

O'ZBEKISTON RESPUBLIKASI
OLIV VA O'RTA MAXSUS TA'LIM VAZIRLIGI

QARSHI DAVLAT UNIVERSITETI
FIZIKA –MATEMATIKA FAKULTETI
“AMALIY MATEMATIKA VA INFORMATIKA” KAFEDRASI

“5130200” – Amaliy matematika va informatika” ta’lim yo’nalishi bo’yicha
bakalavr darajasini olish uchun

Negmatova Aziza Komilovnaning

**« C++ tilining kutubxonalari va ularning
programma tuzishdagi ahamiyati»**

mavzusida yozgan

BITIRUV MALAKAVIY ISHI

Ilmiy rahbar:

o’q. F.Qlicheva

“Himoya qilishga ruxsat beraman”
Fizika-matematika fakulteti dekani
_____ prof. A.Tashatov

“ ___ ” _____ 2015-y.

QARSHI-2015-yil

MUNDARIJA:

Kirish	4
I BOB. C++TILI HAQIDA BOSHLANG'ICH TUSHUNCHA.....	6
1.1. C++ tili tarixi.....	6
1.2. C++ tilidagi programma tuzilishi va uning kompilyatsiyasi	9
1.3.C++ tilida obyektga mo'ljallangan yondashuv.....	12
1.4. Konstruktorlar va destruktorelar	20
II BOB. C VA C++ TILIDA MAVJUD KUTUBXONALAR RO'YXATI.....	26
2.1. Ma'lumotlarni kiritish va chiqarish va akslantirish funksiyalari bilan	27
ishlovchi kutubxonalar	27
2.2. Satrlar va simvollar bilan ishlovchi kutubxonalar	33
2.3. Matematik funksiyalar bilan ishlovchi kutubxonalar	41
2.4. Vaqt va sana bilan bog'liq kutubxonalar	43
III BOB. DASTURIY VOSITALAR QURISHDA C++ TILI KUTUBXONALARIDAN FOYDALANISH	46
3.1. Belgilar va satrlar bilan ishlovchi kutubxonalardan foydalanish.....	46
3.2. Dasturlashda math.h kutubxonasining imkoniyatlaridan foydalanish.....	48
3.3. Dasturiy vositalarda vaqt va sanaga bog'liq funksiyalarni qo'llash.....	50
XULOSA	56
FOYDALANILGAN ADABIYOTLAR VA INTERNET RESURSLARI.....	57

Kirish

“Islohat-islohat uchun emas, avvalo inson uchun, uning farovon hayoti uchun xizmat qilishi kerak”.

I.A. KARIMOV.

Tadqiqot mavzusining dolzarbligi: Obyektga yo'naltirilgan dasturlash tillaridan biri bo'lgan C++ tilining imkoniyatlarini batafsil o'rganish raqobatbardosh dasturiy vositalar qurish imkoniyatini yaratadi. Buni hisobga olgan holda ushbu bitiruv malakaviy ishida C++ tilida mavjud kutubxonalar va ulardagi imkoniyatlarni o'rganish masalasi qo'yilgan. Bu til yetarlicha boy kutubxonaga ega bo'lib, ular yordamida dasturiy vositalar qurish ancha oson kechadi. Shu jihatdan ham ishda qo'yilgan masala hozirgi kundagi muhim ahamiyat kasb etuvchi masalalardan hisoblanadi.

Ishning maqsadi: C++ tilining amaliy dasturlar qurishda keng qo'llaniladigan kutubxonalarini o'rganish va ularni dasturiy vositalar qurishga tatbiq etish.

Ishda qo'yilgan vazifalar:

- C++ tilining imkoniyatlarini batafsil o'rganib chiqish va ishda yoritish;
- C va C++ tillarida mavjud kutubxonalar va ulardagi imkoniyatlarni ;
- Amaliy dasturlar qurishda C++ tili kutubxonalaridan foydalanishni yo'lga qo'yish.

