibosilganda	
	DragLeave	To'xtashharakatibo'lganda	
	DragOver	Ketishharakatibo'lganda	
	GiveFeedback	Teskarialoqaqo'shish	
6	Focusguruhi	Formadafokuslarniboshqarishuchunbo'lib, tahodisano'zichigaoladi	6
	Activated	Fokusaktivbo'lganda	
	Enter	Fokusdama'lumotkiritilganda	
	Leave	Fokusdato'xtashsodirbo'lganda	
	Validated	Tekshirishtugaganidanso'ng	
	Validating	Tekshirishvaqtida	
7	Keyguruhi	Tugmalarbilashishlashhodisalariuchunbo'lib, tahodisadaniborat	6
	KeyDown	Tugmakelganda	
	KeyPress	Tugmabosilganda	
	KeyUp	Tugmaketganda	
	PreviewKeyDown	Tugmakelganda	
8	Layoutguruh i	Tartibguruhiformaniatrofidagilarbilanishlashtartibini nazoraqilashuchunrejalashtirilgan	
	DpiCahged	Joriyformatartibio'zgarganda	
	DpiCahgedAfterParent	Merosbergansinfartibio'zgargandankeyin	
	DpiCahgedBeforeParent	Merosbergansinfartibio'zgargandanoldin	
	Layout	Formatartibinio'rnatish	
	MdiChildActivate	Formasinfiningmerosxo'rifaolbo'lganda	
	Move	Formako'chirilganda	
	Resize	Formao'lchaminio'zgartirganda	
9	Mouseguruhi	Sichqonchabilanbo'ladiganhodisalarbo'lib, tahodisadaniborat	6
	MouseDown	Sichqonchakelganda	
	MouseEnter	Sichqonchajoriyholatida	
	MouseHover	Sichqonchaozginaharakatida	
	MouseLeave	Sichqonchagato'xtashbo'lganda	
	MouseMove	Sichqonchako'chganda	

	MouseUp	Sichqonchaketganda	
1	Property	Formaningxususiyatlario‘zgargandabajariladigan	12
0	Changedguruhi	hodisasibor	
	AutoSizeChanged	Avtomatiko‘lchamo‘zgarganda	
	AutoValidateChanged	Avtomatikteکشiribbo‘lganda	
	BackColorChanged	Fonrangio‘zgarganda	
	EnabledChanged	Holato‘zgarganda	

Hodisalarnio‘rnatishjudaqulayhisoblanadi.

Keraklihodisaningo‘ngtomonigasichqonchaniikkimartabossangiz, hodichamosbo‘lgandasturfranmentiniyozishuchunfunksiyayaratibberadi. Birmisolbilanko‘ribchiqamiz.

Buninguchun MessageBox sinfining show funksiyasidanfoydalanamiz. Bufunksiyastatikfunksiyabo‘lib, System::Windows::Formsnomlarfazosigategishli, 21 variantdayozilishimumkin. Unatijasifatida DialogResult tipiniqaytaradi. Uningargumentlari:

Text - Stringtipiniqabulqiladivamatnixabarsifatidachiqaradi

Caption - Stringtipiniqabulqiladivaxabarsarlavhasinichiqaradi

Buttons -

MessageBoxButtonstipiniqabulqiladivaxabarniqabulqilishtugmalarinichiqaradi, AbortRetryIgnore, OK, OKCancel, RetryCancel, YesNo, YesNoCancel qiymatlarniqabulqiladi.

Icon -

MessageBoxIconstipiniqabulqiladivaxabarniikonkasinichiqaradi, Asterisk, Error, Exclamation, Hand, Information, None, Question qiymatlarniqabulqiladi

defaultButton -

MessageBoxDefaultButtonstipiniqabulqiladivaxabarnitugmalarinibelgilaydi, Button1, Button2, Button3 qiymatlarniqabulqiladi.

Options -

MessageBoxOptionstipiniqabulqiladivaxabarnichiqishvarintlarinianiqlaydi, DefaultDesktopOnly, RightAlign, RtlReading, ServiceNotification qiymatlarniqabulqiladi.

Bufunksiyaning boshqa argumentlari ham bor, ularni keyinchalik o'rganish mumkin.

MessageBox sinfining show funksiyasidan foydalanish:

```
MessageBox::Show("Xabar", "Sar...",  
MessageBoxButtons::OK, MessageBoxIcon::Information  
);
```

Forma hodisalaridan foydalanish uchun 3 tasiniko'rich bchiqamiz:

1. Click hodisasini ishlatish uchun, uning o'ng tomoniga sichqonchanib osamiz.

Forma sinifidatayyor Form1_Click() nomli funksiyayaratiladi.

Bufunksiyanii chida quyidagicha fragmentni yozamiz:

```
MessageBox::Show("Click  
bo'ldi", "Xabar", MessageBoxButtons::OK,  
MessageBoxIcon::Information);
```

Shuningdek forma ning InitializeComponent()

funksiyasini chida avtomatik tarzda this->Click += gcnew

System::EventHandler(this,

&Form1::Form1_Click); hodisa qo'shib qo'yilgan.

Agar bu hodisa nito'liqo'chirmoqchibo'lsangiz, Click

hodisasining o'ng tomoniga yozilgan Form1_Click() nio'chirib, enter

tugmasini bosish yetarli o'zi avtomatik InitializeComponent()

funksiyasini chida gihodisa nio'chiradi.

Dasturning qismidan funksiyani o'chirish mumkin.

Quyidagi kikki hodisa ni ishlatish uchun bir masala olamiz.

Yordam tugmasi bosilganda, yordam tugmasi bolidi deb xabarchiqsiz.

Foydalanuvchi hade bjavob bersa, oynaga qaytadi. Agaryo'q deb bossa,

yolg'onchimisiz debchiqadiva foydalanuvchi, faqat javob nitanlaydi.

Javob tanlaganda oynaning sarlavhasi yolg'onchide bo'zgarsin.

Oynaholathodisalariorqalib tekshirib,

yana oynasarlavhasio'zgardidegan habarnichiqarsin.

2. HelpButtonClicked hodisasini ishlatish uchun,

uning o'ng tomoniga sichqonchanib osamiz.

Forma sinifidatayyor Form1_HelpButtonClicked

()

nomli funksiyayaratiladi.

Bufunksiyanii chida quyidagicha fragmentni yozamiz:

```
System::Windows::Forms::DialogResult result;  
result =  
MessageBox::Show("HelpBuutonClicked", "Xabar", MessageB  
oxButtons::YesNo, MessageBoxIcon::Hand,
```

```

MessageBoxDefaultButton::Button1);
        if(result ==
System::Windows::Forms::DialogResult::No) {
            MessageBox::Show("Yolg'nchimi siz
!", "Xabar", MessageBoxButtons::OK, MessageBoxIcon::Warn
ing);

            Form1::Text = "Yolg'nchi";
        }
    }
}

```

Shuningdek formaning InitializeComponent() funksiyasini ichida avtomatik tarzda this->HelpButtonClicked += gcnew System::ComponentModel::CancelEventHandler(this, &Form1::Form1_HelpButtonClicked); hodisa qo'yilgan.

3. TextChanged hodisasini ishlatish uchun, uning o'ng tomoniga sichqonchanib osamiz. Formasini fidatay yoki Form1_TextChanged() nomli funksiyayaratiladi. Bu funksiyani ichida quyidagicha fragmentni yozamiz:

```

MessageBox::Show("Forma sarlavhasi
o'zgardi", "Xabar", MessageBoxButtons::OK,
MessageBoxIcon::Information, MessageBoxDefaultButton::
Button3, MessageBoxOptions::RightAlign);

```

Shuningdek formaning InitializeComponent() funksiyasini ichida avtomatik tarzda this->TextChanged += gcnew System::EventHandler(this, &Form1::Form1_TextChanged); hodisa qo'yilgan.

Bu hodisalarni ishlatib ko'rsangiz, ancha narsalarni farq qilib ko'rish kerak.

Matn da bir joyda forma, bir joyda oyna, bir joyda form deb ishlatilgan bo'lsa, ularning hammasi bir ma'noda bo'ladi. Shubhan formaning xususiyatlarini bahodisalar bilan tanib chiqdingiz va uni o'rganishni davom ettirish kerak. Chunki, keyingi barcha ishlariz shukabi formalar bilan bog'liq. Keyinchalik formaning ma'lum bir xususiyatini bahodisalar bilan ishlaganda. Albatta unita'kidlabaytib o'tib ketamiz.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN
SAVOLLAR.

1. Integrallashganishlabchiqarishmuhitlariningqaydayturlarinibilasiz?
2. Qachondanboshlabintegrallashganmuhitlarjadalrivojlangan?
3. Turbo muhitlarnisanabbering.
4. Ob'ektgayo'naltirilgantilnibirinchibo'libintegrallashganmuhitlargakimvaqaysifirmajoriyqilgan.
5. Qandaymuhitdazamonaviymatnmuharrirlarikodiavtomatikbajarilishinita'minlash (koditugatish), muharririmuhtdajoriyyozilgankodisintaktikto'g'rivauningdavomibo'lishim mumkin.
6. Taniqli NetBeans integrallashganmuhitnidastlab Java dasturlashuchunqaysiuniversitetlabasiloyihasisifatidayaratgan.
7. Vizual Studio 97 haqidanimalarnibilasiz?
8. Visual Studio 2012 haqidanimalarnibilasiz?
9. ISO xalqarostandartlaribilanstandartlashtirilganvabirnecha implementations egabo'lganob'ektgayo'naltirilganko'ptillidasturlashplatformaniayting?
10. MS Visual Studio muhitoyinasinechtaqismdaniborat?
11. Windows ilovalarniyaratishuchunqaysitugmachalarmajmuasinibosishyetarli.
12. Qaysivositabirnechaqo'llab-quvvatlanadigantillardaharqandayyozilgandasturlarijrosiniboshqaradi.
13. CLR Empty Project qandayloyihayaratishuchunkkerak.
14. Windows Forms Control Library qandayloyihayaratishuchunkkerak.
15. Loyihaning External Dependencis papkasidaqandayfayllarsaqqlanadi.
16. [FILE] menyusiningvazifalarinisanabbering.
17. [TOOLS] menyusiningvazifalarinisanabbering.
18. Menyularvauskunalarpanelifoydalanuvchilarqandaypapkabuyruqlargakirishuchunbiryo'ldir.
19. Birbuyruqyaratishda, buninguchunbiravvalnimayaratishkerak.
20. Yangimenyularvauskunalarmajmuasianiqlashuchun, Visual Studiodaqandayjadvalfaylorqaliularnitasvirlashkerak.
21. Qatorlargaizohqo'yishtugmasiniayting.
22. Satrgaizohqo'shishtugmasiniayting.
23. Tegishlifunksiyavariantlarinichaqirishtugmasiniayting.
24. [Ctrl+Shift +.] tugmachalarmajmuasiqandayamalbajaradi.
25. [Ctrl+Shift+Tab] tugmachalarmajmuasiqandayamalbajaradi.

26. Oynauchun yangi form1 sinfi yaratiladi.
 Businfqaysisinfiningmerosxo'ri hisoblanadi
27. Kontekstmenyu oqaliformaningqanday funksiyalarini bajarish mumkin.
28. Formaning rejimlarini o'rnatishqa erdajoylashgan.
29. Ichki vatashqima'lumotlar bilan ishlash hodisalariga erdajoylashgan
30. MessageBox sinfining nechta argumentlaribor, vahar biriga izohbering.



AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	Form xususiyatlariga hodisalariga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan. 👉 Bundasturda giba'zi o'zgartirishlarini topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.
№ - sizning tug'ilgan kuningiz	
1. Formayaratilgan va uni o'lchamlariga (№*50, (№%2)*100) qiymatlarni o'rnatilgan.	
2. Formaning (№%6) - hodisa bajarilganini tekshiring? (0- Click, 1- DoubleClick, 2 - MouseClick, 3- MouseDoubleClick, 4- ResizeBegin, 5 – ResizeEnd)	

3. Formaning ((№/4)%6) - hodisasibajarinlanginitekshiring, bundaMessageBox::Showfunksiyasining (№%5)+1 taargumentidanfoydalanig ? (0- Click, 1- DoubleClick, 2 - MouseClick, 3- MouseDoubleClick, 4- ResizeBegin, 5 – ResizeEnd)
4. Formaning ((№/4)%8) - hodisasibajarinlanginitekshiring, bundaMessageBox::Showfunksiyasining (№%5)+1 taargumentidanfoydalanig ? (0-FormClosed, 1- FormClosing, 2-HelpButtonClicked, 3- Load, 4-Shown, 5-Activated, 6-Move, 7-Resize)
5. Formaning ((№/4)%7) - hodisasibajarilanginitekshiring, bundaMessageBox::Showfunksiyasining (№%5)+1 taargumentidanfoydalanig ? (0-MouseDown, 1-MouseEnter, 2-MouseHover, 3-MouseLeave, 4-MouseMove, 5-MouseUp, 6-AutoSizeChanged)

6. Formaning (N%8) - hodisasibajarilanga, tekshiringvanoto'g'rideganjavobda (M%7) - hodisanibajartiring, bundaMessageBox::Showfunksiyasining (N%3)+1 taargumentidanfoydalanig ?

Mouse – (0-MouseDown, 1-MouseEnter, 2-MouseHover, 3-MouseLeave, 4-MouseMove, 5-MouseUp, 6-AutoSizeChanged), NoMouse – (0-FormClosed, 1- FormClosing, 2-HelpButtonClicked, 3-Load, 4-Shown, 5-Activated, 6-Move, 7-Resize)

7. Formaning xususiyatlarini $((N_0 * 2) \% 12)$, $((N_0 + 6) \% 12)$, $((N_0 / 6) \% 12)$ bajarib tekshiring.

0- BackColor

1- BackgroundImage

2- Cursor

3- Font

4- FormBorderStyle

5- RightToLeft

6- Text

7- UseWaitCursor

8- Enabled

9- Language

10- locked

11- AutoScroll

- | |
|---|
| 11. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring,
nimalarnio‘rgandingizvabirtakrorlang.
12. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang. |
|---|

3.2. Komponentalar bilan ishlash.

 Komponenta tushunchasi va xususiyatlari, Standart komponentasining xususiyatlarini va hodisalari, Additional komponentasining xususiyatlarini va hodisalari, System komponentalarini komponentasining xususiyatlarini va hodisalari, Toolbox oynasi, Yangi komponentalarni qo‘shish, komponentalarni o‘zarob bog‘lash bo‘yicha nazariy ma’lumotlarni keltirilgan bo‘lib, nazariy bilimlarni asoslash uchun dasturlar tuzib ko‘rsatilgan. Bilimlarni mustahkamlash uchun 30 tan ariy savol va amaliy ko‘nikma va malakalarni rivojlantirish uchun 8 assisment topshiriq berilgan.

Butopshiriqlarni bajarish mavzu mustahkamlash uchun xizmat qiladi.

 **Kalitsa o‘zlar.** IDE, Visual C++, user interface, OYD, komponenta, xususiyatlari, hodisalari, Toolbox, Tab, Add Tab, Delete Tab, Mantlarni tahrirlash, Ma’lumotlarni ko‘rsatish (faqat o‘qish uchun), CheckBox, NumericUpDown, Label, Button, Additional komponentasi, Panel, TabControl, System komponentasi, MenuStrip, ContextMenuStrip.

 **Bilish shart bo‘lgan tushunchalar.** sinf va sinf ob’ekti, xususiyat, hodisa, forma, komponenta dasturlashga oid dastlabki tushunchalar hamda C++ tilini qo‘llab quvvatlovchi muhitda ishlashni bilish lozim.

 **Bil bolasi.** Visual C++ muhitida dasturlash, integrallashgan ishlab chiqarish muhiti, Toolbox, Komponenta tushunchasi va xususiyatlari, Toolbox oynasini, Toolbox oynasini tahrirlash, Yangi komponentalarni qo‘shish, Komponentaning xususiyatlarini guruhlari, loyiha boshqaruvi oynasini orqali ko‘rish, Komponentaning usullari (methods) va hodisalari (events), Standart komponentasining xususiyatlarini va hodisalari, Toolbox oynasidagi [Common Controls] Tab, Standart komponentalarni guruhlari, komponentalarni oynaga joylashtirish, Additional komponentasining xususiyatlarini va hodisalari, Additional komponentalarining guruhlari, Panel komponentasi, groupBox komponentasi, tabControl komponentasi, splitContainer komponentasi,

tableLayoutPanel komponentasi, flowLayoutPanel
 komponentasikabilarningxususiyatlariningvazifalari, System
 komponentasiningxususiyatlarivahodisalari, System
 komponentalariningguruhlari, Menyuvaukunalarpanellarini,
 Components tab komponentalari, contextMenuStrip
 ob'ektio'rnatishvakomponentalarbilanbog'lashusullari,
 shuningdekba'zibirxususiyatlarivahodisalario'rganishingiz mumkin.

REJA

1. Komponentatushunchasivaxususiyatlari.
2. Standartkomponentalariningxususiyatlarivahodisalari.
3. Additionalkomponentalariningxususiyatlarivahodisalari.
4. System komponentalariningxususiyatlarivahodisalari.

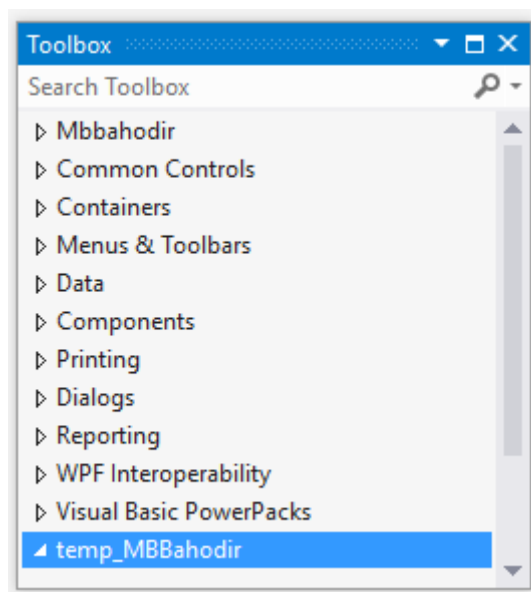
KIRISH

MS Visual Studio
 daishlashtamoyliningasosiyqulayliklaribukomponentalaridir.
 Foydalnuvchining UI nitahlilqilibqaralsa,
 undajudako'plabkomponentalarjoylashtirilgan. Bujudaosondir. MS Visual
 Studio Visual C++
 daturlivazifalarniamalgaoshirishkomponentalarimuvjud. Ularniasosan
 Toolbox debhamyuritiladi. Toolbox
 dagiharbirkomponentaningnomlarfazosivasinflarimavjud.
 Ulardanfoydalangandashusinflardanmerosxo'rolibyaratiladivabiroynada
 birxiltipdagikompanentadanbirnechtasidanfoydalanishmumkin.

Komponentatushunchasivaxususiyatlari. Komponentabu MS
 Visual Studio ningengasosiyqurolidir.
 Dasturchibularorqalitez dasturlashva IDE
 muhitningimkoniyatlaridanfoydalaniladi. Komponentalarto'plamini
 Toolbox debyuritiladi.

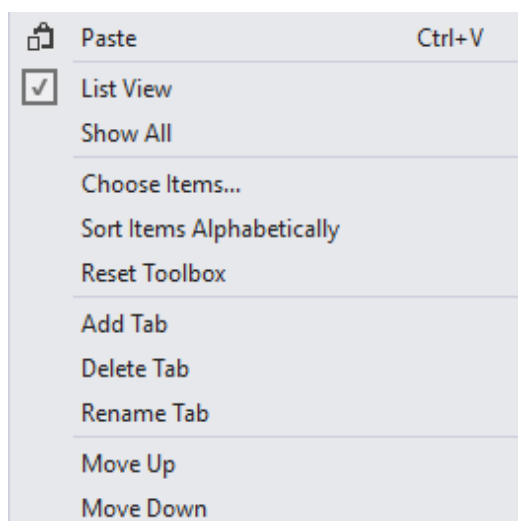
Toolbox
 interaktivoy nasimuhitningixtiyoriyjoyidajoylashganbo'ls himimumkin.
 Loyihalovchiunio'zigamosqilibjoylashtirishmumkin.
 Agarmuhitningoynasida Toolbox oynasiniko'r mayotganbo'lsangiz,
 menyudanfoydalanib, [view → Toolbox] buyruqlaribajaring. [Ctrl + W,
 X] tugmachalarmajmuasinihambosishorqali Toolbox
 oynasigao'tish/chaqirishmumkin.
 Muhitkomponentalarsoninechtadegansavolgajavobberishdanboshlaymiz.
 Muhitintegrallashganmuhitliginibilamiz,

shuninguchunundagikomponentalarsonioldindanxechkimaytaolmaydi. Yangikutubxonalarbijoriyqilishyangikomponentalarniolibkiradi. Standartfoydalanuvchilaruchun Toolbox oynasi 10.1-rasmdatasvirlangan.



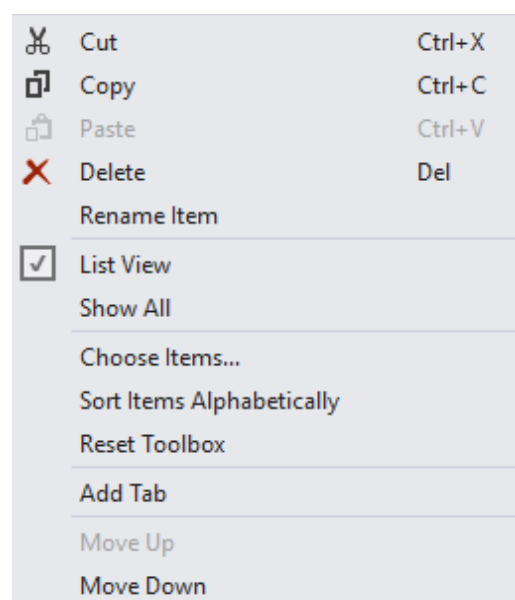
10.1-rasm. Toolbox oynasi.

Toolbox oynasitahrirlashmumkin. Buninguchunsichqonchaningo‘ngtugmasini Tab ga, ya‘nikomponentalarguruhiningnomigabosib, 10.2-rasmdagikontekstmenyuni, Tab ningichigakiribkomponentaniustigabossak 10.3-rasmdagikontekstmenyunichaqiramiz.



10.2-rasm.

komponentalarguruhiuchunkontekstmenyu.



10.3-rasm.

Komponentauchunkontekstmenyu.

Rasmlargaqarasak, ulardabirxilnomlibuyruqlaruchraydi, bularbutun Toolbox oynasiuchunxizmatqiladiva 5 tabo‘limdaniborat. Harbo‘limgavauningbuyruqlarigato‘xtalibo‘tamiz:

1- Bo‘lim.

Komponentalarustidaamalbajarishuchunmo‘ljallanganbo‘lib, [Cut] –
komponentanibufergaolishvajoyidano‘chirishuchun, [Copy] –
komponentaningnuxsasiniolishshuchun, [Paste] –
komponentaningolingannuxsasinijoylashtirish,
bunikomponentaguruhigaqo‘llashmumkin, [Delete] –
komponentaniguruhdano‘chiribtashlash, [Rename item] –
komponentanominio‘zgartirish.

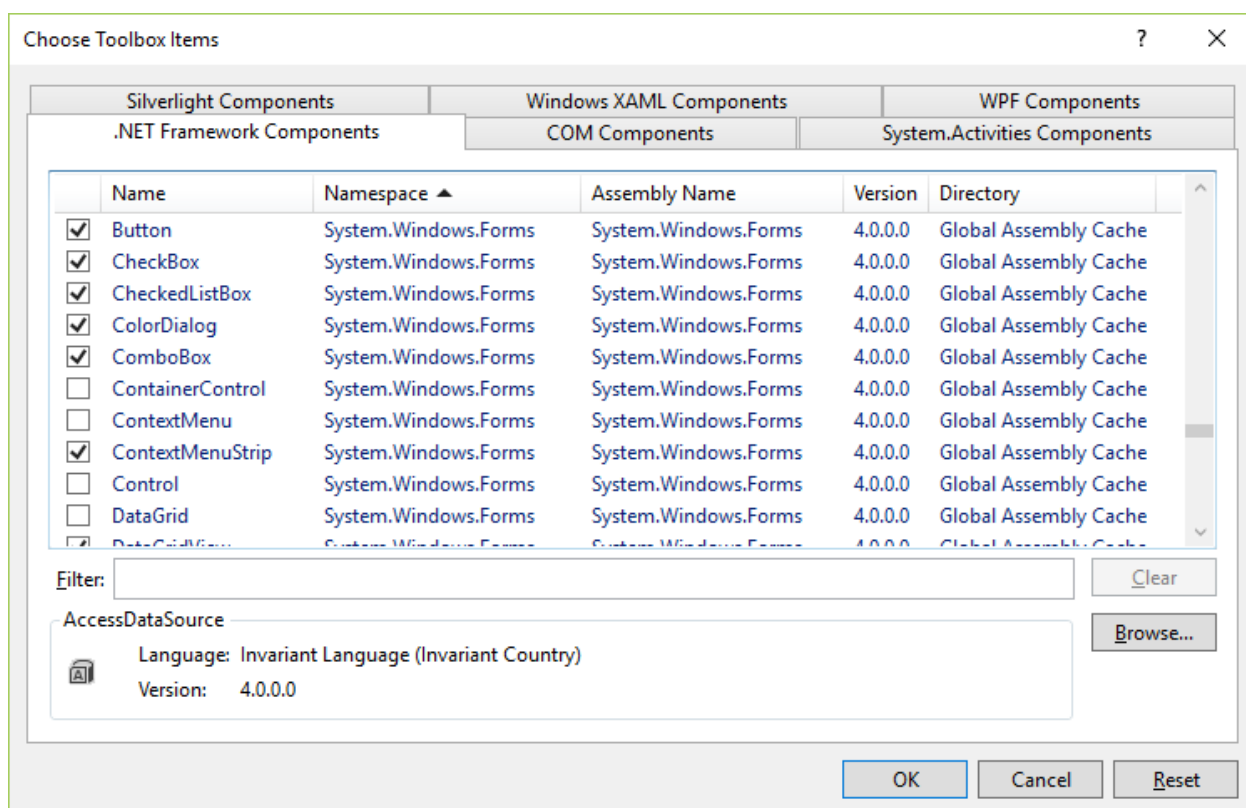
2- Bo‘lim. Toolbox oynasigaxizmatqiladi.[List View] –
tanlaganguruhkomponentalariniro‘yxatqilibchiqaradi, o‘chirilganbo‘lsa,
uskunalar to‘plamisifatidako‘rsatadi. [Show All] –
muhitdagibarchafaollashtirilgankomponentalarchiqaribberadi.
komponentlarjudako‘pligiuchunbuishlatisho‘tamushkulishdebhisoblaym
an. Tajribamdamenhambularninghammasiniishlatolmaganman.

3- Bo‘lim. Toolbox oynasigaxizmatqiladivakomponentalarhosilqilishuchunishlatiladi.
[Choose Items..] – yangikomponentalarniqo‘shishuchunishlatiladi, [Sort
Items Alphabetically] – komponentalarnialfavitbo‘yichasaralash, [Reset
Toolbox] – Toolbox
komponentalarnisozlamalarinistandartshaklgakeltirish.

4- Bo‘lim. Toolbox oynasiga Tab larni,
ya‘nikomponentaguruhlariustidaishlashimkoniyatiniberadi. [Add Tab] -
yangikomponentaguruhiqo‘shish, [Delete Tab] -
komponentaguruhiqo‘chirish, [Rename Tab] -
komponentaguruhiqaytanomlash,
standartnomlaganlariniahambuaamallarbilano‘zgartirishmumkin.

5- Bo‘lim.
komponentaguruhivakomponentalarningo‘rinlarinialmashtirishuchunishl
atiladi. [Move Up] – yuqorigako‘chirish, [Move Down] –
pastgako‘chirish.

Bularningbarchasimuhitningdasturchigamoslashuvchanligibildiradi.
Yangikomponentalarniqo‘shishko‘ribchiqaylik,
bubuyruqbosilishibilan 10.4-
rasmdatasvirlanganmuloqotoynasichiqadi.



10.4-rasm. Yangikomponentalarniqo'shish.

Burasmdankomponentalarningturlariko'rishmumkin, harbirbo'limi 1000 danortiqsinflargaega. Harbirsinfesa, bittakomponentahisoblanadi. Agartizimdamavjudbo'lmagankomponentalarnihamshuoyanaorqaliqu'shi shmumkin, masalan, MySql, oqimlarbilanishlash, OpenCV, OpenMP, Fuzzy ToolBox kabikomponentalarniqo'shishmumkin.

Komponentaningxususiyatlari.Harbirkomponentaningshundayxu susiyatlaribor.

Eslabko'ringOYDdasinfniyaratishvaqtidanimalaryaratilaredi.

Xuddishuningdek, harbirkomponentasinf, uningxususiyatlari (properties), usullari (methods) vahodisalari (events) mavjud.

Komponentaningxususiyatlariquyidagiguruhlargabo'linadi (10.1-jadvalgaqarang)

10.1-jadval. Komponentaningxususiyatlari

1	Accessibilit yguruhi	Buguruhdagixususiyatlarbirkomponentadao'zgarishlarn ianiqlashvao'rnatishuchunishlatiladi.
2	Appearanc eguruhi	Komponentaningtashqiko'rinishiuchunxususiyatlarnio'r natishgamo'ljallangan.
3	Behaviorgu ruhi	Komponentaningrejimxususiyatlarinio'rnatishgamo'ljallangan.
4	Dataguruhi	Komponentaniichkivatashqima'lumotlarbilanbog'lashx ususityayalariuchunishlatiladi.
5	Designguru	Komponentanilo'yihalashxususiyatlariuchunishlatiladi

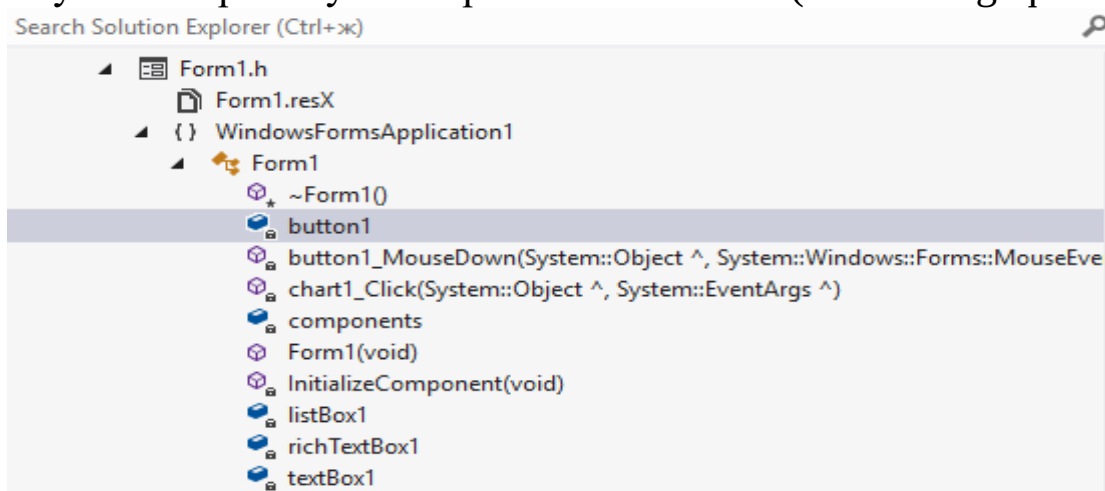
	hi	
6	Focusguruhi	Komponentadafokuslarniboshqarishxususiyatlariuchunishlatiladi
7	Layout guruhi	Komponentanitartiblashxususiyatlariuchunishlatiladi
8	Miscguruhi	Boshqaxususiyatlarnio‘rnatish, komponentaniturigaqarabo‘zgaribturadi.

Buxususiyatguruhlarikomponentatipigamoslashganholdadoimiyo‘zgaribturadivatipgamosxususiyatlarnianiqlabberadi.

Komponentaxususiyatlargaqiymatlarmaxsus, to‘plamdantantlash, sonli, matnli, mantiqiykabima’lumtiplarqimatinioladi. Agarmurakkabtipbo‘lsa, albattamuloqotoynasiorqalitanlashmumkin.

Komponentaxususiyatlarigaqiymatlarniberishvizuatamalgaoshiriladivaasosiiformagauningdasturfragmentlariyozibketiladi.

Buniloyihaboshqaruvoynasiorqaliko‘rishmumkin (10.5-rasmgaqarang)



10.5-rasm. Komponentalarningqo‘shilganligi.

OynaningInitializeComponentfunksiyasidahamkomponentalarningxususiyatlariko‘shilganiniko‘rishmumkin.

```
void InitializeComponent(void)
{
    this->button1 = (gcnew
System::Windows::Forms::Button());
    this->listBox1 = (gcnew
System::Windows::Forms::ListBox());
    this->SuspendLayout();
    //
    // button1
    //
    this->button1->Location =
```

```

System::Drawing::Point(208, 73);
    this->button1->Name = L"button1";
    this->button1->Size =
System::Drawing::Size(75, 23);
    this->button1->TabIndex = 0;
    this->button1->Text = L"button1";
    this->button1->UseVisualStyleBackColor
= true;
    this->button1->MouseDown += gcnew
System::Windows::Forms::EventHandler(this,
&Form1::button1_MouseDown);
    //
    // listBox1
    //
    this->listBox1->FormattingEnabled =
true;
    this->listBox1->Location =
System::Drawing::Point(363, 62);
    this->listBox1->Name = L"listBox1";
    this->listBox1->Size =
System::Drawing::Size(120, 95);
    this->listBox1->TabIndex = 1;

    // Form1 ...
    //

}

```

Komponentaningusullari (methods) vahodisalari (events)
quyidagiguruhlargabo‘linadi (10.2-jadvalgaqarang)

10.2-jadval. Komponentaningusullari (methods) vahodisalari
(events)

1	Actionguruhi	Komponentauchunglobalhodisalarfaollashtirishvama’lumbiralgoritmyozish
2	Appearanceguruhi	Komponentaningichkihodisalariuchunishlatiladivama’lumbiralgoritmyozish
3	Behavior	Komponentaholatlariuchunhodisalnifaollashtirish

	guruhi	
4	Dataguruhi	Ichki vatashqima'lumotlar bilan ishlash hodisalarini faollashtirish
5	Drag drop guruhi	Komponentaning harakatlanish hodisalarini faollashtirish
6	Focus guruhi	Komponentada fokuslarni boshqarish hodisalarini faollashtirish
7	Key guruhi	Komponentaning mostugmalar bilan ishlash hodisalarini faollashtirish
8	Layout guruhi	Komponentada atrofidagi lar bilan ishlash tartibini nazora qilish hodisalarini faollashtirish
9	Mouse guruhi	Komponentada sichqoncha bilan bo'ladigan hodisalarini faollashtirish
10	PropertyChanged guruhi	Komponentaning xususiyatlari o'zgarganda bajariladigan hodisalarini faollashtirish

Bu usullar va hodisalarining odatdagi hammasi ham vizuallash imkonini beradi. Shuning uchun agar komponentaning biror funksiya On bilan boshlangan bo'lsa, bu usul, e'tibor bilan tugagan funksiyalar bo'lsa, bu usul ko'proq hodisalar hisoblanadi.

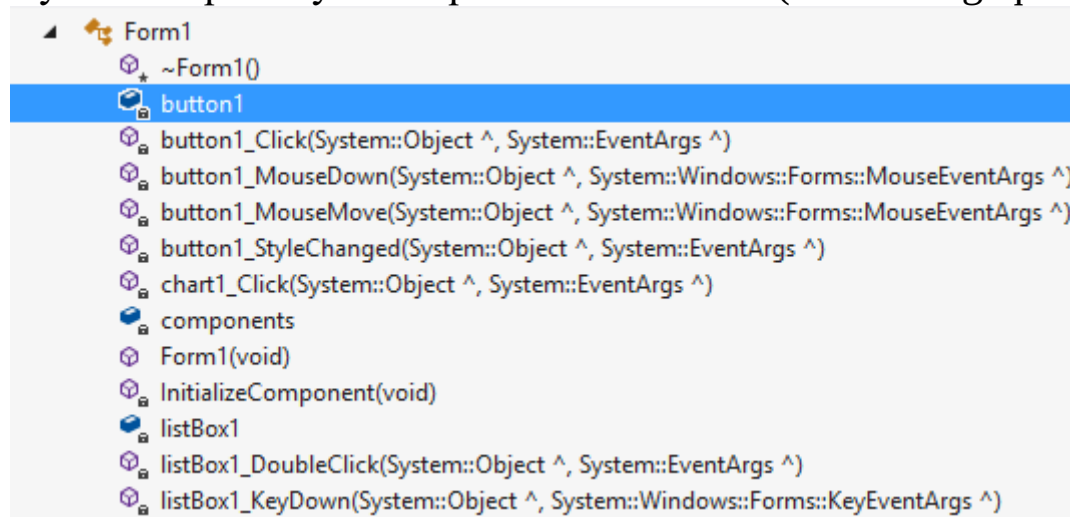
Guruhlar komponenta tipiga moslashgan holda doimiyo'zgarib turadigan tipga mos hodisalarini aniqlab beradi.

Komponenta hodisalarini funksiyasi sifatida yaratiladigan dasturchi uchun kerakli o'zining algoritmini yozadi.

Komponenta hodisalarini o'z shish vizual amalga oshiriladigan asosiy formaga uning dastur fragmentlarini yozib ketiladi.

Amma dasturchi o'z funksiyaning algoritmini yozish kerak.

Buni loyihani boshqaruv vositasidan o'z qalbi bilan mumkin (10.6-rasmga qarang)



10.6-rasm. Komponentalarning hodisalari.

Oynaning `InitializeComponent` funksiyasida ham komponentalarning hodisalariko‘shilganiniko‘rish mumkin.

```
void InitializeComponent(void){
    this->button1 = (gcnew
System::Windows::Forms::Button());
    this->listBox1 = (gcnew
System::Windows::Forms::ListBox());
    this->SuspendLayout();
    //
    // button1
    //
    // xususiyatlar
    this->button1->Click += gcnew
System::EventHandler(this, &Form1::button1_Click);
    this->button1->MouseDown += gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::button1_MouseDown);
    this->button1->MouseMove += gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::button1_MouseMove);
    this->button1->StyleChanged += gcnew
System::EventHandler(this,
&Form1::button1_StyleChanged);
    //
    // listBox1
    //
    // xususiyatlar
    this->listBox1->DoubleClick += gcnew
System::EventHandler(this,
&Form1::listBox1_DoubleClick);
    this->listBox1->KeyDown += gcnew
System::Windows::Forms::KeyEventHandler(this,
&Form1::listBox1_KeyDown);
    //
    // boshqa komponentalar
    // Form1
    // ... xususiyatlar
```

```

        // hodisalar
        this->Controls->Add(this->listBox1);
        this->Controls->Add(this->button1);
        // ... xususiyatlar
    }
private: System::Void
button1_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) { }
private: System::Void
listBox1_DoubleClick(System::Object^ sender,
System::EventArgs^ e) {}
private: System::Void
listBox1_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventArgs^ e) { }
private: System::Void
button1_MouseMove(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) { }
private: System::Void button1_Click(System::Object^
sender, System::EventArgs^ e) { }
private: System::Void
button1_StyleChanged(System::Object^ sender,
System::EventArgs^ e) { }

```

Shuningdek,
baʼzi komponentalar faol va faol boʻlmagan holatlarda boʻlishi mumkin.
Buhuddi windows
oyinlaridagi buyruqlarning foydalanishiga oʻxshash ishlatiladi.
Komponentalarni oʻz joyida ishlatishni ham bilish kerak va ularni qachon ishlatish kerakligini bilish ham muhim.

Standart

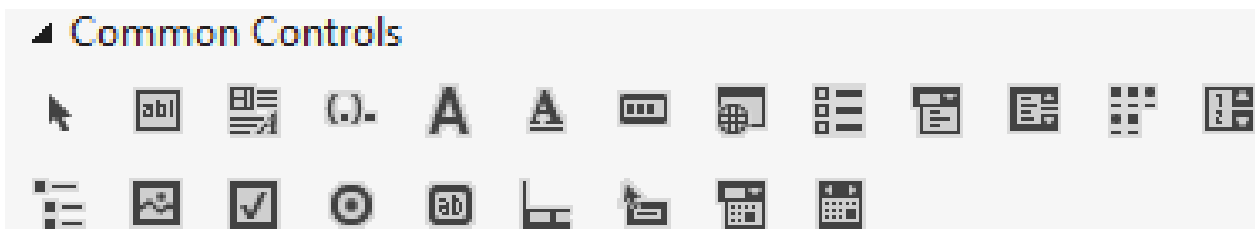
komponentasining xususiyatlarivahodisalari. Standart komponentalarga Toolbox oynasidagi [Common Controls] Tab dagi komponentalarni kiradi. Ularni tarkibini oʻzgartirish mumkin, yaʼni yangilarini qoʻshish va keraksizlarini oʻchirib tashlash mumkin. Shuningdek, agar maxsus ishlab chiqilgan yangi komponentalarni guruh boʻlsalardan ham foydalanish mumkin. Standart komponentalarni quyidagi guruhlarga boʻlinadivashuasosida ularni n qus xususiyatlarivahodisalari 95% bir xil boʻladi.

10.3-jadval. Standart komponentalarni guruhlarivavazifalari.

Komponentning guruhi	Komponentning nomi	Vazifasi va izoh
Mantlarni tahrirlash	TextBox	Matnni tahrirlash imkonini beradi, dastur ishlayotgan vaqtda foydalanuvchi yoki algoritmi yordamida matnni tahrirlash.
	RichTextBox	Oddiy va RTF formatida matnni tahrirlash imkonini beradi, dastur ishlayotgan vaqtda foydalanuvchi yoki algoritmi yordamida matnni tahrirlash.
	MaskedTextBox	Foydalanuvchi tomonidan kiritilayotgan matnlarni masxas belgisi bilan himoyalaydi
Ma'lumotlarni ko'rsatish (faqat o'qish uchun)	Label	Foydalanuvchi tomonidan to'g'ri dan to'g'ri tahrirlanmaydigan matn ko'rsatadi.
	LinkLabel	Veb-linksifatidagi matn ko'rsatadigan foydalanuvchi joriy matn ni bosganda boshqa oyna ga yoki veb-sayt ga havola faollashtiradi.
	ProgressBar	Foydalanuvchi uchun joriy amal bajarilish jarayonini ko'rsatadi.
Veb-sahifa ni ko'rsatish	WebBrowser	Foydalanuvchi oynasida veb-sahifa larni orqali harakat qilish imkonini beradi.
Ro'yxatdant anlash	CheckedListBox	Har bir katak bilan birgabo'lgan elementlarning takrorlanadigan ro'yxatini ko'rsatadi.
	ComboBox	Qalqib chiquvchi elementlar ro'yxatini ko'rsatadi
	ListBox	Matnlarni va grafik elementlar ro'yxatini ko'rsatadi
	ListView	Elementlarni to'rt xil ko'rinishdan birida ko'rsatadi. Ko'rinishlarga faqat matn, kichik piktogramma, matn, kattapiktogramma, matn va batafsil ko'rinish kiradi.
	NumericUpDown	Foydalanuvchilari yuqoridagilarni pastga tugmalari yordamida o'zgartirish mumkin bo'lgan raqamlar ro'yxatini ko'rsatadi.
	TreeView	Qo'shimcha kataklari yoki piktogrammalar bilan matn

Komponentning guruhi	Komponentning nomi	Vazifasi va izoh
	Window	aniborat bo'lish imkon bo'lgan tugunob'ektlarining yerarxiya ko'plamini ko'rsatadi.
Grafiklarni tasvirlash	PictureBox	bitmaps va piktoqramma kabi tasvir fayllarini ko'rsatadi.
parametrli qiymat	CheckBox	Matn chunki katakchavayorliqni ko'rsatadi. Odatda parametr larni belgilash uchun ishlatiladi.
	CheckedListBox	Har bir katak bilan birga bo'lgan elementlarning kaytari ladigan ro'yxatini ko'rsatadi.
	RadioButton	Yoqilgan yoki o'chirilish imkon bo'lgan tugmanini ko'rsatadi.
	Tracker	Foydalanuvchilarga shkalabo'ylab "Thumb" ni harakatlantirib shkala qiymatlarini o'rnatish imkonini beradi.
Buyruqlar	Button	Boshlanadi, to'xtaydi yoki jarayonni bekor qiladi.
	LinkLabel	Veb-linksifatidagi matnini ko'rsatadigan foydalanuvchi joriy matnni bosganda boshqa oyna ga yoki veb-saytga havola faollashtiradi.
	NotifyIcon	Fonda ishlaydigan dasturni ifodalovchi vazifalarni pan eli holatida barcha masidagi belgini ko'rsatadi.
Foydalanuvchi uchun yordam	HelpProvider	Elementlari uchun pop-up yordam oynasini yoki tez koryordam oynasini beradi.
	ToolTip	Foydalanuvchi nazorat bilan hovers nazorat maqsadiga bo'yicha qisqacha tavsifini ko'rsatadigan bir pop-up oyna beradi.
Vaqt	DateTimePicker	Foydalanuvchilarga sana yoki vaqtni tanlash imkonini beradigan grafik taqvimni ko'rsatadi.
	MonthCalendar	Foydalanuvchilarga sana qatorini tanlash imkonini beradigan grafik taqvimni ko'rsatadi.

Standart komponentlarni guruhlashning asosiy maqsadi, guruhlarga mos xususiyatlar va hodisalarini bir xil bo'ladir. Bu komponentlarning Toolbox oynasida quyidagicha ko'rinishga ega (10.7-rasmga qarang)



10.7-rasm. Standartkomponentalarko‘rinishi.

Bukomponentalarnioynagajoylashtirishuchunsichqonchaningchapb ittabosiladivaoy nagakeltiribchaptomoniyanabirmartabosiladivaoy nadata nlangankomponentaningnusxasipaydobo‘ladi.

Bunusxaniob‘ektdebaytamiz.

Chunkieslabko‘ring,

sinfningnusxasibuob‘ektdebaytiladi.

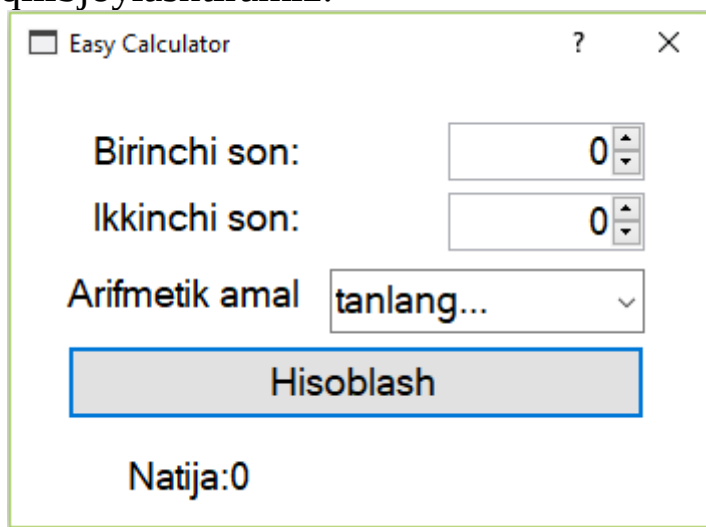
Ob‘ektningustigasichqonchaningo‘ngtugmasinibossangiz, ubilanbajarishmumkinbo‘lganamallarniko‘rasiz.

Agarob‘ektdannusxaolsangizuningsupersinfining, ya‘nishuob‘ektipidagiboshqabirob‘ektyaratiladi.

Ikkita birxil tipdagi ob‘ektlarning xususiyatlarivahodisalariga alohida alohida ishlov beriladi.

Standartkomponentalarnixususiyatvahodisalariniko‘ribchiqishuchunarifmetikamallarnibajaruvchidasturyaratishmasalasiniolamiz.

Mantiqiyjihatdansonlarnikiritishuchun 2 ta NumericUpDown, matnliyo zuvlaruchun 4 ta Label, hisoblashuchun bitta Button, amallaruchun bitta CheckBox komponentlariniolamizvaoy naga quyidagi 10.8-rasmdagidekqilibjoylashtiramiz.



10.8-rasm.

Standartkomponentalarnixususiyatvahodisalaridanfoydalanish.

Bumasalanilo yihasi fatida amalga oshirishuchun quyidagi qadamlarba jariladi:

1-qadam. Formaxususiyatlarinio‘rnatish: BackColor xususiyatiga oq rang; font xususiyatiga 14 o‘lchamliyo zuv; Icon

gamaxsusikinka; MaximizeBox xususiyatiga false qiymat; MinimizeBox xususiyatiga false qiymat; Size xususiyatigayangio‘lchamlar; StartPosition–CenterScreen holati; Text xususiyatiga “Easy Calculator” matnni yoziladi;

Buniqandayajratishshumkin, ya‘nio‘zgarganxususiyatlarni, qiymatiningmatniyog‘onqorarangakirganlario‘zgartirilganhisoblanadi. Buo‘zgarishlarnidasturfragmentko‘rinidakeltiishhammumkin.

Budasturfragmentiniqo‘ldayozishkerakemas, ammotushunarlibo‘lishiuchunkeltiramiz.

BuniInitializeComponentfunksiyasiichidakerakliizohlarbilankeltirilganbo‘ladi.

```
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(351, 225);
this->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->HelpButton = true;
this->Icon = (cli::safe_cast<System::Drawing::Icon^
>(resources->GetObject(L"$this.Icon")));
this->MaximizeBox = false;
this->MinimizeBox = false;
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
this->Text = L"Easy Calculator";
```

2-qadam.

Label1

ob’ektingxususiyatinio‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label1->Text = L"Birinchi son:";
```

3-qadam.

Label2

ob’ektingxususiyatinio‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label2->Text = L"Ikkinchi son:";
```

4-qadam.

Label3

ob’ektingxususiyatinio‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label3->Text = L"Natija:";
```

5-qadam.

numericUpDown1

ob’ektingxususiyatinio‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->numericUpDown1->TextAlign =  
System::Windows::Forms::HorizontalAlignment::Right;
```

6-qadam. numericUpDown2
ob'ektingxususiyatinio'rnatishdasturfragmentisifatidakeltiramiz.

```
this->numericUpDown2->TextAlign =  
System::Windows::Forms::HorizontalAlignment::Right;
```

7-

qadam. comboBox1ob'ektingxususiyatlarinio'rnatishdasturfragmentisi
fatidakeltiramiz.

```
this->comboBox1->Items->AddRange(gcnew cli::array<  
System::Object^ >(5) {L"[+] (qo'shish)", L"[-]  
(ayirish)", L"[*] (ko'paytirish)",  
L"[/] (butun bo'lish)", L"[%]  
(qoldiq bo'lish)"});  
this->comboBox1->Text = L"tanlang...";
```

8-

qadam. label4ob'ektingxususiyatinio'rnatishdasturfragmentisifatidakel
tiramiz.

```
this->label4->Text = L"Arifmetik amal";
```

9-

qadam. button1ob'ektingxususiyativahodisanio'rnatishdasturfragmenti
sifatidakeltiramiz.

```
this->button1->Text = L"Hisoblash";  
this->button1->Click += gcnew  
System::EventHandler(this, &Form1::button1_Click);
```

button1_Click funksiyasigaquyidagialgoritmniyozamiz.

```
int one =  
Convert::ToInt16(Math::Round(numericUpDown1->Value,  
0));  
int two = Convert::ToInt16(numericUpDown2->Value);  
int result = 0;  
label3->Text = "Natija:";  
  
switch(comboBox1->SelectedIndex){  
    case 0: result = one + two; break;  
    case 1: result = one - two; break;  
    case 2: result = one * two; break;  
    case 3: result = one / two; break;
```

```

        case 4: result = one % two; break;
        default: result = 0;
                break;
    }
    label3->Text += result.ToString();

```

Algoritmgazohbrishshartemas, chunkiko‘pkamchiliklar mavjud, masalan, bo‘lisharifmetikamalida. Ammoba‘zixulosalarnichiqarishmumkin. [numericUpDown1->Value] ob’ektningtanlanganhaqiqiytipdagiqiymatiniqaytaruvchixususiyat, [Convert::ToInt16] – ixtiyoriytipnibutuntipgaalmashtiribberadi, [label3->Text] – ob’ektningmatnxususiyatiniio‘zgartirishmumkin, [comboBox1->SelectedIndex] – tanlagichningtanlanganindeksiniqaytaradi, [label3->Text += result.ToString();] – ob’ektningmatningamatnqo‘shishvatipnisatrlitipgao‘tkazishko‘rsatilgan.

Dasturfragmentida 3 tabelgi [::] – cinfningusullarigamurojaatqilish, [->] – ob’ektningxususiyatigamurojaatqilish, [.] ob’ektningqiymatiustidabajarilishikerakbo‘lganamal. Visual C++ dajudako‘pusul, vafunksiyalarniishlatishtamoyllaribor, ularnifaqatkompilyatorningimkoniyatigaqarabajratishmumkin.

Standartkomponentalarningba‘zixususiyatlarivahodisalari, ularuchunba‘ziusullarniishlatiyuko‘rdik. Bularnichuqurroqo‘rganishuchunko‘proqamaliyishlarnibajarishkerak.

Additional

komponentasiningxususiyatlarivahodisalari. Bustandartkomponentalarnibilashishlashuchunqo‘shimchakomponentlardir.

Bularasosanboshqaruvelementlarniguruhlashuchunishlatiladi.

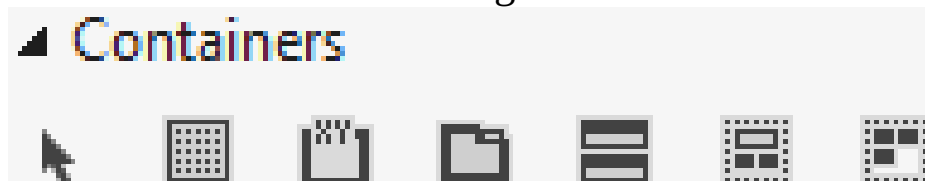
Quyidagijadvaldaqo‘shimchakomponentalarnivavazifalarinikeltiramiz.

10.4-jadval. Additional komponentalariningguruhlarivavazifalari

Komponenta-ningguruhi	Komponentaning nomi	Vazifasivaizoh
Boshqaruvla relementlarn iguruhlash	Panel	elementlarniboshqarishuchunto‘plaminig uruhlaydi
	GroupBox	Nomlanganelementlarniboshqarishto‘plaminiguruhlaydi
	TabControl	Guruhlanganob’ektlarnisamaralitashkileti shvakiritishuchunsohalisahifanitaqdimeta di.

	SplitContainer	Birbiridan ajratilgan ikki panel beradi. SplitContainer – Splitter o'rnigamo'ljallangan.
	TableLayoutPanel	Kontent dinamik satr va ustunlar bo'rt panelini ifodalaydi.
	FlowLayoutPanel	Dinamik ravishda kontentni gorizontal yoki vertikal joylashtiradigan panelni ifodalaydi.

Bu komponentalar Containers nomli tabda joylashgan. Ularning ko'rinishi 10.9-rasmda keltirilgan.



10.9-rasm. Containers komponentalari.

Bu komponentalarining xususiyat va hodiralari alohida to'xtalib o'tamiz. Vazifalar elementlarini guruhlab boshqarishdan iborat, shu maqsadda ishlatiladigan xususiyat va hodisalarini keltiramiz.

Panel komponentasi. Panel – bu oddiy to'rt tomonidan chegaralangan hududga o'xshaydi. Uni butun sohani egallashi uchun o'ng tomonidagi [Panel Task] dagi [dock in parent container] ni bosish yetarli. [undock in parent container] ni tanlab, oldingi holatiga kelish mumkin.

Ob'ektni harakatlantirish uchun chaptomondagi **4** tatomonga o'naltirilgan tugma chadan foydalaniladi.

AutoSize - xususiyati true/false qiymatlarini qabul qiladi. Odatda chin qiymat ko'proq ishlatiladi, chunki ob'ektga joylashtirilgan boshqaruvi elementlarini nigo'lchamlariga mos holda o'lchamni sozlaydi. Joriy holatda yolg'on qiymatga ega bo'ladi.

Cursor - xususiyati sichkonchani kursorko'rinishini o'rnatish imkonini beradi.

BorderStyle - ob'ektni atrofdan chegaralarining ko'rinishlarini o'rnatish uchun ishlatiladi.

AutoScroll – ob'ektda birlashtirilgan elementlarni to'liq ko'rinishga, harakatlaguvchi tugmalarni o'rnatish va olib tashlash uchun ishlatiladi. Bu xususiyat true/false qiymatlarini qabul qiladi.

AutoScrollMargin - bu xususiyat harakatlanuvchi tugma chalarining masofalarini o'rnatida, Width va Height ichki xususiyatlarini beror.

AutoScrollMinSize	-
buxususiyatharakatlanuvchitugmachalarengkichiko‘lchamlarinio‘rnatadi . Wight va Height ichkixususiyatlaribor.	
Dock	—
ob’ektnioynaningqaysiqismidajoylashtirishkerakliginio‘rnatadi.	
Margin	—
ob’ektchetqismlaridaqoldirilishikerakbo‘lganmasofalarnio‘rnatadi. Uning 5 taxususiyatibor, barchasi, chap, o‘ng, yuqorivapast.	
groupBox komponentasi. groupBox	-
nomlaganelementlargauruhinibirlashtirishhuchunishlatiladi.	
Ob’ektniharakatlantirishhuchunchaptomondagi	4
tatomongayo‘naltirilgantugmachadanfoydalaniladi.	
Text - buxususiyatelementlargauruhininominioynagachiqaradi	
AutoSize - xususiyati true/false qiymatlarniqabulqiladi. Elementlarningo‘lchamlarigamosob’ektgao‘zlchamajratadi.	
BackColor – ob’ektningfonxususiyatiranginio‘rnatadi.	
BackgroundImage - ob’ektningfonxususiyati Image	
ob’ektlarnio‘rnatadi.	
Dock	-
ob’ektnioynaningqaysiqismidajoylashtirishkerakliginio‘rnatadi.	
FlatStyle – ob’ektningstilxususiyatinio‘rnatadi.	
Font	—
buxususiyatob’ektningyozuvlariuchunaniqbirdandartyozuvnio‘rnatishuchunishlatiladi.	
MaximunSize – ob’ektningengkattao‘lchaminibelgilaydi.	
MinimumSize - ob’ektningengkichiko‘lchaminibelgilaydi.	
Padding	—
ob’ektgachabo‘lganmasofalarnio‘rnatishhuchunishlatiladi. Uning 5	
taxususiyatibor, barchasi, chap, o‘ng, yuqorivapast.	
RightToLeft – ob’ektningmatninio‘ngtomonidano‘rnatish.	
TabIndex — ob’ektningtabtugmasibosilganda, nechinchinavbatdabo‘lishinibelgilashnio‘rnatadi.	
tabControl komponentasi. tabControl	
buoddiyto‘rttomonidanchegaralanganhududgao‘xshaydi.	
Unigayangisahifaqo‘shishyokisaxifanio‘chirishhuchuno‘ngtomonidagi [tabControl Task] dagi [Add Tab] va[Remove Tab] nibosishyetarli.	
Ob’ektniharakatlantirishhuchunchaptomondagi	4
tatomongayo‘naltirilgantugmachadanfoydalaniladi.	

Anchor

ob'ektniaftoryokioynabo'yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto'rttaqiymatio'ng, chap, yuqori, pastboryuBularningo'zaroaralashuvidanfoydalaniladi.

Appearance

buxususiyatob'ektningtugmachalariningko'rinishinio'rnatishuchunishlatiladi. Uningqiymatlari [Normal], [Buttons], [Flat Buttons] qiymatlarniqabulqiladi.

DrawMode

ob'ektningxususiyatinichizilganformasinio'zgartrishuchunishlvtiladi.

TabPages

buxususiyatto'plamqiymatlarniqabulqiladivauningnechtasahifabo'lishini ta'minlaydi.

Ubilanishlagandaharbirsahifaningalohidaxususiyatlarigaishlovberishmu mkin.

TabPage1

buob'ektningsaxifaxususiyatlarnio'rnatishuilatiladigganob'ekthisoblana divapanelob'ektidekxususiyatlargaega.

tabStop

ob'ektbo'yichatabtugmasiningharakatlanishio'rnatishvao'chirishqiymati nio'rnatadi.

splitContainer

komponentasi.

splitContainer

ikkigaaajratilganelementlarniboshqarishuchunto'plamnihosilqiladi.

Unibutunsohaniegalashiuchuno'ngtomonidagi [splitContainer Task] dagi [dock in parent container]nibosishyetarli. [undock in parent container]nitanlab, oldingiholatigakelishmumkin. Shuningdek, [Horizontal Splitter Orientation]

ob'ektramkalarinigorizontalo'rnatadiva [Vertical Splitter Orientation] esavertikalo'rnatadi. Ob'ektniharakatlantirishuchunchaptomondagi 4 tatomongayo'naltirilgantugmachadanfoydalaniladi.

Anchor

ob'ektniaftoryokioynabo'yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto'rttaqiymatio'ng, chap, yuqori, pastboryuBularningo'zaroaralashuvidanfoydalaniladi.

BackColor - ob'ektningfonxususiyatiranginio'rnatadi.

BorderStyle

ob'ektningatrofchegaralariningko'rinishlarinio'rnatishuchunishlatiladi.

FixedPanel

— ob'ektuchunasosiypanelnio'rnatishuchun, ya'niikkitapaneldanbirasosiybo'lishikerak.

isSplitterFixed

ob'ekt uchun asosiy panelni o'rnatish amal qilish yoki qilmasligini o'rnatadi. xususiyati true/false qiymatlarni qabul qiladi.

Orientation

ob'ektning panellarini horizontal yoki vertikal o'rnatish uchun ishlatiladi. Ikkita Horizontal va Vertical qiymat qabul qiladi.

splitContainer1.Panel1

ob'ektning panel xususiyatlariga ishlov berish uchun panel xususiyatlarini o'chib beradivau uchun xususiyatlarni o'rnatish mumkin.

Panel1.Coolapsed

– ob'ektda panel 1 ni to'liq qilib o'rnatadi. Bunday xususiyat barcha panellari uchun o'rnatiladi.

Panel1.MinSize

ob'ektda panelning eng kichik o'lchamini o'rnatishdan iborat.

RightToLeft – ob'ekt panellarini o'ngdan o'rnatish xususiyati.

tableLayoutPanel – komponentasi. tableLayoutPanel

bujad va kabitartiblangan panellarni o'plamiorqali elementlarni birlashtirish ob'ektidir.

AutoScoroll

– ob'ektda birlashtirilgan elementlarni to'liq ko'rinmasa, harakatlaguvchi tugmalarni o'rnatish va olib tashlash uchun ishlatiladi. Buxususiyat true/false qiymatlarni qabul qiladi.

AutoScorollMargin

buxususiyat harakatlanuvchi tugma chalarchetmasofalarini o'rnatida, Wight va Height ichki xususiyatlaribor.

AutoScorollMinSize

buxususiyat harakatlanuvchi tugma chalarenig kichik o'lchamlarini o'rnatadi. Wight va Height ichki xususiyatlaribor.

CellBorderStyle

ob'ektning yacheykalarning chet chegaralarining stilini belgilaydi.

ColumnCount – ob'ektdagi ustunlar sonini o'rnatish.

Columns

ob'ekt ustunlari uchun xususiyatlarni o'rnatish uchun alohida muloqot oynasi ochiladi.

GrowStyle

ob'ektga ustun yoki qator yoki ikkalasini qo'shish xususiyatini o'rnatish.

RowCount – qatorlar soni

Rows

ob'ekt qatorlari uchun xususiyatlarni o'rnatish uchun alohida muloqot oynasi ochiladi.

FlowLayoutPanelkomponentasi. **FlowLayoutPanel** – ketmaketlikkabitartiblanganpanellarto‘plamiorqalielementlarnibirlashtiri shob’ektidir.

Anchor - ob’ektniaftoryokioynabo‘yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto‘rttaqiymatio‘ng, chap, yuqori, pastboryuBularningo‘zaroaralashuvidanfoydalaniladi.

BackColor - ob’ektningfonxususiyatiranginio‘rnatadi.

AutoSize - xususiyati true/false qiymatlarniqabulqiladi. Elementlarningo‘lchamlarigamosob’ektgao‘zlchamajratadi.

WrapContents - ob’ektdasontentlarniketmaketliginiustuvorqilishyoki qilmasliknio‘rnatadi . Xususiyat true/false qiymatlarniqabulqiladi.

Buqo‘shimachakomponentalarningxususiyatlarihaqidama’lumotlar niberildi. Ularninghodisalaribirbirigao‘xshashbo‘lib, standartkomponentalarningdanfarqijudakam. Shuninguchunhodisalarigato‘xtalaibo‘tirmaymiz. Shuningdek, qo‘shimachakomponentalarjudako‘p, buniko‘rishuchun Tollbox oynasigasichqonchaningo‘ngtugmasinibosingva [Shown All] buyrug‘initanlang.

System **komponentasiningxususiyatlarivahodisalari.**Bukomponentalargaasosa noperatsiontizimbilanishlaydigankomponentalarkiradi. Ularnixususiyatlarimurakabhisoblanadi. Chunkibirginamenyuyaratishuchunstandartvaqo‘shimchakomponentalar hamfoydalaniladi. Shuninguchunumumiyholdabukomponentalarningvazifalarinikeltiribo‘ta miz.

10.5-jadval. System komponentalariningguruhlarivavazifalari

Komponenta ningguruhi	Kompon entaning nomi	Vazifasivaizoh
Bo‘yruqlar	ToolStrip	Microsoft Windows, Microsoft Office, Microsoft Internet Explorer bo‘lishimumkinuskunalarmajmuasiyokifoydalan uvchiinterfeysiniyaratadi.Toolstrip control Asboblarpanteliboshqaruvinialmashtirishgamo‘lj allangan.

	ToolStrip Container	ToolStrip elementlarini ajralmasto'plamini beradi (birikib, sohasida gorizontal yoki vertikal almashish)
Ma'lumotni ksettirish (faqato'qishelementlari uchun)	StatusStrip	Windowsning holat satrini beradi. Statusstrip - Statusbar o'rnini bosadi. Statusstrip maxsus xususiyatlar tartibini o'z ichiga oladi,
Menyuni boshqarish elementlari	MenuStrip	Forma uchun menyutizimini beradi. Menustrip - Mainmenu o'rnini bosadigan yuqoridagi konteyner hisoblanadi. Bundan tashqari, asosiy ishlov berish va hujjat interfeysi imkoniyatlarini beradi. Funktsionaligi ToolStripItem olingan bo'lsa-da, ToolStripDropDownItem va ToolStripMenuItem bilan birga ishlatiladi.
	ContextMenuStrip	Kontekst menyuni ifodalaydi. ContextMenuStrip bu ContextMenu o'rnini bosadi. ContextMenuStrip bilan birga ishlatiladigan qancha noma'ngi tugma sinib olishda avtomatik ravishda kontekst menyusi namoyon bo'ladi.



10.10-rasm.

Menyu va skunalar panellarini yaratish uchun komponentlarning ko'rinishi

Menyu va skunalar panellarini yaratish uchun komponentlardan tashqari tizim bilan ishlaydigan komponentlarga quyidagi 10.6-jadvaldagi komponentlar ham qiradi, ular Components tab ga kiritilgan. 10.6-jadval. Components tab ning komponentlari.

Komponentning nomi	Vazifasi va izoh	
Backgr	Alohida ajratilgan oqim uchun amal bajariladigan	

BackgroundWorker	BackgroundWorker komponentaning nusxasini yaratadi	
DirectoryEntry	Active Directory xizmatibilano 'zarobog'lanib birlashiva uning ierarxiyasidan ob'ektyokitugunlarni inkapsulyatsiyaoladigan DirectoryEntry komponentaning nusxasini yaratadi	
DirectorySearcher	Active Directory daso'rovlarni bajarish imkonini beradigan DirectorySearcher komponentaning nusxasini yaratadi	
ErrorProvider	Formada elementni boshqarishdagixatofoydalanuvchigako'rsatish uchun ErrorProvider komponentaning nusxasini yaratadi	
EventLog	Tizim va foydalanuvchining hodisalar jurnaliga bog'lanish imkonini beradigan EventLog komponentaning nusxasini yaratadi	
FileSystemWatcher	Ruxsat berilgan papkava fayllarga o'zgartirishlarni kiritish imkonini beradigan FileSystemWatcher komponentaning nusxasini yaratadi	
HelpProvider	Tezkor va foydalanuvchining yordam oynasini chaqirish imkonini beruvchi HelpProvider komponentaning nusxasini yaratadi	
ImageList	Image ob'ektlarining to'plamini boshqarish usullarini ta'minlovchi ImageList komponentaning nusxasini yaratadi	
MessageQueue	Navbatdagixabarlar ni o'qish imkoniyatini beruvchi MessageQueue komponentaning nusxasini yaratadi	
PerformanceCounter	Windowsning hisoblagichlari bilan birlashib foydalaniladi, yangi toifalar va misollarni yaratish, hisoblagichlardan qiymatlarni o'qish va hisoblagich ma'lumotlariga asoslangan hisob-kitoblarni bajarish uchun PerformanceCounter komponentaning nusxasini yaratadi	
Process	Tizimdagijarayonlarni bilan bog'liq ma'lumotlarni o'xtatish, ishgatushirish va o'zgartirish uchun Process, komponentaning nusxasini yaratadi	
SerialPort	sinxron va oqea imkoniyatlar uchun SerialPort komponentaning nusxasini yaratadi	
ServiceController	Mavjud xizmatlarni, shu jumladan, boshlang'ich va o'xtatish xizmatlarini boshqarish va ularga buyruqlar berish uchun ServiceController komponentaning nusxasini yaratadi	
Timer	Windows	

	ilovalar uchun vaqtga asoslangan vazifalarni kiritish uchun Timer komponentaning nusxasini yaratadi	



10.11-rasm. Components tab gakiritilgan komponentalarning ko'rinishi.

Bu komponentalardan kontekstni yaratish va boshqa ob'ektlarga o'rnatishni ko'rib chiqamiz.

Masala. Oynada ikki kontekstni yaratish kerak. Birinchisi oyna uchun, ikkinchisi RichTextBox ob'ekti uchun bo'lsin.

Birinchi kontekstda oynayopish, kattalashtirish, kichiklashtirish, yig'ishtirish, tugmachalarni yoqish va o'chirish bo'yruqlari bo'lsin. Ikkinchi kontekstda matnni tahrirlash uchun buferga nusxalash, buferga nusxalash va o'chirish, nusxa o'lish, ob'ektni o'ralash, gorizontaal va vertikal harakatlanish tugmachalarni qurish va o'chirish ham alla ribo'lsin.

Yangi loyihani yaratamiz va biroyunabo'ladi. Oynaning bir chetiga RichTextBox ni o'rnatamiz. Menyu va uskunalartabidan contextMenuStrip ob'ektidan ikki ta o'rnatamiz. Ular formada ko'rinishi mumkin. Ular loyiha ning ishchi maydoni pastki qismida joylashadi (10.12-rasmga qarang).

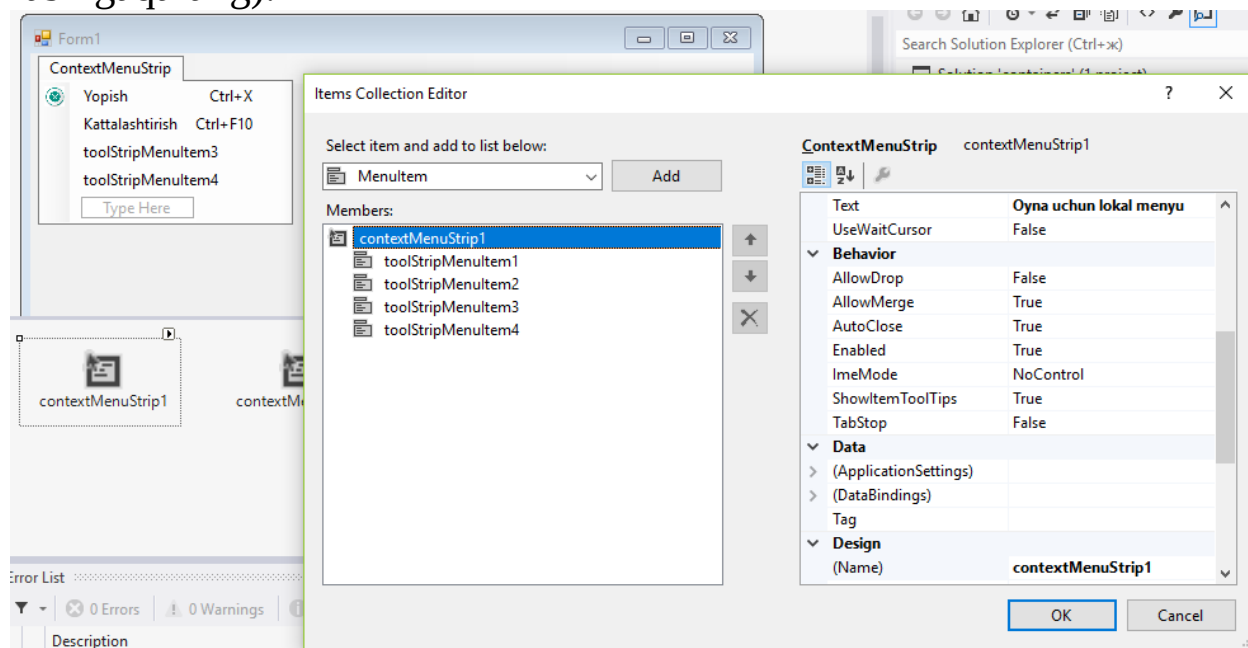


10.12-rasm. contextMenuStrip ob'ekti o'rnatilgan ko'rinishi.

Bularni kerakli ob'ektlar bilan bog'laymiz.

Oynaga birinchi kontekstni yuni [contextMenuStrip1], [RichTextBox1] ob'ektiga ikkinchi kontekstni yuni [contextMenuStrip2] o'rnatamiz. Buning uchun oynaning [ContextMenuStrip] xususiyatiga [contextMenuStrip1] niva RichTextBox1 ob'ektning [ContextMenuStrip] xususiyatiga [contextMenuStrip2] ni o'rnatamiz. Agar bitta kontekstni ikki ta ob'ektga o'rnatilsa, kontekstning buyruqlariga joriy qilingan algoritmlarni bajaraveradi.

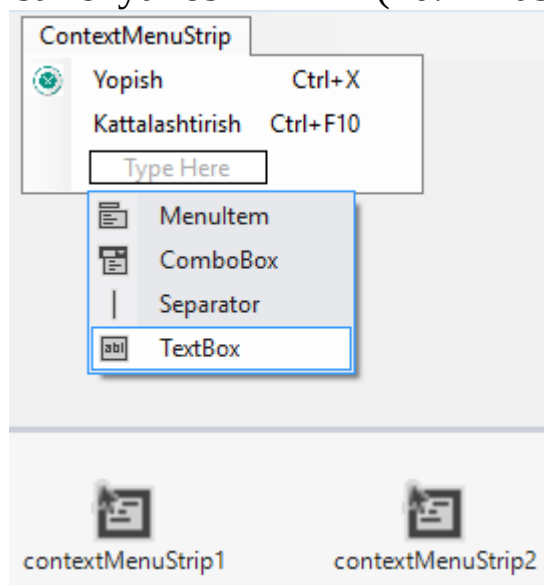
O'rnatilgan kontekst menyu ob'ekti [ContextMenuStrip]
 xususiyati uchun ichki xususiyatlarni, ya'ni [contextMenuStrip]
 xususiyatlarni o'rnatish uchun ochib beradi.
 Buxususiyatlarni hammasiga to'xtalib o'tmaymiz. Faqat Items
 xususiyatiga to'xtalamiz.
 Shu xususiyatdan langanda quyidagicha muloqot oynasichiqadi (10.13-
 rasmga qarang).



10.13-rasm. Kontekst menyubuyruqlarini o'rnatish.

E'tibor qaratsangiz,
 muloqot oynasida qilingan o'rnatishlar muloqot oynasida emas balki asosiy oynada ko'rinadi. Agar bu muloqot oynasida ishlash noqulay bo'lsa,
 asosiy ishchi maydonida ham ishlash mumkin.

Buning keraklik kontekst menyu ob'ekti uchun qancha bir martaba
 osilsa, oynada kontekst menyuni hosil bo'ladi (10.14-rasmga qarang).



10.14-rasm. Oynadakontekstmenyugaishlovberish.

[Type Here] tugmasiorqalijoriy [MenuItem], [ComboBox], [Separator], [TextBox] - 4 taelementlarniqo'shishmumkin.

Qo'shilganharbirelementningo'ngtugmasinibosib,

ubilanma'lumbajarilishimumkinbo'lganamallaribajarishmumkin.

Shuningdek, [properties] deganxususiyatlargaolibkiradi. [Text] - xususiyatigakeribbuyruknominiberishmumkin.

Agarharfbilanmurojjetqilishnixoxlasangiz, kerakliharfoldiga [&] belgisiniqo'yishkerak. Inteaktivtugmao'rnatishuchunesa [ShortcutKeys] xususiyatigakeraklitugmalarkombinatsiyasinio'rnatishmumkin.

Ikonkao'rnatishuchun [Image]

xususiyatgakeraklirasmnio'rnatishmumkin.

Shundayqilibkeraklibarchabuyruqlarniamalgaoshirishmumkin.

Buyruqlarniguruhlashuchun [Separator] komponentasinitanlashkerak.

Buyruqlargaalgortimyo'zishshuchunularniustigasichqonchaniikkimartabo sishyetarliyoki hodisalarigakeribhodisasigayozishmumkin.

Agaryuqoridakontekstmenyugaishlovberishtushunarlibo'lganbo'lsa , masalanio'zingiztugatibqo'ying.

Agarketmaketlikasosidabajarsangizalbattamasalanihechbo'lmasa, algoritmlarsizhalqilishingizlozim.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Toolbox interaktiv oynasini mauchunkera?
2. Qaysitugmachalarmajmuasini ham bosish orqali Toolbox oynasiga o'tish/chaqirish mumkin
3. Toolbox oynasini tahrirlash mumkinmi? Mumkin bo'lsa qanday va mauchun.
4. Nima Toolbox oynasi uchun xizmat qiladiva 5 tabo'limdan iborat.
5. [List View] va [Show All] buyruqlarini mauchunkera.
6. Toolbox interaktiv oynasida nechta komponenta mavjud?
7. Toolbox interaktiv oynasidagi komponentalarining o'rnini almashtirish va qaytanomlash mumkinmi?
8. Komponenta xususiyatlarining asosiy nechta guruhi bor.
9. Komponentalarning Focus guruhi ni mauchun ishlatiladi.
10. Asosiy oynaga qo'shishlgan komponentalar ro'yxatini qaer dano'ri sh mumkin.

11. Oynaning InitializeComponent funksiyasida ham komponentalarni
ng xususiyatlarini ko‘shilganini ko‘rish mumkinmi?

12. Komponentaning nechta usullari (methods) va hodisalari (events)
guruhibor?

13. Komponentaning Layout guruhi nima uchun ishlatiladi?

14. Standart komponentalarga Toolbox oynasida qaysi Tab
dagi komponentalar kiradi.

15. Mantlarni tahrirlash komponentalarini sanab bering?

16. Ro‘yxatdantlash komponentalarini sanab bering?

17. Komponentalarni oynaga joylashtirish uchun qanday amal bajarish
kerak?

18. Label1 ob’ekting text
xususiyatini o‘rnatish qanday amalga oshiriladi?

19. comboBox1 ob’ekting Items
xususiyatlarini o‘rnatish qanday amalga oshiriladi?

20. [numericUpDown1->Value]
ob’ekting tanlangan qanday tipdagi qiymatini qaytaruvchi xususiyat.

21. [comboBox1->SelectedIndex] —
tanlagichning tanlangan qanday qiymatini qaytaradi.

22. Additional
komponentasining xususiyatlarini va hodisalarini nima uchun ishlatiladi.

23. TabControl komponentasining vazifasini ayting.

24. Panel komponentasining xususiyatlariga misollar keltiring.

25. Panel
komponentasini oynaning qaysi qismida joylashtirish kerakligini o‘rnatadigan
xususiyatni ayting.

26. Buxususiyat o‘rnatilgan qiymatlarni qabul qiladigan va uning nechta sahifa
o‘rnatilganini ko‘rsatadi.

U bilan ishlaganda har bir sahifaning alohida xususiyatlariga ishora berish mu
mkin. Gap qaysi komponentaning qaysi xususiyati haqida ketmoqda.

27. System
komponentasining xususiyatlarini va hodisalarini sanab bering.

28. Menyuni boshqarish elementlarini uchun ishlatiladigan komponentala
rni sanab bering.

29. Menyuvastaklar panelini yaratish uchun komponentalardan tas
hqi tizim bilan ishlaydigan komponentalar qaysi tabda kiritilgan.

30. Kontekst menyuni ishlatishni tushuntirib bering.



ASSISMENTTOPShIRIG'I	
	Komponentalar bilan ishlashga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajararing. ☞ Bundasturdagibayozatdagi o'zgartirishlar niti qisqartirilgan bo'sqich amalga oshiriladi.
№ - sizning tug'ilgankuningiz	
1. Oddiy kalkulyator yaratish. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.	
2. Sanoq sistemalar uchun arifmetik amallarni bajaruvchi kalkulyator yaratish. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.	

Komponentalar bilanlashga oid berilgan quyidagi dastur bo'yicha yechim berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.

✎ Bunday dasturda giba'zi o'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.

№ - sizning tug'ilgankuninggiz

1. Oddiykalkulyatoryaring.

Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

This image shows a blank sheet of white paper with horizontal blue ruling lines. On the left side, there are short vertical blue lines that serve as margins, creating a series of narrow columns. The paper is oriented vertically and appears to be a standard notebook or ledger page.

2. Sanoq sistemalar uchunarifmetikamallarnibajaruvchikalkulyatorya rating. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

3. Menyugaegaloyihayarating. Umenyuoinxususiyatlarivahodisalariniboshqarsin. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.
4. Kontekstmenyugaegaloyihayarating. Umenyuoinxususiyatlarivahodisalariniboshqarsin. Interaktiv tugmalar va klaviatura tugmalar bilan ishlash. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

- | |
|---|
| <p>7. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring,
nimalarnio‘rgandingizvabirtakrorlang.</p> <p>8. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang.</p> |
|---|

3.3. Muloqotoynalaribilanishlash.

 Muloqot oynasitushunchasivaxususiyatlari, diaologtabkomponentalariiningxususiyatlarivahodisalari, messageBox vauning Show usulidanfoydlanish, uningparametrlarivaqiymatlari, shaxsiymuloqotoynalariniyaratishbo‘yichanazariyvaamaliybilimlarniha mdaasoslashuchundasturlartuzibko‘rsatilgan.

Bilimlarnimustahkamlashuchun 40
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 7
assismenttopshirig‘iberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

 *Kalitso‘zlar.* IDE, Visual C++, user interface, OYD, komponenta, xususiyatlari, hodisalari, Toolbox, Tab, muloqotoynasi, ColorDialog, FontDialog, OpenFileDialog, PrintDialog, PrintPreviewDialog, FolderBrowserDialog, SaveFileDialog, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, HelpNavigator, Object, DialogResult, InvalidEnumArgumentException, IWin32Window tiplari.

 *Bilishshartbo‘lgantushunchalar.* sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llab quvvatlovchimuhitdaishlashnibilishlozim.

 *Bilibolasiz.* Visual C++ muhitidasturlash, integrallashganishlabchiqarishmuhiti, Toolbox, muloqotoynasitushunchasivaxususiyatlari, ColorDialog, FontDialog, OpenFileDialog, PrintDialog, PrintPreviewDialog, FolderBrowserDialog, SaveFileDialog muloqotoynalaribilanishlash, MessageBox sinfivauning Show funksiyasiqabulqilishimumkinbo‘lgan String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, HelpNavigator, Object, DialogResult, InvalidEnumArgumentException, IWin32Window tiplarivaqiymatlari, funksiyasiniyozilishvariantlari, muloqotoynasiniloyihaboshqaruvoynaorqaliyaratishvamuloqotoynalarig aqo‘yiladigantalablar, o‘rnatishvakomponentalarbilanbog‘lashusullari, shuningdekba‘zibirxususiyatlarivahodisalario‘rganishingizmumkin.

REJA

1. Visual C++ muhitidamuloqotoynalari.
2. Muloqotoynalarinisoqlash.
3. Muloqotoynalariboshqarishelementlari.
4. Muloqotoynalariniyaratish.

KIRISH

Foydalanuvchilarbilantizimningmuloqotiniinteraktivamalgaoshiriuc
hunmuloqotoynalarikerak. Muloqotoynalari 3
takattaguruhlargabo‘linadi. Tizimli,
ya‘niOTbilanishlashgamo‘ljallangan,
interaktivxabarlarniberishvaaniqjavoblarniolishuchunmo‘ljallangan,
dasturchiningyokifoydalanuvchiningtashabbusibilanyaratiladiganmuloqo
toynalaribor.

Visual C++ muhitidamuloqotoynalari. Yuqoridaaytibotilgandek,
muloqotoynalarinyaratishularnimaqsadlaridankelibchiqqanholdaalgaos
hiriladi. Buoynalarningoziningtalablarimavjudki, shotalablarbajarilsa,
umuloqotoynasibolaoladi. Butalablargaquyidagilarkiradi:

1. Muloqotoynasining sarlavhasibolishivaundafaqatoynaniyopishtug
masiningbo‘lishilozim,

tizimlimenyuvaboshqatugmalarbo‘lishimumkinemas.

Istesnotariqasidabazihollarda, yordamtugmasinijoylashtiishmumkin.

2. Muloqotoynasiteskarialoqagamo‘ljallanganligiuchun,
unishartinibajarmasdantizimningboshqaoynasigao‘tishmumkinemas.

3. Teskarialoqaningbirstaturlariniamalgaoshiruvchitugmalarbo‘lis
hikerak.

4. Muloqotoynalarasosiyoyndanhardoimkichikbo‘lishishart.

5. Muloqotoynaningasosiy maqsadianiqkelitirilishikerak, yokituri,
nimamunosabatuchunmuloqotoynasichiqqanligi.

6. Muloqotoynasidanboshqamuloqotoynasigao‘tishmumkinemas,
asosiyoynagao‘tishlozim.

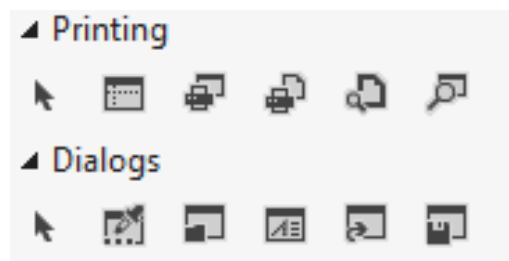
7. Muloqotoynamurojaatqilingandayaratilishivateskarialoqaqabulqili
ngandanso‘ngxotiradano‘chiribtashlanishilozim.

8. Muloqotoynagadinamikxotiralarbo‘lishimumkinemas.

Ushbutalablarnibajarganharqandayoynamuloqotoynasihisoblanadi.

Visual

C++daOTbilanmuloqotqilishgamo‘ljallanganmuloqotoynalariga
[Dialogs] tab dagiva [Printing] tab dagibarchakomponentalarkiradi
(11.1-rasmgaqarang).



11.1-rasm. Tizimlimuloqotoynalariyaratishkomponentalari.

Visual

C++ninghujjatlarigaqarasangiztizimlimuloqotoynalariuchunyagonamuloqotoynalariro'yxatituzilgan.

Buro'xatgaquyidagijadvalgakeltirilgankomponentalarkiradi.

11.1-jadval.Visual

C++daOTbilanmuloqotqilishgamo'ljallanganmuloqotoynalariyaratishgamo'ljallangankomponentalarningvazifalari

№	Komponent nomi	Vazifasi
1	ColorDialog	Foydalanuvchilarinterfeysselementiranginio'rnatishimkoniniberuvchiranglarpalitrasiuchunmuloqotoynasiniko'rsatadi.
2	FontDialog	Foydalanuvchilargakeraklikkomponentauchunshriftvauningxususiyatlarinio'rnatishimkoniniberuvchimuloqotoynasiniko'rsatadi.
3	OpenFileDialog	Foydalanuvchilaruchunfaylnitanlashimkoniniberadiganmuloqotoynasiniko'rsatadi.
4	PrintDialog	Foydalanuvchilargaprinternitanlashvauningxususiyatlarinio'rnatishimkoniniberuvchimuloqotoynasiniko'rsatadi.
5	PrintPreviewDialog	FoydalanuvchilaruchunchopqilishdaPrintDocumentboshqaruv elementiningko'rinishiningko'rsatishimkoniniberadiganmuloqotoynasiniko'rsatadi.
6	FolderBrowserDialog	Foydalanuvchilaruchunpapkalko'rish, yaratishvatanlashimkoniniberadiganmuloqotoynasiniko'rsatadi.
7	SaveFileDialog	Foydalanuvchilaruchunfaylnisaqlashimkoniniberadiganmuloqotoynasiniko'rsatadi.

Bukomponentalarmaxsusxususiyatgaasoslagankomponentalarbilashishlatiladi.

Interaktivxabarlarniberishvaaniqjavoblarniolish Visual C++ da MessageBoxsinfimavjud.

Businfbilanbarchaixtiyoriyturdagimuloqotoynalariyaratishmumkin.
 Sinfningnomlarfazosi System.Windows.Forms
 bo‘libhisoblanadivakutubxonasi System.Windows.Forms.dll
 hisoblanadi. Bumuloqotoynasiformasinfiningmerosxo‘rihisoblanadi.
 Unda 21 taturlikombinatsiyalishow funksiyachibor.
 Uningquyidagiparametrlaribor.

11.2-jadval. MessageBox sinfnining show funksiyasiparamertlari

Paramertno mi	Tipi	vazifasi
text	String	Muloqotoynasiningxaba ri
caption	String	Muloqotoynasining sarla vhasi
buttons	MessageBoxButtons	Teskari aloqanita‘minala shtugmalariturlarinianiqlash
icon	MessageBoxIcon	Muloqotoynalariningiko nkalariturlarinianiqlash
defaultButton	MessageBoxDefaultButton	Teskari aloqanita‘minala shuchun joriy tugmalaritu rlarinianiqlash
options	MessageBoxOptions	Muloqotoynalariningam allariturlarinianiqlash
helpFilePath	HelpNavigator	HelpNavigator ob’ekting qiymatlari uc hun foydalaniladi
param	Object	Yordam tugmasibosilgan da ID fiymatnianiqlashimkoni niberuvchi parametr
Returns	DialogResult	Muloqotoynalarining qiy matlari qaytarish uchun fo y dalaniladigan DialogResult tipidagi turlarinianiqlash
Exceptions	InvalidEnumArgumentExcep tion	MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButt on

		ob'ektlarini birinchi qabul qiluvchi kengaytirilgan ob'ekt tipi
owner	IWin32Window	muloqot oynasini egasi Iwin32window amalga oshirish.

Bu parametrlarning o'ziga mos qiymatlari oldindan aniqlab berilgan bo'lib, oddiy soddako'rinishda muloqot oynasini yaratish imkon beradi. Keyinroq businformatlari qiymatlari va ularga ishlov berishni ko'rib chiqamiz.

Foydalanuvchi tomonidan yaratiladigan muloqot oynasi forma kabi yaratiladi va ularni loyihalashda turchining yoki foydalanuvchining xoxishiga qarab amalga oshiriladi.

Interaktiv muloqot oynalarikabi interaktiv tugmalarni yaratish va ularni boqarish, kerakli ma'lumotlarni olish uchun ishlatiladi. Shuni ham inobatga olish kerakki yaratiladigan muloqot oynasitalablargamoskelish kerak.

Bu talablarni amalga oshirish uchun formaning xususiyatlariga ishlov berish, lohim bo'lsa, asosiy oyndakerakli xususiyatlarni o'rnatish mumkin. Bunda ham formaning show usuli mavjud bo'lib. Shu orqali forma chaqiriladi.

Sinfning nomlar fazosi System.Windows bo'lib hisoblanadi va kutubxonasi PresentationFramework.dll hisoblanadi. Bu muloqot oynasi forma sinfining meros xo'ri hisoblanadi.

.NET 5 Preview 1, .NET Core 3.1, 3.0 va .NET Framework 4.8 4.7.2 4.7.1 4.7 4.6.2 4.6.1 4.6 4.5.2 4.5.1 4.5 4.0 3.5 3.0 variantlarida qo'llab quvvatlanadi.

Foydalanuvchi muloqot oynalarini yaratishni keyinroq batafsil ko'rib chiqamiz.

Muloqot oynalarini sozlash. Bu muloqot oynalarini sozlash uchun tizim muloqot oynalaridan foydalanish va ularga ishlov berish nazarda tutilgan. Yuqoridagi keltirilgan 7 ta muloqot oynalaridan foydalanishlar, xususiyatlarini va hodisalarini boshqarish to'g'risida to'xtalamiz.

1. ColorDialog muloqot oynasi. Bu oyna foydalanuvchilar interfeys elementirangini o'rnatish imkonini beruvchi ranglar palitrasini uchun muloqot oynasini ko'rsatadi.

Bu komponentdan forma ga o'rnatilganda hech qanday ko'rinish hosil bo'lmaydi, ammo formaning ichki tuzilmasiga qo'yshiladi. Formaning ishchi holatidagi formasining pastki qismida uning ob'ekti yaratil

adi. `solorDialog1` ob'ektini yaratish orqali boshqariladi. Uning xususiyatlarini va hodisalarini ham mavjud valoyiha o'ynasida foydalanuvchi xususiyatlarini o'ynasiga qo'yadi. U yerdan kerakli xitoyi yoy nanshu ob'ektga mos xususiyat va hodisalarini o'rnatish mumkin.

`ColorDialog` muloqot o'ynasidan foydalanish uchun `ColorDialog()` konstruktorini ishga tushirish lozim.

Businfnings xususiyatlari, usullari va hodisalarini mavjud.

11.3-jadval. `ColorDialog` muloqot o'ynasining xususiyatlari

<code>AllowFullOpen</code>	Maxsus ranglarni qilash uchun muloqot o'ynasini foydalanish mumkin yoki yo'qligini o'rnatish
<code>AnyColor</code>	Muloqot o'ynasida asosiy ranglarni majmuining barcha mavjud ranglarni ko'rsatish yoki yo'qligini o'rnatish
<code>CanRaiseEvents</code>	Komponentaga bir hodisa o'rnatish mumkinligini aniqlash.
<code>Color</code>	foydalanuvchi tomonidan tanlangan rangni o'rnatish.
<code>Container</code>	Komponentini o'z ichiga olgan <code>IContainer</code> ni o'rnatish
<code>CustomColors</code>	Muloqot o'ynasidagi ko'rsatilgan maxsus ranglarni o'plamini o'zgartirish yoki o'rnatish.
<code>DesignMode</code>	Komponenta joriy dizayn rejimida ekanligini ko'rsatadigan qiymatni o'zgartirish.
<code>Events</code>	komponentaga ilova qilinadigan hodisalar ro'yxatini o'zgartirish.
<code>FullOpen</code>	Muloqot o'ynasini ochilganda maxsus ranglarni yaratish uchun ishlatiladigan boshqaruvi elementlarini ko'rsatuvchi qiymatni o'zgartirish yoki o'rnatish.
<code>Options</code>	<code>ColorDialog</code> boshlash uchun xususiyatlarni o'zgartirish.
<code>ShowHelp</code>	Yordam tugmasi rang muloqot o'ynasidagi paydoy yoki yo'qligini o'zgartirish
<code>Site</code>	Komponentaning <u>ISite</u> ni o'zgartirish yoki o'rnatish.
<code>SolidColorOn</code>	Muloqot o'ynasidagi qattiq ranglarni tanlash uchun foydalanuvchilarni cheklash yoki yo'qligini o'rnatish

ly	
Tag	nazorathaqidama'lumotlarnio'zichigaolganob'ektsozlash.