Mavzuning o'rganilish darajasining qiyosiy tahlili: Ishda qo'yilgan masala o'zining keng ko'lamlil amaliy qo'llanmalariga ega bo'lganligi uchun bir qator yetuk olimlar bu kabi masalalar bilan shug'ullanishgan. Hozirgi kunda ham ilmiy taqiqotlar olib borilmoqda. Bu sohada ish olib borgan olimlardan bir qanchasini keltirib o'tish mumkin. Masalan, G.Shildt,

R.Sedjvik, Sh.F.Madraximov, S.M.Gaynazarov, M.E.Abramyan, A.A.Xoldjigitov, U.E.Adambayev va boshqalar.

Olimlardan G. Shildt C ++ tilidagi kutubxonalarini guruhlarga ajratib, o'zining kitobida batafsil bayon etgan.

Tadqiqotning ilmiy yangiliklari: -Amaliy dasturlar qurishda C++ tilining <ctype.h>, <string.h>, <math.h> va <time.h> kutubxonalaridan samarali foydalanishni yo'lga qo'yish.

Tadqiqotning predmeti va obyekti: C++ tilida mavjud imkoniyatlardan foydalanib, dasturiy vositalar quriladigan turli soha masalalari.

Tadqiqotning ilmiy ahamiyati:

- mavzu bo'yicha ilmiy-uslubiy, nazariy adabiyotlarni o'rganish;
- ta'lim to'g'risida davlat hujjatlari, DTS talablari, ilg'or mutaxassis olimlarning fikrlarini o'rganish;
- mavzuga aloqadar mavjud internet resurslaridan foydalanish.
- C++ tilining mavjud imkoniyatlardan faydalanish.

Bitiruv malakaviy ishning tarkibi va tuzilishi: Bitiruv malakaviy ishi kirish, uchta bob, xulosa, foydalanilgan adabiyotlar hamda internet resurslar ro'yxatidan iborat.

I BOB. C++TILI HAQIDA BOSHLANG'ICH TUSHUNCHA

1.1. C++ tili tarixi

C++ dasturlash tili C tiliga asoslangan. C esa o'z navbatida B va BCPL tillaridan kelib chiqqan. BCPL 1967- yilda Martin Richards tomonidan tuzilgan va operatsion sistemalarni yozish uchun mo'ljallangan edi. Ken Thompson o'zining B tilida BCPL ning ko'p xossalarini kiritgan va B da UNIX operatsion sistemasining birinchi versiyalarini yozgan. BCPL ham, B ham tipsiz til bo'lgan. Ya'ni o'zgaruvchilarning ma'lum bir tipi bo'lmagan - har bir o'zgaruvchi kompyuter xotirasida faqat bir bayt yer egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun sonmi, kasrli sonmi yoki harfdekmi, dasturchi vazifasi bo'lgan [3,5].

C tilini Dennis Ritchie B dan keltirib chiqardi va uni 1972 yili ilk bor Bell Laboratoriyasida, DEC PDP-11 kompyuterida qo'lladi. C o'zidan oldingi B va BCPL tillarining juda ko'p muhim tomonlarini o'z ichiga olish bilan bir qatorda o'zgaruvchilarni tiplashtirdi va bir qator boshqa yangiliklarni kiritdi. Boshlanishda C asosan UNIX sistemalarida keng tarqaldi. Hozirda operatsion sistemalarning asosiy qismi C/C++ da yozilmoqda. C mashina arxitekturasiga bog'langan tildir. Lekin yaxshi rejalashtirish orqali dasturlarni turli kompyuter platformalarida ishlaydigan qilsa bo'ladi.

1983 yilda, C tili keng tarqalganligi sababli, uni standartlash harakati boshlandi. Buning uchun Amerika Milliy Standartlar Komiteti (ANSI) qoshida X3J11 texnik kommitet tuzildi va 1989 yilda ushbu standart qabul qilindi. Standartni dunyo bo'yicha keng tarqatish maqsadida 1990 yilda ANSI va Dunyo Standartlar Tashkiloti (ISO) hamkorlikda C ning ANSI/ISO 9899:1990 standartini qabul qilishdi. Shu sababli C da yozilgan dasturlar kam miqdordagi o'zgarishlar yoki umuman o'zgarishlarsiz juda ko'p kompyuter platformalarida ishlaydi.

C++ 1980 yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan tarzda tuzildi. C++ juda ko'p qo'shimchalarni o'z ichiga olgan, lekin eng asosiysi u obyektlar bilan dasturlashga imkon beradi.

Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoqda. Buni ta'minlash uchun obyektli dasturlash g'oyasi ilgari surildi. Xuddi 70-yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi obyektlar orqali tuzish dasturlash sohasida inqilob qildi.

Dasturlash tillari bilan shug'ullanadigan dasturchilar ba'zi bir terminlar bilan juda ko'p to'qnash kelishadi. C++ dasturlash tili ham bundan mustasno emas. Oldin shu haqida, undan so'ng esa C++ ni ishlatish uchun nimalar kerakligini ko'rib chiqamiz.

Demak terminlarni ko'rib chiqamiz:

Kompilyator – bu o'zgartirish degan ma'noni beradi. Ya'ni dasturlash tilida yozilgan dastur (C++ bo'lsa, *.c, *.cpp)ni kompyuter tushunadigan tilga o'zgartirib, uni ishlashini ta'minlaydi. Bu degani dastur kompyuterda to'liq ishlaydi. Bundan ko'rinib turibdiki, C++ da dastur tuzish uchun kompilyator o'rnatish zarur. Linux operatsion tizimida kompilyator o'rnatilgan bo'ladi (GCC, G++).

Kompilyatsiya – o'zgaruvchi jarayon, ya'ni yuqori pog'onali dastur kodlari (misol uchun C++ da tuzilgan kod)ni quyi pog'onali ishlovchi kodga aylantirish jarayoni.

Komponovshik (linker) – ishlovchi modul yaratish. Projektda ishlatiladigan obyekt fayllarni to'plab yagona ishlovchi modulga aytiladi.

IDE (integrallashgan ishlovchi muhit) – Har bir dasturlash tilida o'z ishchi muhiti bo'ladi. Bu muhitda kompilyator bilan birga kod yozishga mo'ljallangan joy bo'ladi. Bu ishchi muhitda bir necha dastur kodlarini ham yozsa bo'ladi va bu muhit avtomat xatolarni ham ko'rsatib beradi. Bu ishchi muhitda ma'lum kutubxonalar joylashgan bo'lib, kod yozishda kodni to'g'ri tashkil qilish uchun ma'lum bir yordamlar ekranga chiqadi. Ishchi muhitga oddiy bloknot misol bo'ladi, lekin, bu muhitning kompilyatori yo'q, shuning uchun alohida kompilyator o'rnatiladi. Bunda 2 ta dastur o'rnatib bir biriga moslashtirish kerak bo'ladi. Bu 2 ishni yengillashtirish uchun, juda ko'p ishchi muhit dasturlari yaratilgan va ular kompilyatorlar bilan integrallashgan bo'ladi.

C++ dasturlash muhitida eng keng tarqalgan IDE bu – **Visual C++** hisoblanadi. Bu dastur o‘z ichiga dastur kodlarini yozishda juda muhim bo‘lgan .NET klasslari, ishchi muhit va albatta kompilyator integrallashtirilgan bo‘ladi.

Quyida hozirgi zamonda eng ommabop bo‘lgan C++ muhitlari va kompilyatorlari ro‘yxatini keltirib o‘taman.

Dev-C++ - C/C++ dasturlash tillari uchun ochiq integrallashgan dasturlash muhiti. Bu distributivga MinGW nomli kompilyator kiradi. Bu dastur o‘zi Delphi dasturlash tilida tuzilgan. GPL litsenziyasi orqali tarqatiladi. Bir paytlar Linux versiyasi ham mavjud edi, hozir faqat Windows. Hozir yangi versiyalari ishlab chiqilmayapti, uning o‘rniga wxDev-C++ nomi bilan ishlab chiqilmoqda.

C++ Builder – C++ dasturlash tili uchun mo‘ljallangan integrallashgan dasturlash muhiti hisoblanadi. Dastlab Borland Software korxonasi tomonidan ishlab chiqilgan, undan so‘ng uning bir qismi bo‘lgan CodeGear tomonidan. C++ BuilderSTL, VCL, CLX, MFC obyekt kutubxonalarini o‘z ichiga oladi. Delphi dasturlash tili tomonidan ishlab chiqilgan. Bu muhit drag-and-drop harakati bilan vizual shaklni paydo qilishi mumkin, undan tashqari WYSIWYG redaktori ham mavjud.