11.4-jadval. ColorDialog muloqotoynasiningusulari

CreateObjRef(Type)	Ob'ekt bilan muloqot qilish uchun ishlatiladigan proksi ishlab chiqarish uchun zarur bo'lgan barcha tegishli ma'lumotlarnio'zichigaolganob'ektyaratadi.
Dispose()	Komponent tomonidan ishlatiladigan barcha resurslarni chiqaradi.
Dispose(Boolean)	Tarkibiy qisim tomonidan ishlatiladigan boshqarilmaydigan resurslarni chiqaradi va ixtiyoriy ravishda boshqariladigan resurslarni chiqaradi.
Equals(Object)	Belgilangan ob'ekt joriy ob'ektga teng yoki yo'qligini aniqlaydi.
GetHashCode()	Standart hash funksiyasi sifatida xizmat qiladi.
GetService(Type)	Komponent yoki uning konteyneri tomonidan taqdim etilgan xizmatni ifodalovchi ob'ektni qaytaradi.
GetType()	Joriy ob'ekt turi aniqlanadi.
HookProc(IntPtr, Int32, IntPtr, IntPtr)	Umumiy muloqotoynasiga xos funksiyalarni kiritish uchun belgilangan umumiy muloqotoynasiga protsedura belgilaydi.
MemberwiseClone()	Joriy ob'ektning oddiy nusxasini yaratadi.
MemberwiseClone(Boolean)	Joriy MarshalByRefObject ob'ektning oddiy nusxasini yaratadi.
OwnerWndProc(IntPtr, Int32, IntPtr, IntPtr)	Umumiy muloqotoynasiga maxsus funksiyalarni qo'shish uchun belgilanadigan oynatartibni belgilaydi.
Reset()	Ularning standart xususiyatlariga barcha imkoniyatlarini ishga soladi, oxirgi tanlangan qorarang va ularning standart xususiyatlariga maxsus ranglar moslashtiriladi

RunDialog(IntPtr)	Umumiy muloqot oynasini belgilaydi.
ShowDialog()	Umumiy muloqot oynasini ishlatadi.
ShowDialog(IWin32Window)	Umumiy muloqot oynasini ishlatadi.
ToString()	Muloqot oyna qiymatini satrga o'tkazish.

11.5-jadval. ColorDialog muloqot oynasining hodisalari

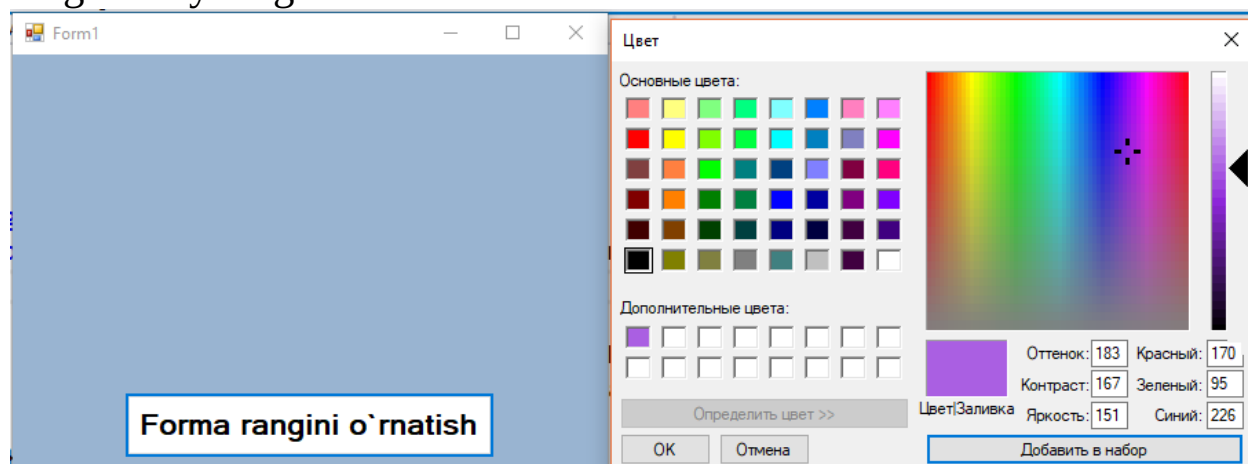
Disposed	Komponentni metodigacha qirib qo'yilgan va yuzaga keladi	Dispose()
HelpRequest	Foydalanuvchi umumiy muloqot oynasidagi yordam tugmasini bosganida sodir bo'ladi.	

ColorDialog muloqot oynasidan foydalanish uchun forma ga birtugma o'rnatamiz va uning Click hodisasi yordamida chaqiramiz. Tanlangan rang safora maning tugma ning fonini o'zgartirsin.

Tugmaning Click hodisasi quyidagicha dastur fragmentini o'rnatamiz.

```
if(colorDialog1->ShowDialog() ==
::System::Windows::Forms::DialogResult::OK)
    Form1::BackColor = colorDialog1->Color;
    button1->BackColor = colorDialog1->Color;
```

Dastur fragmentida muloqot oyna chaqirilgan davataskari aloqasi ok ob'ekti ni qaytarsa forma tugma ning mos xususiyatlar ranglarini o'zgartirish algoritmi yozilgan.



11.2-rasm. ColorDialog muloqot oynasidan foydalanish.

ColorDialog muloqot oynasining xususiyatlari, usullari va hodisalarini masalaning ahamiyatiga qarab ishlatish mumkin.

2.FontDialog

muloqotoynasi.Buoyina

foydalanuvchilargakeraklikkomponentauchunshriftvauningxususiyatlarini o'rnatishimkoniniberuvchimuloqotoynasiniko'rsatadi.

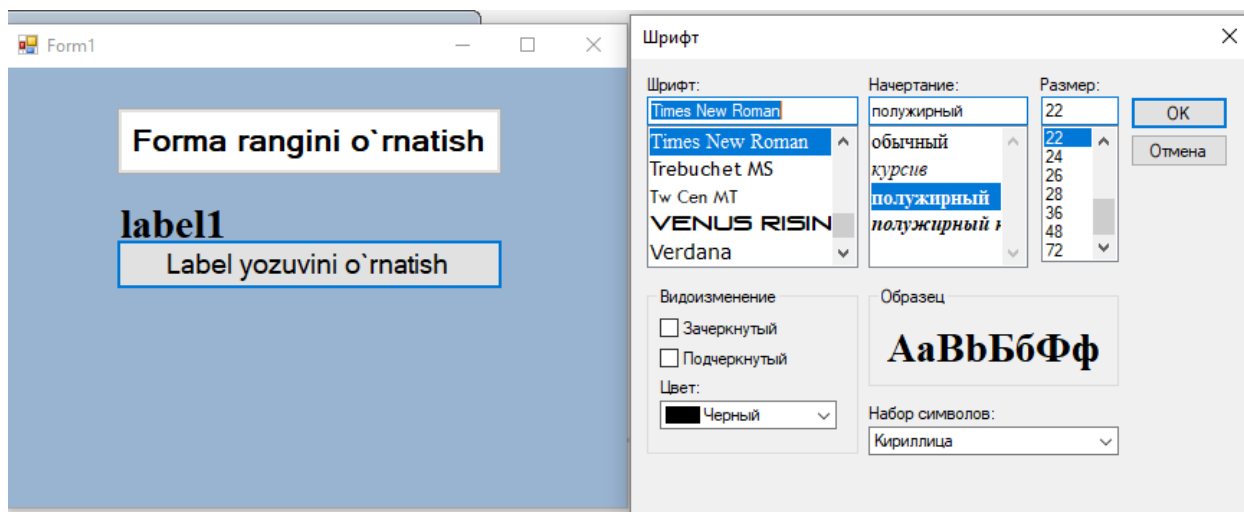
Bukomponentdaniformagao'rnatilgandahechqandayko'rinishhosilbo'lmaydi, ammoformaningichkituzilmasigaqo'yshiladi. Fomraningishchiholatidagiformasiningpastkiqismidauningob'ektiyaratiladi.

Bumuloqotoynasiniishlatishuchunformagabir Label va button ob'ektlarinio'rnatamiz. Tugmabosilganda Label ob'ektningmatniniyozuvxususiyatlarinio'rnatishniko'ribchiqamiz. Buninguchuntunmaningbosilgandaxususiyatigaquyidagidasturfragmentni iyozi shiyetarli.

```
fontDialog1->ShowColor = true;
fontDialog1->Font = label1->Font;
fontDialog1->Color = label1 ->ForeColor;
Color color = label1->ForeColor;
System::Drawing::Font^ font = label1->Font;
System::Windows::Forms::DialogResult result =
fontDialog1->ShowDialog();
if(result ==
::System::Windows::Forms::DialogResult::OK) {
    label1->Font = fontDialog1->Font;
    label1->ForeColor = fontDialog1->Color;
}
```

Dastudadastlab Label ob'ektningrangvayozuvlarinisqalaboladi, chunkiyozuvni formatlashmuloqotoynasichaqirilgandajoriyholatniolishuchun.

Dasturdarangvayozuvqiymatlarinisqalashuchuno'zgaruvchilarnianiqlash olibhamkeltirilgan. Muloqotoynasining [OK] hodisasibajarilgandayozuvvarangnio'zgartirishko'rsatilgan.



11.3-rasm. FontDialog muloqotoynasidanfoydalanish.

Businfningxususiyailari, usullarivahodisalarimavjud. Ularamaliyvazifalarnibajarishdafoydalanishmumkinvamustaqiloʻrganish lozim. Chunkibirvazifaniamalgaoshirishuchunturlixilalgoritmlardanfoydalanish mumkin.

3.OpenFileDialog

muloqotoynasi.Buoyna

foydalanuvchilaruchunfaylnitanlashimkoniniberadiganmuloqotoynasinik oʻrsatadi. Bumuloqotoynasihamyuqoridagidekfoydalaniladi. Shuningdek, sinfningshosxususiyatlari, usullarivahodisalarimavjud. Farqlixususiyatlarnikuoʻrsatishuchunbirmisololamiz.

Undatugmabosilganda label matnigafaylningtoʻliqyoʻliniolishvaoʻrnatishuchunquyidagidasturfragme ntiniyoziladi.

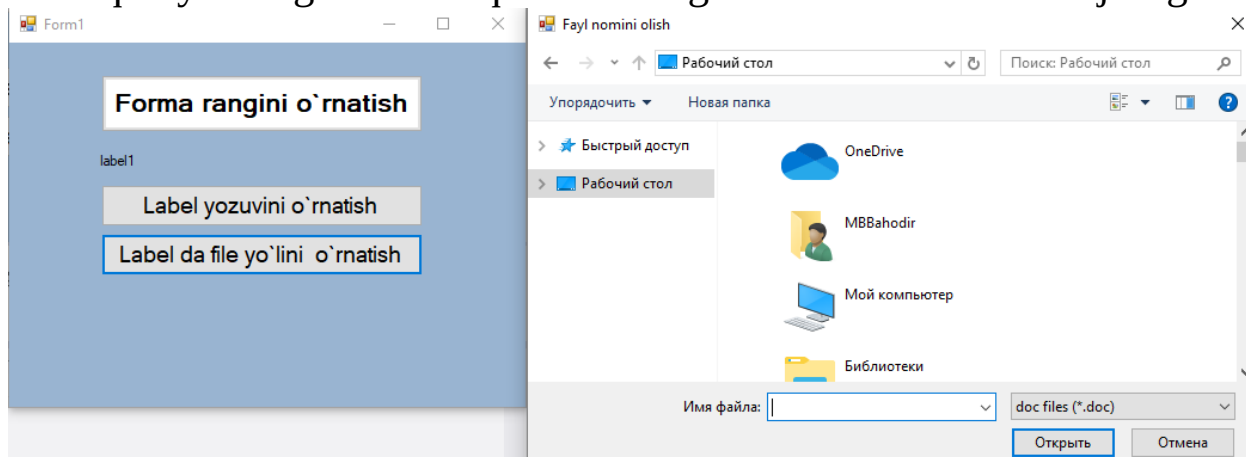
```
openFileDialog1->InitialDirectory = "c:\\";
openFileDialog1->Filter = "txt files (*.txt)|*.txt|doc files (*.doc)|*.doc,*.docx|All files (*.*)|*.*";
openFileDialog1->FilterIndex = 2;
openFileDialog1->RestoreDirectory = true;
openFileDialog1->Title = "Fayl nomini olish";
if ( openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK ){
    if ( openFileDialog1->OpenFile() != nullptr ){
        label1->Text = openFileDialog1->FileName;
    }
}
```

Dasturning birinchi satri joriy katalog o'raltiladi. Shuningdek, joriy foydalanuvchining kerakli kataloglarini ham o'rnatish qiymatlaribor. So'ng, fayllarni tip bo'yicha filtrlash o'rnatiladi. Hozirda 3 ta format o'rnatilgan.

Keyingi satri da fayl kengaytmalaridan qaysi biri joriy bo'lib chiqishini belgilash amalga oshirilgan.

Ko'rsatilgan katalogni faollashtirish bajarilgan va muloqot oynasining sarlavhasi dakerakli matn joylashtirilgan.

Muloqot oynaning teskari aloqasi tekshirilgan va shu asosida amal bajarilgan.



11.4-rasm. FontDialog muloqot oynasidan foydalanish.

Odatda bu muloqot oynasima'lum tizim shafayllarini matnini joylashtirish uchun ishlatiladi.

Buni qanday amalga oshirish mumkin.

Buning uchun kichik bo'lsaham matn muharriri yaratish lozim va unga OT dan oddiy manli fayllarni yuklab olish mumkin.

Fayl ichida gima'lumotlarni olish uchun quyidagicha dastur fragmenti yozish mumkin:

```
openFileDialog1->Filter = "text files (*.txt)|*.txt|cpp files (*.cpp)|*.cpp|All files (*.*)|*.*";
openFileDialog1->FilterIndex = 2;
openFileDialog1->RestoreDirectory = true;
openFileDialog1->Title = "Fayl nomini olish";
openFileDialog1->FileName = "";
if ( openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    if ( (myStream = openFileDialog1->OpenFile())
!= nullptr ){
        System::IO::StreamReader ^ sr = gcnew
```

```

System::IO::StreamReader(openFileDialog1-
>FileName);
    richTextBox1->Text = sr->ReadToEnd();
sr->Close();
}
}

```

Bu amaliy dastur fragmentini amaliyot gasinabkoʻrish orqali tahlil qilish.

4. PrintDialog muloqot oynasi. Bu oyna foydalanuvchilarga printerni tanlash va uning xususiyatlarini oʻrnatish imkonini beruvchi muloqot oynasi nikaʻrsatadi. Oynaning judakoʻpxususiyatlari, usullari va hodisalaribor.

Bularni mustaqilishlarni bajarish vaqtida amalga oshirish mumkin. Shuningdek, bu muloqot oynani ishlatish usullari ham judakoʻp. Odatda dastur chining loyiha holatidan kelib chiqqan holda foydalaniladi.

```

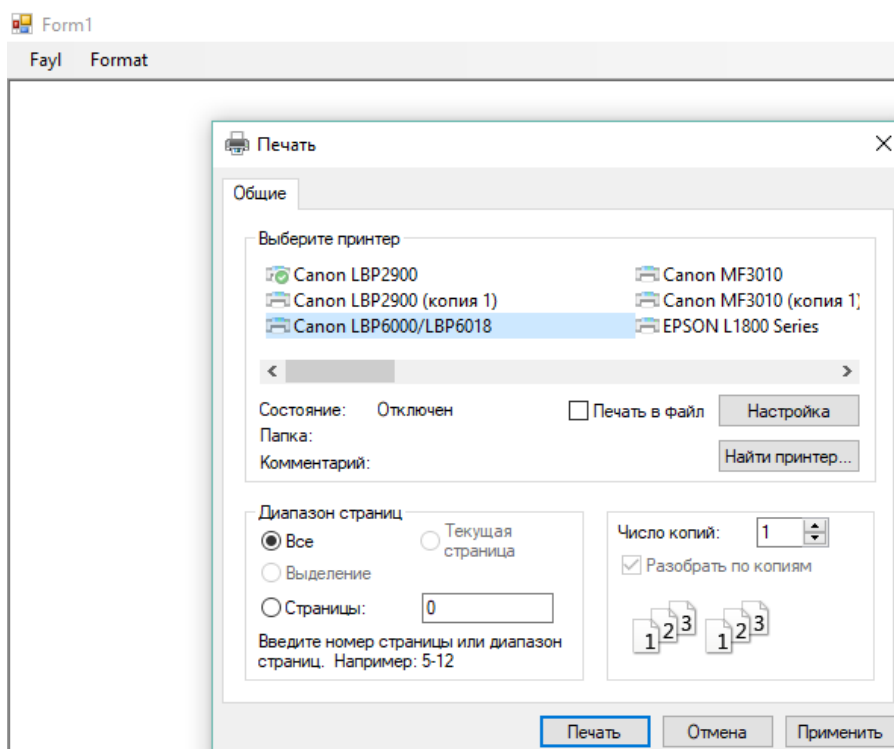
    printDialog1->AllowSomePages = true;
    printDialog1->ShowHelp = true;
    if ( printDialog1 == nullptr )
        System::Windows::Forms::MessageBox::Show("pnull"
);
    System::Windows::Forms::DialogResult result =
printDialog1->ShowDialog();

System::Windows::Forms::MessageBox::Show(result.ToString());
    if ( result ==
System::Windows::Forms::DialogResult::OK )
    {
        // docToPrint->Print();
        System::Windows::Forms::MessageBox::Show("Chop
qilish boshlanadi");
    }

```

Dastur fragmentidamaʼlumotlarni chop qilish uchun avval unimaʼlumbirchi quvchi oqimga yozish va oqim niesa, hujjat formatiga joylashtirish kerak. Hujjat niesa, sahifalarni sozlab docToPrint obʼekti yaratilish kerak. Ammo chop qilishning turlari parametrlarini ishlatish uchun muloqot oynasini chiqarib beradi.

Kerakli xususiyatlarni oʻrnatgandan soʻng amalni bajarish mumkin.



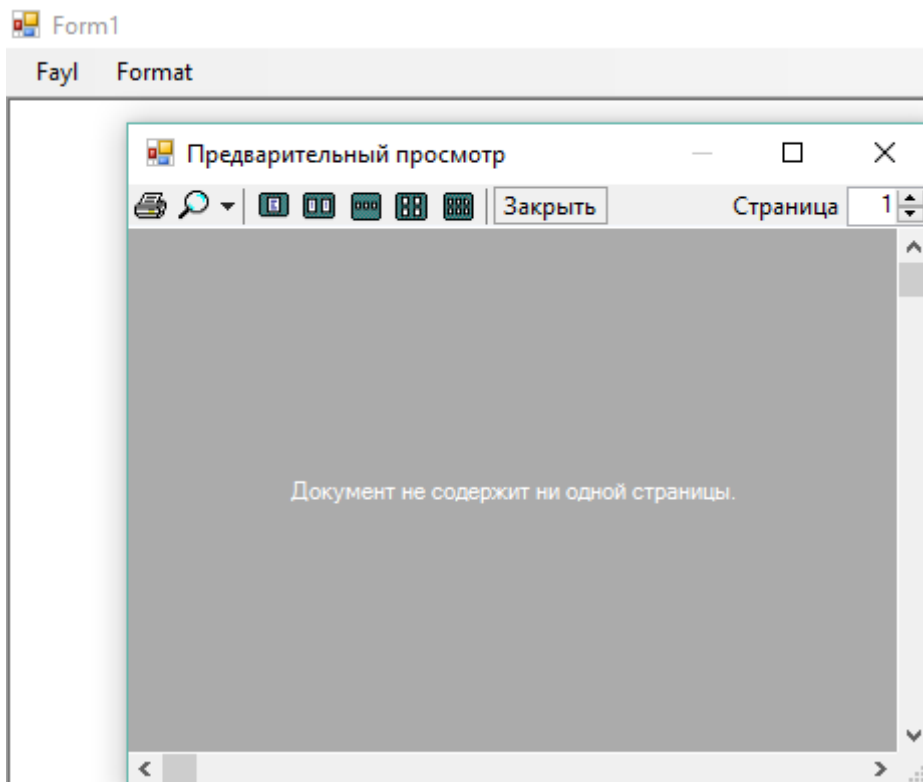
11.5-рasm. PrintDialog muloqotoynasidan foydalanish.

Dasturfragmentinibirloyihajoylashtirib,
ishlatsangizrasmdagidekishlashikerak.

5.PrintPreviewDialog muloqotoynasi. Buoyana -
foydalanuvchilaruchunchopqilishda PrintDocument
boshqaruvelementiningko‘rinishiningko‘rsatishimkoniniberadiganmuloq
otoynasiniko‘rsatadi.

```
printPreviewDialog1->MinimumSize =
System::Drawing::Size( 375, 250 );
printPreviewDialog1->UseAntiAlias =
true;
printPreviewDialog1->Document =
document;
printPreviewDialog1-
>ShowDialog();
```

Bumuloqotoynasiamyuqoridagimuloqotoynasikabisozlashlarnibaja
rshgandanso‘ngfoydalanishmumkin. Dasturfragmentidaqarasangiz
document ob’ektiniyaratishlozim.
Chopqilishninigob’ektidanfarqqilganxoldaoqimdagima’lumotniformatla
shvaunichiqishiniA4 shaklgakeltrishorqaliamlgaoshiriladi.



11.6-rasm. PrintPreviewDialog muloqotoynasidanfoydalanish.

6.FolderBrowserDialog muloqotoynasi.Buoyna

foydalanuvchilaruchunpapaklarko‘rish,
yaratishvatanlashimkoniniberadiganmuloqotoynasiniko‘rsatadi.
Bundanma‘lumotlarniko‘chirishda,
papaklarnitaqqsolashdaishlatishmumkin.
Muloqotoynasiningxususiyatlari, usullarivahodisalarmavjud.
Ularningba‘zilariniquyidagidasturfragmentigakeltiribo‘tamiz.

```
System::IO::Stream ^ myStream;
System::Windows::Forms::DialogResult result =
folderBrowserDialog1->ShowDialog();
if ( result ==
System::Windows::Forms::DialogResult::OK ) {
    System::String^ folderName =
folderBrowserDialog1->SelectedPath;
    openFileDialog1->InitialDirectory = folderName;
    openFileDialog1->FileName =
String::Concat(folderName,"\\1.cpp");
    if ( (myStream = openFileDialog1->OpenFile())
!= nullptr ){
        System::IO::StreamReader ^ sr = gcnew
System::IO::StreamReader(openFileDialog1-
>FileName);
```



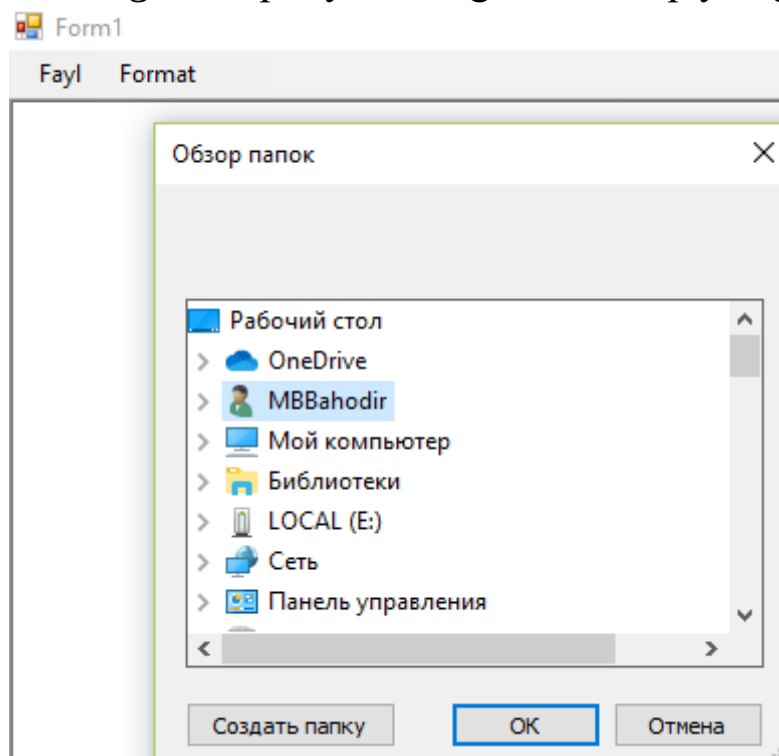
```

        richTextBox1->Text = sr->ReadToEnd();
        sr->Close();
    }
}

```

Dasturfragmentidapapkauchunmuloqotoynasichaqirilganvako‘rsatilganpapkadan 1.cpp fayliniyuklabkelgan. Bundayholatfaylmenejerlaruchunpapkadagifayllarningro‘yhatinihamolislmumkin.

Dasturfragmentidabir StreamReader oqim yaratilgan. Oqimfaylningma’lumotlarinio‘qishuchun yaratilgan. Oqimningkonstruktoriaosida sr oqimob’ektiyaratiladi. Bu oqimning ReadToEnd funksiyasiorqali richTextBox gama’lumotlarjoylashtiriladi. FolderBrowserDialog muloqotoynasiningko‘rinishiquyidagicha:



11.7-rasm. FolderBrowserDialog muloqotoynasidan foydalanish

7. SaveFileDialog

muloqotoynasi.

Buoyna foydalanuvchilaruchun faylni saqlash imkonini beradigan muloqotoynasini ko‘rsatadi.

Bu ham asosan matnli va maxsus tuzilmalarima’lumotlarini saqlash uchun ishlatiladi. Ma’lumot qanaqatuzilmadayozi saqlansa, shunday tuzilmada o‘qiladi.

```

System::IO::Stream^ myStream;
        MemoryStream^ userInput = gcnew
        MemoryStream();

```

```

        richTextBox1->SaveFile( userInput,
RichTextBoxStreamType::PlainText );
        userInput->WriteByte( 32 );

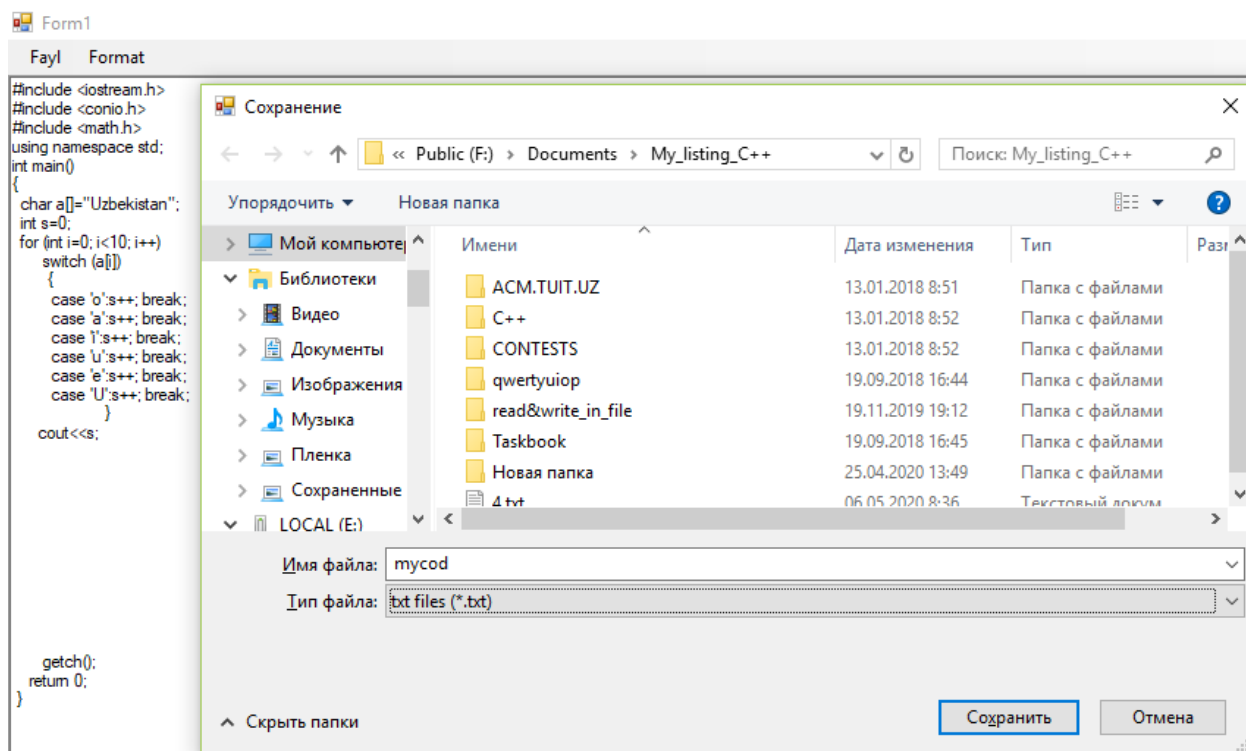
        // saveFileDialog1->CreatePrompt = true;
        // saveFileDialog1->OverwritePrompt = true;
        // saveFileDialog1->FileName = "myText";
        // saveFileDialog1->DefaultExt = "txt";

        saveFileDialog1->Filter = "txt files
(*.txt)|*.txt";
        saveFileDialog1->FilterIndex = 2;
        saveFileDialog1->RestoreDirectory = true;
if ( saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK ){
        myStream = saveFileDialog1->OpenFile();
        userInput->Position = 0;
        userInput->WriteTo( myStream );
        myStream->Close();
}

```

Dasturfragmentidaikkixiloqimiyaratilgan. myStream -
 birinchisian‘anaviyoqimvaMemoryStream -xotiraloqimiyaratilgan.
 Birima’lumotlarniolishvaikkinchisima’lumotlarnixotiragayozish,
 berilgannombilannomlashganqaratilgan.
 Positionbupoizitsiyaaniboshlash, ammoolibtashlasahamishlaydi.

Bumuloqotoynasinafaqatmatnlima’lumotlarnibalki,
 foydlanuvchiningxoxlaganma’lumotnimatnyokibinarkodlashorqalisaqlas
 hmumkin. Muloqotoynasiningko‘rinishiquyidagirasmdakeltirilgan.



11.8-rasm. **SaveFileDialog** muloqotoynasidan foydalanish

Ko'ribchiqilgan muloqotoynalaribarchasi OT bilan ishlashgamo'ljalla nganligiko'rinib, turibdi. OT qanday sozlangan bo'lsa, bu muloqotoynalar ham shu rejimda ishlaydi.

Oynalarga deyari lo'zgartirish kiritish shart emas, xuddi kibarchasi kutilganidek yaratilgan gao'xshaydi. Faqat eng kattam u ammo silokalizatsiya qilish.

Muloqotoynalariboshqarish elementlari.

Bunda interaktiv muloqt qilish oynalaritushiniladi. Ularni yaratish va boshqarish dasturchining xoxishiga qarab amalga oshiriladi. Yuqorida aytibo'tganimizdek, Visual C++ da MessageBox sinfi haqida yuqoridaga aytibo'tgan edik. Unda 21 taturlik kombinatsiyali show funksiyachibordigini ham. Shuningbu funksiyalar va ularning parametrlari hamda parametrlarining qiy matlaritog'risima'lumotlarni va muloqotoynalarini yaratishgavaishlovberishga e'tiborni qaratamiz.

1.Show(String) fknutsiyasi. Bir argumentli funksiya bo'lib, belgilangan matn bilan xabarko'rsatadigan muloqotoynasini yaratish uchun ishlatiladi. Uning kiruchi parametri System::String tipidabo'lib, System::Windows::Forms::DialogResult tipidagi qiymat qaytaradi.

```
if( MessageBox::Show("Bu oddiy sohow") ==
System::Windows::Forms::DialogResult::OK)
    this->Close();
```

2.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String) funksiyasi. Buko‘pparamerlibo‘lib, belgilanganpaoametlarasosidainteraktivmuloqotoynasiniyaratishuchunis hlatiladi. Parametlariquyidagiqiymatlariqabalqiladi.

IWin32Window Interface - System.Windows.Forms nomlarfazosiva System.Windows.Forms.dll kutubxonasidanfoydalanib, Win32 HWND nijoriyqilishinterfeysiberadi. IWin32Window sinfinterfeysidanmerosxo‘roladi. System.Windows.Forms.Control va System.Windows.Forms.NativeWindow umumiyruxsatsinflariinterfeyslariniishlatadi, qiymatlarsifatida ComVisibleAttribute, GuidAttribute, InterfaceTypeAttribute tiplariniishlatadi.

MessageBoxButtons - Enum tipidagiparametrdir. System.Windows.Forms nomlarfazosiva System.Windows.Forms.dll kutubxonasidanfoydalanadi.

Muloqotoynalaridako‘rsatilishikerakbo‘lgantugmalarnianiqlaydi.

Qiymatlari public enum class MessageBoxButtons ta‘luqlidir.

11.6-jadval.MessageBoxButtons qabulqiluvchiqiymatlar

No	Qiymatnomi	vazifasi
1	AbortRetryIgnore	Muloqotoynasida Abort, Retry Ignore tugmalarinio‘rnatish
2	OK	Muloqotoynasida OK tugmasinio‘rnatish
3	OKCancel	MuloqotoynasidaOK va Cancel tugmalarinio‘rnatish
4	RetryCancel	MuloqotoynasidaRetry va Cancel tugmalarinio‘rnatish
5	YesNo	MuloqotoynasidaYes va No tugmalarinio‘rnatish
6	YesNoCancel	MuloqotoynasidaYes, Nova Cancel tugmalarinio‘rnatish

Muloqotoynasiga MessageBoxButtons tugmalarinio‘rnatishdasturfragmenti:

```
if ((MessageBox::Show("Joriy oynani yopishni
xoxlaysizmi?",
                    "Xabar", MessageBoxButtons::YesNo) ==
System::Windows::Forms::DialogResult::Yes)){
    this->Close();
}
```

MessageBoxIcon – Enum tipidagi ob'ekt bo'lib, u ham yuqoridagidek, nomlar fazosida kutubxon bilan ishlaydi. Muloqot oynasidako'rsatish uchun muloqot oynalarining turlarini belgilovchi konstantalarni aniqlaydi.

11.7-jadval. MessageBoxIcon qabul qiluvchi qiymatlar

N	Qiymat nomi	belgisi	vazifasi
1	Asterisk		Qandaydir hodisa haqida faqat xabar beruvchi muloqot oynasi uchun ikonka
2	Error		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqidama'lumot beruvchi muloqot oynasi uchun ikonka
3	Exclamation		Qandaydir hodisa haqida faqat ogohlantirish haqidama'lumot beruvchi muloqot oynasi uchun ikonka.
4	Hand		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqidama'lumot beruvchi muloqot oynasi uchun ikonka
5	Information		Qandaydir hodisa haqida faqat xabar beruvchi muloqot oynasi uchun ikonka
6	None		Ikonkasiz muloqot oynasi uchun
7	Question		Qandaydir hodisa haqida faqat savolga javob beruvchi muloqot oynasi uchun ikonka
8	Stop		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqidama'lumot beruvchi muloqot oynasi uchun ikonka
9	Warning		Qandaydir hodisa haqida faqat ogohlantirish haqidama'lumot beruvchi muloqot oynasi uchun ikonka.

MessageBoxIcon ni ishlatish uchun joriy formani yopish uchun muloqot oynani ishlatish uchun dastur fragmentini keltiramiz.

Buning uchun formaning FormClosing hodisasiga quyidagi dastur fragmenti ni yozamiz:

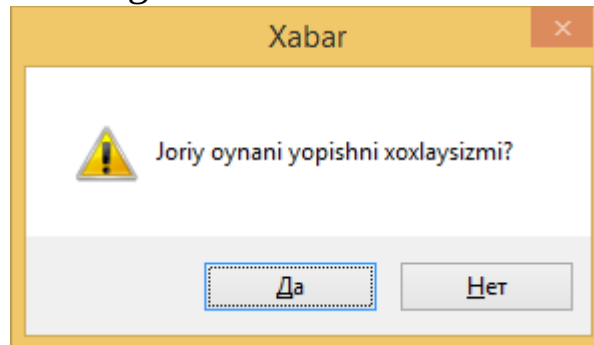
```
System::Void Form1_FormClosing(System::Object^
sender, System::Windows::Forms::FormClosingEventArgs^
e) {
    if ((MessageBox::Show(
        "Joriy oynani yopishni
xoxlaysizmi?",
        "Xabar",
        MessageBoxButtons::YesNo,
        MessageBoxIcon::Exclamation) ==
```

```

System::Windows::Forms::DialogResult::No)) {
    e->Cancel = true;
}
}

```

Dasturfragmentidahodisaningto‘liqyozilishikeltirilganliginingsaba biundahodisagaishlrvberilgan.



11.9-rasm. MessageBoxIcon bilanmuloqotoynasiniyaratish.

Mazkuroynadayo‘qtugmasibosilsa, formayopilmaydivaaksinchahatugmasibosilsa, formayopiladi.

MessageBoxDefaultButton - Enum tipidagiob‘ektbo‘lib, uhamyuqoridagidek, nomlarfazosivakutubxonabilanishlaydi. Muloqotoynasidastandarttugmalarnio‘rnatuvchio‘zgarnaslarnibelgilaydi

11.7-jadval. MessageBoxDefaultButton qabulqiluvchiqiymatlar

Nö	Qiymatno mi	vazifasi
1	Button1	Muloqotoynasidastarndartbirinchitugmanijoriytugmasifatid ao‘rnatish
2	Button2	Muloqotoynasidastarndartikkinchitugmanijoriytugmasifati dao‘rnatish
3	Button3	Muloqotoynasidastarndartuchinchitugmanijoriytugmasifati dao‘rnatish

Tugmabosilgandaformaniyopishniso‘rashmuloqotoynasiniyaratish dasturfragmentinikeltiramiz.

```

String^ message = "Joriy oynani yopishni  
xoxlaysizmi?";
String^ caption = "Xabar";
MessageBoxButtons buttons = MessageBoxButtons::YesNo;
System::Windows::Forms::DialogResult result;
System::Windows::Forms::DialogResult mayli =
System::Windows::Forms::DialogResult::Yes;

```

```

result = MessageBox::Show( this, message, caption,
buttons, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button2);
        if ( result == mayli )
        {
            this->Close();
        }

```

Odatd tugmalarning farqini aniqlash ancha murakkab, agar OT maskabo'lsa, farqlash mumkin.

MessageBoxOptions - Enum tipidagi ob'ekt bo'lib, u ham yuqoridagidek, nomlar fazosida kutubxon bilan ishlaydi. Muloqot oynasidagi standart xususiyatlarni o'rnatish uchun foydalaniladi.

11.8-Jadval . MessageBoxOptions

N	Qiymat nomi	vazifasi
1	DefaultDialogOptions	Oddiy muloqot oynalariga o'xshaydigan faollikni maydonlarga o'rnatiladi.
2	RightAlign	Muloqot oynasining ma'lumotini to'g'ri joylashtirish
3	RtlReading	Muloqot oynasidagi ma'lumotni o'ngdan chapga qarab o'qish uchun moslab joylashtirish.
4	ServiceNotification	Oddiy muloqot oynalariga o'xshaydigan faollikni maydonlarga o'rnatiladi va javob bermasdan o'tib ketish mumkin xususiyati mavjud

Tugma bosilganda forma ni yopishni so'rash muloqot oynasini yaratish dastur fragmentini keltiramiz.

```

String^ message = "Joriy oynani yopishni xoxlaysizmi?";

        String^ caption = "Xabar";
        MessageBoxButtons buttons =
        MessageBoxButtons::YesNo;
        System::Windows::Forms::DialogResult result;

        System::Windows::Forms::DialogResult mayli =
        System::Windows::Forms::DialogResult::Yes;

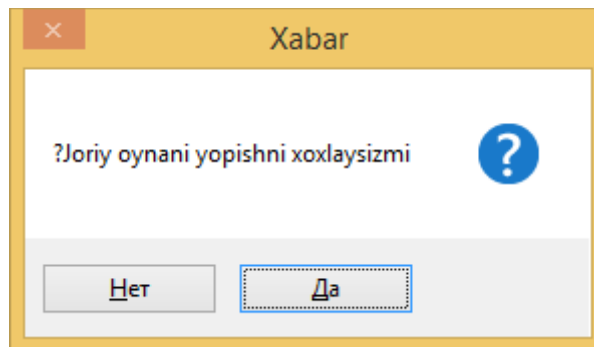
```

```

        result = MessageBox::Show( this,
message, caption, buttons,
MessageBoxIcon::Question,MessageBoxDefaultButton::But
ton1,MessageBoxOptions::RtlReading);
        if ( result == mayli )
        {
            this->Close();
        }
    }

```

Dasturfragmentini ishlatib ko'rib, natijasini tahlil qilish va nimalarni o'ngdanchapga qarabo'qish o'zgarganini ko'rish mumkin.



11.10-rasm. MessageBoxOptions bilan muloqot oynasini yaratish.

Barcha muloqot oynalarikabi yuqoridagi keltirilgan dasturfragmentlarid anko'rinib turibdiki, DialogResult tipini qaytaradi.

3. Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String) funksiyasi. Bu funksiyaga yordam fayli va ichki indeks fayllarga ko'rsatkich yaratadi.

```

String^ message = "Joriy oynani yopishni
xoxlaysizmi?";
String^ caption = "Xabar";
MessageBoxButtons buttons = MessageBoxButtons::YesNo;
System::Windows::Forms::DialogResult result;
System::Windows::Forms::DialogResult mayli =
System::Windows::Forms::DialogResult::Yes;

result = MessageBox::Show( this, message, caption,
buttons,
MessageBoxIcon::Question, MessageBoxDefaultButton::But
ton1, MessageBoxOptions::RightAlign,
"d:\tut.chm", HelpNavigator::KeywordIndex, "iv");
        if ( result == mayli )

```

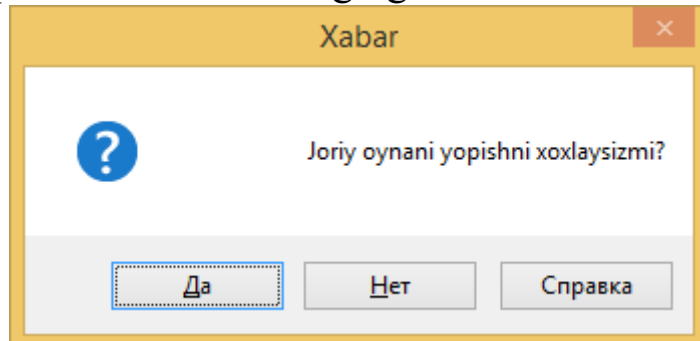


```

    {
        this->Close();
    }

```

Dasturfragmentini ishlatib ko'rib,
natijasini tahlil qilish va nimalarni o'zgarganini ko'rish mumkin.



11.11-rasm. Yordam tugmalari muloqot oynasini yaratish.

Dasturfragmentida e'tibor bilan qarasangiz,
MessageBoxButtons::YesNo tashlatilgan, ammo, muloqot oynasida 3
tugma chiqqan,
demak yordam tugmasining faol holatga kelganining ko'rish mumkin.

Show
usulining boshqayangi parametrlarni qabul qilmaganligi uchun ularni yozilis
hivada dastur fragmentlaridan keltiramiz.

4.Show(IWin32Window, String, String, MessageBoxButtons,
MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions,
String)

```

result = MessageBox::Show( this, message, caption,
buttons,
MessageBoxIcon::Question, MessageBoxDefaultButton::But
ton1, (MessageBoxOptions)0,
"d:\tut.chm", HelpNavigator::KeywordIndex, "iv");

```

5.Show(String, String, MessageBoxButtons, MessageBoxIcon,
MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)

```

System::Windows::Forms::DialogResult r3 =
MessageBox::Show( message, caption,
MessageBoxButtons::OK, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
(MessageBoxOptions)0, "mspaint.chm",
HelpNavigator::Index );

```

6.Show(String, String, MessageBoxButtons, MessageBoxIcon,
MessageBoxDefaultButton, MessageBoxOptions, String, String)

```
System::Windows::Forms::DialogResult r7 =  
MessageBox::Show(message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, "mspaint.chm",  
"mspaint.chm::/paint_brush.htm" );
```

7.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions)

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
MessageBoxOptions::RightAlign );
```

8.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String)

```
System::Windows::Forms::DialogResult r1 =  
MessageBox::Show( " message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, "mspaint.chm" );
```

9.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, Boolean)

```
System::Windows::Forms::DialogResult r =  
MessageBox::Show(message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, true );
```

10.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)

```
System::Windows::Forms::DialogResult r4 =  
MessageBox::Show( this, message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, "mspaint.chm",  
HelpNavigator::Index );
```

11.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton)

```
result = MessageBox::Show( this, message, caption,
```

```
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
MessageBoxOptions::RightAlign );
```

```
12.Show(IWin32Window, String, String, MessageBoxButtons,  
MessageBoxIcon)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question );
```

```
13.Show(String, String, MessageBoxButtons, MessageBoxIcon,  
MessageBoxDefaultButton)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1 );
```

```
14.Show(IWin32Window, String, String, MessageBoxButtons)
```

```
result = MessageBox::Show( this, message, caption,  
buttons);
```

```
15.Show(String, String, MessageBoxButtons, MessageBoxIcon)
```

```
result = MessageBox::Show(message, caption, buttons,  
MessageBoxIcon::Question);
```

```
16.Show(IWin32Window, String, String)
```

```
result = MessageBox::Show( this, message, caption);
```

```
17.Show(IWin32Window, String)
```

```
result = MessageBox::Show( this, message);
```

```
18.Show(String, String, MessageBoxButtons, MessageBoxIcon,  
MessageBoxDefaultButton, MessageBoxOptions)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
MessageBoxOptions::RightAlign );
```

```
19.Show(IWin32Window, String, String, MessageBoxButtons,  
MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions,  
String, HelpNavigator, Object)
```

```
System::Windows::Forms::DialogResult r6 =  
MessageBox::Show(this, message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, "mspaint.chm",
```

```
HelpNavigator::KeywordIndex, "ovals" );
```

Buusullarni.NET5 Preview 1, .NET Core3.1 3.0va.NET Framework

4.8, 4.7.2, 4.7.1, 4.7, 4.6.2, 4.6.1, 4.6, 4.5.2, 4.5.1, 4.5, 4.0, 3.5, 3.0, 2.0, 1.1kabivarintlarda qo'llabquvatlanadi.

Muloqotoynalarini yaratish. Foydalanuvchi tomonidan muloqotoynalarini yaratish oyna formasiga ishlov berish asosida amalga oshiriladi. Forma oyna sigamuloqotoynasini o'rnatish uchun quyidagi qadamlaribajaris hlozim.

1-qadam. Menyudan foydalanib, [menu] → [project] → [addClass] → [CLR] → [Windows Form] buyruqlar ketma ketligi asosida yangi forma qo'shiladi. Yoki, [menu] → [project] → [add New item] → [UI] → [Windows Form] hambajarsabo'ladi.

2-qadam. Yaratilgan yangi forma oyna sigi o'tib, forma ga quyidagi xususiyatlar beriladi. [AutoSize] xususiyatning qiymatini [true] ga, [StartPosition] xususiyatining qiymatini [CenterParent] ga, [FormBorderStyle] xususiyatining qiymatiga [none], [FixedDialog], [FixedToolWindows], [SizableToolWindows] larning birini o'rnatish mumkin.

3-qadam. Forma oyna sinini o'zini o'z xoliga dekkloyihalashingiz mumkin. Masalan, bir label, inputBox vabir button joylashtiramiz, ularni ham kerakli xususiyatlarini o'rnatamiz.

4-qadam. Asosiy forma ga o'tib, unga #include "MyForm.h" sarlavha faylni qo'shamiz.

Bumuloqotoynasi uchun yaratilgan forma oyna bilan ishlash uchun kerak.

5- Qadam. Asosiy forma ga bir label vabir button joylashtiramiz. Tugmaning klik hodisasida muloqotoyna sinini chaqirish dastur fragmentini yozamiz.

```
// MyForm myForm;  
// myForm.Show();  
// myForm.ShowDialog();  
  
MyForm^ myFormWith = gcnew  
MyForm();  
// myFormWith->Show();  
myFormWith->ShowDialog();
```

Dasturfragmentida izohga olib qo'yilgan yordam ham muloqot oynalarni yaratish va chaqirish mumkin. Ammo ularni farqlar mavjud. Shuning uchun ularni hammasini shu dasturfragmentida keltirdik. Birinchi qatorda formani `MyForm myForm` oddiyob'ektsifatida yaratilgan. Uning birinchi `Show()` usuli hisoblanib, bu usul bilan muloqot oynasini chaqirish mantiqan xato, chunki itizim oynani yaratadivastunlikni asosiy forma ga beradi. Bu holda yaratilgan oyna birlahzagako'rinadixolos. Ikkinchi usulibu `ShowDialog()` usulyaxshiyondashuvlardan bo'lib, muloqot formasigao'rnatilgan barcha xususiyatlarga rioya qiladi hamda foydalanish mumkin. Ikkinchi `MyForm()` konstruktordan foydalanib, `gcnew` operatoriasosida yaratilgan, uning birinchi funksiya `Show()` hamoldingisigao'xshashbo'lib, ammuostunlikni foydalanuvchiningo'ziga qo'yib beradi, ya'nifoydalanuvchi muloqot oyna ga javob bermas danturib, asosiy forma oynasigao'tishim mumkin. Ikkinchisiesabu `ShowDialog()` usulyaxshiyondashuvlardan bo'lib, muloqot formasigao'rnatilgan barcha xususiyatlarga rioya qiladi hamda foydalanish mumkin.

6-qadam.

Muloqot oynasini tugmasida quyidagicha algoritmni yozamiz. Muloqot oynasiga kiritilgan ma'lumotni asosiy forma ga olibo'tish uchun avval `public: System::String^ email;` kabibiro'zgaruvchi yaratib olamiz.