Borland C++ - C/C++ dasturlash tillarida dastur tuzish uchun mo‘ljallangan muhit bo‘lib, Borland firmasi tomonidan ishlab chiqilgan. Muhitning har bir versiyasi ma'lum bir dasturlash tili standarti asosida ishlovchi kompilyator bilan integrallashgan holda chiqarilgan. Dastlab, DOS operatsion tizimi muhitida ishlovchi dasturlarni yaratish uchun, keyinchalik Windows muhitida ishlovchi dasturlarni yaratish uchun mo‘ljallangan versiyada ishlab chiqildi.

Microsoft Visual C++ - Microsoft firmasi shu muhitni yaratdi. C++ dasturlash tili uchun mo‘ljallangan va Microsoft Visual Studio ni bir qismi sifatida yoki o‘zi yakka Visual C++ Express Edition shaklida namoyon bo‘ldi. Microsoft QuickC muhitini almashtirish vazifasini bajardi.

GNU Compiler Collection(GCC) – GNU proyeksi asosida yaratilgan, har xil turdagi dasturlash tillarining kompilyatorlar yig‘indisi hisoblanadi. Bu dastur ochiq hisoblanadi va dastlab faqat S uchun yaratilgan edi. Keyinchalik

kuchayib C++, Objective-C, Java, Fortran va Ada dasturlash tillari uchun ham ishlab chiqildi.

C++ dan tashqari boshqa ko'p obyektli dasturlashga yo'naltirilgan tillar paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xeroxning Palo Altoda joylashgan ilmiy-qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalk da hamma narsa obyektarga asoslangan. C++ esa gibril tildir. Unda C ga o'xshab strukturali dasturlash yoki yangicha, obyektlar bilan dasturlash mumkin. Yangicha deyishimiz ham nisbiydir. Obyektli dasturlash falsafasi paydo bo'lganiga ham yigirma yildan oshayapti.

C++ funksiya va obyektning juda boy kutubhonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubhonasidagi tayyor obyekt/funksiyalarni qo'llashni o'rganishdir.

1.2. C++ tilidagi programma tuzilishi va uning kompiyatsiyasi

C++ tilida programma yaratish bir nechta bosqichlardan iborat bo'ladi. Dastlab, matn tahririda (odatda programmalash muhitining tahririda) programma matni teriladi, bu faylning kengaytmasi «.cpp» bo'ladi. Keyingi bosqichda programma matni yozilgan fayl kompilyatorga uzatiladi, agarda programmada xatoliklar bo'lmasa, kompilyator «.obj» kengaytmali obyekt modul faylini hosil qiladi. Oxirgi qadamda komponovka (yig'uvchi) yordamida «.exe» kengaytmali bajariluvchi fayl programma hosil bo'ladi. Bosqichlarda yuzaga keluvchi fayllarning nomlari boshlang'ich matn faylining nomi bilan bir xil bo'ladi [3,5].

C++ tilida matnli fayl shaklida tayyorlangan dastur uchta qayta ishlash bosqichlaridan o'tadi.

Matnni preprocessor direktivalari asosida o'zgartilishi. Bu jarayon natijasi yana matnli fayl bo'lib preprocessor tomonidan bajariladi.

Kompilyatsiya. Bu jarayon natijasi mashina kodiga o'tkazilgan obyektli fayl bo'lib, kompilyator tomonidan bajariladi.

Bog'lash. Bu jarayon natijasi to'la mashina kodiga o'tkazilgan bajariluvchi fayl bo'lib, bog'lagich(komponovshik) tomonidan bajariladi.

Kompilyatsiya jarayonining o'zi ham ikkita bosqichdan tashkil topadi. Boshida preprotssessor ishlaydi, u matndagi kompilyatsiya direktivalarini bajaradi, xususan #include direktivasi bo'yicha ko'rsatilgan kutubxonalaridan C++ tilida yozilgan modullarni programma tarkibiga kiritadi. Shundan so'ng kengaytirilgan programma matni kompilyatorga uzatiladi. Kompilyator o'zi ham programma bo'lib, uning uchun kiruvchi ma'lumot bo'lib, C++ tilida yozilgan programma matni hisoblanadi. Kompilyator programma matnini leksema (atomar) elementlarga ajratadi va uni leksik, keyinchalik sintaktik tahlil qiladi. Leksik tahlil jarayonida u matni leksemalarga ajratish uchun «probel ajratuvchisini» ishlatadi. Probel ajratuvchisiga - probel belgisi (—), '\t' - tabulyatsiya belgisi, '\n' - keyingi qatorga o'tish belgisi, boshqa ajratuvchilar va izohlar (kommentariylar) kiradi.

Programma matni tushunarli bo'lishi uchun izohlar ishlatiladi. Izohlar kompilyator tomonidan «o'tkazib» yuboriladi va ular programma amal qilishiga hech qanday ta'sir qilmaydi.

C++ tilida izohlar ikki ko'rinishda yozilishi mumkin.

Birinchisida "/*" dan boshlanib, "*/" belgalari bilan tugagan barcha belgilar ketma-ketligi izoh hisoblanadi, ikkinchisi «satriy izoh» deb nomlanadi va u "/" belgilardan boshlangan va satr oxirigacha yozilgan belgilar ketma-ketligi bo'ladi. Izohning birinchi ko'rinishida yozilgan izohlar bir necha satr bo'lishi va ulardan keyin C++ operatorlari davom etishi mumkin.

Programmaning 1-satrida #include.. preprotssessor direktivasi bo'lib, programma kodiga oqimli o'qish/yozish funksiyalari va uning o'zgaruvchilari e'loni joylashgan «iostream.h» sarlavha faylini qo'shadi. #include fayl nomi direktivasi dasturning shu direktiva o'rniga qaysi matnli fayllarni qo'shish kerakligini ko'rsatadi.

#include <fayl nomi> direktivasi dasturga kompilyator standart kutubxonalariga mos keluvchi sarlavhali fayllar matnlarini qo'shish uchun mo'ljallangandir. Bu fayllarda funktsiya prototipi, tiplar, o'zgaruvchilar, konstantalar ta'riflari yozilgan bo'ladi. Funktsiya prototipi funktsiya qaytaruvchi tip, funktsiya nomi va funktsiyaga uzatiluvchi tiplardan iborat bo'ladi. Misol uchun, cos funktsiyasi prototipi quyidagicha yozilishi mumkin: double cos(double). Agar funktsiya nomidan oldin void tipi ko'rsatilgan bo'lsa, bu funktsiya hech qanday qiymat qaytarmasligini ko'rsatadi. Shuni ta'kidlash lozimki, bu direktiva dasturga standart biblioteka qo'shilishiga olib kelmaydi. Standart funktsiyalarning kodlari

bog'lash ya'ni aloqalarni tahrirlash bosqichida, kompilyatsiya bosqichidan so'ng amalga oshiriladi.

Keyingi qatorlarda programmaning yagona, asosiy funksiyasi -main() funksiyasi tavsifi keltirilgan. Shuni qayd etish kerakki, C++ programmasida albatta main() funksiyasi bo'lishi shart va programma shu funksiyani bajarish bilan o'z ishini boshlaydi.

Programma tanasida konsol rejimida belgilar ketma-ketligini oqimga chiqarish amali qo'llanilgan. Ma'lumotlarni standart oqimga (ekranga) chiqarish uchun quyidagi format ishlatilgan:

```
cout << <ifoda>;
```

Bu yerda <ifoda> sifatida o'zgaruvchi yoki sintaksisi to'g'ri yozilgan va qandaydir qiymat qabul qiluvchi til ifodasi kelishi mumkin (keyinchalik, burchak qavs ichiga olingan o'zbekcha satr ostini til tarkibiga kirmaydigan tushuncha deb qabul qilish kerak).

Masalan:

```
Int uzg=324;
```

```
Cout<<uzg; //butun son chop etiladi
```

Berilganlarni standart oqimdan (odatda klaviaturadan) o'qish quyidagi formatda amalga oshiriladi:

```
cin>> <o'zgaruvchi>;
```

Bu yerda <o'zgaruvchi> qiymat qabul qiluvchi o'zgaruvchining nomi.

Misol:

```
int Yosh;
```

```
cout << "Yoshingizni kiriting_";
```

```
cin>>Yosh;
```

Butun turdagi Yosh o'zgaruvchisi kiritilgan qiymatni o'zlashtiradi. Kiritilgan qiymatni o'zgaruvchi turiga mos kelishini tekshirish mas'uliyati programma tuzuvchisining zimmasiga yuklanadi.