```
email = textBox1->Text;
this->Close();
```

7-qadam.

Asosiy formaning tugmasining hodisasigayozilgan dasturfragmentida vomi dan quyidagini qo'shib qshyamiz.

```
label1->Text = myFormWith->email;
```

8-qadam.

Loyihani ishlatib yaratilgan, yaratilgan muloqot oynani ishlashiniko'rish mumkin.

Muloqot oynani yuqoridara jarayon yaratish uchun uning ayni konstruktor yozish ham mumkin. Uni quyidagicha amalga oshiriladi.

```
public: MyForm(System::String^ title)
{
    InitializeComponent();
    _title = title;
}
public: System::String^ _title;
```

```
//...
private: System::Void MyForm_Load(System::Object^
sender, System::EventArgs^ e) {

        this->Text = _title;
    }
// Aspsiy formada esa
        MyForm^ myFormWith = gcnew
MyForm("Xabar");
        myFormWith->ShowDialog();
        label1->Text = myFormWith->email;
```

Bundayimkoniyatbilanfoydalanuvchiuchunixtiyoriymuloqotoynasi niyaratishmumkin.

Muloqotoynalaridanfoydalanishdasturfoydalanuvchilarigako‘plabq o‘layliklaryaratibberadi.
Harbirdasturdabirasosiyoyonagakamidafunksionalimkoniyatigaqarab 4-5 tamuloqooyinalaribo‘ladi.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Axborottizimlaridamuloqotoynalariningqandayguruhlaribor.
 2. Muloqotoynalarigaqo‘yiladigantalablarbormivabo‘lsa, nimauchun?
 3. Muloqotoynasidateskarialoqadegandanimanitushunasiz?
 4. Muloqotoynadanimalaraniqko‘rsatilishikerak?
 5. [Dialogs] tab
 6. oynasidajoylashganmuloqotkomponentalarinisanabbering?
 7. ColorDialogkomponentasiningvazifasiniayting?
 8. Qaysikomponentafoydalanuvchilaruchunpapkalarko‘rish, yaratishvatanlashimkoniniberadiganmuloqotoynasiniko‘rsatadi.
 8. Interaktivxabarlarniberishvaaniqjavoblarniolish Visual C++
- daqandaysinfimavjud.

9. MessageBox sinfnining show funksiyasiparamertlarinisana bbering?
10. MessageBoxOptions qanday ob'ekttipivanimavazifanibajaradi.
11. MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton ob'ektlarinibiriniqabulqiluvchikengaytirilganob'ekttipinominiayting.
12. Foydalanuvchitomonidanyaratiladiganmuloqotoynasiqandayyaratiladi.
13. Interaktivmuloqotoynalarikabinimalarniyaratishvaularniboqarish, keraklima'lumotlarniolishuchunishlatiladi.
14. Qaysisinfningnomlarfazosi System.Windows vakutubxonasi PresentationFramework.dll hisoblanadi.
- 15. ColorDialog muloqotoynasiningasosiyvazifasivaob'ektiqandayyaratiladi.**
16. ColorDialog muloqotoynasidanfoydalanishuchunqandaykonstruktorniishgatushishlozim.
17. CanRaiseEventsxususiyatiningvazifasiniayting?
18. komponentagailovaqilinadiganhodisalarro'yxatinioladixususiyatnominiayting?
19. Muloqotoynasiochilgandamaxsusranglarniyaratishuchunishlatiladiganboshqaruvelementlariko'rinibturishiniko'rsatuvchiqiymatnioladiganynyokio'rnatadiganxususiyatnominiayting?
20. Dispose()buqandayusul?
21. Komponentayokiuningkonteyneritomonidantaqdimetilganxizmatniifodalovchiob'ektniqaytaradiganhodisaniayting?
22. Disposedhodisaningvazifasiniayting?
23. Foydalanuvchiumumiymuloqotoynasidagiyordamtugmasinibosgandasodirbo'ladiganusulnominiayting?
24. Foydalanuvchilargakeraklikkomponentauchunshriftvauningxususiyatlarinio'rnatishimkoniniberuvchimuloqotoynasininominiayting?
25. PrintPreviewDialog muloqotoynasiqaysitabdajoylashganvavazifasiniyaniqaytibbering.
26. Dasturfragmentidabir StreamReader oqiminimauchunyaratiladi.
27. Win32Window Interface nimauchunishlatiladi.
28. MuloqotoynasidaOK va Cancel tugmalarinio'rnatishqaydayamalgaoshiriladi.
29. MessageBoxIcon muloqotoynalaridanimauchunishlatiladi.

30. Qandaydir hodisa haqida faqat xato amal bajarilganligi haqidama'lu mot beruvchi muloqotoynasi uchun konkanomivayozlashini tushuntirib bering.

31. Muloqotoynasidastandard tugmalarni o'rnatuvchi o'zgarishlarni belgilaydigan parametrni ayting?

32. RtlReading qaysi parametr uchun ishlatiladi va muloqotoynasidanimahodisa sodir bo'ladimi.

33. System::Windows::Forms::DialogResult r6 = MessageBox::Show(this, message, caption, MessageBoxButtons::OK, MessageBoxIcon::Question, MessageBoxDefaultButton::Button1, (MessageBoxOptions)0, "mspaint.chm", HelpNavigator::KeywordIndex, "ovals") – dastur fragmentini tushuntirib bering

34. Asosiy forma ganyang forma qanday qo'shiladi

35. Nima uchun [StartPosition] xususiyatining qiymatini [CenterParent] ga tenglashtiriladi.

36. ShowDialog() va Show() usullarining farqi nima?

37. Oddiy ob'ektni yaratish va konstruktor asosida ob'ektni yaratish farqi nima?

38. Muloqotoynalarni yaratishda foydalanuvchi yangi konstruktor yaratishim mumkinmi?

39. [menu] → [project] → [add New item] → [UI] → [Windows Form] ketma ketlik qanday amal bajaradi.

40. Muloqotoynasidagima'lumotlarni qanday qilib asosiy forma ga o'tkazishim mumkin.



AMALIY KO'NIKMA VA AMALAKALARNI ANIQLASH HAMDARI VOJLANTIRISH UCHUN ASSISMENT TOPSHIRIG'I.

ASSISMENT TOPSHIRIG'I	
☐	Muloqotoynalar bilan ishlashga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan. ☞ ☐ Bu vazifa larni kichik loyiha sifatida bajariladi, kichik loyiha deganda kamida 100 tavazifa sibo'lganda tushuniladi. Jamoalar bilan ishlashga maqsadga muvofiq.

3. Intellektual turli ma'lumotlarni kiritishni ta'lab qiluvchi IQ testlardan iborat dastur tuzing. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.
4. C++ dasturlash muhitini yarating. C++ kompilyatorini vakutub'onalarini tayyor tizimdan oling. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.
5. Barcha turdagi mulqot oynalarini o'z ichiga olgan, interaktiv savol-javob tizimini yarating va savollarni, javoblarni faylga saqlaydigan dastur tuzing. Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

DrawLines, DrawPolygon, DrawRectangle, DrawString, FillPie, FillEllipse, Chart, ChartAreas, Series, ChartType, Visual Basic Power Packs, PrintForm, LineShape, OvalShape, RectangleShape, DataRepeater.

✍Bilishshartbo'lgantushunchalar. Grafika, geometriya, xarakterning paydobo'lishi, sinf va sinf ob'ekti, xususiyat, hodisa, forma, komponenta dasturlashga oid dastlabki tushunchalar hamda C++ tilini qo'llab quvvatlovchi muhitda ishlatishni bilish lozim.

✍Bil bolasiz. Visual C++ muhitida dasturlash, Visual C++ ning grafikimkoniyatlari uchun Graphics sinfi xususiyatlarini va usullari, Graphics sinfi usul va resursidatavirlarni turlixildachizish, matn joylashtirish, rasmlarni yaratish va saqlash, tahrirlash amallari, Chart komponenta xususiyat va hodisalari, turlitipli diagrammalarni hosil qilish usullari, Funktsiyalarni grafiklarini qurish va Visual Basic Power Packs komponentalarini va ularni ishlatish, tasvirlarni harakatlantirish, oqimlarni animatsiyalarni yaratish gatalablar, o'rnatish va komponentalar bilan bog'lash usullari, shuningdek, Clip, ClipBounds, CompositingMode, CompositingQuality, DpiX, DpiY, Display, Document, Inch, Millimeter, Pixel, Point, World, Pen, DrawArc, DrawBezier, DrawCurve, DrawEllipse, DrawImage, DrawLines, DrawPolygon, DrawRectangle, DrawString, FillPie, FillEllipse, Chart, ChartAreas, Series, ChartType, Visual Basic Power Packs, PrintForm, LineShape, OvalShape, RectangleShape, DataRepeater usullar va boshqalar xususiyatlari, hodisalar o'rganishingiz mumkin.

REJA

1. Visual C++ ning grafikimkoniyatlari
2. Graphics sinfi usul va resursidatavirlarni qurish
3. Chart komponentalari.
4. Funktsiyalarning grafiklarini qurish.
5. Visual Basic Power Packs komponentalarini va ularni ishlatish.

KIRISH

Visual C++ o'z grafik tuzilma limug'it bo'lib hisoblanadi. Grafik deganda ixtiyoriy narsani kompyuterda piksellardahosil qilish tushuni ladi. Piksel da ikkita argumentlaribor, ya'ni ko'ordinatalari $A(x, y)$ ko'rinishida. Har qanday IDE muhitlarida grafikabilashishimkonibor. Ularturlicha nomlanishim mumkin. Ammo, ularning ruchkasi (qalami) vamo'y qalamiboladi. GUI

ko‘rinishdaishlashto‘liqgrafikabilanbog‘liqbo‘lib, texnikaninggrafikrejimdaishlashinita‘minlaydi. Hozirdadeyarlibarchafoydalanuvchilargrafikimkoniyatlaridanfoydalanibishlaydilar.

Visual

C++ninggrafikimkoniyatlari. Grafikimkoniyatgaegabo‘lgantizimlardaasosan, nuqta, chiziq, to‘rtburchak, aylana, ko‘pburchakkabishakllarniqamraboladi. GUI asosidagibarchaelementlargadiqqatbilanqarasangizshugrafiktuzilmalarda niboratbo‘ladi.

Visual

C++ningimkoniyatlariniko‘rishuchunGraphics

sinfiamurojaatqilamiz.

Businfningnomlarfazosi System.Drawing vakutubxonasi System.Drawing.Common.dll bo‘libhisoblanadi.

Chizishuchun

GDI+

modulniinkapsulyatsiyaqiladivabusinfndanmerosxo‘rolishmumkinemas.

Businf MarshalByRefObject, IDisposable, System::Drawing::IdeviceContextgaasoslanganbo‘lib, Object → MarshalByRefObject → Graphics sinfiningmerosxo‘rihisoblanadi.

Grafikasinfiekrangatasvirlarnichizishusullarinio‘ziichigaqamarabolangan.

Grafikamuayyanqurilmanikontekstbilanbog‘lashuchunhamxizmatqiladi.

Grafikob‘ektyordamidako‘pturlishaklvachiziqlarchizishmumkin.

Chiziqlarvashakllarchizishuchunmaxsus DrawGraphicalElement usullarnio‘rganishlozim. Buusullar DrawLine, DrawArc, DrawClosedCurve, DrawPolygon va DrawRectangle o‘zichigaoladi. Chiziqlarvashakllarchizishuchunqalamyordamidavashakllarnito‘ldirishuchunmo‘yqalamyordamidaamalgaoshiriladi.

Grafikasinfimkoniyatlariniuningxususiyatlarivausulariorqaliko‘rsa tibo‘tamiz. Shuningyangiparaemetkelsa, ungahamto‘xtalibo‘tiladi.

12.1-jadval.Grafikasinfixususiyatlari.

Xususiyaitnom i	vazifasi
Clip	Grafikaningchizilganchegarasinicheklaydiganchegaranio ladiyokio‘rnatadi.
ClipBounds	Grafikaningkesishchegarasinichegaralovchi RectangleF tuzilishinioladi.
CompositingM	Kompozittasvirlarchizishholatinioladiyokio‘rnatadi.

ode	
CompositingQuality	Grafikagachizilgankompozitsiontasvirlarningko'rsatishsifatini o'rnataadi.
DpiX	Ushbugrafikaninggorizontalo'lchaminioladi.
DpiY	Ushbugrafikaningvertikalo'lchaminioladi.
InterpolationMode	Ushbugrafikalarbilanbog'liqinterpolyatsiyarejiminioladiyoki o'rnataadi.
IsClipEmpty	Bugrafikakesishsohasinibo'shyokiyo'qliginiko'rsatib, birqiymatoladi.
IsVisibleClipEmpty	Bugrafikaaniqkesishsohasinibo'shyokiyo'qliginiko'rsatib, birqiymatoladi.
PageScale	Bugrafikauchunsahifamodulivabirlikmodulio'rtasidache garasinisozlash.
PageUnit	Bugrafikasahifakoordinatalariniuchunishlatiladigan o'lchov birligisozlash.
PixelOffsetMode	Bugrafikako'rsatishpaytidaPikseloffsetqandayko'rsatilganqiymatsozlash.
RenderingOrigin	Bugrafikako'rsatishrejimini o'rnataadi.
SmoothingMode	Grafikalaruchunko'rsatishsifatinioladiyoki o'rnataadi.
TextContrast	Matnko'rsatishuchun gamma qiymatinibelgilash.
TextRenderingHint	Bugrafikabilanbog'liqmatnuchunko'rsatishrejimini o'rnataadi.
Transform	Grafikalaruchungeometriko'zgarishiningnusxasinioladiyoki o'rnataadi.
VisibleClipBounds	Grafikninganiqkesishsohasinitekslashuchunto'rtburchakoladi.

Grafikasinfiusulariquyidagilardaniborat:

1. AddMetafileComment(Byte[]) - Rasmmetafayligaizohqo'shadi. BundaByte[] belgilimassivbo'lib, <System::Byte> ^ data tipidaaniqlanadivamasalan, array<Byte>^metaCom = {(Byte)'T',(Byte)'e',(Byte)'s',(Byte)'t'}; kabianiqlanishimumkin.

2. BeginContainer() - Grafikaninghozirgiholatibilangrafikkonteynernisaqlaydivayangigrafikkonteynerniochadivaishlatadi.

BeginContainer(Rectangle, Rectangle, GraphicsUnit) - Grafikjoriyholatibilanbirgrafikkonteynersaqlaydi, belgilanganparametrlar o'zgartirishbilanyangigrafikkonteynerdanfoydalanadiv aochadi. Bunda Rectangle to'rtburchak bo'lib, Rectangle(0,0,200,200) kabi aniqlanadi. Birinchito'rtburchakkonteyneruchunshkalanivaikkinchisikonteyneruchunsohao'zgarishirianiqlaydi. GraphicsUnit - konteyneruchuno'lchovbirliginianiqlashuchunishlatiladi. Uning enum bo'lib, quyidagi 12.2-jadvalgakeltirilganqiymatlarnioladiv GraphicsUnit::Pixel ko'rinishdafoydalaniladi.

12.2-jadval. GraphicsUnit qiymatlari.

Qiymat nomi	vazifasi
Display	Ko'rsatishqurilmasiningo'lchovbirliginibelgilaydi. Odatda video displeylaruchunpiksellarvaprinterlaruchun 1/100 dyuym.
Document	O'lchovbirligisifatidahujjatbirligini (1/300 dyuym) belgilaydi.
Inch	O'lchovbirligisifatidadyuymnibildiradi.
Millimeter	Millimetrlar o'lchovbirligisifatidabelgilaydi.
Pixel	O'lchovbirligisifatidaqurilmapikselinibelgilaydi.
Point	O'lchovbirligisifatidaprinterningnuqtasini (1/72 dyuym) belgilaydi.
World	Koordinatatizimibirliginio'lchovbirligisifatidabelgilaydi.

BeginContainer(RectangleF, RectangleF, GraphicsUnit) - grafikjoriyholatibilanbirgrafikkonteynersaqlaydi, belgilangansohalio'zgartirishbilanyangigrafikkonteynerdanfoydalanadiv aochadi. Bininchiikkita parametrlartuzilmabo'lib,

ikkixilaniqlanishimumkin:

RectangleF(PointF(30.0F,40.0F),SizeF(50.0F,100.0F)) va RectangleF(float x, float y, float width, float height); GraphicsUnit - esa konteyneruchuno'lchovbirliginianiqlashuchunishlatiladi.

3. Clear(Color) - butunchizmayuzasinitozalaydivabelgilanganfonrangibilanto'ldiradi. Color tuzilmabo'libuningqiymatlarigakengaytirishamali [::] bilanmurojaatqiladilar.

4.CopyFromScreen(Int32, Int32, Int32, Int32, Size) - Piksellito'rtburchakkamosranglima'lumotlarniekrandangrafikaningchiz mayuzasigabit-blokliuzatishniamalgaoshiradi. CopyFromScreen(int

sourceX, int sourceY, int destinationX, int destinationY,
 System::Drawing::Size blockRegionSize) -
 bundax, ulartu rrtburchakning moskoordinatalariva Size
 sohaning o'lchamidir. Hamtuzilmabo'lib, Size(int width, int height)
 ko'rinishdayaratiladi.

CopyFromScreen(Int32, Int32, Int32, Int32, Size,
 CopyPixelOperation) -

Piksellito'rtburchakkamosranglima'lumotlarniekrandangrafikaningchiz
 mayuzasigabit-blokliuzatishniamalgaoshiradi.

Buningparemetrlariyuqoridagidekbo'lib,

CopyPixelOperationdagimanbarangiyakuniyranggaolibkeladiganmaqsad
 rangibilanbirlashtirilganliginianiqlaydiva Blackness, CaptureBlt,
 DestinationInvert, MergeCopy, MergePaint, NoMirrorBitmap,
 NotSourceCopy, NotSourceErase, PatCopy, PatInvert, PatPaint,
 SourceAnd, SourceCopy, SourceErase, SourceInvert, SourcePaint,
 Whiteness kabiqiymatlarniqabulqiladi.

CopyFromScreen(Point, Point, Size) -
 Ekrandangrafikaningchizmayuzasigapiksellito'rtburchakkamosranglima'
 lumotlarnibit-blokliuzatishniamalgaoshiradi. Point -

bunuqtayaratuvc hituzilmabo'lib, o'zining 3ta Point(Int32), Point(Int32,
 Int32), Point(Size) konstruktorlarimavjud.

CopyFromScreen(Point, Point, Size, CopyPixelOperation) -
 ekrandangrafikaningchizmayuzasigapiksellito'rtburchakkamosranglima'
 lumotlarnibit-blokliuzatishniamalgaoshiradi.

5. CreateObjRef(Type) -
 ob'ekt bilan muloqot qilish uchun ishlatiladigan proksi ishlab chiqarish uchun
 zarur bo'lgan barcha tegishli ma'lumotlarni o'z ichiga olgan ob'ektyaratadi.
 Type - requestedType tipidagi abstrakt tip, array<Type^> { int::typeid,
 int::typeid }); kabianiqlanadi.

6. Dispose() - Bugrafika tomonidan ishlatiladigan barcha resurslarni
 null qiladi.

7. DrawArc. Bu funksiya cho'lchovlisohani ifodalovchi yoynichizadi.
 Uning quyidagi variantlaribor:

DrawArc(Pen, Int32, Int32, Int32, Int32, Int32, Int32) - void
 DrawArc(System::Drawing::Pen ^ pen, int x, int y, int
 width, int height, int startAngle, int sweepAngle);

DrawArc(Pen, Rectangle, Single, Single) - void
 DrawArc(System::Drawing::Pen ^ pen,


```
System::Drawing::Rectangle rect, float startAngle,
float sweepAngle);
```

```
DrawArc(Pen, RectangleF, Single, Single) - void
DrawArc(System::Drawing::Pen ^ pen,
System::Drawing::RectangleF rect, float startAngle,
float sweepAngle);
```

```
DrawArc(Pen, Single, Single, Single, Single, Single, Single) - void
DrawArc(System::Drawing::Pen ^ pen, float x, float y,
float width, float height, float startAngle, float
sweepAngle);
```

DrawArc funksiyasida Pen buqalambo‘lib, uninguchun Pen sinfi ishlatiladi. Pen ning 4 takonstruktor mavjud bo‘lib, Pen(Brush), Pen(Brush, Single), Pen(Color), Pen(Color, Single) konstruktorlardan foydalaniladi,

xususiyyat va usullarilarni o‘zgartirish orqali o‘rnatish mumkin. Brush bumo‘y qalambo‘lib, Brush sinfi ishlatiladi. Brush ning Brush() konstruktor mavjud bo‘lib,

xususiyyat va usullarilarni o‘zgartirish orqali o‘rnatish mumkin. Funksiyaning qolgan parametrlari barchasi yuqoridagi keltirilgan parametrlar kabi amalga oshiriladi.

8. DrawBezier. Bu funksiya berilgan 4 tanuqtali soha uchun splayinchizadi. Uning 3 taturibor:

```
DrawBezier(Pen, Point, Point, Point, Point) - void
DrawBezier(System::Drawing::Pen ^ pen,
System::Drawing::Point pt1, System::Drawing::Point
pt2, System::Drawing::Point pt3,
System::Drawing::Point pt4);
```

```
DrawBezier(Pen, PointF, PointF, PointF, PointF) - void
DrawBezier(System::Drawing::Pen ^ pen,
System::Drawing::PointF pt1, System::Drawing::PointF
pt2, System::Drawing::PointF pt3,
System::Drawing::PointF pt4);
```

```
DrawBezier(Pen, Single, Single, Single, Single, Single, Single,
Single, Single) - void DrawBezier(System::Drawing::Pen ^
pen, float x1, float y1, float x2, float y2, float
x3, float y3, float x4, float y4);
```

9. DrawBeziers. Bu funksiya berilgan nuqtalar sohasi uchun splayinchizadi. Uning 2 taturibor:

```

        DrawBeziers(Pen,          Point[])          -          void
DrawBeziers(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);
        DrawBeziers(Pen,          PointF[])          -          void
DrawBeziers(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);

```

Funksiyada Point[] - ikkio'lchovlitekislikdanuqtabelgilaydiganxva y koordinatalaruchuntartiblijuftlikniifodalaydimassivdir.

10. DrawClosedCurve. Bu funksiyayopiqegrichiziqchizishgamo'ljallangan. Uning 4 taturibor:

```

        DrawClosedCurve(Pen,          Point[])          -          void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::Point> ^ points);
        DrawClosedCurve(Pen, Point[], Single, FillMode) - void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::Point> ^ points, float
tension,          System::Drawing::Drawing2D::FillMode
fillmode);
        DrawClosedCurve(Pen,          PointF[])          -          void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::PointF> ^ points);
        DrawClosedCurve(Pen, PointF[], Single, FillMode) - void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::PointF> ^ points, float
tension,          System::Drawing::Drawing2D::FillMode
fillmode);

```

Bu funksiyada FillMode tipibor, busohanibo'yaashturinibelgilaydi. FillMode ning 2 xilusulibor, Alternate - Muqobilto'ldirishva Winding o'rabto'ldirishrejimlarinibelgilaydi.

11. DrawCurve. Bu funksiyaberilganparametrlarasosidaspalyn (egrichiziq) chizadi. Uning 7 taturimavjud.

```

        DrawCurve(Pen, Point[])          -          void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);
        DrawCurve(Pen, Point[], Int32, Int32, Single) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points, int offset, int
numberOfSegments, float tension);

```

```

    DrawCurve(Pen, Point[], Single) - void
    DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points, float tension);
    DrawCurve(Pen, PointF[]) - void
    DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);
    DrawCurve(Pen, PointF[], Int32, Int32) - void
    DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, int offset, int
numberOfSegments);
    DrawCurve(Pen, PointF[], Int32, Int32, Single) - void
    DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, int offset, int
numberOfSegments, float tension);
    DrawCurve(Pen, PointF[], Single) - void
    DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, float tension);

```

Bufunksiyaning barcha parametrlaritasniflangan.

12. DrawEllipse.

Bufunksiyaberilgan parametrlarasosida ellipischizadi. Uning 4 taturimavjud.

```

    DrawEllipse(Pen, Int32, Int32, Int32, Int32) - void
    DrawEllipse(System::Drawing::Pen ^ pen, int x, int y,
int width, int height);
    DrawEllipse(Pen, Rectangle) - void
    DrawEllipse(System::Drawing::Pen ^ pen,
System::Drawing::Rectangle rect);
    DrawEllipse(Pen, RectangleF) - void
    DrawEllipse(System::Drawing::Pen ^ pen,
System::Drawing::RectangleF rect);

```

```

    DrawEllipse(Pen, Single, Single, Single, Single) - void
    DrawEllipse(System::Drawing::Pen ^ pen, float x,
float y, float width, float height);

```

Bufunksiyaning barcha parametrlaritasniflangan.

13. DrawIcon.

Bufunksiyaberilgan parametrlarasosidabelgilanganbelgibo'yichachizadi. Uning 3 taturimavjud.

```

    DrawIcon(Icon, Int32, Int32) - void
DrawIcon(System::Drawing::Icon ^ icon, int x, int y);
    DrawIcon(Icon, Rectangle) - void
DrawIcon(System::Drawing::Icon ^ icon,
System::Drawing::Rectangle targetRect);
    DrawIconUnstretched(Icon, Rectangle) - void
DrawIconUnstretched(System::Drawing::Icon ^ icon,
System::Drawing::Rectangle targetRect);
    Bufunksiyaningbarchaparametrlaritasniflangan.

```

14. DrawImage.

Bufunksiyaberilganparametrlarasosidabelgilangantasvirnichizadi. Uning 30 taturimavjud.

```

    DrawImage(Image, Int32, Int32)- void
DrawImage(System::Drawing::Image ^ image, int x, int
y);
    DrawImage(Image, Int32, Int32, Int32, Int32) - void
DrawImage(System::Drawing::Image ^ image, int x, int
y, int width, int height);
    DrawImage(Image, Int32, Int32, Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image, int x, int
y, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Point) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Point point);
    DrawImage(Image, Point[]) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes) - void DrawImage(System::Drawing::Image ^
image, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort, Int32) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback, int callbackData);
    DrawImage(Image, PointF) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::PointF point);
    DrawImage(Image, PointF[]) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit) -
void DrawImage(System::Drawing::Image ^ image,
cli::array <System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort, Int32) -
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback, int callbackData);
    DrawImage(Image, Rectangle) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle rect);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit) - void DrawImage(System::Drawing::Image ^
image, System::Drawing::Rectangle destRect, int srcX,
int srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image,

```

```

System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);

```

```

DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort, IntPtr) -

```

```

void DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback, IntPtr callbackData);

```

```

DrawImage(Image, Rectangle, Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit) - void DrawImage(System::Drawing::Image ^
image, System::Drawing::Rectangle destRect, float
srcX, float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,

```

```

System::Drawing::GraphicsUnit          srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback);

```

```

    DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort, IntPtr) -
void    DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit          srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback, IntPtr callbackData);

```

```

    DrawImage(Image, RectangleF) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::RectangleF rect);

```

```

    DrawImage(Image, RectangleF, RectangleF, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

    DrawImage(Image, Single, Single) - void
DrawImage(System::Drawing::Image ^ image, float x,
float y);

```

```

    DrawImage(Image, Single, Single, RectangleF, GraphicsUnit)
- void DrawImage(System::Drawing::Image ^ image, float
x, float y, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

    DrawImage(Image, Single, Single, Single, Single) - void
DrawImage(System::Drawing::Image ^ image, float x,
float y, float width, float height);

```

Bufunksiyaningparametrlariko‘pchiligitasvirlargaishlovberishbilanb
 og‘liq. Shuninguchuno‘maishvaqtidazarurlarparametrlarikeltirilgan.
 Shungdek, birxulosashundaniboratki,
 meno‘zimbularninghammasiniishlatibko‘rmaganman.

15. DrawImageUnscaled.

Bufunksiyaberilganparametrlarasosidatasvirnibelgilanganjoydaaslkattali
 gidanfoydalanibchizadi. Uning 5 taturimavjud.


```
DrawImageUnscaled(Image, Int32, Int32) - void
DrawImageUnscaled(System::Drawing::Image ^ image, int
x, int y);
```

```
DrawImageUnscaled(Image, Int32, Int32, Int32, Int32) - void
DrawImageUnscaled(System::Drawing::Image ^ image, int
x, int y, int width, int height);
```

```
DrawImageUnscaled(Image, Point) - void
DrawImageUnscaled(System::Drawing::Image ^ image,
System::Drawing::Point point);
```

```
DrawImageUnscaled(Image, Rectangle) - void
DrawImageUnscaled(System::Drawing::Image ^ image,
System::Drawing::Rectangle rect);
```

```
DrawImageUnscaledAndClipped(Image, Rectangle) - void
DrawImageUnscaledAndClipped(System::Drawing::Image ^
image, System::Drawing::Rectangle rect);
```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

16. DrawLine. Bufunksiyaberilgan parametrlarasosidachiziqchizadi. Uning 4 taturimavjud.

```
DrawLine(Pen, Int32, Int32, Int32, Int32) - void
DrawLine(System::Drawing::Pen ^ pen, int x1, int y1,
int x2, int y2);
```

```
DrawLine(Pen, Point, Point) - void
DrawLine(System::Drawing::Pen ^ pen,
System::Drawing::Point pt1, System::Drawing::Point
pt2);
```

```
DrawLine(Pen, PointF, PointF) - void
DrawLine(System::Drawing::Pen ^ pen,
System::Drawing::PointF pt1, System::Drawing::PointF
pt2);
```

```
DrawLine(Pen, Single, Single, Single, Single) - void
DrawLine(System::Drawing::Pen ^ pen, float x1, float
y1, float x2, float y2);
```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

22. DrawLines. Bufunksiyaberilgan parametrlarasosida ulanadigan chiziqchizadi. Uning 2 taturimavjud.

```

        DrawLines(Pen, Point[]) - void
DrawLines(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);

```

```

        DrawLines(Pen, PointF[]) - void
DrawLines(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);

```

Bufunksiyaining barcha parametrlari talqil qilingan.

17. DrawPath.

Bufunksiya berilgan parametrlar asosida bog'langan chiziqlar va egrichiziqlar qatorini chizadi. Bu sinf meros xo'r bo'lishi mumkin emas.

```

        DrawPath(Pen, GraphicsPath) - void
DrawPath(System::Drawing::Pen ^ pen,
System::Drawing::Drawing2D::GraphicsPath ^ path);

```

18. DrawPie. Bufunksiya berilgan parametrlar asosida koordinata jufti, kengligi, balandligi va ikkita radial chiziq bilan belgilangan ellips bilan belgilangan nok shaklini chizadi. Uning 4 taturim mavjud.

```

        DrawPie(Pen, Int32, Int32, Int32, Int32, Int32, Int32) - void
DrawPie(System::Drawing::Pen ^ pen, int x, int y, int
width, int height, int startAngle, int sweepAngle);

```

```

        DrawPie(Pen, Rectangle, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen,
System::Drawing::Rectangle rect, float startAngle,
float sweepAngle);

```

```

        DrawPie(Pen, RectangleF, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen,
System::Drawing::RectangleF rect, float startAngle,
float sweepAngle);

```

```

        DrawPie(Pen, Single, Single, Single, Single, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen, float x, float y,
float width, float height, float startAngle, float
sweepAngle);

```

Bufunksiyaining barcha parametrlari talqil qilingan.

19. DrawPolygon.

Bufunksiya berilgan parametrlar asosida ko'p shaklini chizadi. Uning 4 taturim mavjud.

```

        DrawPolygon(Pen, Point[]) - void
DrawPolygon(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);

```

```

    DrawPolygon(Pen, PointF[]) - void
DrawPolygon(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);
    Bufunksiyainingbarchaparametrlarita'lilqilingan.
20. DrawRectangle.
Bufunksiyaberilganparametrlarasosidato'rtburchakshaklinichizadi.
Uning 3 taturimavjud.
    DrawRectangle(Pen, Int32, Int32, Int32, Int32) - void
DrawRectangle(System::Drawing::Pen ^ pen, int x, int
y, int width, int height);
    DrawRectangle(Pen, Rectangle) - void
DrawRectangle(System::Drawing::Pen ^ pen,
System::Drawing::Rectangle rect);
    DrawRectangle(Pen, Single, Single, Single, Single) - void
DrawRectangle(System::Drawing::Pen ^ pen, float x,
float y, float width, float height);
    Bufunksiyainingbarchaparametrlarita'lilqilingan.
21. DrawRectangles.
Bufunksiyaberilganparametrlarasosidaulanadiganto'rtburchakshaklinichi
zadi. Uning 3 taturimavjud.
    DrawRectangles(Pen, Rectangle[]) - void
DrawRectangles(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Rectangle> ^ rects);
    DrawRectangles(Pen, RectangleF[]) - void
DrawRectangles(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::RectangleF> ^ rects);
    Bufunksiyainingbarchaparametrlarita'lilqilingan.
22. DrawString.
Bufunksiyaberilganparametrlarasosidashriftob'ektlarbilanbelgilanganjoy
dabelgilanganmatnchizadi (joylashtiradi). Uning 6 taturimavjud.
    DrawString(String, Font, Brush, PointF) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::PointF point);
    DrawString(String, Font, Brush, PointF, StringFormat) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::PointF point,
System::Drawing::StringFormat ^ format);

```

```

    DrawString(String, Font, Brush, RectangleF) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::RectangleF layoutRectangle);

```

```

    DrawString(String, Font, Brush, RectangleF, StringFormat) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::RectangleF layoutRectangle,
System::Drawing::StringFormat ^ format);

```

```

    DrawString(String, Font, Brush, Single, Single) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush, float x,
float y);

```

```

    DrawString(String, Font, Brush, Single, Single, StringFormat) -
void
DrawString(System::String ^ s,
System::Drawing::Font ^ font, System::Drawing::Brush
^ brush, float x, float y,
System::Drawing::StringFormat ^ format);

```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

23. EndContainer.

Bufunksiyaberilgan parametrlarasosidajoriy grafik konteynerni yopadiva
BeginContainer()

usuli uchun koʻrsatki chorqalisanlangan bugrafik holatini chizadi.

```

EndContainer(GraphicsContainer) - void
EndContainer(System::Drawing::Drawing2D::GraphicsCont
ainer ^ container);

```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

24. EnumerateMetafile.

Bufunksiyaberilgan parametrlarasosidamatn nimeta faylga uzatadivabelgilan
joydachizadi (joylashtiradi). Uning 36 taturim mavjud.

```

EnumerateMetafile(Metafile, Point,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Point destPoint,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);

```

```

EnumerateMetafile(Metafile, Point,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile

```

```

^    metafile,    System::Drawing::Point    destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile,                                Point,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^    metafile,    System::Drawing::Point    destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback,                                IntPtr                                callbackData,
System::Drawing::Imaging::ImageAttributes                                ^
imageAttr);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^    metafile,    System::Drawing::Point    destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback,                                IntPtr                                callbackData,
System::Drawing::Imaging::ImageAttributes                                ^
imageAttr);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^    metafile,    System::Drawing::Point    destPoint,
System::Drawing::Rectangle                                srcRect,
System::Drawing::GraphicsUnit                                srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^    metafile,    System::Drawing::Point    destPoint,
System::Drawing::Rectangle                                srcRect,
System::Drawing::GraphicsUnit                                unit,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback,                                IntPtr                                callbackData,
System::Drawing::Imaging::ImageAttributes                                ^
imageAttr);

```

```

EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc    ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^

```



```

destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback,          IntPtr          callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);

```

```

EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);

```

```

EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc, IntPtr,
ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);

```

25. Equals.

Belgilanganob'ektjoriyob'ektgatengyokiyo'qliginianiqlaydi.
 Equals(Object) -
 Belgilanganob'ektjoriyob'ektgatengyokiyo'qliginianiqlaydi.

26. ExcludeClip.

Bufunksiyaberilganparametrlarasosidako'rsatilganmaydonniistisnoqilish
 uchunushbugrafikaningkliphududiniyangilaydi. Uning 2 taturimavjud.

```

ExcludeClip(Rectangle) - void
ExcludeClip(System::Drawing::Rectangle rect);
ExcludeClip(Region) - void
ExcludeClip(System::Drawing::Region ^ region);
Bufunksiyainingbarchaparametrlarita'lilqilingan.

```

27. FillClosedCurve.

Bufunksiyaberilganparametrlarasosidayopiqegrichiziqisplaynnibo'yyalg
 anholdachizadi. Uning 6 taturimavjud.

```

FillClosedCurve(Brush, Point[]) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points);
FillClosedCurve(Brush, Point[], FillMode) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillmode);
FillClosedCurve(Brush, Point[], FillMode, Single) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillmode, float
tension);

```

```

FillClosedCurve(Brush, PointF[]) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points);
FillClosedCurve(Brush, PointF[], FillMode) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillmode);
FillClosedCurve(Brush, PointF[], FillMode, Single)- void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillmode, float
tension);

```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

28. FillEllipse.

Bufunksiyaberilgan parametrlarasosida ellipsisniboʻyialgan holda chizadi.
Uning 4 taturimavjud.

```

FillEllipse(Brush, Int32, Int32, Int32, Int32) - void
FillEllipse(System::Drawing::Brush ^ brush, int x,
int y, int width, int height);
FillEllipse(Brush, Rectangle) - void
FillEllipse(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect);
FillEllipse(Brush, RectangleF) - void
FillEllipse(System::Drawing::Brush ^ brush,
System::Drawing::RectangleF rect);
FillEllipse(Brush, Single, Single, Single, Single) - void
FillEllipse(System::Drawing::Brush ^ brush, float x,
float y, float width, float height);

```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

29. FillPath.

Bufunksiyaberilgan parametrlarasosida bogʻlangan va egri chiziqlar boʻyialgan holda chizadi.

```

FillPath(Brush, GraphicsPath) - void
FillPath(System::Drawing::Brush ^ brush,
System::Drawing::Drawing2D::GraphicsPath ^ path);

```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

30. FillPie.

Bufunksiyaberilgan parametrlarasosida bir juft koordinatalar, kenglik,

balandlikvaikkitaradialchiziqbilanbelgilanganellipsnibo‘yalganholdachizadi. Uning 3 taturimavjud.

```
FillPie(Brush, Int32, Int32, Int32, Int32, Int32, Int32) - void
FillPie(System::Drawing::Brush ^ brush, int x, int y,
int width, int height, int startAngle, int
sweepAngle);
```

```
FillPie(Brush, Rectangle, Single, Single) - void
FillPie(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect, float startAngle,
float sweepAngle);
```

```
FillPie(Brush, Single, Single, Single, Single, Single, Single) -
void FillPie(System::Drawing::Brush ^ brush,
float x, float y, float width, float height, float
startAngle, float sweepAngle);
```

Bufunksiyainingbarchaparametrlarita‘lilqilingan.

31. FillPolygon.

Bufunksiyaberilganparametrlarasosidamaydonnibo‘yalganholdachizadi. Uning 4 taturimavjud.

```
FillPolygon(Brush, Point[]) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillMode);
```

```
FillPolygon(Brush, Point[], FillMode) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points);
```

```
FillPolygon(Brush, PointF[]) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillMode);
```

```
FillPolygon(Brush, PointF[], FillMode) - void
FillRectangle(System::Drawing::Brush ^ brush, int x,
int y, int width, int height);
```

Bufunksiyainingbarchaparametrlarita‘lilqilingan.

32. FillRectangle.

Bufunksiyaberilganparametrlarasosidato‘rtburchaknibo‘yalganholdachizadi. Uning 4 taturimavjud.

```
FillRectangle(Brush, Int32, Int32, Int32, Int32) - void
FillRectangle(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect);
```

```

FillRectangle(Brush, Rectangle) - void
FillRectangle(System::Drawing::Brush ^ brush,
System::Drawing::RectangleF rect);
FillRectangle(Brush, RectangleF) - void
FillRectangle(System::Drawing::Brush ^ brush, float
x, float y, float width, float height);
FillRectangle(Brush, Single, Single, Single, Single) - void
FillRectangles(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Rectangle> ^ rects);

```

Bufunksiyaining barcha parametrlari ta'riflangan.

33. FillRectangles.

Bufunksiya berilgan parametrlar asosida ulanadigan to'rtburchakni bo'lgan holda chizadi. Uning 2 taturim mavjud.

```

FillRectangles(Brush, Rectangle[]) - void
FillRectangles(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::RectangleF> ^ rects);
FillRectangles(Brush, RectangleF[]) - void
FillRegion(System::Drawing::Brush ^ brush,
System::Drawing::Region ^ region);

```

Bufunksiyaining barcha parametrlari ta'riflangan.

34. FillRegion.

Bufunksiya berilgan parametrlar asosida soha ni bo'lgan holda chizadi.

```

FillRegion(Brush, Region) - void
FillRegion(System::Drawing::Brush ^ brush,
System::Drawing::Region ^ region);

```

Grafika sinfining 34 ta usullarini varinatlari asosida keltirdik.

Bu usullarni va xususiyatlar ni tasvirlarni qurish orqali ko'rib chiqamiz.

Graphics

sinfi usullari asosida tasvirlarni qurish. Tasvirlarni qurish uchun **PictureBox** komponentasidan foydalanamiz.

Chiziqli chizish. Bunda komponentaning berilgan joyida, rangli chiziqli chizishni ko'ramiz. Buning uchun 1 ta komponenta, 1 ta tugma va 4 ta textBox komponentalarini oynaga qulay qilib joylashtiramiz.

1-qadam.

Oynaning

Form1_Load

hoda asosida quyidagi algoritmnini joylashtiramiz.

```

this->Text = "DrawLine - chiziqli chizish";
button1->Text = "CHIZISH";

```

2-qadam.

KomponentavateTextBoxxususiyatlarinisozlashamallarnibajarishmumkin.

3-qadam. Chiziqchizishuchunnuqtalargaatabint
myPoint[4];o'zgaruvchisiolamiz.

4-qadam. Komponentaning Paint
deganhodisasigaquyidagialgoritmnijoylashtiramiz.

```
e->Graphics->DrawLine(System::Drawing::Pens::Red,  
myPoint[1], myPoint[2], myPoint[3],  
myPoint[4]);
```

5-qadam.

Tugmaningbutton1_Clickhodisasidaquyidagialgoritmnijoylashtirami
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);  
myPoint[2] = Convert::ToInt32(textBox2->Text);  
myPoint[3] = Convert::ToInt32(textBox3->Text);  
myPoint[4] = Convert::ToInt32(textBox4->Text);  
pictureBox1->Refresh();
```

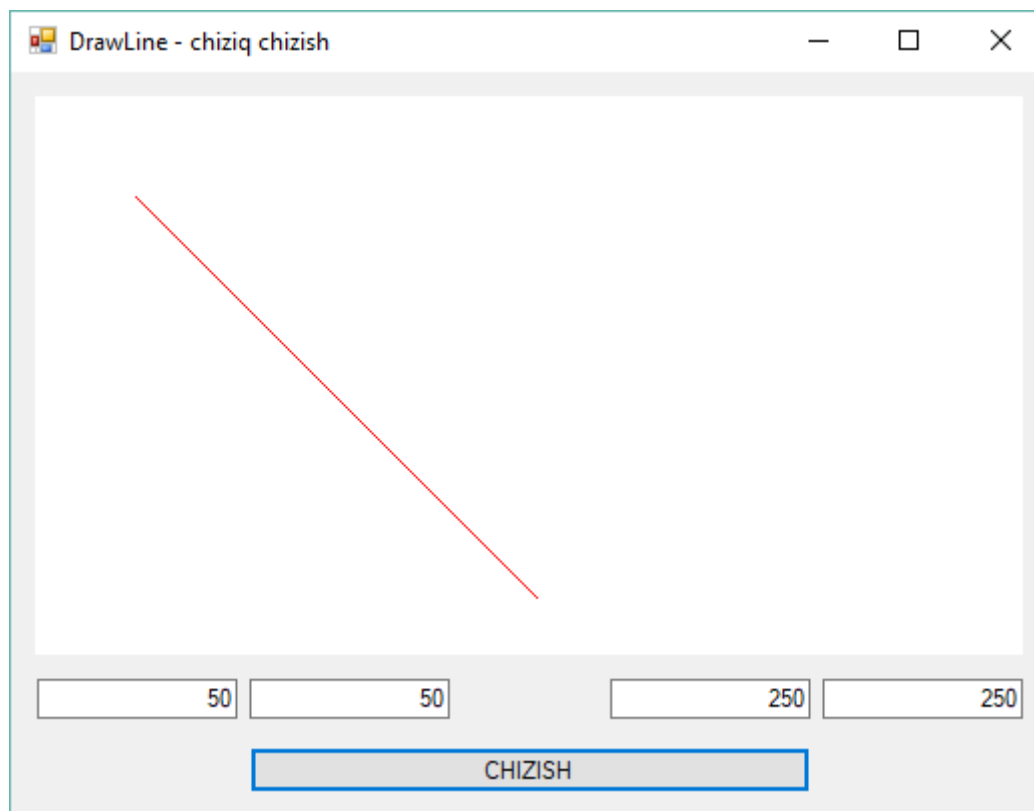
Bundamatndansontipigao'tishamalidanfoydalanilganvaharbirtugma
bosilgandakomponentaningchizmasio'zgaribturadi.

Agarloyihaniishgatushursangiz, textBoxlargakeraklisonlarnikiritib,
chiziqchizishmumkin.

Komponentaningo'lchamidankattachiziqlarnichizishmumkinammoko'ri
nmaydi. Unibrshqarishlozim,

ya'nikeraklikattalikdanoshibketgandafoydalanuvchigamuloqotoynasibila
nxabarberishmumkin, buni
textBoxlarningfokusio'zgargandaamalgaoshirishmaqsadgamuvofiq.

Chiziqchizishningnatijasi 12.1-rasmdakeltirilgan.



12.1-rasm. Chiziqchizishnatijasi.

Uchburchakchizish. Bundak komponentaning berilgan joyida, rangli chiziqlar orqali uchburchak chizishni ko'ramiz. Buning uchun 1 takomponenta, 1 tatugmava 6 tateTextBox komponentalarini oynaga qulay qilib joylashtiramiz. Chunki 3 tachiziqning tutashtirsak, uchburchak hosil bo'ladi.

1-qadam. Oynaning Form1_Load hodisasida quyidagi algoritmnijoylashtiramiz.

```
this->Text = "DrawLine - uchburchak chizish";
button1->Text = "CHIZISH";
```

2-qadam.

Komponenta va textBox xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. Chiziqchizish uchun nuqtalarga a tabint myPoint[6]; o'zgaruvchisi o'lamiz.

4-qadam. Komponentaning Paint degan hodisasiga quyidagi algoritmnijoylashtiramiz.

```
e->Graphics->DrawLine(System::Drawing::Pens::Red,
                        myPoint[1], myPoint[2], myPoint[3],
                        myPoint[4]);
e->Graphics->DrawLine(System::Drawing::Pens::Black,
                        myPoint[3], myPoint[4], myPoint[5],
                        myPoint[6]);
```

```
e->Graphics->DrawLine(System::Drawing::Pens::Blue,
                        myPoint[1], myPoint[2], myPoint[5],
myPoint[6]);
```

5-qadam.