Bir paytning o'zida probel () vositasida bir nechta va har xil turdagi qiymatlarni oqimdan kiritish mumkin. Qiymat kiritish <enter> tugmasini bosish bilan tugaydi. Agar kiritilgan qiymatlar soni o'zgaruvchilar sonidan ko'p bo'lsa, «ortiqcha» qiymatlar bufer xotirada saqlanib qoladi.

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

```
int main ()
```

```

{
int x,y;
float z;
cin>>x>>y>>z;
cout<< "O'qilgan qiymatlar\n";
cout<<x\t'<<y\t'<<z;
return 0;
}

```

O'zgaruvchilarga qiymat kiritish uchun klaviatura orqali

10 20 3.14 <enter>

harakati amalga oshiriladi. Shuni qayd etish kerakki, oqimga qiymat kiritishda probel ajratuvchi hisoblanadi. Haqiqiy sonning butun va kasr qismlari ' ' belgisi bilan ajratiladi.

1.3.C++ tilida obyektga mo'ljallangan yondashuv

Obyekt — mavhum (abstrakt) mohiyat bo'lib, u bizni o'rab turgan haqiqiy olamning tavsiflariga ega. Obyektlarni yaratish va ular ustida manipulyatsiyalar olib borish C++ tilining qandaydir alohida imtiyozi emas, balki obyektlarning tavsifi va ular ustida o'tkaziladigan operatsiyalarni kodli konstruksiyalarda o'zida mujassamlantiradigan dasturlash metodologiyasi (uslubiyoti) ning natijasidir.

Obyektga mo'ljallangan dasturlash (OMD) — bu dasturchining asosiy diqqatini obyektlarni ishlatish tafsilotlariga emas, balki obyektlar o'rtasidagi aloqalarga qaratadigan metodika.

Har bir tushunchaga aloqador va uning asosini tashkil etuvchilari mavjud bo'lgani kabi C++ tilining obyektga yo'naltirilgan yondashuv qismida sinflar, komponentlar va obyektlar tushunchalari mavjud. Quyida ular haqida to'xtalib o'tamiz.

Sinf jismoniy mohiyatga ega emas, tuzilmaning e'lon qilinishi uning eng yaqin analogiyasidir. Sinf obyektini yaratish uchun qo'llangandagina xotira ajralib chiqadi. Bu jarayon ham sinf nusxasi (class instance) ni yaratish deb ataladi.

C++ tilining har qanday obyektini bir xil atributlarga, shuningdek, ushbu sinfning boshqa obyektlari bilan funktsionallikka ega. O'z sinflarini yaratish hamda ushbu sinflar obyektlarining xulq-atvori uchun to'liq mas'uliyat dasturchi zimmasiga yuklanadi. Biror bir muhitda ishlar ekan, dasturchi standart sinflarning kattagina kutubxonasi (masalan, C++ Builder Vizual Komponentlar Kutubxonasi) ga kirish huquqiga ega bo'ladi.

Odatda, obyekt qandaydir unikal holatda mavjud bo'lib, bu holat obyekt atributlarining joriy qiymatlari bilan belgilanadi. Obyekt sinfining funktsionalligi ushbu sinf ekzemplari (nuxsasi) ustida o'tkaziladigan operatsiyalar bilan belgilanadi.

C++ tilida sinfni aniqlash uchun ma'lumotlar a'zolari ustida ish olib boradigan hamda obyektlarning xulq-atvorini belgilaydigan ma'lumotlar a'zolari va metodlarini inkapsulatsiyalash talab qilinadi.

C++ Builder komponentlar (components) tushunchasini ham kiritadi. Komponentlar — maxsus sinflar bo'lib, ularning xususiyatlari obyektlar atributlarini tashkil qiladi, ularning metodlari esa komponentli sinflarning tegishli nusxalari ustidagi operatsiyalarni amalga oshiradi. Metod tushunchasi odatda komponentli sinflar tarkibida qo'llanadi va tashqi tomondan oddiy sinfning funktsiya-a'zo atamasidan farq qilmaydi. C++Builder tili komponentlarning turi va funktsional xulq-atvorini nafaqat metodlar yordamida, balki komponentlar sinflarigagina xos bo'lgan xususiyatlar vositasida ham manipulyatsiya qilish imkonini beradi. C++ Builder muhitida ishlar ekansiz, siz ilovani loyihalash bosqichida ham, uni bajarish bosqichida ham komponentli obyekt ustida ishlash (manipulyatsiya qilish) mumkin ekanini tushunib yetishingiz ayon.