Tugmaningbutton1_Clickhodisasidaquyidagialgoritmnijoylashtirami
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);
myPoint[2] = Convert::ToInt32(textBox2->Text);
myPoint[3] = Convert::ToInt32(textBox3->Text);
myPoint[4] = Convert::ToInt32(textBox4->Text);
myPoint[5] = Convert::ToInt32(textBox3->Text);
myPoint[6] = Convert::ToInt32(textBox4->Text);
pictureBox1->Refresh();
```

Agarloyihaniishgatushursangiz, textBoxlargakeraklisonlarnikiritib,
uchburchaknichizishmumkin.

Ellipischizish.Bundakomponentaningberilganjoyida,
rangliellipischizishniko‘ramiz. Buninguchun 1 takomponenta, 1
tatugmava 4 tatetextBox
komponentalarinioynagaqulayqilibjoylashtiramiz.

1-qadam. Oynaning Form1_Load
hodisasidaquyidagialgoritmnijoylashtiramiz.

```
this->Text = "Ellipse - chizish";
button1->Text = "CHIZISH";
```

2-qadam.

KomponentavateTextBoxxususiyatlarinisozlashamallarnibajarishmumkin.

3-qadam. Chiziqchizishuchunnuqtalargaatabint
myPoint[4];o‘zgaruvchisiolamiz.

4-qadam. Komponentaning Paint
deganhodisasigaquyidagialgoritmnijoylashtiramiz.

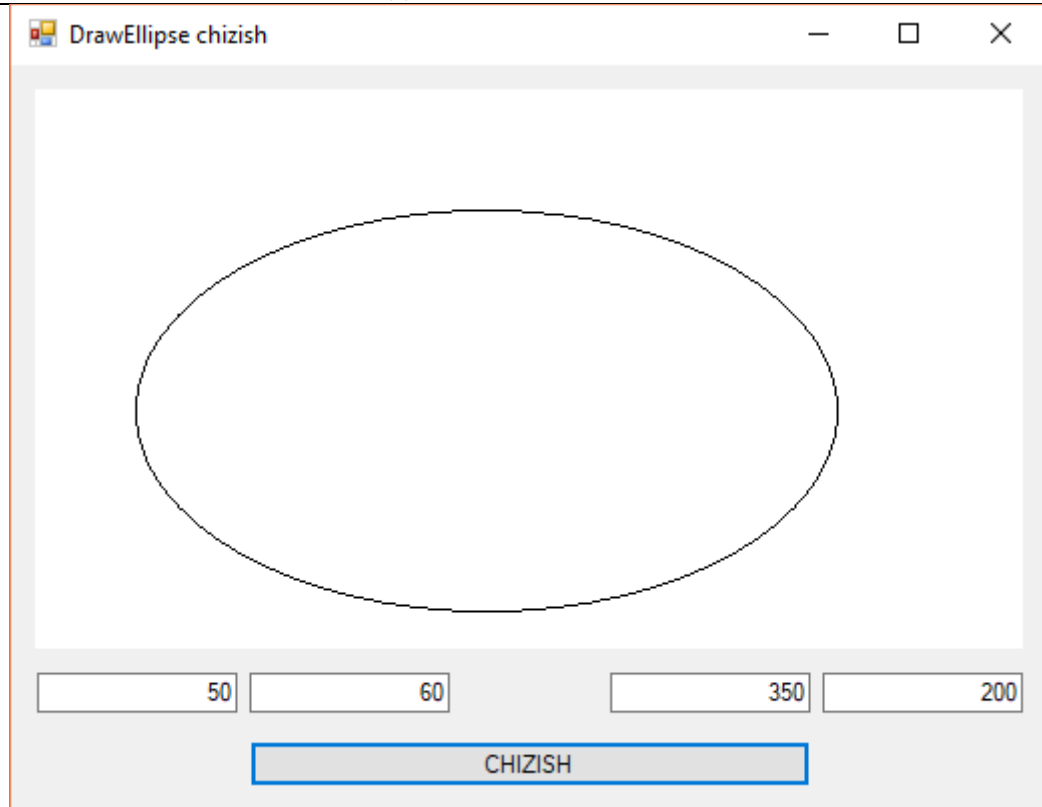
```
Pen ^ pen = gcnew Pen(Color::Black);
e->Graphics->DrawEllipse(pen,
                        myPoint[1], myPoint[2], myPoint[3],
myPoint[4]);
```

5-qadam.

Tugmaningbutton1_Clickhodisasidaquyidagialgoritmnijoylashtirami
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);
myPoint[2] = Convert::ToInt32(textBox2->Text);
```

```
myPoint[3] = Convert::ToInt32(textBox3->Text);
myPoint[4] = Convert::ToInt32(textBox4->Text);
pictureBox1->Refresh();
```



12.2-rasm. DrawEllipsechizishnatijasi.

Yuqoridakeltirilganchizishlarorqaliboshqaixtiyoriyshakllarnihamchizishmumkin. Turlishakllarnichizibbo‘yyashniko‘ramiz.

Turlishakllarnibo‘yyash.Bundakkomponentaningberilganjoyida,bo‘yyalganshakllarnichizishniko‘ramiz. Buninguchun**Brush** – mo‘yqalamdanfoydalanib, 1 takomponenta, 1 ta label va 1 ta Combox komponentalarinioynagaqulayqilibjoylashtiramiz.

1-qadam. Oynaning Form1_Load hodisasidaquyidagialgoritmnijoylashtiramiz.

```
this->Text = "Shaklni bo‘yash";
label1->Text = "Kerakli shaklni
tanlang:";
comboBox1->Text = "Shakllar";
comboBox1->Items-
>Add("To‘rtburchak");
comboBox1->Items->Add("Ellipis");
comboBox1->Items->Add("Doira");
comboBox1->Items->Add("Qirqilgan
doira ");
```

2-qadam.

KomponentavaComboBoxxususiyatlarinisozlashamallarnibajarishmumkin.

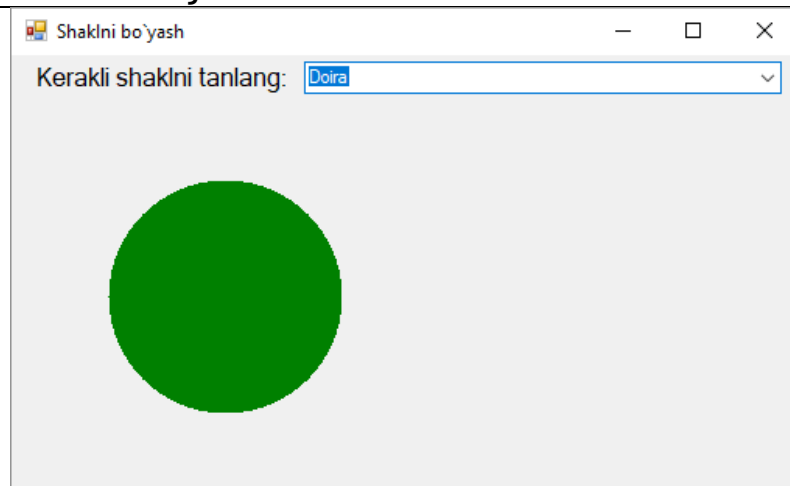
3-qadam.

ComboBox

komponentasiningcomboBox1_SelectedIndexChangedhodisasidaquyidagialgoritmnijoylashtiramiz.

```
Graphics ^grp = pictureBox1->CreateGraphics();
Brush ^ brsh = gcnew SolidBrush(Color::Green);
grp->Clear(SystemColors::Control);

switch (comboBox1->SelectedIndex){
    case 0:
        grp->FillRectangle(brsh,50,50,150,150);
break;
    case 1:
        grp->FillEllipse(brsh,50,50,300,150); break;
    case 2:
        grp->FillEllipse(brsh,50,50,150,150); break;
    case 3:
        grp->FillPie(brsh,50,50,150,150,150,100);
break;
    default:
        grp->Clear(SystemColors::Control);break;
}
```



12.3-rasm. Turlishakllarnibo'yyashnatijasi.

Ko'pburchaklarnichizish. Bundakomponentaningberilganjoyida, bo'yalganfoydalanuvchishaklarnichizishniko'ramiz.

Buninguchun **Brush** – mo‘yqalamdan foydalanib, 1 takomponenta, 1 ta button komponentasini oynaga qulay qilib joylashtiramiz.

1-qadam. Oynaning Form1_Load hodisasida quyidagi algoritmnijoylashtiramiz.

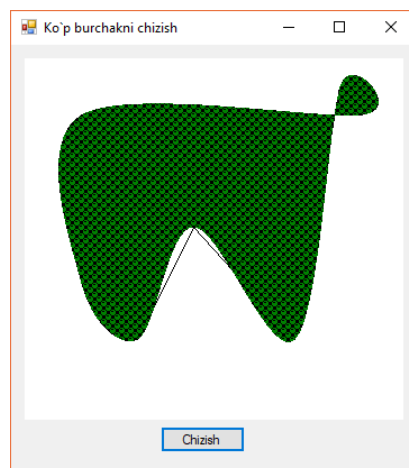
```
this->Text = "Ko‘p burchakni  
chizish";  
button1->Text = "Chizish";
```

2-qadam. Komponentava button xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. button komponentasining button1_Click hodisasida quyidagi algoritmnijoylashtiramiz.

```
Graphics^ graf = pictureBox1->CreateGraphics();  
Pen^ gPen = gcnew Pen( Color::Black,1);  
HatchBrush^ HBrush = gcnew HatchBrush(  
HatchStyle::Sphere, Color::Green, Color::Black );  
  
Point point1 = Point(50,50);  
Point point2 = Point(50,200);  
Point point3 = Point(100,250);  
Point point4 = Point(150,150);  
Point point5 = Point(240,250);  
Point point6 = Point(280,25);  
Point point7 = Point(300,50);  
array<Point> ^Points =  
{point1,point2,point3,point4,point5,point6,point7};  
  
graf->DrawPolygon( gPen, Points );  
graf->FillClosedCurve( HBrush, Points);
```

Bu algoritmdagi HatchBrush sinfi ishlatish uchun using namespace System::Drawing; Drawing2D; nomlar fazosini qo‘shib qo‘yish kerak. Algoritmda 7 tanuqtalarni birlashtirish orqali shakl chiziladi.



12.4-rasm. Turlishakllarnibo‘yyashnatijasi.

Hodisalarorqalishaklchizish. Hamma "Paint" kabidasturlarnibiladi. Uningengajoyibxususiyatlaridanbirisichqonchabilanchiziqlarchizishdir. Buniamalgaoshirishhuchunkomponentningsichqonchabilashishlash "**MousDown**",

"**MausUp**" i "**MouseMove**" hodisalaridan foydalanish mumkin.

Dasturning algoritmi g‘oyasi quyidagicha:

foydalanuvchi sichqonchaning chaptugmasini bosganda kursor orqasiga juda ko‘p kichik kvadratlarchizilab boshlaydi.

Bu kvadratlarning hajmi koddako‘rsatilgan. Bundantashqari, "tugma" tugmachasini rasimga tushiradigan shaklga ko‘chirishingiz kerak.

Bunda komponentaning berilgan joyida, sichqoncha hodisalar orqalishaklarni chizish niko‘ramiz.

Buning uchun **Brush** – mo‘yqalamdan foydalanib, 1 ta komponenta, 1 ta button komponentasini o‘ynaga qulay qilib joylashtiramiz.

1-qadam. Oynaning Form1_Load hodisasida quyidagi algoritmni joylashtiramiz.

```
button1->Text = "Tozalash";
this->Text = "Shakl chizish";
```

2-qadam. Komponenta va button xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. button komponentasining button1_Click hodisasida quyidagi algoritmni joylashtiramiz. Bu algoritm komponentaning tozalash uchun ishlatiladi.

```
Graphics ^ grp = pictureBox1->CreateGraphics();
grp->Clear(SystemColors::Window);
```

4-qadam.

Komponentaning sichqonchahodisalarini qaytaishlash uchun mantiqiy biroʻzgaruvchi olinadi, uning qiymati [0] boʻlsin. `Bool Drow = false;`

5-qadam.

Sichqonchak komponentaning ustiga kelgandamantiqiy oʻzgaruvchining qiymati 1 gaoʻzgaradi. Chunki chizishni boshlash uchun. Buni amalga oshirish uchun sichqonchani `pictureBox1_MouseDown` ho-disiga `Drow = true;` ni yozib qoʻyamiz.

6-qadam.

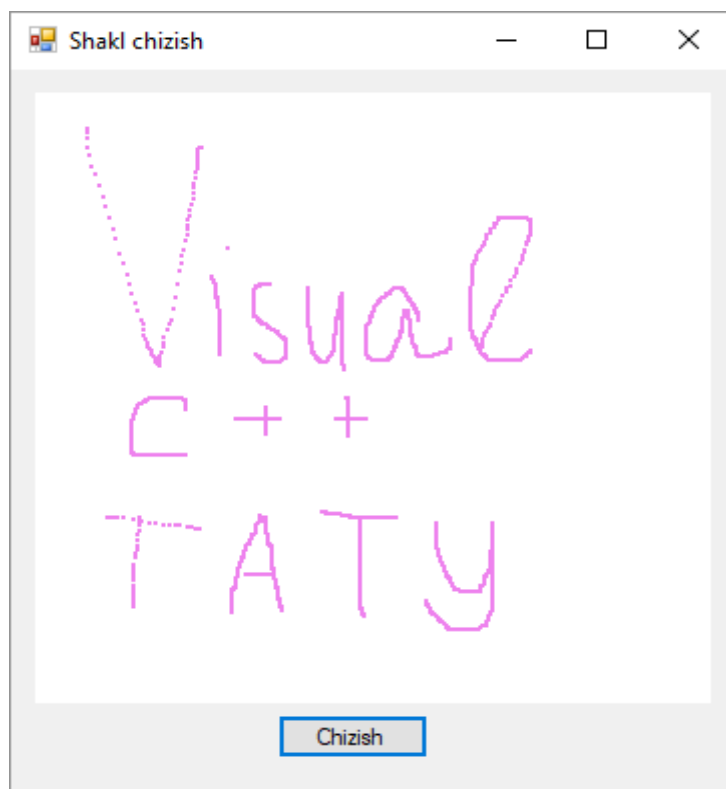
Sichqonchak komponentaning ustidan ketgandamantiqiy oʻzgaruvchining qiymati 0 gaoʻzgaradi. Chunki chizishni tugatish uchun. Buni amalga oshirish uchun sichqonchani `pictureBox1_MouseUp` ho-disiga `Drow = false;` ni yozib qoʻyamiz.

7-qadam.

Sichqonchak komponentaning ustida kelgandachizishi uchun quyidagi algoritmni kiritamiz.

```
Graphics^ graf = pictureBox1->CreateGraphics();  
    if (Drow == true){  
        graf->FillEllipse(Brushes::Violet, e->X, e->Y,  
3,3); // mo'y qalam qalinligi  
    }
```

Algoritmda shakllarni hosil qilish uchun juda kichik ellipschalardan foydalanamiz.



12.5-rasm. Sichqonchabilanturlishakllarnichizish.

Rasmlarnio‘zgartirish. Buninguchun fayllarni yuklash **openFileDialog** o‘gva fayllarni saqlash **saveFileDialog** komponentalaridan foydalanamiz.

Buninguchun **Brush** – mo‘yqalamdan foydalanib, 1 ta komponenta, 3 ta button komponentasini oynaga qulay qilib joylashtiramiz. Button larning birinchisi, rasmni tahrirlashni tozalashni amalga oshiradi, ikkinchisi, tahrirlangan rasmni saqlash uchun, uchinchisi, rasmni tahrirlash uchun yuklashni amalga oshiradi. `using namespace System::Drawing::Drawing2D;` ni qo‘shish b qo‘yish kerak.

1-qalam.

Rasmlarnio‘zgartirish uchun ishlatiladigan o‘zgaruvchilarni e‘lon qilamiz.

```
private: Bitmap ^ bmp_for_draw;
private: Point start_point;
private: bool Draw;
public: Pen^ pen_for_draw;
private: String ^ full_name_of_image;
```

2-qadam.

Dasturoynasining `Form1_Load` hodisasi gadastlab kisozlamalarni o‘rnatamiz.

```
this->Text = "Rasmni tahrirlash";
button1->Text = "Tozalash";
button2->Text = "Rasmni saqlash";
button3->Text = "Rasmni yuklash";
pen_for_draw = gcnew Pen(Color::Black, 4);
pen_for_draw->StartCap =
System::Drawing::Drawing2D::LineCap::Round;
pen_for_draw->EndCap =
System::Drawing::Drawing2D::LineCap::Round;
```

Bundak komponentalarni sozlash va bir qalamni yaratib olish, uning `StartCap`, `EndCap` xususiyatlarini o‘rnatish amalga oshiriladi.

3-qadam.

Komponentdagi rasmni tozalash uchun `button1_Click` hodisasi ga algoritmni yozamiz.

```
Graphics^ grp = pictureBox1->CreateGraphics();
grp->Clear(SystemColors::Window);
```

4-qadam.

Sichqoncha komponentaning ustidagi kichik chizish uchun `pictureBox1_MouseDown` hodisasi ga quyidagi algoritmni kiritamiz


```

if (e->Button ==
System::Windows::Forms::MouseButtons::Left){
    Draw = true;
    start_point = e->Location;
}

```

5-qadam.

Sichqonchakomponentaningustidaketgandachizishnito‘xtatishuchun
pictureBox1_MouseUp hodisasigaquyidagialgoritmnikiritamiz

```

if (e->Button ==
System::Windows::Forms::MouseButtons::Left){
    Draw = false;
}

```

6-qadam.

Komponentagakeraklirasmniyuklashuchun
button3_Clickhodisasigaquyidagialgoritmniyoziadi

```

OpenFileDialog ^ open_dialog = gcnew
OpenFileDialog();
open_dialog->Filter = "Image
Files(*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PN
G|All files (*.*)|*.*";
if (open_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    try{
        full_name_of_image = open_dialog->FileName;
        bmp_for_draw = gcnew Bitmap(open_dialog-
>FileName);
        //pictureBox1->Size = bmp_for_draw->Size;
        pictureBox1->SizeMode =
PictureBoxSizeMode::StretchImage;
        pictureBox1->Image = bmp_for_draw;
        pictureBox1->Invalidate();
    } catch(Exception^ e) {
        System::Windows::Forms::DialogResult rezult =
MessageBox::Show("Tanlangan faylni ochib bo'lmaydi "+
e->ToString(),"Diqqat", MessageBoxButtons::OK,
MessageBoxIcon::Error);
    }}

```

Bundagiasosinuqsonshundaniboratki,
agarrasmningo‘lchamlarikomponentaningo‘lchamlaribilanbixilbo‘lsa,

yaxshichizadi,
foydalanuvchiekraniningproporsiyasiniolibchizadi.

birxilbo‘lmasa,

7-qadam. Komponentadagirasnmnisaqlashuchun
button2_Clickhodisasigaquyidagialgoritmniyoziladi

```
if (pictureBox1->Image != nullptr){
String ^format = full_name_of_image-
>Substring(full_name_of_image->Length - 4, 4);
SaveFileDialog ^savedialog = gcnew SaveFileDialog();
savedialog->Title = "Rasmni saqlash ...";
savedialog->OverwritePrompt = true;
savedialog->CheckPathExists = true;
savedialog->Filter = "Image Files(*.BMP)|*.BMP|Image
Files(*.JPG)|*.JPG|Image Files(*.GIF)|*.GIF|Image
Files(*.PNG)|*.PNG|All files (*.*)|*.*";
savedialog->ShowHelp = true;
// If selected, save
if (savedialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    try{
bmp_for_draw->Save(savedialog->FileName,
System::Drawing::Imaging::ImageFormat::Jpeg);
} catch(Exception ^ e) {
System::Windows::Forms::MessageBox::Show("Impossible
to save image"+e->ToString(), "FATAL ERROR",
MessageBoxButtons::OK, MessageBoxIcon::Error);
} } }
```

8-qadam.

Sichqonchakomponentaningustidakelgandachizishiuchunquyidagialgorit
mnikiritamiz.

```
if (e->Button ==
System::Windows::Forms::MouseButtons::Left){
Graphics ^graf = Graphics::FromImage(pictureBox1-
>Image);
graf->DrawLine(pen_for_draw, start_point, e-
>Location);
start_point = e->Location;
pictureBox1->Invalidate();
}
```

Dasturni ishlatganda quyidagi oyna chiqadi.



12.6-rasm. Rasmlarni tahrirlash.

Matnlarni tasvirkabi joylashtirish. Bunday komponentaning belgilan gan joyida, aniqo'ldan ham livar angli matn ni joylashtirish niko'ramiz. Buning uchun komponentaga 3 ta textbox va 1 ta botton komponentlarini qulay qilib joylashtiramiz. Birinchi matn yozish uchun, qolgan ikkita matn ni joylashtirish nuqtasi uchun.

1-qadam. Textbox nirejimini multiline rejimigava scrolBars xususiyatiga Vertical qiymat o'rnatiladi.

2-qadam. Oynaning Form1_Load hodisasida quyidagi algoritm ni joylashtiramiz.

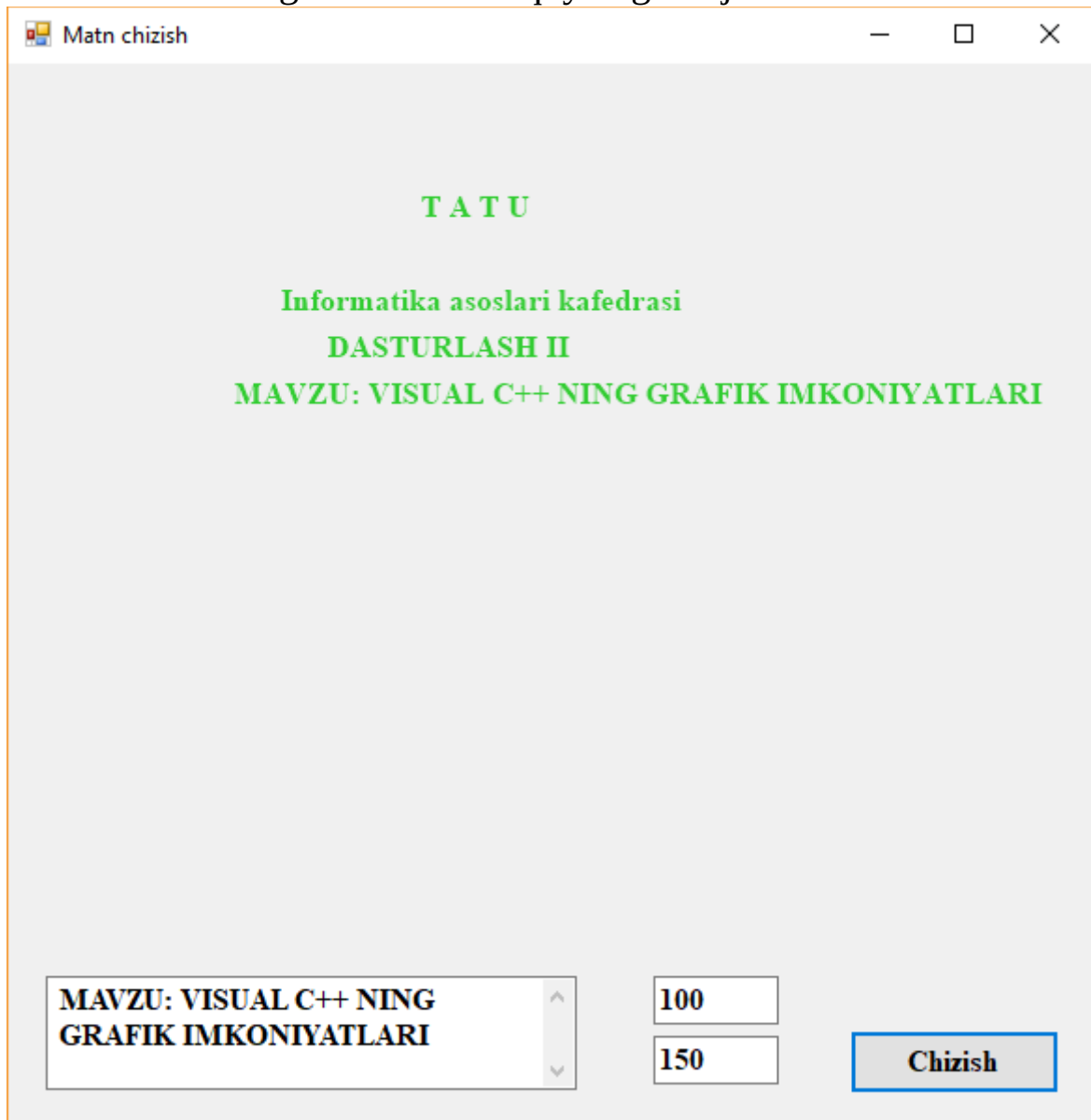
```
Font = gcnew System::Drawing::Font("Times New Roman",  
12, FontStyle::Bold);  
button1->Text = "Chizish";  
this->Text = "Matn chizish";
```

3-qadam. Komponentava button xususiyatlarini sozlash amallarni bajarish mumkin.

4-qadam. button komponentasining button1_Click hodisasida quyidagi algoritm ni joylashtiramiz. Bular algoritm komponentaning tozalash uchun ishlatiladi.

```
String^ Text = String::Format("{0}", textBox1->Text);
Brush^ brsh = gcnew SolidBrush(Color::LimeGreen);
Graphics^ G = pictureBox1->CreateGraphics();
G->TextRenderingHint =
System::Drawing::Text::TextRenderingHint::AntiAlias;
int xP = Convert::ToInt16(textBox2->Text);
int yP = Convert::ToInt32(textBox3->Text);
G->DrawString(Text, Font, brsh, xP, yP); //
joylashtirish manzili
```

Dasturni ishga tushirami va quyidagi natijani olamiz



12.7-rasm. Matnlarni joylashtirish.

Matnlir asmlarni yaratish. Matnlir asmlarni yaratish uchun 1 takomponenta, 4 ta label, 4 ta button, 4 ta textbox larni qulay qilib joylashtiriladi.

1-qadam. 2 ta Textbox nirejimini multiline rejimigava scrolBars xususiyatiga Vertical qiymat beriladi.

2-qadam. Oynaning Form1_Load hodisasida quyidagi algoritmi joylashtiramiz.

```
this->label1->Text = "| Yuqoridagi  
matn";  
  
label2->Text = "|";  
label3->Text = "| Pastdagi matn";  
label4->Text = "|";  
button1->Text = "Tozalash";  
button2->Text = "Chizish";  
button3->Text = "Saqlash";  
button4->Text = "Rasm yuklash";
```

3-qadam.

Komponenta va button xususiyatlarini sozlash amallari bajarish mumkin.

4-qadam. TextBox3 va TextBox4 ni mos ravishda TextBox1 va TextBox2 lar orqasiga tashlaymiz.

5-qadam. Formaning formBorderStyle xususiyatiga SizableToolWindow beriladi.

Forma o'lganliklari o'zgartirish uchun, buni boshqacha amalga oshirish mumkin.

6-qadam. TextBox3 va TextBox4 ni shrift o'lganliklari 12, TextBox1 va TextBox2 lar boshqacha shrift o'lganliklari beriladi. Chunki dastur ishlatilganda asosiy ko'ringan matnlar ko'ringan matnlariga o'zgartiriladi.

7-qadam. Zaruriy o'zgaruvchilarni aniqlash.

```
private: Bitmap^ bmp_for_draw;  
private: String^ full_name_of_image;
```

8-qadam. Oynaning Form1_Load hodisasida quyidagi algoritmi joylashtiramiz.

```
Text = "Matnni rasmga joylashtirish";  
Font = gcnew System::Drawing::Font("Times New  
Roman", 32, FontStyle::Bold);  
this->pictureBox1->SizeMode =  
PictureBoxSizeMode::StretchImage;
```

9-qadam. Komponentaga rasmni yuklash uchun button4_Click hodisasiga quyidagi algoritmi yoziladi

```
OpenFileDialog^ open_dialog = gcnew OpenFileDialog();  
open_dialog->Filter = "Image
```

```

Files(*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG|All files (*.*)|*.*";
if (open_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)      {
    try{
        full_name_of_image = open_dialog->FileName;
        bmp_for_draw = gcnew Bitmap(open_dialog-
>FileName);
        pictureBox1->Image = bmp_for_draw;
        pictureBox1->Invalidate();
    }catch(Exception^ e) {
        System::Windows::Forms::DialogResult result =
MessageBox::Show("Fayl xato tanlandi"+e-
>ToString(),"Warning",MessageBoxButtons::OK,
MessageBoxIcon::Error);
    } }

```

10-qadam. Komponentadagirasnmnisaqlashuchun
button3_Clickhodisasigaquyidagialgoritmniyoziladi

```

if (pictureBox1->Image != nullptr){
    String^ format = full_name_of_image-
>Substring(full_name_of_image->Length - 4, 4);
    SaveFileDialog^ savedialog = gcnew SaveFileDialog();
    savedialog->OverwritePrompt = true; // Agar shu
nomli fayl bo'lsa
    savedialog->CheckPathExists = true; // Agar
noto'g'ri nom kiritilsa
    savedialog->ShowHelp = true;
    savedialog->Filter = "Image Files(*.BMP)|*.BMP|Image
Files(*.JPG)|*.JPG|Image Files(*.GIF)|*.GIF|Image
Files(*.PNG)|*.PNG|All files (*.*)|*.*";
    if (savedialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
        try {
            Bitmap^ MBB = gcnew Bitmap(pictureBox1->Image);
            MBB->Save(savedialog->FileName,
System::Drawing::Imaging::ImageFormat::Jpeg);
        } catch(Exception^ e){
            MessageBox::Show("Rasmni saqlashda xatolik",

```

```
"FATAL ERROR", MessageBoxButtons::OK,  
MessageBoxIcon::Error);  
} } }
```

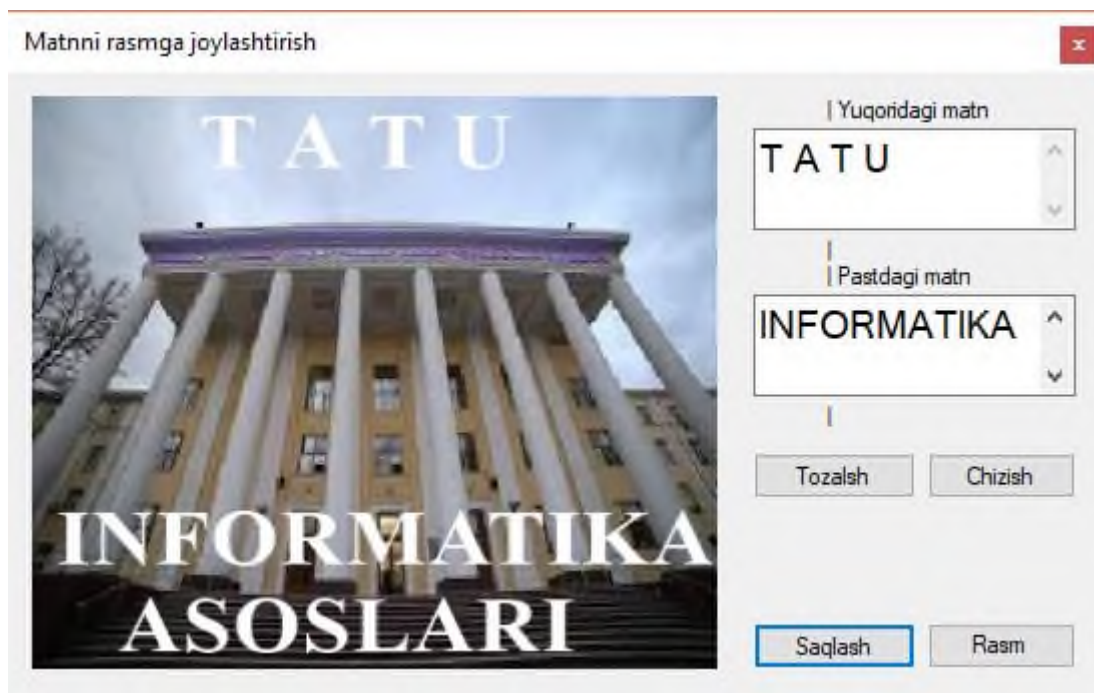
11-qadam. Komponentadagirasmnitozalashuchun
button1_Clickhodisasigaquyidagialgoritmniyoziladi

```
Bitmap^ BM = gcnew Bitmap(pictureBox1->Image);  
Graphics^ C =  
System::Drawing::Graphics::FromImage(BM);  
//Graphics C = CreateGraphics();  
C->Clear(pictureBox1->BackColor);  
pictureBox1->Image = BM;
```

12-qadam. Komponentagamatnlarnijoylashtirishuchun
button2_Clickhodisasigaquyidagialgoritmniyoziladi

```
System::Drawing::Font^ FontTemp = gcnew  
System::Drawing::Font("Times New Roman", 32,  
FontStyle::Bold);  
textBox4->Text = textBox1->Text;  
textBox3->Text = textBox2->Text;  
SaveFileDialog^ savedialog = gcnew SaveFileDialog();  
String^ Text = String::Format("{0}", textBox4->Text);  
String^ Txt = String::Format("{0}", textBox3->Text);  
Brush^ brsh = gcnew SolidBrush(Color::White);  
Bitmap^ MBB = gcnew Bitmap(pictureBox1->Image);  
Graphics^ G =  
System::Drawing::Graphics::FromImage(MBB);  
Graphics^ Q = G;  
G->TextRenderingHint =  
System::Drawing::Text::TextRenderingHint::AntiAlias;  
G->DrawString(Text, FontTemp, brsh, 75, 2);  
Q->TextRenderingHint =  
System::Drawing::Text::TextRenderingHint::AntiAlias;  
Q->DrawString(Txt, FontTemp, brsh, 5, 225);  
pictureBox1->Image = MBB;
```

Dasturniishgatushiramizvaquyidaginatijaniolamiz



12.8-rasm. Matnlarni rasmga joylashtirish.

Graphics

sinfi usulari asosidatasvirlarni qurishning turli usullari qarang chiqdik, bu amallarni yordamida rasmlarni tahrirlash, ishlov berish, matnlarni joylashtirish kabi ixtiyoriy amallarni bajarish mumkin.

Chart komponenta xususiyatlarini baholash. Bu komponenta Data tab bo'limiga joylashtirilgan bo'lib, asosan ma'lumotlarni infografiklarni yaratish uchun ishlatiladi. Infografika uchun ma'lumotlarni to'plam kerak.

Bu komponentaning xususiyatlarini baholash boshqa komponentalarni ikki bo'lim, xuddi o'sha kabi bo'limlarga bo'lingan. Ularning maxsuslarini keltirib o'tamiz.

1. **BorderSkin** xususiyatlariga gruppasi bo'lib, unda komponentaning yonidagi qobgasi olinishi mumkin. Niqob deganda, uning yonidagi ko'rinishi inobatga olingan. Rang (color), rasm (image), stil (style), kengligi (width) kabi xususiyatlar mavjud. Bu xususiyatlarni o'rnatish muammokeltirib chiqarmaydi. Oldingi o'rganganlaringizda bunday xususiyatlardan foydalangan siz. Shuningdek, asosiy niqobbu SkinStyle bo'lib, komponentaning asosiy ko'rinishini o'zgartirish uchun xizmat qiladi. Uning mos qiymatlar ro'yxati mavjud, shunda kerakli sinifni tanlab olinishi mumkin.

2. **Palette** (palitra) — xususiyat yordamida komponentaning infografika ko'rinishidagi shaklini aniqlash mumkin. Uning mos qiymatlar ro'yxati mavjud.

3. PaletteCustomColors

bundahamkomponentaninginfografikaniko‘rsatadiganshaklinitanlashmumkin. Ammofoydalanuvchio‘ziningrangitanlashimimumkin. Bundamaxsusmuloqotoynasiaosidapalitrageaturliranglarniqo‘shishmumkin.

4. Annotations – bundakomponentagaizohlarniyozishmumkin. Buningo‘ziningxususiyatlarimaxsusmuloqotoynasiyordamidao‘rnatiladi.

5. ChartAreas – buxususiyatorqalikomponentagabirnechata chart infografikajoylashtirishmumkin.

Buninghamo‘zimosxususiyatlarinimosmuloqotoynasibilano‘rnatishlozim.

6. Legends

infografikagakeltirilganqiymatlariningjoylashishmaydoni.

Bunihammaxsusmuloqotoynasiasosidatahrrirqilishmumkin

7. Series

– buxususiyatasosiybo‘lib, infografikaningqiymatlarinibelgilovchi, harbirqiymattegishliligini bildiradi.

Buninguchunmaxsusmulovotoynasimavjud.

Bungato‘liqroqto‘xtalibo‘tamiz.

8. Titles

komponentalargajoylashtirilganinfografikalargasarlavhaqo‘yishuchunishlatiladi.

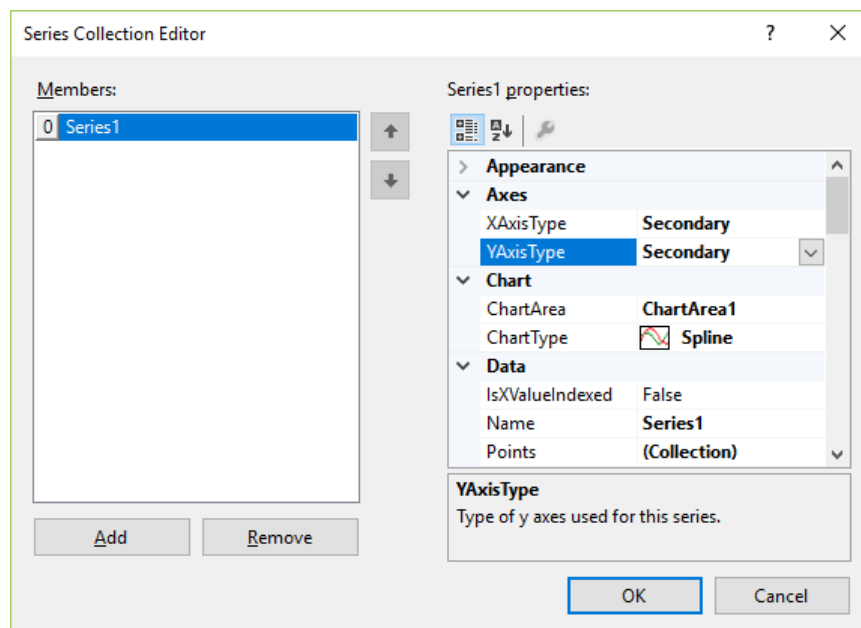
Uningmaxsusmuloqotoynasiorqaliishlovberishorqalishrnatishmumkin.

Series

bukomponentagaqiymatlarniqo‘shishvaunitasvirlashuchunxizmatqiladi.

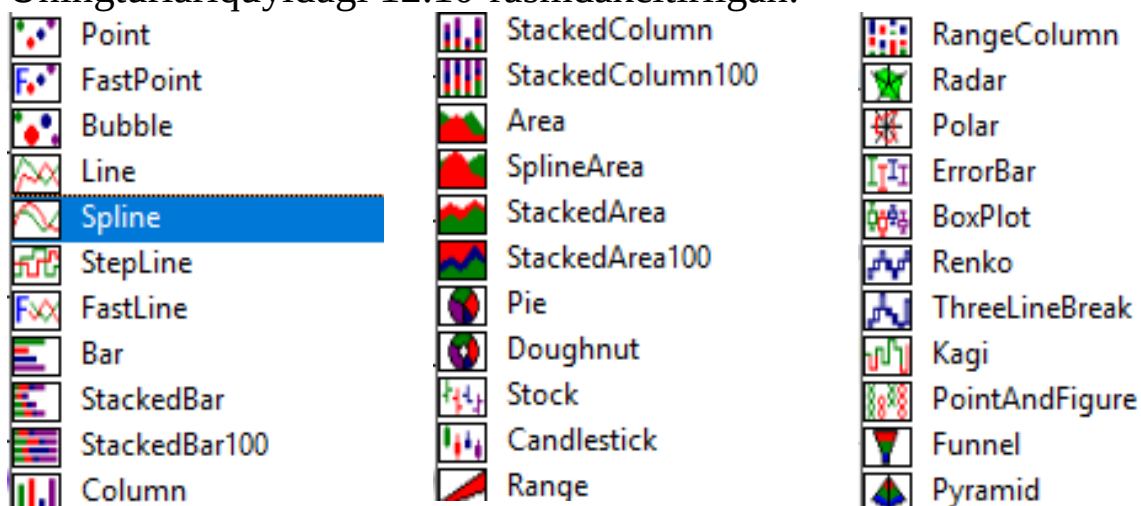
Komponentaningichigajoylashtirilgansohanitahrirlashuchunishlatiladi.

Ungabosgandaquyidagimuloqotoynasichiqadi.



12.9-rasm. Tahrirlashoynasi.

Buoyndayangiqiymatlarmaydoniqo‘shishuchun [Add] tugmasivaunio‘chirishuchun [Remove] tugmasiishlatiladi. Infografikaningasosiyxususiyatlariniboshqarishuchuno‘ngtomondagixususiyatlarpanelidanfoydalanishmumkin. Undainfografikanitaxhrilashuchunzarurbo‘lganbarchaxususiyatlarbor. Ulardanbiribu ChartType bo‘lib, infografikaningturlarinibelgilashuchunxizmatqiladi. Uningturlariquyidagi 12.10-rasmdakeltirilgan.



12.10-rasm. Infografikaningturlari

Shuningdek, infografikaningma‘lumotlari, yozuvlari, qiymatlari, maydoni, qiymatturlari, chegaralaribilanishlashxususiyatlarixammuvjud. Odatdabuxususiyatlarnidasturlashorqalidasturfragmentlaridao‘rnatishfaf oydalanishdasturchigaqulayhisoblanadi. Ammovizualdusturlashningimkoniyatidanfoydalanishuchunbunihamishlatishnio‘rganishlozim.

Funksiyalarnigrafiklariniqurish.
Infografikakomponentasigamosravishdafunksiyalarnigrafikginichizishusullariko‘ribchiqamiz.

Matematikfunksiyalarnigrafiklarinichizishuchunavvalshufunksiyalarnibirsinf gayaratibolamiz.

```
private value class MyFunction
{
    private:
        double _value;
    public:
        double getValuePow(double x) {
            return Math::Pow(x, 2);
        }
        double  getValueX4(double x){
            return x*x*x*x;
        }
        double  getValueKxa(double x, int k, int a){
            return k*x+a;
        }
};
```

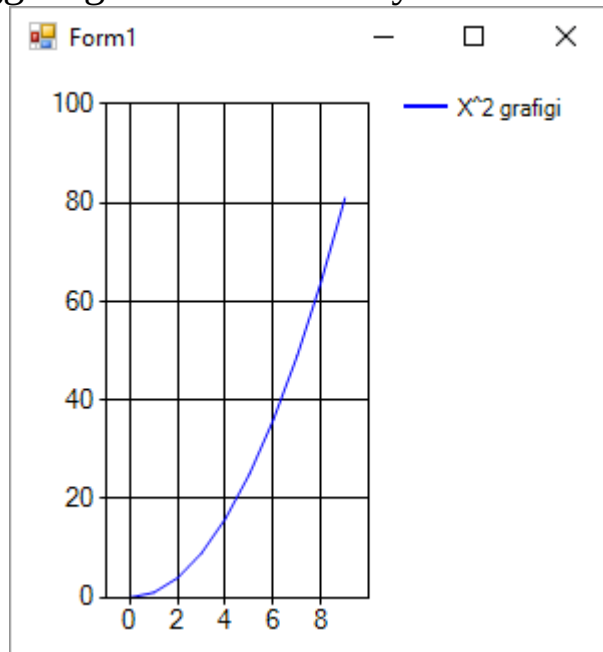
Buko‘rinishdamatematikaningbarchafunksiyalariniyaratibolishyo kito‘g‘ridanto‘g‘rifoydalanishmumkin.

Chart komponentasigagrafiknichizishuchun Form1_Load hodisasigaquyidagichaalgoritmniyozamiz.

```
chart1->Series->Clear();
Series^ series1 = gcnew Series(L"X^2 grafigi");
// rangni tanlash
series1->Color = Color::Blue;
series1->IsVisibleInLegend = true;
series1->IsXValueIndexed = true;
// infografikani turini tanlash
series1->ChartType = SeriesChartType::Line;
// qiymatlar qatlamini qo‘shish
chart1->Series->Add(series1);
// qiymatlarni
MyFunction^ func = gcnew MyFunction();
for (double i = 0; i < 10; i++) {
    series1->Points->AddXY(i, func-
```

```
>getValuePow(i));  
}
```

Agardasturniishlatsak,
bir x^2 funksiyaning grafiginichizishimkoniyatiniberadi.



12.11-rasm. x^2 funksiyaning grafigi.

Birvaqtningo‘zidabirnechta funksiyaning grafiklarinichizishuchuny
uqoridaaniqlangansinfdanfoydalanib, Form1_Load
hodisasigaquyidagichaalgoritmniozamiz.

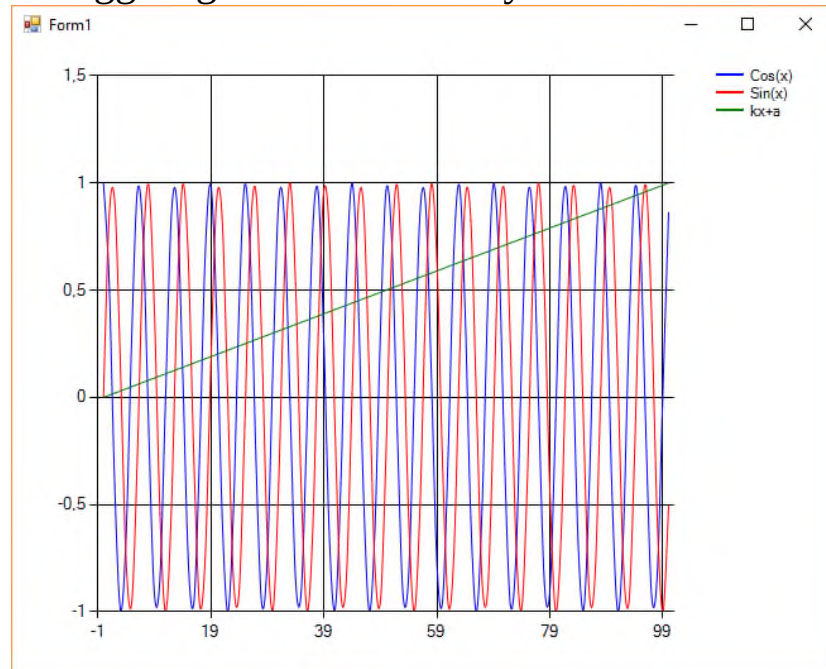
```
chart1->Series->Clear();  
Series^ series1 = gcnew Series(L"Cos(x)");  
Series^ series2 = gcnew Series(L"Sin(x)");  
Series^ series3 = gcnew Series(L"kx+a");  
// rangni tanlash  
series1->Color = Color::Blue;  
series2->Color = Color::Red;  
series3->Color = Color::Green;  
// infografikani turini tanlash  
series1->ChartType = SeriesChartType::Spline;  
series2->ChartType = SeriesChartType::Spline;  
series3->ChartType = SeriesChartType::Spline;  
// qiymatlar qatlamini qo‘shish  
chart1->Series->Add(series1);  
chart1->Series->Add(series2);  
chart1->Series->Add(series3);  
// qiymatlarni
```

```

MyFunction^ func = gcnew MyFunction();
for (double i = 0; i <= 100; i++) {
    series3->Points->AddXY(i, func-
>getValueKxa(i*0.01,1,0));
    series1->Points->AddXY(i, Math::Cos(i));
    series2->Points->AddXY(i, Math::Sin(i));
}

```

Agardasturniishlatsak,
birkatsiyalarninggrafiginichizishimkoniyatiniberadi.



12.12-rasm. Funksiyalarninggrafigi.

Gistogrammagrafiklarinichizishuchunmassivlardanyokiixtiyoriyto
‘plamlardanfoydalanishmumkin. Buninguchun 3 tamassivlamiz,
Form1_Load hodisasigaquyidagichaalgoritmniyozamiz

```

chart1->Series->Clear();
    Title^ title = gcnew Title("Sotilgan texnikalar
soni ");
    chart1->Titles->Add(title);
    Series^ series1 = gcnew Series(L"Telefon");
    Series^ series2 = gcnew Series(L"Kompyuter
texnikasi");
    Series^ series3 = gcnew Series(L"Avtomashina");
    // rangni tanlash
    series1->Color = Color::Blue;
    series2->Color = Color::Red;
    series3->Color = Color::Green;

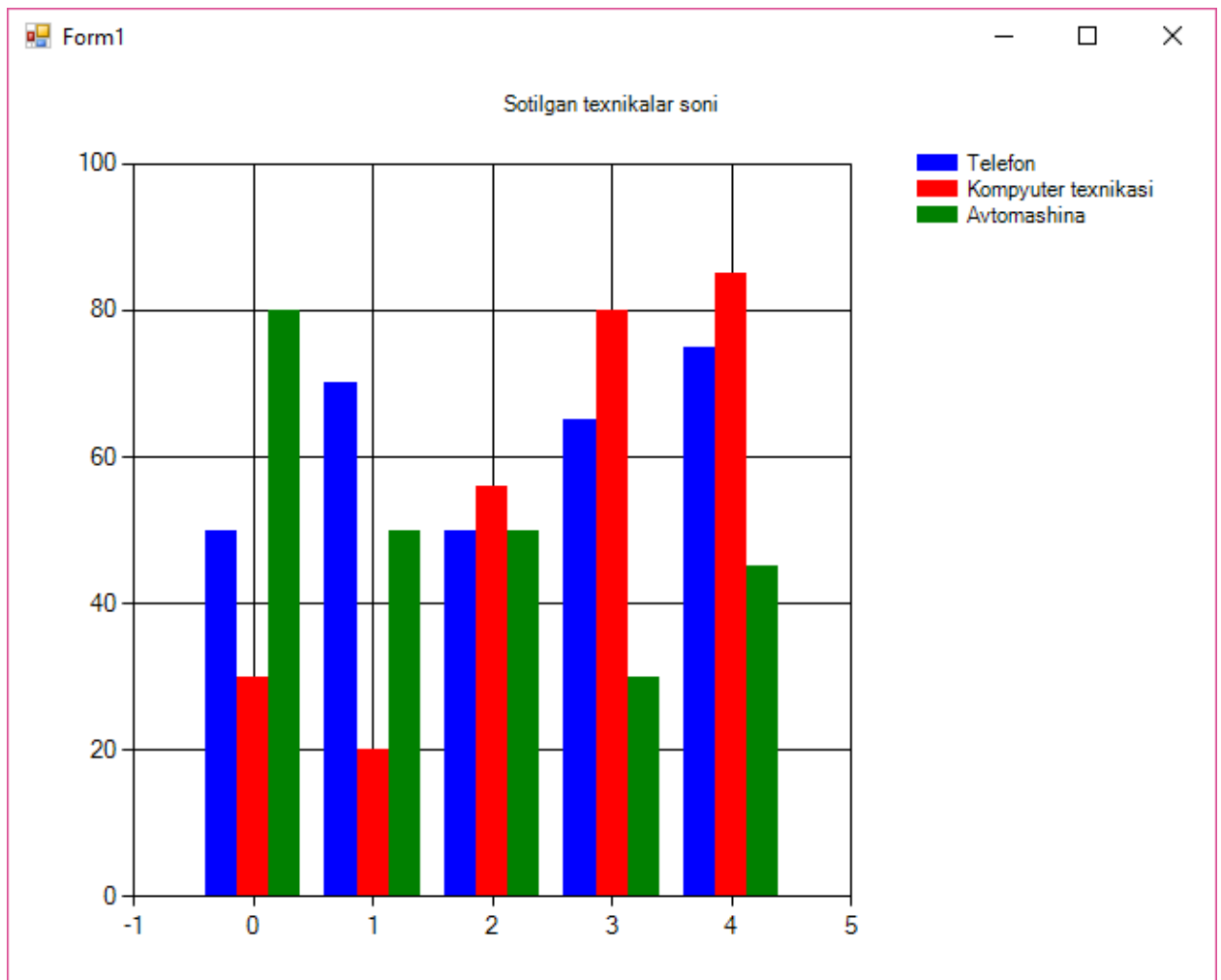
```

```

// infografikani turini tanlash
series1->ChartType = SeriesChartType::Column;
series2->ChartType = SeriesChartType::Column;
series3->ChartType = SeriesChartType::Column;
// qiymatlar qatlamini qo'shish
chart1->Series->Add(series1);
chart1->Series->Add(series2);
chart1->Series->Add(series3);
// qiymatlarni
array<int>^ arr1 = {30,20,56,80,85};
array<int>^ arr2 = {50,70,50,65,75};
array<int>^ arr3 = {80,50,50,30,45};
for (int i = 0; i < arr1->Length; i++) {
    series3->Points->AddXY(i, arr3[i]);
    series1->Points->AddXY(i, arr2->GetValue(i));
    series2->Points->AddXY(i, arr1->GetValue(i));
}

```

Agardastumiishlatsak,
Gistogrammagrafiginichizishimkoniyatiniberadi.



12.13-rasm. Gistogrammagrafiklarinichizish

Grafiklarinialohidasohashachizishuchunmassivlardanyokiixtiyori
yto'plamlardanfoydalanishmumkin. Buninguchun 3 tamassivolamiz,
Form1_Load hodisasigaquyidagichaalgoritmniyozamiz

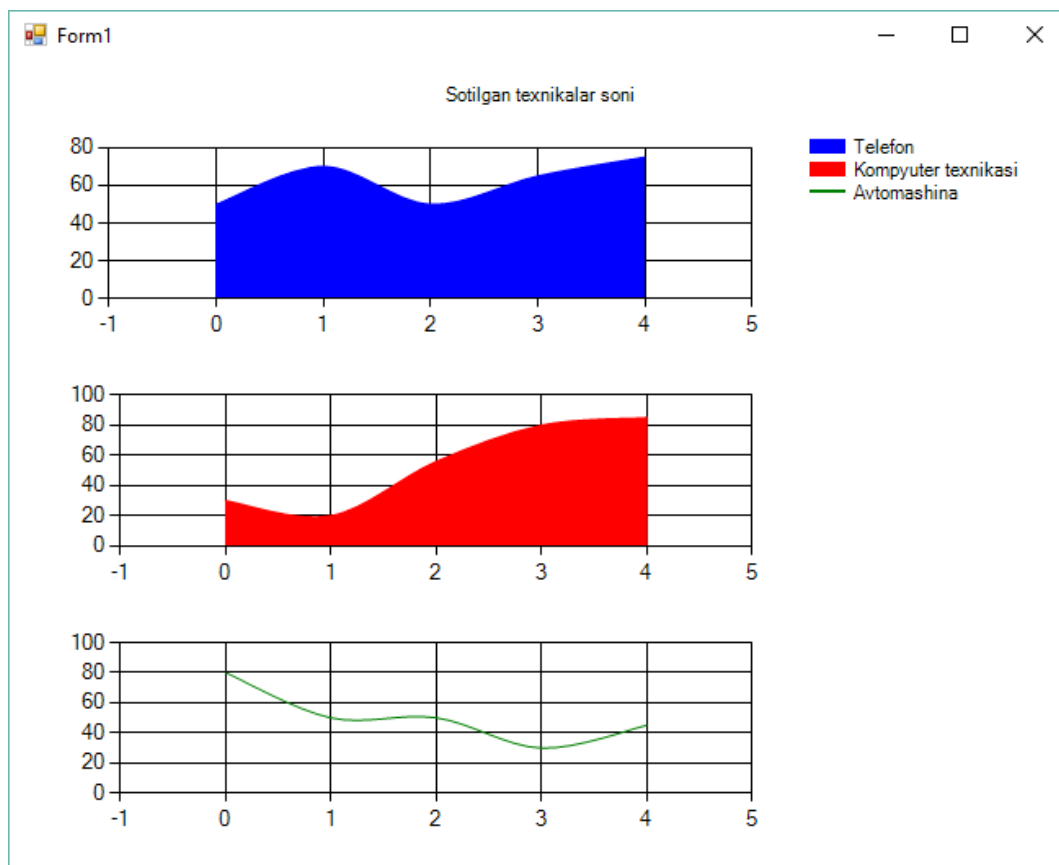
```
chart1->Series->Clear();
ChartArea^ chartArea1 = gcnew ChartArea();
ChartArea^ chartArea2 = gcnew ChartArea();
ChartArea^ chartArea3 = gcnew ChartArea();
chartArea1->Name = "1";
chartArea2->Name = "2";
chartArea3->Name = "3";
this->chart1->ChartAreas->Add(chartArea1);
this->chart1->ChartAreas->Add(chartArea2);
this->chart1->ChartAreas->Add(chartArea3);
Title^ title = gcnew Title("Sotilgan texnikalar
soni ");
chart1->Titles->Add(title);
Series^ series1 = gcnew Series(L"Telefon");
```

```

Series^ series2 = gcnew Series(L"Kompyuter
texnikasi");
Series^ series3 = gcnew Series(L"Avtomashina");
// rangni tanlash
series1->Color = Color::Blue;
series2->Color = Color::Red;
series3->Color = Color::Green;
// infografikani turini tanlash
series1->ChartType = SeriesChartType::SplineArea;
series2->ChartType =
SeriesChartType::SplineRange;
series3->ChartType = SeriesChartType::Spline;
series1->ChartArea = chartArea1->Name;
series2->ChartArea = chartArea2->Name;
series3->ChartArea = chartArea3->Name;
// qiymatlar qatlamini qo'shish
chart1->Series->Add(series1);
chart1->Series->Add(series2);
chart1->Series->Add(series3);
// qiymatlarni
array<int>^ arr1 = {30,20,56,80,85};
array<int>^ arr2 = {50,70,50,65,75};
array<int>^ arr3 = {80,50,50,30,45};
for (int i = 0; i < arr1->Length; i++) {
    series3->Points->AddXY( i, arr3[i]);
    series1->Points->AddXY( i, arr2-
>GetValue(i));
    series2->Points->AddXY( i, arr1-
>GetValue(i));
}

```

Dasturfragmentini tahlil qilish orqali yangi infografika sohalarni qo'shi-
sh va ularni qiymatlar o'plam bilan bog'lashni ko'rish imkonligi mavjud.
Agar dasturni ishlatib, Gistogramma grafigini chizish imkoniyatini beradi.



12.14-rasm. Bitta Chart gabirnechtasohani joylashtirish.

Chart

komponentasi yordamida foydalanuvchining ixtiyoriy ma'lumotlarini infografikasini yaratish mumkin.

Bujudak ta'imkoniyatlik komponenta hisoblanadi.

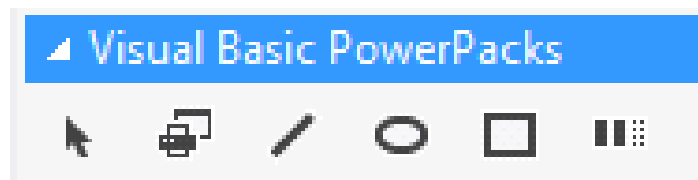
Buni xammasini o'rganish uchun ko'proq amaliyot qilish lozim.

Yuqorida a'niqlangan ba'zi ixtiyorliklarini keltirgan holda komponentaning xususiyatlarini va hodisalarini bo'yicha nazariy va amaliy ma'lumotlarni keltirdik.

Visual Basic Power Packs
komponentalarini va ularni ishlatish. Bu komponentalar guruhi alohida bo'lib, ularni Microsoft Visual Basic Power Packs deb atashadi. Buni yangi variantlarini ham internetdan olish va o'rnatish mumkin.

Asosiy vazifa shiforini belgilash uchun ishlatiladi, shuning uchun barcha komponentalar gabor bo'lgan hodisalar mavjud.

Bu Microsoft Visual Basic Power Packs nomlar fazosida joylashgan bo'lib, Visual Basic Power Packs elementlar uchun sinflari mavjud. Visual Basic Power to'plam elementlari qo'shimcha Windows formalar elementlari uchun mo'ljallangan. Ularda standart bepul plugin-smaylik sifatida ishlatilgan va endi Visual Studio tarkibiga kiritilgan.



12.15-rasm. Visual Basic Power Packs komponentalari.

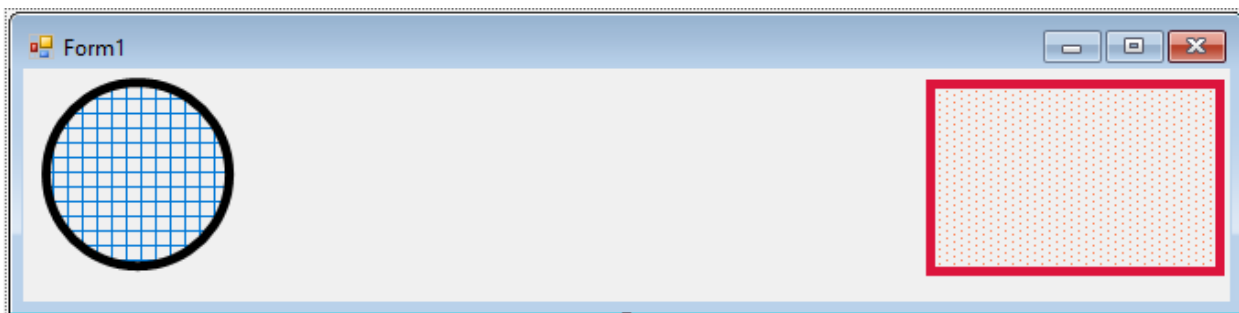
12.3-jadval. Visual Basic Power Packs komponentalari

Rasmi	Komponentanomi	vazifasi
	PrintForm	Formanichopqilishgaruxsatberishniboshqaradi
	LineShape	Gorozantal, vertikal, diognalchiziqlarniboshqarishnita'minlaydi
	OvalShape	Ovalko'rinishdagiob'ektlarniboshqarishnita'minlaydi
	RectangleShape	To'rtburchakko'rinishdagiob'ektlarniboshqarishnita'minlaydi
	DataRepeater	Talabasosidaformatlanganma'lumotlarniko'rsatadi

OvalShape komponentasiningba'zixususiyatlarivahodisalari

1. BackColor – komponentaningorqaranginio'rnatadivaqiymatlariranglarto'plami.
2. BackStyle - komponentaningorqastilinio'rnatadivaqiymatlari 2 tastatikberilgan.
3. BorderColor - komponentaningchegararanginio'rnatadivaqiymatlariranglarto'plami.
4. BorderStyle - komponentaningchegarastilinio'rnatadivaqiymatlari6tastatikberilgan.
5. BorderWight - komponentaningchegararangqalinliginio'rnatadivaqiymatlarisonlar.
6. FillColor - komponentaningaktivbo'lgandagirangnio'rnatadivaqiymatlariranglar.
7. Location – komponentaningjoylashuvo'rninio'rnatish..
8. SelectionColor - komponentaningtanlanganbo'lgandagirangnio'rnatadivaqiymatlariranglar.

Birloyihayaratingvaunda 12.16-rasmkabiformanitayyorlang.



12.16-rasm. Loyihako‘rinishi.

Yaratilgan ovalShape1 ni formabo‘yicha xarakatlanishni masalasini ko‘ramiz. Sizning formadagi 2 ob’ektni yaratishda ularning xususiyatlarini bilantirish ibol gansiz.

Buning uchun loyihaga bir Timer1 ob’ektidan joylashtiramiz. Tez bajarilishi uchun uning interval xususiyatiga 1 qiymatni kiritamiz. Timer1 ob’ekti ning ustiga sichqonchani ikki marta bosib, timer1_Tick hodisasiga quyidagicha algoritmni kiritamiz.

```
if(ovalpos <= (this->Size.Width - ovalShape1-
>Size.Width)){
    ovalpos += 5;
    ovalShape1->Location = Point(ovalpos, ovalShape1-
>Location.Y);
}
```

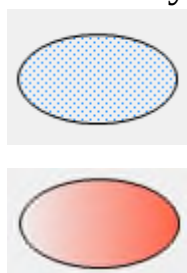
Bunda ovalpos qiymati 0 ga teng bo‘lgan butun son tipidagi o‘zgaruvchi. Location – bu ob’ektning joylashini belgilaydigan juft qiymatga ega Point qiymatni qabul qiladi.

Shuningdek, Form1_Load hodisasida quyidagi algoritmni yozamiz.

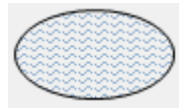
```
ovalpos = 0;
timer1->Start();
```

Agarda sturni ishga tushirsangiz ovalShape1 ob’ekti ning harakatini ko‘rasiz.

Shakllarning FillColor, FillGradientColor, FillGradientStyle, FillStyle xususiyatlaridan foydalanib, turli xil tugmalarni ham yaratish mumkin.



FillColor	MenuHighlight
FillGradientColor	Maroon
FillGradientStyle	ForwardDiagonal
FillStyle	Percent20
FillColor	Control
FillGradientColor	Tomato
FillGradientStyle	Horizontal
FillStyle	Solid



FillColor		ActiveCaption
FillGradientColor		White
FillGradientStyle		Horizontal
FillStyle		Wave
FillColor		ActiveCaption
FillGradientColor		OrangeRed
FillGradientStyle		Horizontal
FillStyle		SolidDiamond

Berilgan button tugmanibezash masalasini qaraymiz. Buning uchun forma gap bittatugmajoyla shtiramiz va uning xususiyatlarini quyidagicha aniqlaymiz.

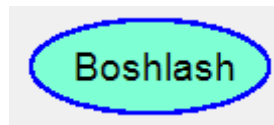
```
using namespace System::Drawing::Drawing2D;
using namespace System::Drawing::Text;
// ...
this->button1->Location = System::Drawing::Point(115,
438);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(121, 49);
this->button1->TabIndex = 0;
this->button1->Text = L"button1";
this->button1->UseVisualStyleBackColor = true;
```

Tugmaning button1_Paint hodisasi ga o'tib, quyidagi algoritimni yozamiz.

```
Pen^ pen = gcnew Pen(Color::Blue, 10);
Brush^ brush = gcnew SolidBrush(Color::FromKnownColor
(KnownColor::Control));
SolidBrush ^ brushinside = gcnew SolidBrush(Color::
Aquamarine);
Graphics ^ gr = e->Graphics;
gr->FillRectangle(brush, 0, 0, button1-
>Width, button1->Height);
gr->FillEllipse(brushinside, 0, 0, button1->Width,
button1->Height);
gr->DrawEllipse(pen, 0, 0, button1->Width, button1-
>Height);
GraphicsPath^ path = gcnew GraphicsPath();
button1->Region = gcnew
System::Drawing::Region(path);
String^ text = "Boshlash";
System::Drawing::Font^ drawfont = gcnew
```

```
System::Drawing::Font("Arial", 14);
SolidBrush^ exbrush = gcnew SolidBrush(Color::Black);
RectangleF rect = RectangleF(button1->Width/2-
40,button1->Height/2-10,button1->Width, button1-
>Height);
gr->DrawString(text, drawfont, exbrush,rect);
```

Bundachizishuchunqalam - Pen , mo'yqalam - Brush, to'liqbo'yashuchunmo'yqalam - SolidBrush yaratibolamiz. Keralishakllarnichizishob'ektininitugmaningo'lchamigamoslabtayyorlay miz. Tugmaningustigamatnjoylagtirishuchunyanabirob'ektyaratamiz, ungamatn, matnrangi, formativaegallashsohasiniko'rsatib, DrawString bilantugmaningustigachizamiz.



12.17-rasm. Tugmanibezashgami sol.

Yangiyaratilgantugmachalarorqalishakllarnioqimorqaliboshqarish masalasiniqaraymiz. BuninguchunLoyihaformasiga, 1 ta PictureBox, 2 ta button tugmaniyuqoridagidekqilibjoylashtiramiz. PictureBox ning BackColor xususiyatigaoqrangnio'ratimiz. Shaklningxarakatlantirishuchununingtayanchnuchtasiniko'chirib, oldingisioqbilanbo'yaladivakeyingisihosilqilinadi, shundayqilibiteratsiyadavometaveradi.

Ikkitaberilgantugmalarniyuqoridagidekqilib, moslabolamiz.

Oqimniboshqarishuchunavvalbirtugmaniyaratibolamiz.

Bufunksiyachizilganshaklni PictureBox bo'ylabharakatlanishvaoldinkeltirilganiteratsiyaniamalgaoshirishuchunk erak. Funksiyaningalgoritmiquyidagicha:

```
public: void Go(){
    int R = 30;
    // button2->BeginInvoke(gcnew
    InvokeDelegateSetEnabled(this,
    &Form1::setEnabledToButton), false);
    Bitmap ^ image = gcnew Bitmap(pictureBox1->Width,
    pictureBox1->Height);
    Graphics^ g = Graphics::FromImage(image);
    for (int x = 0; x <= pictureBox1->Width - 2 * R;
    x += 2){
        g->Clear(Color::White);
```

```

        g->FillEllipse(Brushes::Aqua, x, 0, 2
* R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int y = 0; y <= pictureBox1->Height - 2 * R;
y += 2){
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua,
pictureBox1->Width - 2 * R, y, 2 * R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int x = pictureBox1->Width - 2 * R; x >= 0;
x -= 2){
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua, x,
pictureBox1->Height - 2 * R, 2 * R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int y = pictureBox1->Height - 2 * R; y >= 0;
y -= 2)
    {
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua, 0, y, 2
* R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    // button2->BeginInvoke(gcnew

```

```
InvokeDelegateSetEnabled(this,
&Form1::setEnabledToButton), true);
}
```

Bu funksiyani ishlatish uchun qiymatni to'g'ri dan to'g'ri berib bo'lmay
dishing uchun birdelgant (funksiyako'rsatkichsaqllovchi)
vaqiymatni funksiyalaratamiz.

```
delegate void InvokeDelegateSetImage(Bitmap^ image);
public: void setImageToPictureBox(Bitmap^ image){
    pictureBox1->Image =
image;
}
```

Formaning ochiq joyiga oqim yaratib olamiz.

```
using namespace System::Threading;
// ...
private: Thread^ myThread;
private: Thread^ myThreadOne;
```

Birichitugmabosilgandamazkuroqimlarni ishlatish uchun button1_Click hodisasi quyidagi algoritmi yoziladi.

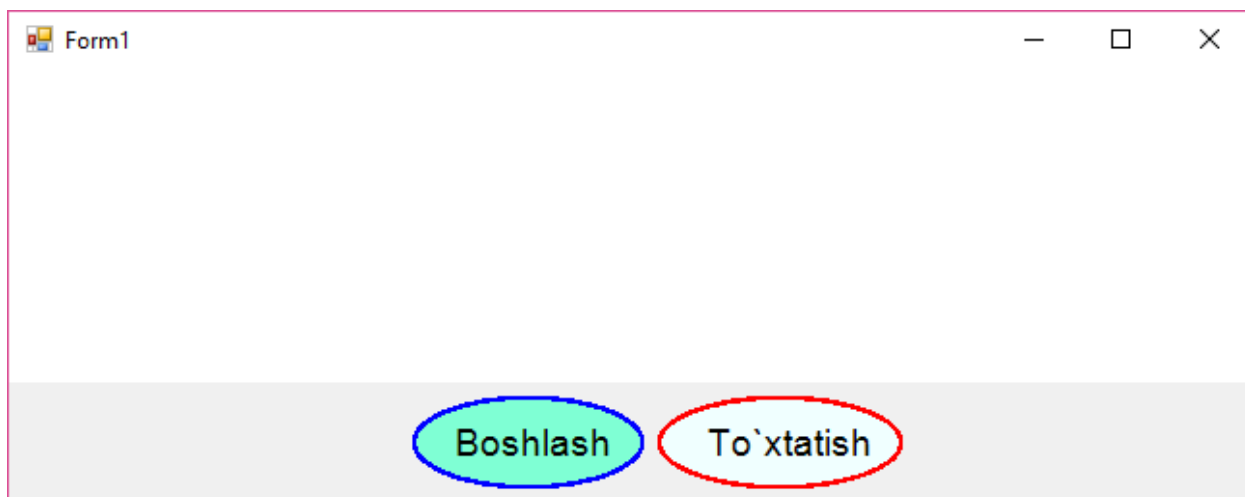
```
myThread = gcnew Thread(gcnew
ThreadStart(this, &Form1::Go));
myThread->Start();
myThreadOne = gcnew Thread(gcnew
ThreadStart(this, &Form1::GoOne));
myThreadOne->Start();
```

Oqimni saqlab tuurish uchun yaratilgan funksiyada
Thread::Sleep(10); dan foydalanilgan.

Oqim ishlatishgan paytdan ishlaydi,
ammo formayopilganda uning ishini to'xtatish lozim. Shuning uchun oqimni
to'xtatish uchun button2_Click va Form1_FormClosing
hodisalariga quyidagi algoritmi kiritamiz.

```
try{
    myThread->Abort();
    myThreadOne->Abort();
}
catch(...){}
```

Loyihani ishlatish hirsangiz, quyidagi 12.18- rasm hosil bo'ladi.



12.18- rasm. Oqimlarasosidaxarakatniboshqarish

Dasturlashboshlashtugmasinibosgandaikkita paralel bo'lgan shakllari ning harakatini ko'rasiz. To'xtash bosilganda ularni to'xtatishni ko'rasiz. Xarakat davom etatgan vaqtda ham formani yopiqishni tugallash mumkin.

Umuman olganda grafikabilanishlash uchun geometriyani yaxshi bilasht alabqilinadi. Visual C++ da grafikani, uning usullari va turlari shakllarni chizish usullari, Chart va Shape sinflarining xususiyatlarini va usullarini ko'rib chiqdik. Bularni muvazimasma'lumotlarda ishlatishni tavsiya qilamiz.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.



1. Chiziqlar va shakllar chizish uchun maxsus qanday usullarni o'rganish lozim
2. DrawLine, DrawArc, DrawClosedCurve, DrawPolygon va DrawRectangle o'z ichiga oladigan sinflarni ayting.
3. Grafikani chizish xususiyatlarini sanab bering.
4. Grafikani chizishda koordinatalarini uchun ishlatiladigan o'lchov birligini ayting.
5. GraphicsUnit qiymatlarini sanab bering.

6. -

butunchizmayuzasinitozalaydivabelgilanganfonrangibilanto‘ldiradiganus
ulningnominiayting.

7. Bufunksiyaberilganparametrlarasosidatasvirnibelgilanganjoydaasl
kattaligidanfoydalanibchizadi. Funksiyanominiayting.

8. Chiziqchizishniqandayamalgaoshiriladi.

9. Uchburchakchizishniqandayamalgaoshiriladi

10. Ellipsischizishniqandayamalgaoshiriladi

11. Turlishakllarnibo‘yyashniqandayamalgaoshiriladi

12. Ko‘pburchaklarnichizishniqandayamalgaoshiriladi

13. Hodisalarorqalishaklchizishniqandayamalgaoshiriladi

14. Rasmlarnio‘zgartirishniqandayamalgaoshiriladi

15. Matnlarnitasvirkabijoylashtirishniqandayamalgaoshiriladi

16. Matnlirasmlarniyaratishniqandayamalgaoshiriladi

17. Chart komponentaxususiyativahodisalariahaqidagapiribbering.

18. BorderSkin nima.

19. Qaysixususiyatyordamidakkomponentaninginfografikaniko‘rsatad
iganshaklinitanlashmumkin

20. Qaysixususiyatasosiybo‘lib,
infografikaningqiymatlarinibelgilovchi,
harbirqiymattegishliliginibildiradi

21. ChartType infografikaningturlarinibelgilashuchunxizmatqiladi.
Uningqiymatlarinibilasizmi.

22. Birinfografikadanechtagachagistogrammajoylashtirishmumkin.

23. Chart

komponentasiyordamidafoydlanuvchiningixtiyoriyma‘lumotlariniinfogr
afikasiniyaratishmumkinmi

24. Nimalardastlabbepulplugin-smayliksifatidaishlatilganvaendi
Visual Studio tarkibigakiritilgan

25. Visual Basic Power Packs komponentalarinisanabbering.

26. Visual Basic Power Packs
komponentalariningasosiyvazifalarinimadaniborat.

27. Komponentaningchegararangqalinliginio‘rnatadivaqiymatlarison
lardaniboratbo‘lganxususiyatnominiayting.

28. Shakllarning FillColor, FillGradientColor, FillGradientStyle,
FillStyle xususiyatlaridanfoydalanib, niamqilishmumkin.

29. Tugmningko‘rinishinio‘zgartirishuchunnimaamallarbajario‘kera
k.

30. Rasmlarniharakatlantirishdaoqimlardanqandayfoydalanamiz.



ASSISMENTTOPShIRIG'I	
	Grafikabilanishlashgaoidberilganquyidagidasturb o'yichaberilgantopshiriqlarkeraklifragmentlarias osidabajaring. 👉 Buvazifalarnikichikloyihasisifatidabajariladi, kichikloyihadegandakamida 100 tavazifasibo'lgandasturtushuniladi.
1. Oddiyrasmmuharriridasturinituzing. Loyihabo'yichaqisqachaasosiyfragmentlarniyozing.	
2. Interaktivmuloqotoynalaridanfoydlanib, diagrammalarniyaratuvchidasturyarating. Loyihabo'yichaqisqachaasosiyfragmentlarniyozing.	

3. Intellektual turlimaʼlamotlarni kiritishni talab qiluvchi funksiyalarni afigini yaratish dasturini tuzing. Loyihaboʻyicha qisqacha asosiy fragmentlarni yozing.
4. Tugmalarni bezash orqali kichik kalkulator dasturini tuzing. Loyihaboʻyicha qisqacha asosiy fragmentlarni yozing.

5. Barchaturdagimuloqotoynalarinio‘zichigaolgan, interaktivsavoljavobtiziminiyaratingvajavoblargayasamaharakatl anishvositalaridanfoydalanishdasturtuzing. Loyihabo‘yichaqisqachaasosiyfragmentlarniyozing. _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
6. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring, nimalarnio‘rgandingizvabirtakrorlang. 7. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang.

3.5. OLE,MFC texnologiyalari. Ko‘poynalimuhitlar bilan ishlash.

 Visual C++ningoperatsiontizim, amaliydasturlaribilano‘zarobog‘lanishuchun MFC va OLE texnologiyalariningimkoniyativavariantlari, ularningo‘zarohamkorligi, ko‘poynalidasturlarniyaratishbo‘yichanazariybilimlarkeltirilgan.. Bilimlarnimustahkamlashuchun 15 tanazariysavolberilgan.

 **Kalitso‘zlar.** Operatsiontizim, amaliydastur, integratsiya, o‘zarobog‘lanish, MFC, OLE, SDI, MDI, ko‘poynalidastur..

 **Bilishshartbo‘lgantushunchalar.** Operatsiontizimvaamaliydasturl ardaishlash, sinfvasinfob‘ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llab quvvatlovchimuhitdaishlashnibilishlozim.

✎**Bilibolasiz.** Visual C++ muhitidasturlash, Visual C++ning MFC texnologiyasiimkoniyatlarivavariantlari, OLE texnologiyasiimkoniyatlarivavariantlari, ularningo‘zarobog‘lashuchunloyihayaratish, ko‘poynalidasturlarniyaratishuchun SDI, MDI konstruktorlariusularivaba‘zibirxususiyatlari, hodisalarior‘rganishingiz mumkin.

Reja

1.MFC texnologiyasi.

2.OLE texnologiyasi.

3.Visual C++ muhitidako‘poynalimuhitniyaratish.

MFC texnologiyaci. Microsoft Foundation Class paketi (MFC) Microsoft tomonidanishlabchiqilganbo‘lib, C++ kutubxonalariyordamida Microsoft Windows uchun GUI dasturlarishlabchiqishniengillashtirishuchunmo‘ljallangan. Undakutubxonalarningvasinflariningsonijudako‘p.

MFC texnologiyasi

MFC kutubxonasi, uningasosiyranoqbatchisibo‘lgan Borland VCL kabi, GUI bilanishlashniosonlashtiradidasturasoslariniyaratish, ya‘nima‘lumbirtartibasosidaavtomatikravishdayaratiladi.

Foydalanuvchiinterfeysivabutunlayuningamaliyvatexnikxizmatko‘r satishuchunmuntazamharakatvahodisalarbajariladi(masalan, oynaamallari,

elementlarvadasturo‘zgaruvchilarichkiob’ektlaro‘rtasidama‘lumotlarniy uborish, vahokazo.). Dasturasoslariniishlabchiqqandanso‘ng, dasturchifaqatkodnimaxsusharakatvahodisalaratalabqilinadiganjoylargaki ritishikerak.

Bunday Framework yaxshibelgilangantuzilishgaegabo‘lishikerak, shuninguchun visual C++ uniyaratishvao‘zgartirishuchunyordamchikomponentalarniberadi.

Bundantashqari, MFC Windows API vazifalariturliob’ektgayo‘naltirilganqatlamlariniberadi, ularbilanishlashuchunbirozosondir.

Buqatlamko‘pajralmasob’ektlarniifodalaydi (Windows, vidjet, fayllar, instrumentlar, sinflar, komponentalarvaboshqalar.). Sinflarsifatidajoriyhodisalariniyopishvaxotiraaajratish/bo‘shatishkabimunt azamamallarnibajaradi.

Dasturdoirasigakodqo‘shish. MFC Framework uchundasturkodikiritishuchunikkiyo‘lbor.

Birinchisikutubxonadanmerosmexanizmidanfoydalanadi:

asosiy dastur doirasida tuzilmalar meros qilib olingan sinflar sifatidan amoyon bo'ladi. Businflar dasturidagi lumnuqtalarda chaqiriladigan ko'pgina virtual funksiyalar nitaqinlaydi. Bu funksiyalar ni yanada belgilab (ko'phollarda bazaviy sinflar vazifasini chaqirish kerak), dasturchi o'z kodidagi lumnuqtalarda bajarishni o'z shishi mumkin.

Ikkinchi usul o'ynash usulidir. Ikkinchisi usul o'ynash usulidir. Ikkinchisi usul o'ynash usulidir. Yordamchi instrument Windows — message maps (inglizcha: message map) bilan bog'liq sinflar qoliblarida [message ID — pointer to handler] juftlarini o'z ichiga olgan maxsus massivlarni yaratadi. Qo'shish yoki o'zgartirish yordamchi instrument asosida tegishli xabar orqali o'zgarishlar qilinadi.

MFC ning yaratilish tarixi. MFC ning birinchi versiyasi 1992-yilda Microsoft kompaniyasining 16-bitli C/C++ kompilyatorining gettinchi versiyasi bilan birga chiqarilgan. API funksiyalar yordamida ilovalarni ishlab chiqayotganlar uchun MFC paketini juda katta o'sishni vujudga keltirdi.

1. Dasturlash jarayonini boshqarish. MFC ninge tiborli xususiyatlaridan biri "AFX" prefiks bo'lib, ko'plab funksiyalar, makroslar nomlarida va "STDAFX" standart sarlavha sifatida ilovalar bilan ishlatiladi.

Rivojlanishning dastlabki bosqichida, keyinchalik MFC nomi "dastur" deb nomlangan. AFX doirasida rangaytmalari Microsoft Foundation sinflar nomini o'zgartirish uchun qaror (MFC) kodi AFX murojaatlarni o'zgartirish uchun juda kechqilingan.

2. C++ kompilyator bilan ishlash uchun Borland tomonidan ishlab chiqilgan Object Windows Library (OWL) shu davrda joriy etilgan raqobatchi mahsulot edi. Oxir oqibat Borland rivojlanishdagi o'xtadiva MFC bilan ishlash uchun kutubxona, foydalanish uchun qisqam uddatli litsenziya sotib oldi. Lekin uning mahsulotlari MFC uchun o'z liqqa o'z lib-quvvatlash amalga oshirilmadi.

3. MFC Microsoft markazida Microsoft .Net Framework tanlangan. Biroq, shunga qaramay, MFC hali ishlab chiqilgan uchun rorasi damashhur.

Birinchi lardan bo'lib, Rossiyaning bcssoft kompaniyasida sotib olingan. MFC yangi dizaynli interfeysi o'z lib-quvvatlash va takomillashtirilgan interfeys bilan bog'liq bir nechta boshqa o'zgartirishlarni kiritgan va nazorat qilmoqda. Visual Studio 2008 tizimidan ba'zibir komponentalarni olib, businflar asosida MFC ajralmas qismi qilib yaratildi.

13.1-jadval. MFCmahsulotlarivavarintlari.

Mahsulot nomi	MFC varianti	Joriyqilingan sana
Microsoft C/C++ 7.0	MFC 1.0	1992
Visual C++ 1.0	MFC 2.0 (arxitekturasiyangilangan)	
Visual C++ 1.5	MFC 2.5 (ODBC i dragand-dropniqo‘llabquvvatlovchitexnologiyajoriy qilingan)	
Visual C++ 1.52c	MFC 2.5 (MS Windows3.xuchunyaratilganoxirgiversiya)	
Visual C++ 2.0	MFC 3.0 (ko‘pmasalalivaUnicodeniqo‘llabquvvatlashjoriyqilindi)	
Visual C++ 2.1	MFC 3.1	
Visual C++ 2.2	MFC 3.2	
Visual C++ 4.0	MFC 4.0 (mfc40.dll - Windows 95tarkibigakiritilgan)	1995y. avgust
Visual C++ 4.1	MFC 4.1	
Visual C++ 4.2	MFC 4.2 (mfc42.dll - Windows 98tarkibigakiritilgan)	1998y. mart
eMbedded Visual C++ 3.0	MFC 4.2 (mfc42.dll)	
Visual C++ 5.0	MFC 4.21 (mfc42.dll), MFC 4.2.ningyangilanganligi	
Visual C++ 6.0	MFC 6.0 (mfc42.dll)	1998
eMbedded Visual C++ 4.0	MFC 6.0 (mfcce400.dll)	
Visual C++ .NET 2002 (Visual	MFC 7.0 (mfc70.dll), .NET 1.0	2002 y. fevral

C++ 7.0)		
Visual C++ .NET 2003 (Visual C++ 7.1)	MFC 7.1 (mfc71.dll), .NET 1.1	2003 y. aprel
Visual C++ 2005[2] (Visual C++ 8.0)	MFC 8.0 (mfc80.dll), .NET 2.0	2005 y. Mart
Visual C++ 2008[2] (Visual C++ 9.0)	MFC 9.0.21022 (mfc90.dll), .NET 3.5	2007 noyabr
Visual C++ 2008[2] with Feature Pack	MFC 9.0.30411 (mfc90.dll)	2008 y. Aprel
Visual C++ 2008[2] SP1	MFC 9.0.30729 (mfc90.dll)	2008y. avgust
Visual C++ 2008[2] Security Update (KB971092)	MFC 9.0.30729.4148 (mfc90.dll)	2009y.Iyul
Visual C++ 2010	MFC 10.0.30319.1 (mfc100.dll), .NET 4.0	2010 y. aprel
Visual C++ 2010 SP1	MFC 10.0.40219.1 (mfc100.dll), .NET 4.0	2011y.mart
Visual C++ 2010 + MS11-025	MFC 10.0.30319.415 (mfc100.dll), .NET 4.0	2011 y. aprel
Visual C++ 2012 (Visual	MFC 11.0.50727.1 (mfc110.dll), .NET 4.5	2012 y. iyul

C++ 11.0)		
Visual C++ 2012 Update 1 (Visual C++ 11.0)	MFC 11.0.51106.1 (mfc110.dll), .NET 4.5	2012y. noyabr
Visual C++ 2012 Update 3 (Visual C++ 11.0)	MFC 11.0.60610.1 (mfc110.dll), .NET 4.5	2012y. dekabr
Visual C++ 2013 (Visual C++ 12.0)	MFC 12.0.21005.1 (mfc120.dll), .NET 4.5.1	2013
Visual C++ 2013 Update 2 (Visual C++ 12.0)	MFC 12.0.30501.0 (mfc120.dll), .NET 4.5.1	2014
Visual C++ 2015 (Visual C++ 14.0)	MFC 14.0.23026.0 (mfc140.dll), .NET 4.6	2015
Visual C++ 2015 Update 1 (Visual C++ 14.0)	MFC 14.0.23506.0 (mfc140.dll), .NET 4.6.1	2015 y. noyabr
Visual C++ 2015 Update 2 (Visual C++ 14.0)	MFC 14.0.23918.0 (mfc140.dll)	2016 y. mart
Visual C++ 2015	MFC 14.0.24210.0 (mfc140.dll)	2016y. iyun

Update 3 (Visual C++ 14.0)		
Visual C++ 2015 Update 3 + KB316575 6	MFC 14.0.24212.0 (mfc140.dll)	2016 y. avgust
Visual C++ 2017 (Visual C++ 15.0)	MFC 14.10.25008.0 (mfc140.dll), .NET 4.6.2	2017 y. mart
Hozirgivaqtdahamvariantlariyangilanibborilmoqda.		

OLE texnologiyasi. OLE (Object Linking and Embedding) - Microsoft tomonidan ob'ektlarni boshqarish hujjat va obektlar integratsiya qilish uchun ishlab chiqilgan texnologiya. 1996 yilda Microsoft kompaniyasi ActiveX texnologiyasini qaytanomlagan.

OLE - bir qismini birdastur dan boshqasiga o'tkazish va natijalarni qaytarib olish imkonini beradi. Masalan, shaxsiy kompyuterga o'rnatilgan amaliy tizimda o'z imkoniyatlarini qayta taishlash uchun matn muharriri yoki biron bir ortasvirni OLE texnologiyasi yordamida tasviri muharriri yoki biron bir shayni o'rnatish mumkin.

OLE dan foydalanishning asosiy afzalligi (fayl hajmini kamaytirishdan boshqacha) shundaki, u asosiy faylni, dastur kiritadigan funksiyalarning fayl kutubxonasini yaratish imkonini beradi. Bu fayl qayta ishlab, so'ngi manba hujjatga qaytariladigan barcha dasturiy ma'lumotlarni qayta o'rnatish mumkin.

OLE murakkab hujjatlarni qayta taishlash uchun ishlatiladi, drag-and-Drop interfeysi orqali turli bog'liq bo'lmagan tizimlar o'rtasida ma'lumotlarni uzatish uchun foydalanish mumkin va clipboard operatsiyalarini amalga oshirish uchun. HTML-sahifalarda (gipermatnli markup tilida) yoki matn tahriri dandan foydalanadigan boshqa fayllarda (masalan, XML va SGML) tasvirlar, tovush, video va animatsiya uzatiladigan web-sahifalarda (masalan, Web TV)

multimediya kontent bilan ishlashda amalga oshirishgʻoyasikeng qoʻllaniladi. Biroq OLE texnologiyasi "kattamijoz" arxitekturasidan, yaʼni ortiqcha hisoblash resurslariga ega boʻlgan tarmoq kompyuteridan foydalanadi.

Bu amalga oshirishda fayl turi yoki dastur mijoz mashina mavjud boʻlishi kerak. Masalan, OLE Microsoft Excel jadvallaridan foydalansa, Excel foydalanuvchi mashinasiga oʻrnatilishi kerak.

OLE	1.0	Microsoft	Windows
operatsion tizimining olingan versiyalarida ishlatiladigan			DDE
(dinamik maʼlumotlar mashinavi)		texnologiyasi asosida	1990-
yildan chiqarilgan.			DDE
texnologiyasi ikki taʼshlaydigan dastur oʻrtasida maʼlumotlarni uzatish soni va usullar bilan cheklangan boʻlsa-da,			OLE
ikkita hujjat orasidagi faol ulanishlar ustida ishlay oldi va hatto birturdagij hujjatni boshqarturdagij hujjatga joylashtirdi.			

OLE serverlar va mijozlar virtual funksiya jadval yordamida tizim kutubxonalar bilan oʻzaro bogʻlanadi.

Bu jadvallarda tizim kutubxonasi server yoki mijoz bilan oʻzaro aloqa qilish uchun foydalanish imkon boʻlgan funksiyalar gamoskoʻrsatgichlari mavjud. Kutubxona OLESRV.DLL (server) va OLECLI.DLL (mijoz haqida) dastlab operatsion tizim tomonidan taqdim etiladigan WM_DDE_Execute xabari yordamida bir-biri bilan oʻzaro moslashuv kerak.

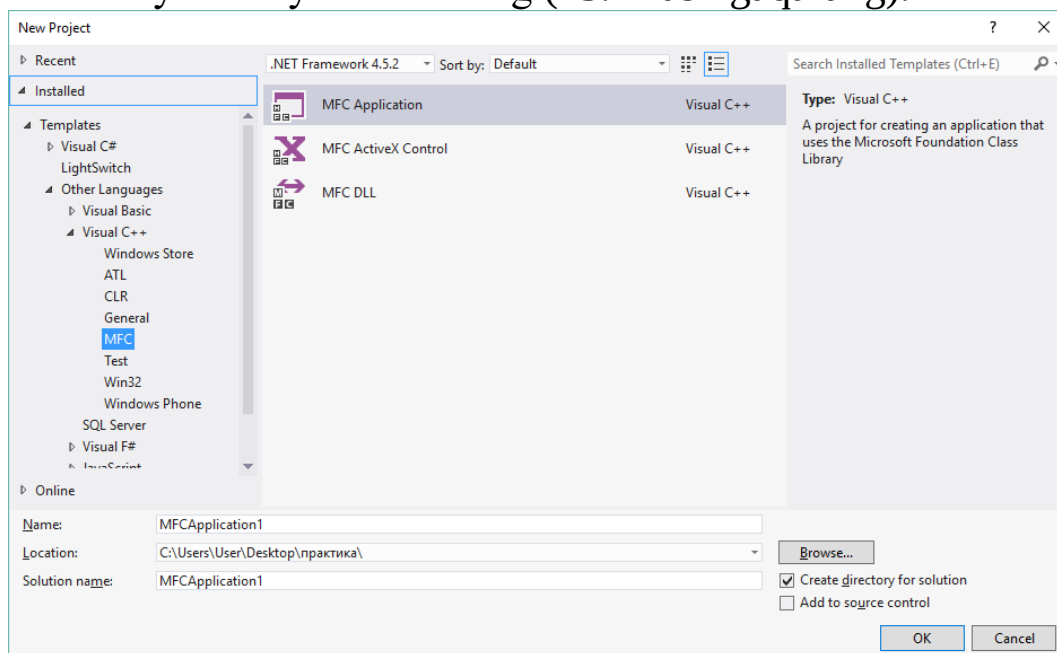
OLE 1.1 keyinchalik dasturiy komponentlar bilan ishlash uchun COM (butlovchi obʼekt modeli) arxitekturasiga aylandi. Keyinchalik, DCOM sifatida maʼlum boʻldi.

Bu erga OLE obʼekti joylashtirilganda Windows formatlarida (bitmap yoki metafile kabi) saqlanadi hamda oʻz formatida saqlanadi. U formatni qoʻllab-quvvatlash OLE dasturi clipboardga koʻchirilgan boshqa hujjatning birligini joylash va foydalanuvchi hujjatda uni saqlash imkonini beradi.

Keyingi evolyusion qadam OLE 2.0 edi, bu avvalgi versiya bilan bir xil maqsad va vazifalarni saqlab qolgan holda OLE 2.0 VTBL yordamida COM arxitekturasiga oʻshilgan boʻlgan. Yangi xususiyatlari Drag-and-Drop avtomatlashtirish, IN-joy faollashtirish va tuzilganlarni saqlash texnologiyalarini oʻz ichiga oladi.

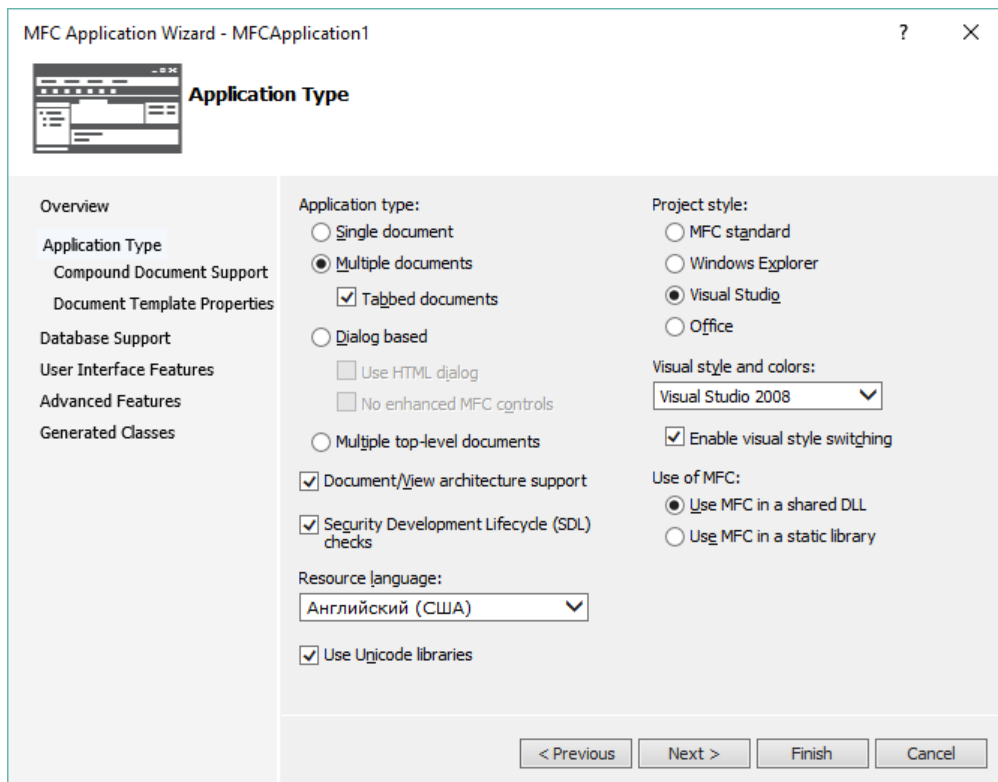
Visual C++ dasturlarida OLE konteyneridan foydalanish.