Komponentlar xususiyatlari (property) bu ma'lumotlar a'zolarining kengayishidir. Garchi ular ma'lumotlarni o'z hollaricha saqlamasalar-da, biroq obyekt ma'lumotlari a'zolariga kirish huquqini ta'minlaydilar. Xususiyatlarni e'lon qilishda C++ Builder `_property` kalit-so'zdan foydalanadi. Voqealar (events) yordamida komponenta o'ziga qandaydir avvaldan belgilangan ta'sir ko'rsatilganini foydalanuvchiga ma'lum qiladi. C++ Builder muhitida ishlab

chiqilayotgan dasturlardagi metodlar asosan ma'lum voqealarning yuzaga kelishida dastur reaksiyasini ularga nisbatan ishga soladigan voqealarning qayta ishlagichlari (events handlers) da qo'llanadi. Windows operatsiya tizimidagi voqealar va ma'lumotlardagi qandaydir o'xshashlikni payqab olish qiyin emas. Bu yerdagi oddiy tipik voqealar klaviaturadagi tugmacha yoki klavishlarni bosishdan iborat. Komponentalar o'z xususiyatlari, metodlari va voqealarini inkapsulatsiyalaydilar. Bir qarashda, komponentalar C++ tilining boshqa obyektli sinflaridan, bir qator xususiyatlarni hisobga olmaganda, hech bir farq qilmaydi. Bu xususiyatlar orasida hozircha quyidagilarni ko'rsatib o'tamiz:

Komponentalarning ko'pchiligi interfeysning foydalanuvchi bilan boshqarish elementi bo'lib, ularning ayrimlari g'oyat murakkab xulq-atvoriga ega. Barcha komponentalar bitta umumiy ajdod — sinf (TComponent) ning bevosita yoki bilvosita avlodlaridirlar.

Komponentalar odatda bevosita qo'llanadi, ya'ni ularning xususiyatlari ustida ish olib boriladi (manipulyatsiyalar o'tkaziladi); ularning o'zlari yangi tarmoq sinflar (sinfchalar) qurish uchun bazaviy sinflar sifatida xizmat qila olmaydi.

Komponentalar faqat new operatori yordamida uyum (heap) ning dinamik xotirasida joylashtiriladi, oddiy sinflar obyektlarida bo'lganidek, stekda emas. Komponentalar xususiyatlari RTTI — dinamik turlar identifikatsiyasini o'z ichiga oladi.

Komponentalarni Komponentalar Palitrasiga qo'shish va shundan so'ng C++Builder vizual ishlanmasining integrallashgan muhitiga tegishli Shakllar Muharriri vositasida ular ustida ishlash (manipulyatsiyalar o'tkazish) mumkin. OMD obyektarning o'zaro aloqasini talablarni biror bir obyektga yoki obyektlararo yuborish sifatida talqin etadi. Talabni olgan obyekt bunga tegishli usulni chaqirish bilan javob beradi. OMD ning SmallTalk kabi boshqa tillaridan farqli o'laroq, C++ tili «talab» tushunchasidan foydalanishni qo'llab-quvvatlamaydi. Talab — obyekt ustida qilinayotgan ish, metod esa kelib tushgan talabga obyektning javobi.

Yaqinroq olib qaralsa, metod — sinf ta'rifiga kiritilgan oddiy funksiya— a'zo. Metodni chaqirib olish uchun ushbu sinf kontekstida yoki biror bir voqeaning qayta ishlagichida funksiya nomini ko'rsatish kerak.

Aynan metodning sinf bilan yashirin aloqasi uni oddiy funksiya tushunchasidan ajratib turadi. Metodni bajarish paytida u o'z sinfining barcha ma'lumotlariga kirish huquqiga ega bo'ladi, garchi ushbu sinf nomining ochiq-oydin spetsifikatsiyasini talab qilmasa ham. Bu ish beistisno har bir metodga yashirin parametrغا, ya'ni sinf nusxasi (ekzemplari) ga o'zgaruvchan ko'rsatkich this ni berish orqali ta'minlanadi. Har gal metod sinf ma'lumotlari a'zolariga murojaat qilganda, kompilyator this ko'rsatkichidan foydalanadigan maxsus kodni generatsiya qiladi.