Visual C++ OLE moslamalarni foydalanish misolida 'riflaydidebatalmish OLE konteynerb yordamida ilovalar MFC platformasida amalga oshiriladi. OLE ob'ektlar eng tez-tez murakkab hujjat qismlari sifatida ishlatiladi. Hujjatlarga Microsoft Office ilovalari (Excel, Word, PowerPoint va boshqalar) yordamida yaratilgan fayllar kiradi. Biroq, Microsoft Office hujjatlarining 'ziboshqailovalarda OLE ob'ektlari sifatida ishlatilish mumkin. Misol sifatida, Visual Studio yangi MFC dastur loyihasini yaratib ko'ring (13.1-rasmga qarang).



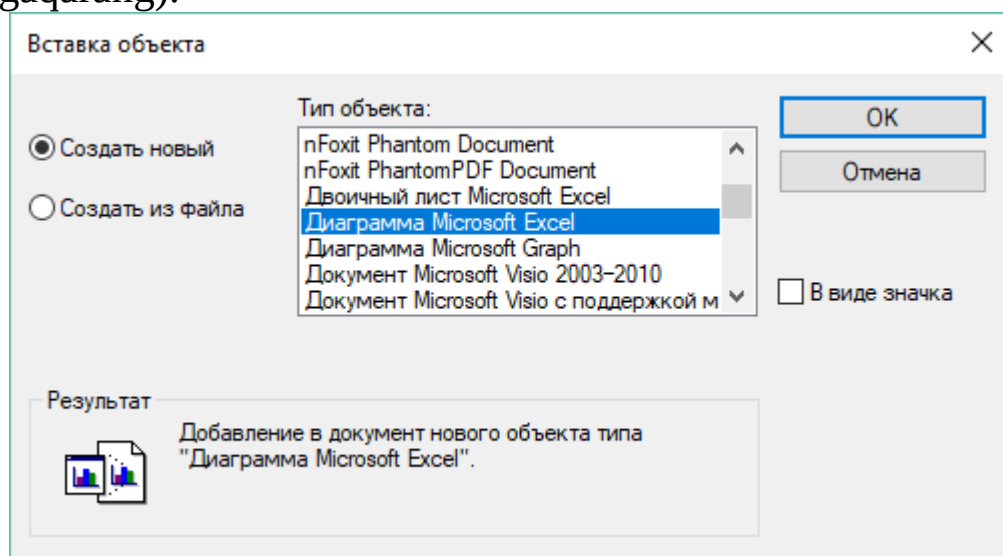
13.1-rasm. MFC application yaratish.

Keyingi qadammuloqot o'ynasida, kelajakda qo'llash asoslarini tanlash. Joriy dastur "hujjat-ko'rinish" arxitekturasidan foydalanib yaratilish kerak. Ilova interfeysi bitta hujjat yoki ko'phujjat bo'lishi mumkin. Shunda so'ng "keyingi" tugmasini bosib va kompozit hujjatlarni konteyner sifatida qo'llab-quvvatlash variantini tanlash lozim. Bundan tashqari, faol hujjat konteyner xususiyatini rnatish kerak. (13.2-rasmga qarang)



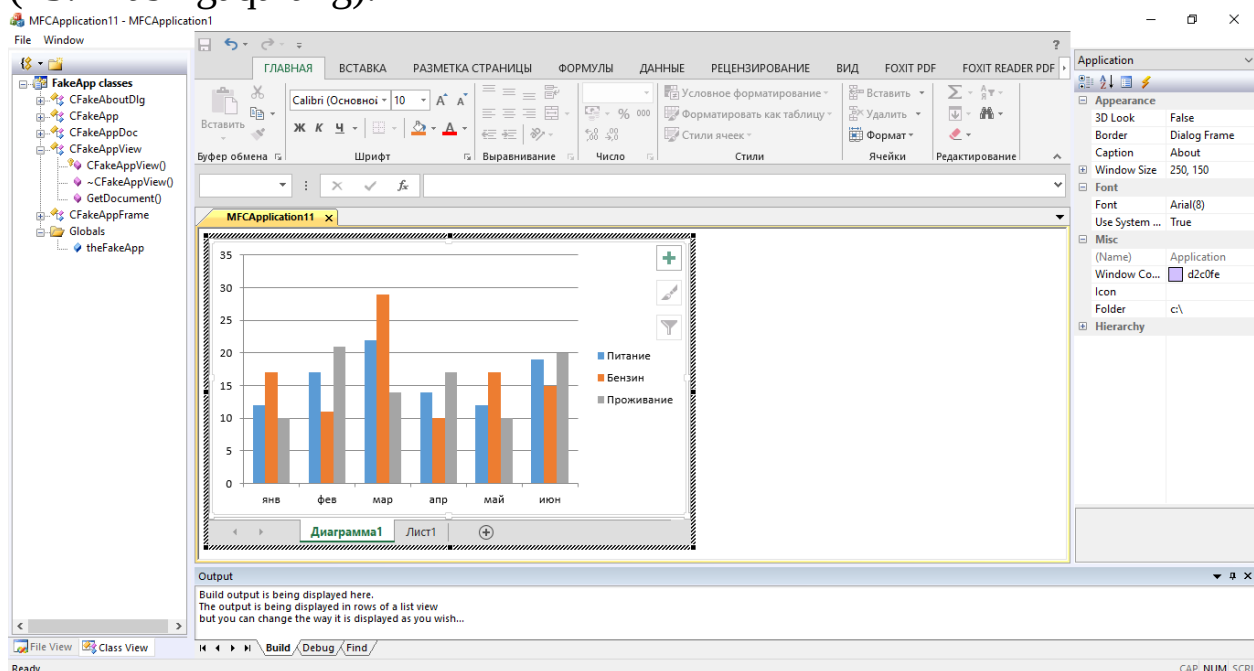
13.2-rasm. Parametrlarni sozlash.

Shundan so'ng tugmasini bosish kerak, yoki keyingi sahifalardan o'tib, zarur hollarda loyiha parametrlarini o'zgartirish mumkin. Natijada siz kompilyatsiya uchun ishlatish mumkin, yangi loyiha hisoblanadi. Dasturoyni asosida Boshmenyubandini tanlash orqali "Edit Insert New Object...". orqali OLE ob'ektlarining ro'yxati bilan muloqot oynasi hosil qilinadi (13.3-rasmga qarang).



13.3-rasm. OLE asosida bog'lanish.

Misolsifatida Microsoft Excel grafikob'ektinitanlasangiz, dasturoynasiob'ektkiritilgandanso'ngshundayko'rinishgaegabo'ladi (13.4-rasmgaqarang).



13.4-rasm. Microsoft Excel bilan bog'lanish.

Bu OLE ob'ektfal, unio'zgartirishmumkin: grafikuslubio'zgartirish, ma'lumotlargao'zgarishlarqilish, vakonteynerichidanuqtaformatinio'rnatishhammumkin.

OLE birkonteyner serialization xususiyatigaega, birfayluchunma'lumotlarnisaqlashvakeyinfayluniqaytaribo'qishmumkin. Ushbuob'ektnifaylgasaqlashuchun "File-Save as" nitanlab, dasturningasosiymenyusidavamuloqotoynasidagifaylnominiko'rsatiladi.

OLE konteyneridanfoydalanib, mavjud Microsoft Office hujjatfayllarini, masalan, Excel elektronjadvallariniyoki PowerPoint taqdimotlarini, shuningdek Word hujjatlarinihamko'rishingizvatahrirlashingizmumkin. Buninguchunyaratilgandasturniishgatushiringva "Edit / Insert New Object" nitanlang. Buorqalibarchaimkoniyatlarniamalgaoshirishmumkin.

Visual C++ muhitidako'poyalimuhitniyaratish. Visual C++ muhitidako'poyalidasturlarniyaratishuchun SDI va MDI ilovalarkonstruktorlarimavjud. Hozirda MDI ilovakonstruktoriorqalilikichikko'poyalidasturyaratamiz. Bundaquyidagiqadamlardanfoydalinadi.

1. Yangiformayaratiladi.
2. Formaningxususiyatlario'zgartiriladi.

IsMdiContainer: True
StartPosition: CenterScreen
Text: **Notice**

3. Yangi formaqo'shishuchun, asosiymenyusida, click Project -> Add Windows Form...

4. Yagonahujjatganomo'rnatish

5. Add tugmasinibosing

6. Asboblarningumumiyelementlaribo'limidanRichTextBoxnitlang

7. Xususiylaroyinasidaquyidagixususiylarlarnio'zgartiring:

(Name): **rtbNotice**
Dock: Fill
Modifiers: Public

8. Birinchishaklniko'rsating

9. Formadao'tingvayangiikkimartasichqonchanibosing

10. Faylningyuqoriqismidayagonahujjat header faylinitahrirlang:

```
#include "SingleDocument.h"
```

11. Kodningpastgatushingvaquyidagichahodisaniamalgaoshirishkodi niyozing.

```
System::Void  
mnuFileNew_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
    SingleDocument ^ document = gcnew  
    SingleDocument;  
    document->Text = "Untitled";  
  
    document->MdiParent = this;  
    document->Show();  
  
    mnuFileClose->Enabled = true;  
}
```

12. Formagaqaytiladi.

13. Asboblarpnelidanyangibirtugmanio'rnatish.

14. Xususiylaroyinasininghodisalarigao'tib, tugmanihodisalarinibosing. click maydoninibosingvao'ngtomonida, mnufilenew_click tanlang

15. Formadao'tingvayangiikkimartasichqonchanibosing

16. Quyidagichatadbiramalgaoshiring:

```

System::Void
mnuFileOpen_Click(System::Object^
sender, System::EventArgs^ e)
{
    if( dlgFileOpen->ShowDialog() ==
System::Windows::Forms::DialogResult::OK
)
    {
        for each( String ^ strFile in
dlgFileOpen->FileNames )
        {
            SingleDocument ^ document =
gcnew SingleDocument;

            document->rtbNotice-
>LoadFile(strFile);

            document->MdiParent = this;
            document->Show();
        }
    }
}

```

17. Formagaqaytiladi.

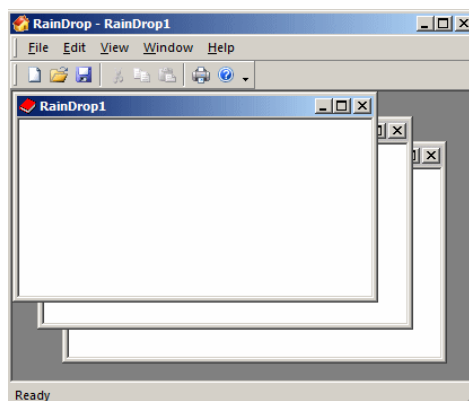
18. Asboblarpneldanyangibirtugmanio‘rnatimg.

19. Xususiyatlaroy nasininghodisalarigao‘tib,
tugmanihodisalarinibosing. click maydoninibosingvao‘ngtomonida,
mnufilenew_click tanlang

20. mnufileopen_click tanlang

21. Formagaqayting.

Natijadayaratilganloyihaniishgatushirsangizquyidagicha,
dasturhosilqilinadi.



NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. MFC nima, uqandayimkoniyat.
2. MFC va API ningbog'liqxususiyatlarinibilasizmi?
3. MFC ningengasosiyxususityainima.
4. Visual Studio 2008 tizimidanqandaykomponentalarniolib, qaysisinflarasosida MFC ajralmasqismi qilib yaratildi.
5. MFC 9.0.21022 (mfc90.dll), .NET 3.5 ningimkoniyatlarinisanabbering.
6. MFC 14.10.25008.0 (mfc140.dll), .NET 4.6.2 ningimkoniyatlarinisanabbering.
7. OLE texnologiyasiningimkoniyatinisanabbering.
8. OLE murakkabhujjatlarniqaytaishlashuchunishlatiladi, qaysiinterfeysiorqaliturlibog'liqbo'lmagantizimlaro'rtasidama'lumotlarniuzatishuchunfoydalanishmumkin.
9. Qaysitexnologiyavauningmetodi Microsoft Windows operatsiontizimini goldingiversiyalaridaishlatiladigan DDE (dinamikma'lumotlaralmashinuvi) texnologiyasiasosida 1990-yildachi qarilgan
10. OLE 1.0 va OLE 2.2 texnologiyalarinifarqinitushuntiribbering.
11. MFC va OLE texnologiyalaribog'liqmivanimauchun.
12. Ko'poynalimuhitlarningavfzalliginimada?
13. Formaxususiyatlarinio'zgartirishnimauchunkkerak.
14. Dock xususiyatininimauchun Fill qiymatgao'zgartiriladi.
15. mnufileopen_click hodisasiningvazifasinitushuntiribbering.

3.6. Kichik loyihalarni yaratish

Visual

C++ ning loyihayaratish usullariva uskunalarni panelidagi elementlardan foydalanishasosidakichik loyihalarni yaratish talablariva uskullari, loyihada foydalaniladigan algoritmlarni ifodalash va loyihalashasosidatahl il qilishni, hamdahi sobotlarni yaratish usullariva elementlari bo'yicha nazariy bilimlar keltirilgan. Bilimlarni mustahkamlash uchun 15 tanazariy savol berilgan.

✍Kalitso‘zlar. Operatsiontizm, amaliydastur, kichikloyiha, axborottizimi, IDEF, DFD, Hayotsikli, algoritm, EPSmetodologiyasi, BRMImetodologiyaisi, hisobot, hisobotpaneli.

✓Bilishshartbo‘lgantushunchalar. Visual

C++ningloyihayaratishusullarivauskunalarpanelidagielementlardanfoydalanish, ishlash, sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

✎Bilibolasiz. Kichikloyihalarniloyihalashtirishusullari, hayotdavri, loyihadagialgoritmlarningifodalashninguchullarivatalablari, Visual C++ muhitidadasturlash, Visual C++ninghisobotlarniyaratishkonstruktorlariusularivaba‘zibirxususiyatlari, hodisalarior‘rganishingizmumkin.

REJA

1. Visual C++ muhitidaturlisohagaoidmasalalarniyechishuchunkichikloyihalarniloyihalashusullari.
2. Kichikloyihalaralgorimlariniyozishusullari.
3. Visual C++ muhitidahisobotshaklinitayyorlashvachopetishelementlari.

KIRISH

Visual C++ muhitidaturlisohagaoidmasalalarniyechishuchunkichikloyihalarniloyihalashusullariasosanaxborottizimlarniloyihalashtirishusullarinazardatutiladi. Shuninguchunaxborottiziminiogob’ektlarivausullari, algoritmlariga, qo‘llanishsohasivachegaralarigaqarabloyihanikatta, kichikloyihalardeyishmumkin. Shuninguchunkichikloyihadegandafunksionalimkoniyatlarisanoqlibo‘lganaxborottizimlaritushuniladi.

Axborottizimlariningloyihalashtirishuchunaxborottiziminihayotdavriniyaxshibilishvatushunishkerak. Chunkiaxborottiziminiloyihalashdauchunhayotdavrimuhimahamiyatgaega.

Axborottiziminihayotdavri. Hayotdavri (HD) tushunchasiauxborottizimlariniloyihalashmetodologiyasiningasosiytushunchalaridanbiridir. Axborottiziminihayotiydari - buaxborottiziminiyaratishfikridanboshlanib, ubutunlayfoydalnishdanchiqqanpaytdatugaydiganzluksizjarayondir. HayotdavriningtuzilishinibelgilovchiasosiystandartGOST ISO/IEC

hayotdavrstrukturasi uchta jarayon guruhiga asoslangan:

1. **Asosiy** (buyurtma, ishlabchiqish, etkazibberish, sinov, texnik xizmat ko'rsatish);

2. **Yordamchi** (asosiy jarayonlarning bajarilishini ta'minlash uchun):

- Hujjatlar uchun

barcha manfaatdor shaxslar tomonidan zarur bo'lgan hujjatlarni ishlabchiqish, tahrirlash, tarqatish va saqlash bo'yicha ishlar;

- konfiguratsiya niboshqarish (configuration management)

quyidagilarni o'z ichiga oladi:

tizimdagidasturiy ob'ektlarning holatini aniqlash va o'rnatish;

ob'ektlarning o'zgarishini aniqlash va niboshqarish;

ob'ektlarning to'liqligi, mosligi va ta'limni ta'minlash;

ob'ektlarni saqlash, tashish va etkazibberishni boshqarish;

- sifatni ta'minlash

yaratilayotgan tizim va amalga oshirilayotgan hayotdaviy jarayonlarining belgilangan talablar va tasdiqlangan rejalarga javob berishini ta'minlash ishlar;

- tekshirish

loyiha amalga oshirilayotgan dasturlar va amalga oshirilgan muvaqqat natijalarning belgilangan talablarga javob berishini tekshirish uchun tegishli shaxs (buyurtmachilik, etkazibberuvchi yoki mustaqil tomon) ning ishi. Shartnoma, jarayon, talablar, loyihaning tizimi, qurilish tizimining hujjatlarni tekshirishni ajrata;

- sertifikatlash

alablarning va yakuniy mahsulotning tizimning funksional maqsadiga to'liq mosligini tekshirish uchun tegishli shaxsning ishi;

- qo'shma tahlil - har qanday (tizim)

holatini yoki natijalarini baholash uchun;

- audit - mustaqil (loyihaga nisbatan)

ekspertlarning korxona faoliyatining qabul qilingan talablar, rejalar va shartnomalar shartlariga muvofiqligini aniqlash ishlar;

- muammoni hal qilish

loyihani amalga oshirish vaqtida topilgan muammolarni tahlil qilish va tuzatish ishlar;

3. **Tashkiliy:**

- loyiha boshqaruvi - rejalashtirish va jarayonlarni boshqarish, shu jumladan monitoring,

tekshirish va hisobot bilan tugallangan ishlarni baholash;

- loyihani fratsuzilmasini yaratish

boshqar qanday jarayon uchun zarur bo'lgan infratsuzilmani tashkil etish va

a' minlash uchun ishlaydi. Infrastruktura texnika dasturiy vositalar, vositalar, usullar, standartlar va tizimni ishlab chiqish, ishlatish yoki xizmat ko'rsatish shartlari bo'lishi mumkin;

- yaxshilash - hayot davri jarayonlarini baholash, kuzatish va yaxshilash uchun ishlash;

- o'quv-kadrlar tayyorlashni rejalashtirish va o'tkazish, shu jumladan o'quv materiallarini ishlab chiqish. Shubhaning biriga,

xodimlarni faqat tizimni boshqaradigan oxirgi foydalanuvchilarga, balki tizimni ishlab chiquvchilari gaham tegishlidir. Masalan,

ishlab chiquvchilarning tashkilotdagi qabul qilingan texnologiyalar va dasturiy vositalar bo'yicha o'qitilishi,

hatto oxirgi foydalanuvchilarning tizim bilan ishlashni o'g'ri amalga oshirish va o'qitishga o'rgatish kerak. Qanchalik paradoksal bo'lsin,

lekin o'qitishning o'g'ri usul va metodlari o'rgatish ham kerak.

A Tyarlash uchun asosiy mezon quyidagilarni o'z ichiga oladi:

- tizimning funksional va tashkiliy tuzilmalari;
- texnika va dasturiy vositalar majmuasining tarkibi va tuzilishi;
- ishlatiladigan uskunalar;
- axborotni qayta ishlash texnologiyalari;
- axborot bazasini yuritish tarkibi, tuzilishi va texnologiyasi;
- kirish va chiqish formalari;
- ma'lumotlarni qayta ishlash algoritmlari.

Axborot tizimni ishlab chiqishdan oldin mijoz va ishlab chiquvchi tizimda qanday funksionallik o'rnatilishini va tizim ichida qanday funksional o'zaro munosabatlarni tashkil etilishini aniqlash kerak.

Funksional modelni ishlab chiqishda (funktional talablarni belgilashda) ko'pgina muammolar paydo bo'lishi mumkin:

- mijoz axborot tizimi oldiga qanday vazifalar qo'yilganini aniqlash fida etolmaydi. Ko'pincha mijoz ham talabni mavajunishakllantirish qanday bilmaydi;

- buyurtmachilarning (turlidarajadagi rahbarlar, texnologiya mutaxassislari, oddiy foydalanuvchilar) kelajak tizimi haqida o'z qarashlariga ega bo'lganlarining tizimga bo'lgan talablari o'zaro farq qiladi. Ushbu holat, ayniqsa, ishlab chiqilayotgan tizim bir nechta avtomatlashtirish ob'ektlarida amalga oshirilganda odatiy holdir;

-
mijozko‘pinchazamonaviyhisoblashtizimlariningimkoniyatlarinibilmayd
ivaavtomatlashtirishjarayoniniqo‘ldabajariladiganelementarfaoliyatniko
mpyuterlargaoddiyuzatishdebbhisoblashgaintiladi. Shubilanbirga,
ularyangitexnologiyalarkelishibilantashkilotichidagibiznes-
jarayonlarni optimallashtirishhaqidao‘ylamaydilar;

-
mijozba‘zifunksiyalarnivirtualmashinalarbajarishimumkinligigaishonma
ydi.

Funksionalmodelqurish, bumuammolarniengavvalhalqilishkerak.
Bujarayonbatafsilko‘ribchiqamiz.

Uniishlabchiqishdaishnimavjudtashkiletishningmodeliavvaloishtav
siflari, farmoyishlari, hisobotlari,
me‘yoriyhujjatlarivaboshqalarasosidaquriladi.

Modelnitahlilqilishzaiftomonlarqaerda,
yangijarayonlarningafzalliklariqandaybo‘lishinivakorxona (kompaniya,
bo‘lim)

ningmavjudtashkilotigaqandaychuquro‘zgarishlarkiritilishinitushunishga
imkonberishilozim.

Faoliyatnisamarasiztashkiletishbelgilaribo‘lishimumkin:

- foydasiz, nazoratdantashqarivaduplikativishlar;
- natijasizishlar;
- samarasizhujjatoqimi (to‘g‘rihujjato‘zvaqtidato‘g‘rijoydaemas)

vaboshqalar.

Modeldatopilgankamchiliklarmodeliniyaratishdatuzatilib,
korxonaningyangitashkilotimodelitaklifqilinishihammumkin.
Bumodelmuammonihalqilishningmuqobilyo‘llarinitahlilqilishvaengyaxs
hisinitanlashuchunishlatiladi.

SADT metodologiyasi (Structured Analysis and Design
Technique - strukturalitahlilvaloyihalashtexnikasi) -
tizimningfunksionalmodeliniqurishuchunmo‘ljallanganusullar,
qoidalarvaprotseduralarmajmui.

BumetodikaniishlabchiqishniXXasrning 60-yillario‘rtalarida
Duglas Ross (Aqsh) boshlabbergan. Keyin, Softech, Inc
tahlilchilartomonidan SADT
yaxshilandi vamumamolarkengko‘lamlihalqilishuchunishlatiladi. Telefon
tarmog‘idasturiyta‘minot, diagnostika,
uzoqmuddatli vastrategikrejalashtirish,
kompyuteryordamishlabchiqarishvaloyihalash,

kompyutertizimikonfiguratsiya, kadrlartayyorlash,
vamoliyaviyvalogistikaboshqarishSADTsamaraliilovalaryaratilaboshlan
di.

KengsohalarSADTmetodikasiningko'pqirraliligivaqudratiniko'rsatadi.
Amerika

qo'shmaShtatlariMudofaavazirliginingintegratsiyalashgankompyuterquv
vatishlabchiqarish (ICAM) dasturiSADTningfoydaliliginitanoldi. Bu
1981-yildauningbirqismini "IDEF0" (Integrated Computer Aided
Manufacturing, ICAM) debnomlab,
dasturiya'tminotniishlabchiqishbo'yicha Federal standart
sifatidachopetishgaolibkeldi.

BunomostidaSADTdanharbiyvasanoattashkilotlaridaminglabmutaxassisl
arfoydalanishgan. MilliystandartlarvatexnologiyalarinstitutidaIDEF0
standartiningso'nggiversiyasi 1993-yilningdekabridachiqarildi.

Shunita'kidlashkerakki, IDEF0
RossiyavaRespublikamizGosstandartomonidanfoydalanishhuchuntavsiy
aetildivamilliydavlatidoralaridafaolqo'llaniladi.

Ushbumetodologiyaaxborottiziminingfunktionaljihatinitavsiflashd
ama'lumotlaroqimigayo'naltirilganusullar (DFD) bilanraqobatlashadi.
Aksincha, IDEF0 imkoniyatlariko'yidagilardanberadi:

- harqandaytizimnitavsiflash, nafaqataxborottizimlari
(dfddasturiya'tminotnitasvirlashhuchunmo'ljallangan);

-
ungaqo'yiladiganyakuniyatalablarnibelgilashdanoldintizimvauningtashqi
muhittavsifiniyaratish. Boshqaso'zlarbilanaytganda,
bumetodologiyasiyordamida, sizasta-
sekinqurishvauningbajarilishinitasavvurqilishqiyinbo'lsaham,
tizimtahlilqilishmumkin.

Shundayqilib, IDEF0
kengdoiradagitizimlarniyaratishningdastlabkibosqichlaridafoydalanishm
umkin.

Shubilanbirgamavjudtizimlarningvazifalarinitahlilqilishvaularnitakomill
ashtirishechimlariniishlabchiqishhuchunfoydalanishmumkin.

IDEF0 metodikasigrafikjarayonnitasvirlashtiligaasoslangan.
IDEF0 notatsiyadagi model
iyerarxiktartiblivao'zarobog'langandiagrammalarto'plamidir.
Harbirdiagrammatizimtavsifiningbirligibo'lib,
alohidavaraqdajoylashgan.

Model (AS-IS, TO-BE yoki SHOULD-BE) diagrammalar 4 turnio‘zichigaolishimumkin

- kontekstdiagrammasi;
- parchalanganda diagrams;
- tugundaraxtdiagrammalari;
- faqatEHMuchungrafiklar (FEO).

Diagrammalardaraxttuzilishiningyuqorisibo‘lmishkontekstdiagramma (top-level diagramma) tizimningmaqsadi (asosiyvazifasi) vauningtashqimuhitbilano‘zarota‘siriniko‘rsatadi.

Harbirmodelfaqatbittakontekstgrafikkaegabo‘lishimumkin.

Asosiyfunksiyanitavsiflagandanso‘ngfunksionaldekompozitsiyaamalgao shiriladi, ya‘niasosiyfunksiyanitashkiletuvchifunksiyalaraniqlanadi.

Bundantashkari,

o‘rganilayotgantizimningkeraklidarajadadetallashtirilgungaqadarfunksiy alarsubfunksiyalargabo‘linadivahokazo.

Sistemaningharbirshundayfragmentinitasvirlovchidiagrammalardekompozitsiondiagrammalardeyiladi.

Harbirdekompozitsiyaseansidanso‘ngeksperttekshiruvseanslario‘tkaziladi

domenekspertlarirealjarayonlaryaratilgandiagrammlargamoskelishiniko‘rsatadilar.

Topilganharqandaynomuvofiqliklartarafetiladi, so‘ngrajarayonlarniyanadabatafsildavometiradi.

Tugundaraxtdiagrammasidafunksiyalar (isho‘rinlari) ningiyerarxi kmunosabatlariko‘rsatilgan, lekinularo‘rtasidagimunosabatlaremas.

Ulardanbirnechtasibo‘lishimumkin, chunkidaraxtharqandaychuqurlikkavaharqandaytugundanqurilishimumkin.

EHMdiagrammalaritizimdayuzberadiganjarayonlarningmuqobilko‘rinishiniko‘rsatishuchunmodelningalohidaqismlariniko‘rsatishuchunmo‘ljallangan (masalan, tashkilotboshqaruvinuqtainazaridan).

Kichikloyihalaralgorimlariniyozishusullari. Tizimvazifalarinibelgilabvaaxborotasoslariniishlabchiqqandanso‘ng, keyingiqadamtizimidaaxborotoqimiharakatiloyihalashtirishiborat.

Tizimningmodelikeraklifunksiyalargaqandayerishishvaunita‘minlashuchunqandayma‘lumotlarishlatilishiniko‘rsatadi. Shundayqilib, modelbevositatizimningfunksionalvaaxborotmodellarigaasoslanadi.

BumodellarodatdaIDEF0

vaDFDdekompozitsiyadiagrammalariningoxirgidarajalaridako‘rsatilganf

unksiyalar (jarayonlar) uchunquriladi.
DFD uchun diagrammayoki diagramma ko'rishidagi model elementar jarayonlarni minispektlarni ko'rishidagi tasvirlashdagi ko'rayaxshiroq echimdir.

Bunday vaziyatlarda funksional imkoniyatni tasvirlashning eng eng tanqilivamash huriyati va uslubiyati bor:

- algoritmlarning blok diagrammalari;
- Jadvalning EPC ;
- BPMN metodologiyasi.

Algoritmlar va BPMN diagrammalarining oqimlari anaviy tizimlar niti zimliyondashuv tamoyillariga asoslangan holda model lashtirish uchun eng mos keladigan funksiyalarni elementar bosqichlarga (protseduralar, operatorlar, harakatlar va hokazolarga) aniqlash bilan ta'riflanadi.) ma'lum ketma-ketlikda (algoritmlar bo'yicha) bajariladi. EPC diagrammalar va BPMN diagrammalar bir harakat yoki ularning majmui ijrosi tizimida sodir bo'lgan voqealar bog'liqlash bo'lgan voqea gahoyat naltirilgan tizimlarini model lashtirish uchun yaxshi vosita hisoblanadi. Bundan tashqari, bu usul va uslubiyatlardan nafaqat xulq-atvorini model lashtirish uchun, balki funksional model lashtirish uchun ham foydalanish mumkin.

Ularda ma'lum mallarni (mantiqiy simvollarini) bajarish shartlarini ko'rsatishga imkon beruvchi elementlarning mavjudligi tizim funksiyalarining mantiqiy ketma-ketligini yaxshi roq tushunishga imkon beradi.

Algoritmlar niti tasvirlashda uzoq vaqt davomida flowcharts (Basic Flowchart) muvaffaqiyatli ishlatilgan. Algoritmlarning blok diagrammalarini qurish GOST 19.701-90 (ISO 5807-85) dasturiy hujjatlarning yagona tizim bilan tartib gasolinadi. Dasturlar, ma'lumotlar va tizimlar uchun algoritmlarning sxemalarini mavjud.

Ushbu davlat standarti "ISO 5807-85" xalqaro standart sifatida tasdiklanilgan. Axborotni qayta ishlash – ma'lumotlar, dastur va tizim oquvchilari, dastur tarmoqlar va ularni tizim resurslarini jadvalari uchun hujjat simvollarini va konvensiyalarini moslashtirilgan.

GOST 19.701-90 gahoyat, diagramma-bu muammoni hal qilishning ta'rif, tahlili yoki usulining grafik tasviri. Tizimning statik va dinamik jihatlari ko'rsatish uchun diagrammalardan foydalanishingiz mumkin.

Davlat standarti daberilgan simvollaridan quyidagilarda sxemalarda foydalanish mumkin:

- ma'lumotlarsxemalari - ma'lumotlarniqaytaishlashketma-ketliginivaularningtashuvchilarinianiqlaydi;

- dasturdiagrammalari - dasturdagiamallarketma-ketliginiko'rsatish (aslidabuan'anaviyma'nodaalgoritmlarningasosiyxususiyati);

- tizimningishlashsxemasidatizimdagiamallarvama'lumotlaroqimlariko'rsatiladi;

- dasturningo'zaroishlashdiagrammalari - dasturlar (modullar) ningaktivlashtirishyo'liniko'rsatishvaularningtegishlima'lumotlarbilano'zarohamkorligiuchun;

- tizimresursdiagrammalari ma'lumotlarbloklarivaqaytaishlashbloklarikonfiguratsiyasiniko'rsatish.

Yuqoridakeltirilganturdagisxemalardanko'rinihturibdiki, ulardannafaqatxulq-atvorob'ektinimodellashtirishuchun, balkifunksional, axborotvakomponentliloyihalashvazifalariuchunhamfoydalanishmumkin

Tizimningfunktionalmodeliniquishdastrukturaviyyondashuvninga sositamoyillari

dekompozitsiyavaiyerarxikalgoritmtamoyillaridanfoydalaniladi.

Funktionalmodeli

buturlidarajadagidetallargaegabo'lgano'zarobog'langandiagrammalarmajmuibo'lib, harbiryangidetaldarajasibilantizimyanadato'liqlashadi.

Diagrammalargrafikbelgilashningquyidagielementlarinio'zichigaolishimumkin:

- data symbollar - ma'lumotlarningmavjudligini, axborotlarningturiniyokima'lumotlarnikiritishvachiqarishusuliniko'rsatish;

- jarayonsimvollari - ma'lumotlarustidabajariladiganamallar;

- chiziq symbols

jarayonlarva/yokisaqlashvositalariorasidagima'lumotlaroqimlarini, shuningdekjarayonlarorasidaginazoratoqimlariniko'rsatadi;

-maxsusbelgilar

diagrammalarniyoziishvao'qishniosonlashtirishuchunishlatiladi.

Semantikmazmunga'rabolinishdantashqariharbirturkumsimvollari (maxsuskategoriylardantashqari)

asosiyvao'zigaxossimvollargabo'linadi.

Asosiybelgijarayonyokima'lumotlartashuvchininganiqturinoma'lumbo'lgandayokihaqiqiyima'lumotlartashuvchisi (jarayoni)

nita‘riflashgahojatbo‘l magandaqo‘llaniladi.

Jarayonyokima‘lumotlartashuvchininganiqturima‘lumbo‘lgandavadiagrammadako‘rsatilishikerakbo‘lgandao‘zigaxossimvoldanfoydalaniladi.

Hodisalijarayonzanjiri (EPC) - biznes-jarayonlarnimodellashtirish, tahlilqilishvaqaytatashkiletish (funksionalmodellashtirish) uchunishlatiladigandiagrammat turi. ShubilanbirgaEPC diagrammalaridanfunksiyalarniamalgaoshirishdatizimningalohidaqismla rifunksionalliknimodellashtirishdafoydalanishvaan‘anaviyoqimlarnialma shtirishhuchunxizmatqilishmumkin.

EPSmetodiAvgust-VilgelmSheertomonidan 1990-yillarboshidaintegrallashganaxborottizimlariarxitekturasimetodikasigaoi dishiningbirqismisifatidaishlabchiqilgan.

EPC notatsiyasidajarayon (funksiya) diagrammasihodisavafunksiyalarningtartiblibirikmasidir. Harbirkunksiyauchunungahamrohbo‘lgandastlabkivaoxirgihodisalar, ishtirokchilar, ijrochilar, moddiyvaaxborotoqimlarianiqlanishihamdaquyidarajalargabo‘linishimumkin.

DFDkabi, EPC metodologiyasiturlinotatsiyalarni (elementlarningsintaksisivasemantikasini) qo‘llabtalqinqilishbilanrivojlanmoqda. EPC (ARIS, Microsoft Visio, Business Studio, Bflow) yordamidabiznes-jarayonlarnimodellashtirishqobiliyatiniamalgaoshirganmetodikamuallifi ham, dasturiyta‘minotishlabchiqaruvchilarihambungaqo‘lurishgan. Blokdigrammasigao‘xshashlikbo‘yichasimvollar (elementlar) grafikyozuvlarinimaqsadgamuvofiqquruhlasmumkin. Quyidagijadvaldaadabiyotvadasturiyta‘minotdaengko‘puchraydiganEPC larivaularningmuqobiltasvirlarimavjud.

EPC yordamidadekompozitsiyajarayonlarnimodellashtirishvaiyerarxikalgoritmlarjarayoninimodellashtirishningklassiktamoyillarigaamalqiladiyu. Dekompozitsiya, alohidadiagrammalardako‘rsatishbilan, IDEF0 diagrammalardaoldindanbelgilanganjarayonlardaishlashgao‘xshashfunksiyalaruchunbajariladi.

1. EPC jarayoniningdiagrammasikamidabirboshlashhodisasibilanboshlashkerak vakamidabirtugashhodisasibilanyakunlandi.

2.

Jarayonrivojlanibborgansarihodisalarvafunksiyalaralmashinibturishikera
k.

3. Diagrammavazifalaritavsiyasoniko‘pibilan 20 tabo‘lishimumkin.
Agargrafikfunksiyalarisoni 20 dansezilarlidarajadaoshsa,
yuqoridarajadagijarayonlarnoto‘g‘rita‘kidlanganvamodelnisoqlashkerak
bo‘ladi.

4.

Hodisalarvafunksiyalarjarayonningborishiniaksettiruvchiqat‘iybittakiruv
chivabittachiquvchialoqani (nazoratoqimini) o‘zichigaolishikerak.

5. Grafikbirtegishsizelementlarbo‘lmasligikerak.

6.

Yuqorimoduldiagrammasidafunksiyaatrofidagihodisalarvamantiqiyopera
torlarfunksisiyadekompozitsiyadiagrammasidaboshlang‘ich /
natijaviyhodisalarvaoperatorlarbo‘lishikerak.

7.

Harbirlashmakamidaikkita kiruvchiulanishgavafaqatbittachiquvchi,
ayirmalar -
faqatbittakiruvchialoqavakamidaikkita chiquvchibo‘lishikerak.
Operatorlarbirvaqtningo‘zidabirnechtakiruvchivachiquvchibog‘lanishlar
gaegabo‘laolmaydi.

8.

Mantiqiyoperatorlarfaqatfunksiyalarniyokifaqathodisalar nibirlashtirishiy
okibirlashtirmasligimumkin.
Birvaqtningo‘zidafunksiyavahodisanibirlashtiraolmaydi.

Visual

C++

muhitidahisobotshaklinitayyorlashvachopetishelementlari.Buninguc
hunma’lumotlarbazasinibilishtalabqilinadi.

Quyidagichamasalaberilganbo‘lsin.

Masala.

Universitetma’lumotlarbazasiberilganbo‘lsin.

Undatalabavasemestrnomlijadvallarbo‘lsin.

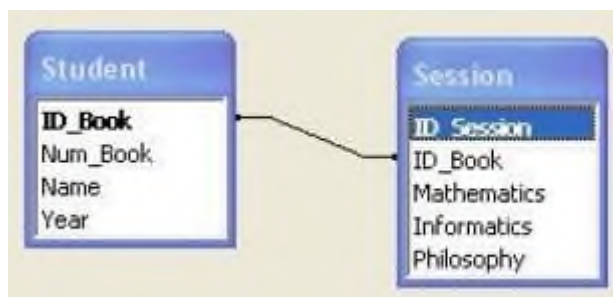
14.1-jadval.Talabajadvaliningtuzilishi

Maydonnomi	tipi	Izoh
ID_Book	Butunson (int)	Birlamchikalitmaydon
Num_Book	matnli (10 belgi)	Talabaninggreytingdaftariraqami
Name	matnli (30 belgi)	Talabaningismisharifi
Year	Butunson (int)	Tug‘ilganyili

14.2-jadval.Semestrijadvaliningtuzilishi

Maydonnomi	tipi	Izoh
ID_Session	Butunson (int)	Birlamchikalitmaydon
ID_Book	Butunson (int)	reytingdaftariraqami
Mathematics	Butunson (int)	Matematikabahosi
Informatics	Butunson (int)	Informatikabahosi
Philosophy	Butunson (int)	Falsafabahosi

Ushbujadvallarningo‘zaromunosabatiquyidagibog‘lanishorqaliifod alanganbo‘lsin.



14.1-rasm. Jadvallarningrelyatsionbog‘lanishi.

Berilganma‘lumotlarasosidahisobotlardanfoydalanishnitashkilqilib beruvchidasturniishlabchiqishniko‘ramiz.

Quyidagima‘lumotlarnio‘zichigaolganhisobotyaratishlozim:

- Reytingdaftarini
- Talabaningismi – sharifi
- Matematikabahosi
- Informatikabahosi
- Falsafabahosi
- Talabanningo‘rtachabahosi

Joriyhisobotniyaratishdama‘lumotlar Microsoft Access DBMS tomonidanhosilqilinganso‘rovorqaliolinadi. So‘rov Query1 debataladi. Hisobotalohidashakldako‘rsatilishikerak.

Buninguchunquyidagiamallarbajariladi:

1. Microsoft Visual Studio tizimiorqali Windows Forms dasturshabloniyordamidaloynihayaratiladi. Microsoft Visual Studio ishगतushirilgandanso‘ng, sizyangiloyihayaratishkerak. Windows Forms dasturshabloniyordamidayangiloyihaniyaratishvasaqlashningbatafsilolro qytibo‘tilgan.

2. Ma‘lumotlarbazasiniyaratishyokitayyorma‘lumotlarbazasifaylini yuklasholishlozim. Ma‘lumotlarbazasiuchun Microsoft Access ma‘lumotlarbazasiniboshqarishtizimiyordamidayaratiladi. Ma‘lumotlarbazasidaikkitegishlijadvaltalabavasesmestr, shuningdek, hisobotniyaratishuchunishlatiladigan Query1 so‘roviyaratibolishkerak.

3. Ma'lumotlar bazasifaylining loyihaga ulanishi.

Ma'lumotlar bazasifaylini loyihaga ulashdan oldin ushbu faylni loyihafayllar isqatlanadigan papkaga saqlashni tavsiya etamiz. Foydalanish uchun "Education.mdb" ma'lumotlar bazasi, uni standart usulda loyihaga ulash kerak.

Loyihaga ma'lumotlar bazasifaylini ulash uchun "Add Connection" dan foydalaniladi.

Bu amalkontekst meniyudan buyruq ro'yxatida chaqiriladi.

4. Natijada "ma'lumotlar manbasini tanlash"

oynasida ma'lumotlar manbai (Data Source) ni tanlash uchun - Microsoft Access ma'lumotlar bazasifayli ulash kerak. Keyingi oynada "ma'lumotlar bazasifayl nomi" dagi "ulanish qo'shish" ma'lumotlar bazasifayligayo'lni ko'rsatiladi ("Browse" tugmasi orqali).

5. Asosiy loyiha oynasini ishlab chiqish.

Topshiriq shartiga ko'ra hisobot alohida shakldako'rsatish ilozim.

Shuning uchun loyiha ning asosiy shakli 14.2-rasmdako'rsatilganidek shaklga egabooladi.



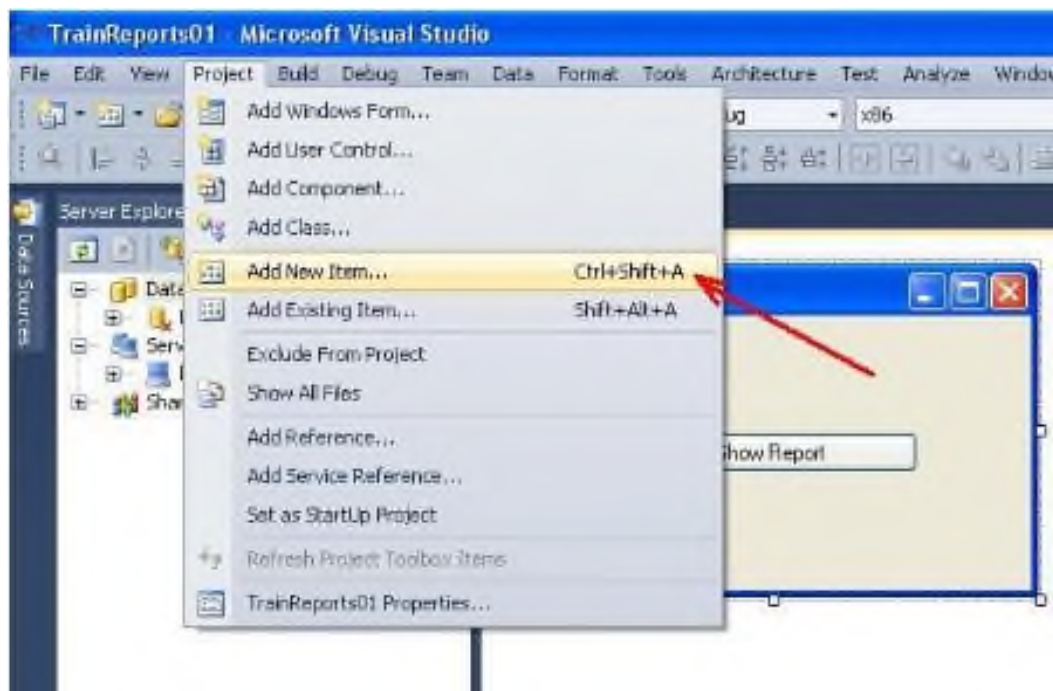
14.2-rasm.

Agar rasmdako'rib turganingizdek, shakl button1 nomidagi faqat bitta tugma turini o'z ichiga oladi. "Show Report" tugmasini bosganda so'ngi hisobot boshqachashakldako'rsatiladi. Quyidagi xususiyatlarini sozlash kerak:

Boshqarish elementi button1 ning Text = «Show Button»

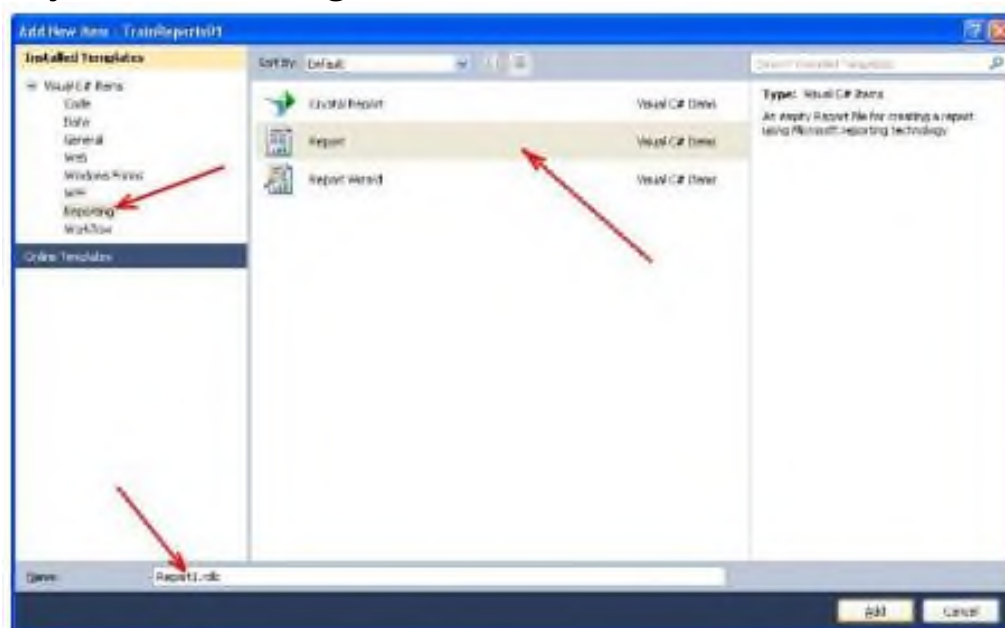
Boshqarish elementi Form1 ning Text = «Report Viewer»

6. Hisobot faylining loyihaga ulanishi. Microsoft Visual Studio dasturida har bir hisobotda «*.rdlc» kengaytmasiga egafaylmavjud. Bufayl hisobotda hosil qilingan ma'lumotlarni o'z ichiga oladi. Microsoft Visual Studio dasturida fayl yaratish uchun loyiha meniyusidan Add New Item buyrug'inicha qiriladi (14.3-rasm).



14.3-rasm. Loyihamenyusidan Add New Item amali

7. Natijada, «Add New Item...» hisobotshabloninitanlashkerakbo‘lganoyna ochiladi (14.4-rasm) va hisobot fayli uchun nom belgilanadi.

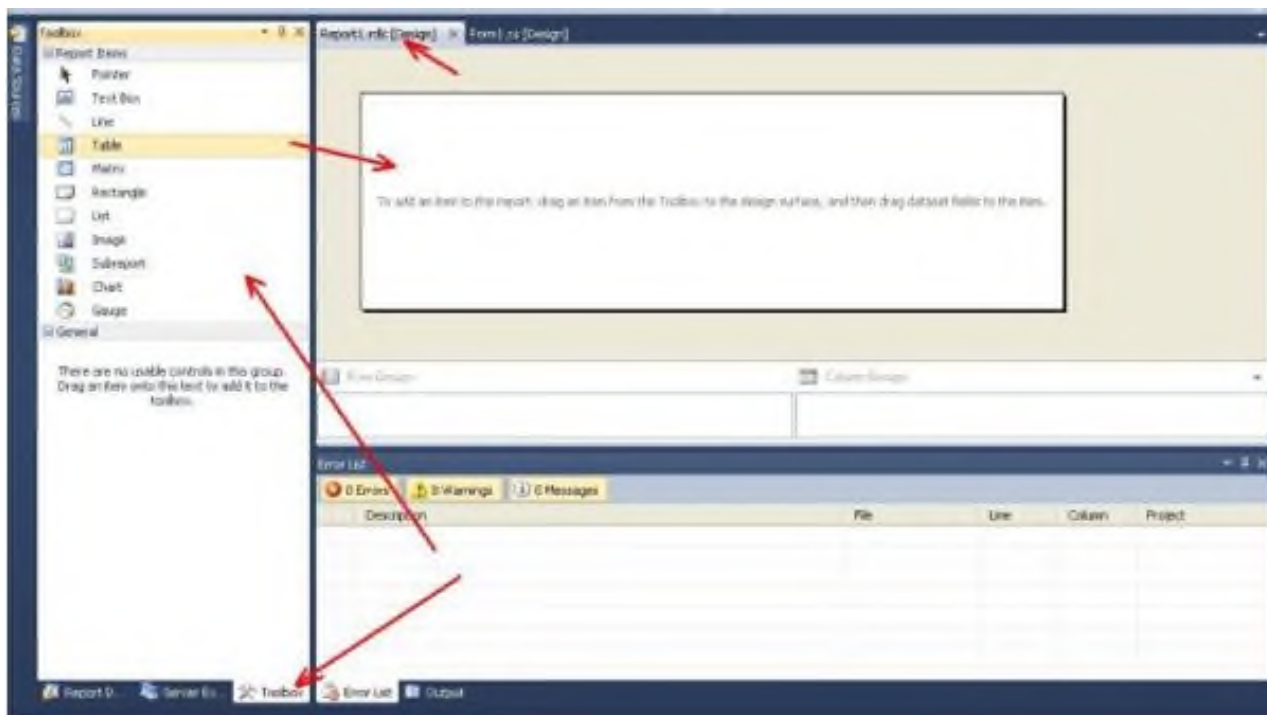


14.4-Rasm. Hisobotshabloninitanlash va hisobotni o‘rnatish

Tanlovnitasdiqlagandanso‘ng Microsoft Visual Studio oynasida hisobotning asosiy oynasi ko‘rsatiladi.

Hisobot oynasida hisobotni loyihalashtirish lozim. Buning uchun quyidagilozim.

1. ToolBox asboblari paneli. ToolBox paneli - hisobot elementlarini loyihalashtirish uchun ishlatiladi.



14.5-rasm. ToolBox panel vahisobotloyihalashtirishvositalari.

2. Hisobot yaratish uchun - Microsoft Visual Studio hisobotini ishlab chiqish uchun bir qatortaklifetiladigan standart shablonlar bo'lar. Boshqaruvlardan turli hisobotlarning ko'rinishini (taqdimotini) loyihalashtirish uchun foydalanishingiz mumkin.

Quyidagi elementlartaklifetiladi:

TextBox - Hisoblashlarni natijasida olingan teglar, maydonlari yoki qiymatlarni ko'rsatadi;

Line - nuqtayoki undanko'pqalinlikka ega bo'lgan chiziqli chizadi;

Table (jadval) - nazorat ustunlar va qatorlarni ko'rsatadi;

Matrix - o'zgaruvchan raqamli, ustunlar va qatorlarni ko'rsatadi;

Rectangle - hisobot elementlari uchun shablonsifatida to'rtburchak chizadi;

List - har bir guruh yoki ma'lumotlar qatori uchun takrorlanadigan hisobot bandlarini ko'rsatadi;

Image - tasvirni bitmap sifatida ko'rsatadi (masalan, logotipi yoki fotosurat);

Subreport - hisobot chegaralarini doirasida bog'liq sub-hisobot ko'rsatadi;

Chart - Ma'lumotlarni har xil turdagi diagrammalarni ko'rsatadi;

Gauge - Maydonyokiifodaningqiymatinichiziqliyioki radial qarama-qarshilikko‘rinishidako‘rsatadi.

Harbinnazoratdanfoydalanishningo‘zigaxosxususiyatlarigakirmasda n, muammonihalqilishhuchunjadvalnazoratidanfoydalanamiz. Bunazoratustunlarsonvaqatorlarbiro‘zgarmaydiganqatorborma’lumotlarn iko‘rsatadi.

Buelementlarningfoydalanishxususiyatlaritizimdabirxilamalgaoshiri ladi.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Visual

C++

muhitidaturlisohagaoidmasalalarniye chishhuchunkichikl oyihalarniloyihalashusullariasosanqandaytizimlarniloyi halashtirishusullarinazardatutiladi.

2. Axborottizimlariningloyihalashtirishhuchunaxborottiziminingq andaydavriniyaxshibilishvatushunishkerak?

3. Hayotdavrlarningtuzilishinibelgilovchiasosiystandar tnominiayting.

4. Hayotdavristrukturasinechtajarayonguruhigaasoslangan.

5. Mustaqil (loyihaganisbatan) ekspertlarningkorxonafaoliyatiningqabulqilingantalablar, rejalarvashartnomashartlarigamuvofiqliginianiqlashishlarinimadebyur itiladi.

6. Strukturalitahlilvaloyihalashtexnikasigaasoslanganmetodolog iyanominiayting.

7. Kichikloyihalaralgorimlariniyozishusullarinechta.

8. BPMNdiagrammalarningvazifasinimadaniborat.

9. AlgoritmlarningblokdiagrammalariniquirishqandayGOSTdast uriyhujjatlariningyagonatizimibilantartibgasolinadi.

10. Ma’lumotlarxemalarinimauchunkerak.

11. Hisobotshaklinitayyorlashvachopetishhuchunnimanibilashkera kvanimauchun.

12. Jadvallarning relyatsion bog‘lanishidegandan imani tushunasiz.
13. Yangi hisobotloyiha siniyaratish uchun qanday amallar bajariladi.
14. Hisobotkonch traktorining elementlarini sanab bering.
15. Hisobotdagi dalga asoslangan ma’lumotlarni chiqarish uchun qaysi elementdagi foydalanish maqsadga muvofiq.

3.7. Foydalanuvchi interfeysini loyihalash.

 Foydalanuvchi interfeysini loyihalash usullari, talablari, ma’lumotlarni eksport qilish vositalari va komponentalari, .NET texnologiyasining hisobotni yaratish konstruktori, testlash va instruksiya yozish usullari, sinovchining xususiyatlarini bo‘yicha nazariy bilimlarni keltirilgan.. Bilimlarni mustahkamlash uchun 15 tanazariysavol berilgan.

 *Kalitso ‘zlar.* Interfeys, UI, WUI, amaliy dastur, ma’lumotlarni kiritish, PUI, integratsiya, o‘zaro bog‘lanish, xisobot, testlash.

 *Bilish shart bo‘lgan tushunchalar.* Operatsion tizim va amaliy dasturlar da ishlash, sinf va sinf ob‘ekti, xususiyat, hodisa, forma, komponenta dasturlashga oid dastlabki tushunchalar hamda C++ tilini qo‘llab quvvatlovchi muhitda ishlashni bilish lozim.

 *Bilib olasiz.* Visual C++ muhitida tuzilgan dasturlarning foydalanuvchi interfeysini loyihalash usullari, talablari, ma’lumotlarni eksport qilish vositalari va komponentalari, .NET texnologiyasining hisobotni yaratish konstruktori, testlash va instruksiya yozish usullari, sinovchining qo‘yiladigan talablarni o‘rganish giz mumkin.

REJA

1. Foydalanuvchi interfeysini loyihalash usullari.
2. Ma’lumotlarni eksport qilish vositalari va komponentalari.
3. Testlash va instruksiya yozish usullari.

Inson-mashina interfeysi (IMI) keng tushunchadir. Inson-operator va u lar boshqaradigan mashinalar bilan o‘zaro aloqasita‘minlashnin g muhandislikechimi larni o‘z ichiga oladi. Inson-mashina interfeysi tizimlarini yaratishda yaqindan ergonomika tushunchalari bilan bog‘liq. IMI loyihalashtirish quyidagilarni o‘z ichiga oladi:

- ish joyini yaratish: ishchining maydon va boshqaruv paneli, qurilmalar va boshqaruv larni joylashtirish, ish joyi va ehtimol ranglar uyg‘unligi.

- operatorning barcha boshqaruv bilan o'zaro aloqasini ta'minlash hisoblanadi:

ularning mavjudligi vazarur harakatlari, kirish samaradorligi va tezligi, mustahkamligi, nazorat harakatlarining (izchilligi), joylashgan o'rni, displeylar va ulardagi teglar hajmi (bularning barchasi mavjudligi).

Inson -

kompyutero'zaro hamkorligi ilmiy ko'pyo'nalishlik kontekstida kompyuter grafika, muhandislik psixologiyasi, ergonomika, tashkilot nazariyasi, kognitiv fan, informatika va boshqalarni o'z ichiga oladi.

Foydalanuvchi interfeysi -

bu foydalanuvchi kompyuter bilan o'zaro aloqa qilish imkonini beruvchi dasturiy va apparat vositalar majmuasidir.

Bu o'zaro muloqotlar dialoglarga asoslanadi. Bu holda dialog deganda inson va kompyutero'rtasida Real vaqt da amalga oshiriladigan va muayyan vazifa birgalikda haletishga qaratilgan axborot almashuvitushuniladi. Axborot almashish va harakatlarni muvofiqlashtirish har bir dialog foydalanuvchi va kompyuterni fizik ulaydigan alohida i/o jarayonlardan iborat. Xabarlarni uzatish orqali axborot almashiladigan azoratsignallari. Xabar-dialog almashinuvida ishtirok etuvchi axborot qismidir. Ular quyidagilardir:

-

odam tomonidan kiritish vositalari yordamida hosil qilinadigan kiritish xabarlari: klaviatura, manipulyatorlar, sichqonchak va boshqalar;

-

kompyuter tomonidan matnlar, audio signallar va/yoki tasvirlar ko'rishida hosil qilinadigan va foydalanuvchiga monitor

ekranida yoki boshqa chiqish qurilmalarida ko'rsatiladigan chiqish xabarlari.

Asosan foydalanuvchi quyidagitarixdagi xabarlarini hosil qiladi:

axborot so'rovi, yordam chiso'rovi, operatsiya yoki funksiya so'rovi, ma'lumot kiritish yoki o'zgartirish, shablon maydonini tanlash va boshqalar. Bunga javoban, uqabul qiladi: maslahatlari yoki yordam, javob talab qilmaydigan axborot xabarlari, harakatni talab qiladigan so'rovlar, javob kerak bo'lgan xato xabarlarini shablon formatida o'zgarishlar.

Dasturlashga protsessual va ob'ektga o'zgartirilgan yondashuvlar bilan taqqoslaganda,

interfeysni ishlab chiqishga protsessual yoki o'zgartirilgan va ob'ektga o'zgartirilgan yondashuvlar mavjud. Protsedura yoki o'zgartirilgan interfeyslar

"protsedura" va "operatsiya" tushunchalariga asoslangan foydalanuvchi o'zaro aloqasining anaviy mo

delidanfoydalanadi. Bu model doirasidadasturiya 'minotfoydalanuvchigategishlima' lumotlarnivauning natijasini foydalanuvchianiqlaydiganma' lumamallarnibajarishimkoniyatini beradi.

Ob'ektyo 'naltirilgan interfeyslarni foydalanuvchi o'zarobiroz boshqa chamodelini foydalanib, manipulyatsiya domenob'ektlarini qaratishim umkin.

Zamonaviy realliklardan farqli o'laroq, birinchi kompyuterlarga grafik foydalanuvchi interfeyslari juda zaif bo'lgan. Shuning uchun, eng boshida, odamlar faqat buyruq qatordan foydalanishgan (CLI yoki buyruq qatorni interfeysi), unda buyruqlar so'rovlari yordamida belgilangan edi. Keyinchalik, bu TU interfeyslarni ishlab, endi operatsion tizimlari o'rnatish jarayonida ishlatiladi. Kompyuterlarning mavjudligi foydalanuvchilarga qulay interfeysni ishlab chiqish zarurligiga olib keladi.

Grafik foydalanuvchi interfeysi - bu kompyuterning tobora ortib borayotgan ishlashi bilan birga mustahkam o'rnatilgan interfeysturi. Yaqin kelajakda, odamlar nutq yordamida kompyuter bilan o'zaro imkonini beradi foydalanuvchi audio interfeyslarni (VUI yoki ovozli foydalanuvchi interfeysi) bo'lishim umkin.

Turli kompyuter o'yinlari tabiiy foydalanuvchi interfeysi (NUI yoki tabiiy foydalanuvchi interfeysi) yaratib beradi. Uning tizimi insonning harakatlarini tahlil qiladi va ularni o'yindagi harakatlarga aylantiradi. Perseptual foydalanuvchi interfeysi (PUI) va aqlli-kompyuter interfeysi (BCI yoki brain-kompyuter interfeysi) hozirda ishlab chiqilmoqda. eng so'nggi rivojlanish odamlarga kompyuterlarni fikrikuchi bilan boshqaris hqobiliyatini ta'minlashga qaratilgan.

Web-ilovalarning foydalanuvchi interfeysi ham katta auditoriya danielib chiqqan holda o'zigacha xususiyatlarga ega. Notiqlik haqida, notiqlik-ritorika haqida shunday fan mavjud. Fikrlash insoniyat bo'rikan, bu mavzu bo'yicha birinchi ishlar Empedokllarga (taxminan miloddan avvalgi v asr) taalluqlidir. Ritorika dan bilamizki, hozirgi odamlar sonining ortishi bilan tomoshabinlarning intellektual darajasi nika maytirish qonuni mavjud. Auditoriya qanchalik katta bo'lsa,

nutqshunchaliksoddabo'lishi,
 iboralarshunchalikqisqabo'lishivanutqdagipauzalarshunchalikmuhimaha
 miyatkasbetadi. Boshqatomondan,
 ijtimoiytarmoqlarningrivojlanishibizgamuloqotmutlaqoog'zakiemasligin
 iko'rsatadivafoydalanuvchiinterfeysiorqalihammuloqotqilishingiz mumki
 n. Yuqoridagilardankelibchiqqanholda, veb-
 dasturinterfeysisoddabo'lishikerakligimantiqankelibchiqadi.
 Ilojiborichaoddiy. Aytaylik,
 ma'lumbirkorxonaichidaishlashuchunmo'ljallangandasturuchuninterfeys
 ishlabchiqyapsiz. Shubilanbirga,
 sizhardoimtizimningkelajakdagifoydalanuvchilariuchuno'quvkurslarinit
 ashkilqilishingiz,
 ekrandanimaborliginivaishlabchiquvchisifatidanimademoqchiekanliging
 izniaytishingiz mumkin.

FI dizayniko'plabyondashuvlarnio'zichigaoladi: kaskadli, ichki,
 tashqi, iterativvaboshqalar.
 Yondashuvlarningharbiriichidabirqatorusullarhammavjud.
 Yuqoridarajadagiyondashuvdanqat'iy nazar,
 yuqoridarajadagivapastdarajadagio'zaroaloqalarniloyihalasho'zigaxosus
 ullarnitalabqiladi. Ob'ektagayo'naltirilganyondashuvtasvirlangan,
 garchiko'plabdizaynbosqichlariuchunan'anaviyrivojlanishusullaribilanal
 mashtirilishim mumkin.
 UshbuyondashuvyordamidaolingandizaynnatijalariFI gayo'naltirilganilo
 valarnihujjatlashtirish, modellashtirish, prototiplash,
 baholashvaamalgaoishirishuchunfoydalidir.

Nostandartyondashuv.
 YuqoridarajadagiloyihatizimibilanFI vafoydalanuvchio'zaro standart
 xattihaqidao'ylashmumkin, lekin tafsilotlarqo'shiladi, deb,
 ustandartierishishuchunyanadaqiyinbo'ladi —
 vabirvaqtningo'zidaizchil-qo'llashturliqismlarigaajratiladi.
 Tafsilotlarodatdakompromisslarvakutilmaganhodisalarbilanbohliqdir .

Foydaliqoida. Butasodifanyokiloyihagamuvofiqsodirbo'lsa,
 nostandartxattiharakatlariqo'llashkerak.

Yangilangangrafik.
 Dastlabkigrafikimkoniyatlarvavizuallashtirishmasalalaridastlabkiloyihal
 ashbosqichlaridaishlabchiqiladi,
 shuninguchunbatafsilloyihalashbugrafikimkoniyatlarnianiqlashnitalabqil
 adi
 (ba'zangrafikadanfoydalanishdastlabkiloyihalashtushunchalaridantashqa

rigachiqadi). Bundantashqari, bubosqichdaekranlarvaxabarlarningyakuniybatafsiltartibitalabqilinadi.

Javobvaqti. Og'irtahlilyokisimulyatsiyaorqali, muhimvaengteztizishlatiladigandasturiyta'minotjoylariuchunjavobvaqtsmetasiniolishkerak.

Oxirgifoysdalanuvchitomonidanamalgaoshirilganharbifunksiyauchunehtimoliyjavobvaqtigaasoslanib, dasturiyta'minotuchunmosteskarialoqamexanizmiishlabchiqiladi.

Interfeysyaratishningasosiytamoyillari

1. Tabiiylik (intuitiv).

Tizim bilan ishlash vazifasini hal qilish jarayonini boshqarish uchun kerakli dasturlar (interfeys elementlarini) topishda foydalanuvchiga qiyinchilik tug'dirmasligi kerak.

2. Mustahkamlik.

Agar foydalanuvchi tizim bilan ishlash uchun zarur bo'lgan barcha ma'lumotlarni o'z ichiga olgan bo'lsa, u holda metodlar tizimning boshqa qismini boshqarish uchun zarur bo'lgan bo'lsa, shuningdek, interfeys orqali tizim bilan ishlash o'rinli bo'lgan, doimiy, qulay me'yorlarga (masalan, Enter klavishi bilan foydalanish) mos kelishi kerak.

3. Ortiqchalilik.

Bu foydalanuvchi faoliyatini yoki tizimni boshqarish uchun zarur bo'lgan minimal ma'lumot kiritish kerak degan ma'noni anglatadi. Masalan, foydalanuvchi ahamiyatsiz sonlarni kiritmasligi kerak (00010 o'rniga 10). Xuddi shunday, foydalanuvchi oldindan kiritilgan yoki avtomatik ravishda tizimdan olinishi mumkin bo'lgan ma'lumotlarni kiritishni talab qila olmaydi. Axborot kiritish jarayonini kamaytirish uchun iloji boricha standart qiymatlardan foydalanish tavsiya etiladi.

4. Yordam tizimiga to'g'ridan-to'g'ri kiritish.

Ish vaqtida tizim foydalanuvchiga kerakli ko'rsatmalar berishi kerak bo'ladi.

Yordam tizimi uchta asosiy jihati bo'yicha javob beradi:

taqdim etilgan buyruqlar sifatini aniqlashi;

xato xabarlarining tabiatini tizimning nima qilayotganini tasdiqlashi.

Xato xabarlarini foydalanuvchi uchun foydalilavtish uchun qayta ko'rib chiqilishi kerak.

Shubilan birga,

bu savollarga foydalanuvchilar uchun imkon qadar muloyim javob berish uchun qayta ko'rib chiqilgan tilda javob berishi kerak.

5. Moslashuvchanlik.

Tizim interfeys turlidagi darajadagi tayyorgarlikka ega foydalanuvchilarga xizmat ko'rsatishi kerak.

atqilishkerak.

Tajribasiz foydalanuvchilar uchun interfeysi yerarxik menyu strukturasiga ajralib foydalanuvchilar uchun buyruqlar, kalit kombinatsiyalar va parametrlar sifatida tashkilotilishi mumkin.

Sayt moslashuvchanligi

foydalanuvchilar osonlik bilan saytida manipulyatsiyadan ortiq qaraqam bilan bezovta holda zarur ma'lumotlarni topish imkonini beradigan sayt materiallari tashkilotilishi hisoblanadi.

Maxsus tadqiqotlarga ko'ra,

o'rtacha bir foydalanuvchi saytda o'ttiz tish soniyasarflaydigan vaqt ichida ularkamdan-kamsahifaning oxiriga o'tadi.

Shu munosabat bilan,

sayt samimiyma'lumotlarni chiqarishni tashkilotilish sohasidagi zamonaviy standartlarga javob bermasa, agar bunday sayt targ'ib qilish, barcha sahifalarni kerakli maqsadga erishish mumkin emas, chunki foydalanuvchi saytda zarur ma'lumotlarni topib bo'lmasa, ular shuncha ko'proq samimiy boshqarilgan saytda.

Saytning eng samarali moslashuvchanligi ta'minlash Internetda axborot taqdim etish xususiyatlarini bilishni talab qiladi.

Foydalanuvchilar,

birinchi navbatda, ularning ko'z dida moddiy ko'rish muhimdir.

Shuning uchun interfeysning maksimal soddaligi va qulayligi Internetdagi ma'lumotlarni tashkil qilishning asosiy tamoyilidir.

Saytda oddiy va samimiy interfeysni yaratish uchun mavjudlikning asosiy tamoyillarini amalga oshirishga imkon beruvchi texnikalardan foydalanish kerak.

Butun sayt uchun navigatsiya umumiy bo'lishi kerak,

sayt logoyu qorichap burchagida joylashtirilgan bo'lishi kerak va logobosish bosha sahifa olib kelishi kerak va veb-saytda header va alt bilgidan aloqama'lumotlarni joylashtirilgan kerak.

Saytda malakali qidiruvni tashkilotilish saytning mavjudligini ta'minlashning muhim elementi hisoblanadi. Qidiruv tugmasi, logotipi dan tashqari, sayt uchun ikkinchi eng muhim element hisoblanadigan u har qanday saytda foydalanuvchi uchun birinchi yordamchidir. Foydalanuvchi birinchi 2 soniya ichida butugmanit o'psa kamida, ular xavfsiz hissa sayt tomonidan keskinliklarning darajasini kamayadi.

Moslashuvchanlik sinov o'tkazish paytida ta'qib qilinishi kerak va uning ma'lum bosqichlarini bor. Sinov o'tkazishning mohiyati quyidagicha:

1. Muammolarni aniqlash;
2. Gipotezani hosil qilish;
3. Test uchun o'lchovni aniqlash;
4. Belgilar va test senariyini aniqlash;
5. Tanlang respondents;

6. Anketalarni to'ldirish;
7. Kirish b'rifingini o'tkazish;
8. Mavjudligini to'ldirish;
9. Intervyu respondentlari;
10. Natijalarni tahlil qilish;
11. Saytdizayni uchun talablarni aniqlash.

Umuman olganda, f'ni ishlab chiqish sohasida foydalanuvchilarning imkoniyatlarini yaxshilashni inobatga olish kerak.

Ma'lumotlarni eksport qilish vositalari va komponentalari. Zamonaviy dasturlar foydalanuvchini qiziq tiradigan ma'lumotlarga kirishning ko'p usullarini ta'minlaydi.

Ma'lumotlar bilan ishlash uchun ixtisoslashgan dasturi lovalari ishlab chiqilgan. Microsoft Office kabi umumiy ofis vositalari bilan integrallashgan ma'lumotlarni chiqarish imkoniyatlarini o'z ichiga oladi.

Foydalanuvchi hozirda ma'lumotlar bazasidan kerak ma'lumotlarni chiqarib eksport qilish imkon.

Biroq, turli xil ma'lumotlar bilan ishlashning eng keng tarqalgan varianti hisoblanadi. Birtomondan, ko'p foydalanuvchilar, ayniqsa, halii hisobot standart qog'oz shaklidan voz kechishga kuch topolmaydi. Boshqatmondan, hisobot, asosan, turli xil xususiyatlarga ega bo'lgan ma'lumotlarni birlashtiruvchi vosita.

Hisobot berish funksiyalarini ta'minlashtirib ko'pincha zamonaviy dasturiy mahsulotlarni ishlab chiqishda paydobo'ladi. Data beacon Toolkit sifatida dastur plugin-smaylik va hisobot qo'llab-quvvatlash vositalari va komponentalari mavjud.

Shunga o'xshash vositalar Microsoft Visual Studio .net (VS.NET)dir. Microsoft Visual Studio .net (CRVS) uchun Crystal Reports deb nomlangan, biznes ob'ektlarini ishlab chiqish va litsenziyalash sxemasini o'yicha etkazib berishga yo'naltirilgan. Yanabir vositasi, Microsoft hisobot xizmatlari (MSRS), Microsoft SQL Server DBMS asoslangan platforma hisoblanadi.

Crystal Reports for Microsoft Visual Studio .NET (Hisobotlarni qurish instrumenti). Crystal Decisions mahsulotlarini Microsoft development tools bilan bog'lashtirishi 1993-yil gaborib taqaladi. O'shanda Crystal Decisions hisobotlarning tarkibiy qismlari avval Visual Basic 3.0 data qidimetilgan edi. 1996 Crystal hisobotlari dasturi Visual Studio

development tools paketining birqismigakiritildi va 2002 dauning birqismibo'ldi VS.NET va. net Framework, rasmiynomi CRVSqabul qilindi. Crystal Reports komponentlarining paydobo'lishi va ularni o'zarori vojlangan varinatlarima vujud. Masalan, VS.NET 2003 VS.NET 2005.

CRVSkomponentlarining asosiy maqsadidasturning hisobot funksiyalarini ishlab chiqishni qo'llab-quvvatlashdir.

Bumaxsus hisobot shablon talablariga muvofiq ishlab chiqilgan bo'ladi.

Hisobot sahifasidama'lumotlarni joylashtirishni belgilovchi riptkengaytmasidan foydalanib, shablonda qaysima'lumotlartuzilmalari, hisobotma'lumotlari uchun manbabo'lib xizmat qilishiko'rsatilgan bo'lishi kerak.

Keyinchalik shablon manbada ma'lumotlarni chiqarib, yakuniy hisobotning tuzilishini shakllantirish uchun foydalanadigan CRVSkomponentlarini kiritishga o'tiladi.

Hisobotni ko'rsatish uchun ishlab chiqilayotgan dasturga ko'shilgan maxsus funksiyalardan foydalanib web va Windows ilovalarini ham qo'llab-quvvatlanadi.

Aslida,

VS.NET da ekspert jarayoni kodibir nechasatrlarni chaqirib kamayadi, lekin " hisobot dizayner " yordamida amalga oshiriladi. Shunday qilib, hisobot shablonini yaratish uchun Microsoft.

net platformasida dasturlashingiz shart emas (MS.NET). xohlasangiz, uni o'zgartirish imkonini beruvchi,

dastur hisobot shablonni o'zichiga olish im mumkin. Bujuda qulay, chunki bu holda, hard oim ishlab chiqilgan dasturni qo'llab-quvvatlash uchun amalga oshirish im mumkin.

Yuqorida aytib o'tilganidek,

CRVSkomponentlarini vebga yo'naltirilgan va muntazam Windows ilovalarini ishlab chiqish uchun ishlatilish im mumkin.

Ma'lumotlar bilan ishlash nuqtainazaridan Crystal hisobotlarkomponentlari ikki rejimda ishlash - surish modeli (Push model) va tortish modelida (Pull model) ishlatilish im mumkin.

Surish modeli - oraliq ma'lumotlartuzilmasi bilan ishlash modeli.

Shu bilan birga hisobotlar uchun ma'lumotlar odatiyl takrorlanish ko'rinishida hosil qilinadi .NET Framework - ma'lumotlar bazasi deb ataladigan ma'lumotlartuzilmasidir.

Dastur hisobotlarni qurish uchun kiritish ma'lumotlarni tayyorlash barchama ntiqamallarga bog'liq. Bu qulay,

chunki ma'lumotlar faqat ma'lumotlar manbaiga bevositakirish yo'libil anem

as, balkimutlaqobiron-birtarzdahosilqilinishimumkin. Biroq, agarhisobotyetarlichakattabo'lsa, judanoqulaytuzilmanito'ldirishingizkerak.

Tortishmodeli - talabbo'yichama'lumotlarniso'rash, ya'nima'lumotlarbazasidanolish. Buholda, Crystal Reports komponentlarigao'tganhisobotshablonmanbadanma'lumotlarniolishbilan bog'liqishlarniamalgaoshiradi.

Ushbuyondashuvko'psahifalihisobotlarninamoyishqilishuchunjudasamar ali, chunkiko'rsatilgansahifalaruchunfaqatma'lumotlarmanbadanolinadi.

Hisobotniishlabchiqarishvaaksettirishdantashqari, CRVSkomponentlariolinganma'lumotlarniPDF, HTML, XLS, RTFvaDOC formatlarigaeksportqilishniqo'llab-quvvatlaydi. Buholda, dasturfayltizimikatalogdabevositazarurhisobotfayllarniyaratishmumkin. Buhisobotfayllarniyaratishningommaviyrejimiuchunjudaqulay, foydalanuvchikeyinchaliknatijalarnimanfaatdortomonlargauzatishuchun ularnioldindanko'rmasdanhisobotlarguruhiniyaratishkerak.

CRVSningafzalliklarivakamchiliklariniqisqachaeslatibo'tamiz. Shubhasizafzalligi. net Framework vahamyaqinintegratsiyabo'ladiVS.NET, qaysiilovatuzuvchirivojlanishvositalaribilanhisobotyaratishimkoninibera di. Crvskomponentlarigaasoslanganechimlar, shuningdek, buxususiyathisobotlarva Microsoft .NET. arxitekturasidajoylashganligisababli, murakkabvainteraktivhisobotlarniamalgaoshirishqiyinemas, chunkiularniinteraktivnazoratelementlarikabi "kublar" CRVSDamavjud.

Ushbumahsulot Microsoft texnologiyalarioldingiavlod (u Win32 va C++yordamidaishlabchiqilgan) bo'yichabutlovchiyadrodayashirinamalgaoshirishgamo'ljallangan, uningishlabchiqaruvchihisoblanadi. Dinamikkutubxonalarko'rsatkichlarnio'rnatishvadasturiyjudaoddiyeksportthisobotlarniyordamidaamalgaoshiriladi. Shubilanbirga, yadroishlaydiganbundayplatforma .netvaan'anaviyxotiraalgoritmdinamikiyig'ish, raqobatkabinostandartinfratuzilmamuummolarilibkelishimumkin. Bumuummonihalqilishyo'lidaharakatqilib, ishlabchiqaruvchi 1.1 versiyasiqismisifatidaerkinfoydalanishgaberilganva .netplatformabosqichma-bosqicho'tishniboshlaganVS.NET 2003). Buo'zgartirish Crystal hisobotlarkomponentlarinifoydalanadivadasturbilanfaqat.

netma'lumotlarni ekspert qilib beruvchi tomonidan tarqatish jarayonini soddalashtirish kerak. CRVS ning platforma ga to'liq o'tish kutilmoqda.

CRVS komponentlarining arxitekturasini yakuniy hisobotish labchi qari sh jarayonlarini ko'p darajali taqsimlashning ssenariysini tashkil etishga imkon berishiga qaramasdan, ushbu mahsulot butun hisobotni boshqarish siklini tashkil qilish uchun mos emas.

MSRS platformasi buni muammoni hal qilish uchun mo'ljallangan vatayyorekspert fayllarini hosil qilishga qaratilgan.

Testlash va instruktsiya yozish usullari. Dasturiy ta'minot testi - nuqsonlarni aniqlash va mahsulot sifatini yaxshilash maqsadida dasturiy ta'minotni vatagishli hujjatlarni tahlil qilish jarayonidir.

Dasturiy ta'minot ishlabchi qisho'nyillarda davomida, turlicha yo'llar bilan sinov sifatiga qafolatni o'rnatish qalijudaya qinlashib qilinadi. Uning bir necha asosiy "sinov davrlari" bor.

O'tgan asrning 50-60-yillarida sinov jarayoni keng rasmiylashtirilgan, bevositadasturiy ta'minot ishlabchi qish jarayonidan ajralgan va "matematik lashtirilgan" edi. Aslida, test ko'proq disk raskad rovdasturlariga kabi bo'lgan. "Mukammal testing 4"-deb talimish birtushuncha bo'ldiki, barcha mumkin bo'lgan kiritish ma'lumotlar bilan kodni ijrosi barcha mumkin bo'lgan yo'llarini tekshirish amalga oshiriladi. Kiritish ma'lumotlar soni juda katta, chunki, mukammal sinov mumkin emas, va buyondashuv bilan hujjatlardagi muammolarni topish qiyin bo'lgan.

70 yillari sinov kichik asosiy g'oyalarasosida bo'lgan: birinchi muayyan belgilangan sharoitlarda (ijobiy testing) dastur faoliyatini isbotlash jarayoni sifatida qabul qilindi va keyin - qat'iy qarama-qarshi: dastur muayyan belgilangan sharoitlarda (salbiy testing) inoperable kanligini isbotlash jarayoni sifatida qaraladi. Nafaqat bu ichki ziddiyat vaqt davomidagi oyib bo'lmaydi, lekin bugungi kunda ko'pmualliflar juda to'g'risinov kichik to'ldiruvchi maqsad bo'lgan deb hisoblaydi.

U aniqlangan muammolarga ko'zi yumishga ruxsat bermaydi.

80-yillarda, dasturiy ta'minot ishlabchi qish sinov joyida asosiy o'zgarish yuz berdi: loyihayaratish yakuniy bosqichlarida biri, test butun rivojlanish sikli davomida tatbiq etildi (dasturiy lifecycle).

Bundantashqari,
oldindanvaularningyuzagakeladiganvaziyatlarnioldiniolishuchunemas,
balkifaqatko‘phollardaruxsatberilgan
"dasturiyta‘minotishlabchiqishmodellari"
iterativortibdasturiyta‘minotishlabchiqishmodelitavsifiniko‘rishmaqsadg
amuvofiq.

90-yillarda,
testlashjarayonibutundasturiyta‘minotishlabchiqishsiklio‘zichigaoladiva
mavjudsinovhollardavasinovmuhitlarqo‘llab-quvvatlash, rejalashtirish,
loyihalash, yaratishvasinovishlarniamalgaoshirishjarayonlarinita‘sir
"sifatkafolati"
debnomlanganyanadakengqamrovlijarayonikabisinovo‘tishboredi.
Sinovsifatjihatidanyangidarajagaetdi,
butabiiyravishdauslubiyotningyanadarivojlanishiga,
sinovjarayoniniboshqarishhuchunjudakuchlivositalarvahozirgiavlodlarigaj
udao‘xshashbo‘lgansinovlarniavtomatlashtirishvositalarinipaydobo‘lishi
gaolibkeldi.

Testlashningboshidaharqandaymutaxassis (tester
hambundanmustasnoemas) ijrochivatalaba.
Sinovholatlarivanuqsonhisobotlariqandayekanliginiyaxshitushunish,
talablarnio‘qiyolish,
birnechtavositlardanfoydalanishvajamoadayaxshiishtiroketishkifoya.
Asta-
sekinsinaluvchiloyihaniishlabchiqishningbarchabosqichlarigakirishboshl
anadi, buesao‘znavbatidaularniyanadato‘laroqanglatadi,
ulardannafaqatfaolfoydalanaboshlaydi,
balkiloyihahujjatlariniishlabchiqib,
yanadamas‘uliyatliqarorlarqabulqilaboshlaydi.

Testlashningasosiymaqsadinimajoziyma‘nodaifodallasangiz,
shundaybo‘ladi: loyihahozirgivaqtdanimagamuhtojligini,
loyihabuzarurnio‘zo‘lchovidaqabulqiladimi, yo‘qmi,
qandayqilibvaziyatniyaxshiroqo‘zgartirishkerakliginitushuntiribberish.
Loyihamenejeriningmaqsadigao‘xshaydi, to‘g‘rimi? Sogi-hidori.
huMuayyanrivojlanishdarajasidanboshlab, ITmutaxassislari,
faqattexnikko‘nikmalarto‘plamlaridavabuko‘nikmalarniqo‘llashningasos
iysohasidafarqqiladi.

Sinovchisifatidaishlashnimuvaffaqiyatliboshlashhuchunqandaytexni
kko‘nikmalarkerak?

1. Chettillarko'nikmalari. Ha, butexnikbo'lmaganmahorat. Buniaksiomadebhisoblashingiz mumkin.

2. .Haqiqiy rivojlangan foydalanuvchi darajasida ishonchlik kompyuter qobiliyatlariga busohada oimorivojlanish kuzatib borish.

3. Dasturlashtirish.

Butartib dabinchinavbatdaha r qanday IT shaxs vada sturiy ta'minot tester hayotini osonlashtiradi. Dasturlash bilimisiz test qilish mumkinmi? Ha, mumkin. Bu, albatta, yaxshi amalga oshirilishi mumkin.

4. Ma'lumotlar bazalariga SQL tili.

Buyer da testlovchi ham torma taxassislar darajasida malakaga ega bo'lish talab etilmaydi, lekin eng ko'ptar qalgan DBMLar bilan ishlashning minimal ko'nikmalariga oddiy so'rovlarni yozish qobiliyati majburiy deb hisoblanishi mumkin.

5. Tarmoqlar va operatsion tizimlar qanday ishlashi haqida tushuncha. Hech bo'lmasa, muammoning tashxislash va iloj bo'lsa, uni o'zingiz hal qilish imkonini beruvchi minimal darajada.

6. Veb-ilovalar va mobil ilovalar tamoyillari haqida tushuncha. Shukunlarda deyarli hammanarsa bunday ilovalar shaklida yoziladivategishl itexnologiyalarni tushunish samarali sinov uchun majburiy bo'ladi.

Xulosasi sifatida testlovchi gatezda ajoyib torma taxassis bo'lish imkonini beruvchi shaxsiy sifatlarini ham qaydetamiz:

1) mas'uliyat va ishlashning o'rtishi;

2) yaxshi mulohaza qot qobiliyatlari, o'z fikrlarini aniq, tez, aniq ifodalash qobiliyati;

3) sabr-toqat, qat'iyat, tafsilotga e'tibor, kuzatish;

4) yaxshi mavhum va analitik fikrlash;

5) nostandart tajribalar o'rnatish qobiliyati, tadqiqot uchun imkoniyat.

Bu fazilatlar gategalik qiladigan odamni topish qiyin, ammo o'z-o'zini rivojlantirish uchun ma'lum birmos yozuvlarni uqtasiga ega bo'lish hard oim foydalidir.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1.



- Insonvamashinaningo‘zaromuloqotinita‘minlashnimadebaytiladi.
2. Inson kompyutero‘zarohamkorligiilmiyko‘pyo‘nalishlarinisanabbering.
 3. Foydalanuvchiinterfeysinima?
 4. Realvaqtdaamalgaoshiriladiganvamuayyanvazifanibirgalikdahaleti shgaqaratilganaxborotalmashuvinimadebaytiladi.
 5. Kiritishvositalariyordamidahosilqilinadigankirishxabarlariningtexnikvositalarinisanibbering.
 6. Interfeysyaratishningasosiytamoyillarisananabbering.
 7. Ma’lumotlarnieksportqilishvositalarivakomponentalaridegandanim anazardatutiladi.
 8. Turlixilxususiyatlargaegabo‘lganma’lumotlarnibirlashtiruvchivosita – bunima.
 9. Crystal Reports for Microsoft Visual Studio .NET tizimihaqidanimalarnibilasiz.
 10. Surishmodeliningvazifasi
 11. Tortishmodeliningvazifasinimadaniborat.
 12. Testlashdegandanimatushuniladi.
 13. Testlashningrivojlanishbosqichlarinisanabbering.
 14. Ikkiyo‘nalishlitestlashjarayoniqaysibosqichgategishli.
 15. Zamonaviytestlashningqulayliklarivakamchiliklari.
 16. Sinovchixodimbo‘lishningxususiyatlarisanabbering?

Foydalanilgan adabiyotlar

1. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. S va C++ tili. “Voris- nashriyot” MCHJ, Toshkent 2013, 488 b.
2. Horstmann, Cay S. C++ for everyone/Cay S. Horstmann. Printed in the United States of America - 2nd ed. 2010. – P. 562.
3. Horton I.-Beginning Visual C++ 2012/ I.Horton. Published simultaneously in Canada. –2012. –P. 988.
4. Алгоритмы. Справочник с примерами на C, C++, Java и Python. Джордж Хайнемай, Гэри Поллис, Стэнли Селков. Москва «Альфа-Книга», 2017. –С254.
5. ASP.NETMVCFramework с примерами на C# для профессионалов. Стивен Сандерсон. Москва 2010. –С. 562.
6. Самоучитель Microsoft Visual Studio C++ и MFC. Сидорина Т.Л. Санкт-Петербург. «БХВ-Петербург» 2009. –С. 600.
7. Си Шарп: Создание приложения для Windows. В.В. Лабор. Минск, Харвест, 2003.
8. Полный справочник по C++. Герберт Шилдт. Москва, 2006.
9. Современные технологии разработки программ, взаимодействующих с базами данных. С.А. Минеев, Ю.Е. Чуманкин. Нежный Новгород, 2018.
10. C/C++. Программирование на языке высокого уровня. Т.А.Павловская. Санкт-Петербург, 2003.
11. Самоучитель C++. Герберт Шилдт. Санкт-Петербург. 2003.
12. C/C++ и MSVisualC++ 2010 для начинающих. Санкт-Петербург. 2011.
13. Программирование под Windows в среде VisualC++ 2005. М.В. Свиркин, А.С. Чуркин. Санкт-Петербург. 2008.
14. Простая графическая задача на MicrosoftVisualC++ с использованием библиотеки MFC. Е.П. Дербакова. Москва 2015.
15. Программирование в VisualC++ с использованием библиотеки MFC. В.В. Васильчиков. Ярославль 2006.
16. MSVisualC++ 2010 в среде .NET. В.В. Зиборов. Санкт-Петербург, 2012
17. Herbert Schildt, Java the complete reference ninth edition, oracle press, 2014.
18. Алексеев А.П. Информатика. 2001. М., СОЛОН-Р, 2001, 364 с.
19. Арипов М.М. Internet ва электрон почта асослари. –Т.: Университет, 2000. -132 б.

20. Страуструп. Б. Язык программирования C++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.
21. Вирт Н. Алгоритмы + структуры данных = программа.- М.:Мир,1985.-405с.
22. Говорухин В., Цибулин В. Компьютер в математическом исследовании. Maple, Matlab, Latex и др. Учебный курс. Питер. 2001. – 624 с.
23. Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург, 2001. под редакцией С.В.Симоновича.
24. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с
25. Павловская Т.С. Щупак Ю.С. C++. Объектно-ориентированное программирование. Практикум.- СПб.: Питер,2005-265с
26. Подбельский В.В. Язык СИ++.- М.; Финансы и статистика-2003 562с.
27. Романчик В.С., Люлькин А.Е. Программирование в C++ BUILDER. Учебное пособие. Мн.: БГУ, 2007. –126 с.
28. Юлдашев У.Ю., Бокиев Р.Р., Зокирова Ф.М. Информатика. – Т.: F.Фулом номидаги нашриёт-матбаа ижодий уйи, 2002. - 237 б.
29. Abduqodirov A.A., Hayitov A.G', Shodiyev R.R. Axborot texnologiyalari // Akademik litsey va kasb hunar kollejlari uchun darslik. -Т.: O'qituvchi, 2003. –152 b.
30. Axmedov A.B., Tayloqov N.I. Informatika. Akademik litsey va kasb hunar kollejlari uchun darslik.-Т.: O'zbekiston, 2001. 2 – nashri. – 272 b.
31. Mo'minov B.B. Microsoft Excel bo'yicha amaliy mashg'ulotlar va ularni bajarish tartibi. Uslubiy qo'llanma. – Buxoro: Ziyo Rizograf, 2008. – 72 b.
32. Mo'minov B.B. Microsoft Excel dasturi bo'yicha laboratoriya ishlari to'plami. Uslubiy qo'llanma. – Buxoro: Ziyo Rizograf, 2008. – 92 b.
33. Mo'minov B.B. Pedagogik dasturiy ta'minot yaratish texnologiyasi. Monografiya. –Buxoro, Buxoro nariyoti, 2010. -168 b.
34. Xaldjigitov A.A., Madraximov Sh.F., Adamboev U.E. Informatika va programmalash. O'quv qo'llanma. O'zMU, 2005 yil, 145 bet.
35. <http://aut.researchgateway.ac.nz/index.jsp> Auckland University of Technology digital library

36. <http://dastur.uz> Kompyuter Dasturlari va Kompyuterda Dasturlashga oid Forum, Habar va Yangiliklar
37. <http://google.com> Google qidiruv tizimi
38. <http://informatika.sch880.ru> Основы логики
39. <http://lex.uz/> O‘zbekiston Respublikasi qonun hujjatlari ma’lumotlari milliy bazasi
40. <http://ru.wikipedia.org/wiki> Википедия — свободная энциклопедия
41. <http://uza.uz> O‘zbekiston milliy axborot agentligi
42. <http://winedt.com> winedt rasmiy portali
43. <http://www.aci.uz> O‘zbekiston aloqa va axborotlashtirish agentligi
44. <http://www.cplusplus.com> cplusplus.com
45. <http://www.edu.uz> O‘zbekiston Respublikasi Oliy va O‘rta maxsus ta’lim vazirligi
46. <http://www.exponenta.ru> Образовательный математический сайт Exponenta.ru
47. <http://www.intuit.ru> Интернет-Университет Информационных Технологий
48. <http://ziyonet.uz> Oliy va O‘rta maxsus ta’lim vazirligi huzuridagi axborot ta’lim portali

MUNDARIJA

KIRISH.....	3
1-BOB. KONTEYNERLAR	4
1.1. Noma'lum tiplar va noma'lum nomlar fazosi.	4
1.2. Konteynerlar (Kollektsiyalar).	67
1.3. Assosiativ va tartiblanmagan assosiativ konteynerlar	98
1.4. Konteynerlarning adapterlari.	160
2-BOB. DASTURLASH TILINING TAKOMILLASHTIRILGAN	
IMKONIYATLARI	219
2.1. Standart algoritmlar va iteratorlar.	219
2.2. Sonli sinflar bilan ishlash	266
2.3. Sintaktik tahlil	313
2.4. Murakkabsaralash algoritmlari. Amaliy dasturlash	355
3-BOB. VISUAL C++ MUHITIDA DASTURLASH.	434
3.1. Visual C++ muhitida dasturlash.	434
3.2. Komponentalar bilan ishlash.	476
3.3. Muloqot oynalari bilan ishlash.	506
3.4. Visual C++ning grafik imkoniyatlari.	538
3.5. OLE, MFC texnologiyalari. Ko'p oynali muhitlar bilan ishlash.	603
3.6. Kichik loyihalarni yaratish	616
3.7. Foydalanuvchi interfeysini loyihalash.	632
FOYDALANILGAN ADABIYOTLAR	645

MO'MINOV Bahodir Boltaeyich

e-mail: Mbbahodir@gmail.com

DASTURLASH II

DARSLIK

O'zbek tilida

Toshkent – «NIHOL PRINT» OK – 2021

Muharrir:	A.Tog'ayev
Tex. muharrir:	F.Tog'ayeva
Musavvir:	B.Esanov
Musahhiha:	O.Muxammadiyeva
Kompyuterda sahifalovchi:	G.Tog'ayeva

9323



№ 7439-765f-47f1-7eal-a683-4648-1314.

Bosishga ruxsat etildi: . Bichimi 60x841 /16.

Shartli bosma tabog'i 38,00. Nashr bosma tabog'i 37,75.

Adadi 100. Buyurtma № 54.

«Nihol print» Ok da chop etildi.

Toshkent sh., M. Ashrafiy ko'chasi, 99/101.