Vorislik: Vorislik g'oyasi obyektlar xulq-atvorini modifikatsiyalash muammosini hal qiladi hamda OMD ga favqulodda kuch va moslashuvchanlik baxsh etadi. Vorislik, deyarli hech qanday cheklanishlarsiz, siz yoki boshqa biron kimsa tomonidan yaratilgan sinflarni izchil qurish va kengaytirish imkonini beradi. Eng oddiy sinflardan boshlab, murakkablik jihatidan asta-sekin ortib boradigan, ammo sozlanishi ham oson, ichki tuzilishi ham oddiy bo'lgan hosila sinflarni yaratish mumkin.

Ayniqsa yirik dasturiy loyihalarni ishlab chiqishda vorislik tamoyilini hayotga izchil tatbiq etish pasayib boruvchi tuzilmaviy dasturlash (umumiydan juz'iyga) texnikasi bilan yaxshi moslashadi hamda ko'p o'rinda bunday yondashuvni rag'batlantiradi. Bunda dastur kodining murakkabligi ancha kamayadi. Hosila sinf (avlod) o'z bazaviy sinfining (otasining) hamda sinflar tabaqalanishidagi o'zining barcha ajdodlarining hamma xususiyatlari, metodlari va voqealarini meros qilib oladi.

Vorislik paytida bazaviy sinf yangi atributlar va operatsiyalar hisobiga yanada o'sadi. Hosila sinfda odatda yangi ma'lumotlar a'zolari, xususiyatlar va metodlar paydo bo'ladi. Obyektlar bilan ishlashda dasturchi odatda konkret masalani hal qilish uchun eng to'g'ri keladigan sinfni tanlaydi hamda undan bitta yoki bir nechta voris avlod yaratadiki, ular o'z otalarida mavjud imkoniyatlardan

ko'proq imkoniyatga ega bo'ladilar. Do'stona funksiyalar hosila sinfga barcha tashqi sinflar ma'lumotlari a'zolariga kirish huquqini olish imkonini beradilar.

Bundan tashqari, meros qilib olinayotgan metodlardan, ularning bazaviy sinfdagi ishi avlodga to'g'ri kelmasa, hosila sinf ortiqcha yuklanishi (overload) mumkin. OMD da ortiqcha yuklanishdan foydalanish har qanaqasiga rag'batlantiriladi, garchi bu so'zning to'g'ri ma'nosidan kelib chiqqanda, odatda ortiqcha yuklanishlar tavsiya qilinmaydi. Agar metod bittadan ortiq bir nomdagi funksiya bilan assotsiatsiyalansa, u ortiqcha yuklangan deb aytiladi. E'tibor bering, sinflar tabaqalanishida ortiqcha yuklatilgan metodlarni chaqirib olish mexanizmi qayta aniqlangan.

Vorislik konsepsiyasini soat haqidagi misolga tatbiq qilish uchun, faraz qilaylik, vorislik tamoyiliga amal qilgan «Casio» firmasi soatning yangi modelini chiqarishga qaror qildi. Aytaylik, bu model tugmachalardan biri ikki marta bosilsa, vaqtni ovozda ayta oladi. Gapiradigan soatlar modeli (OMD atamaları bo'yicha, yangi sinf) ni yangidan yaratish o'rniga muhandislar ishni uning prototipidan boshlaydilar (OMD atamaları bo'yicha bazaviy sinfning yangi avlodini yaratadilar). Hosila obyekt otasining barcha atributlari va funksional- ligini meros qilib oladi. Sintezlangan ovozda aytilgan sonlar avlodning yangi ma'lumotlar a'zolari bo'lib qoladi, tugmachalarning obyektli metodlari esa, ularning qo'shimcha funkcionalligini ishga tushirish uchun ortiqcha yuklatilgan bo'lishi kerak. Tugmachalarning ikki marta bosilish hodisasiga yangi usul javob berib, u joriy vaqtga mos keladigan sonlar ketma-ketligi (yangi ma'lumotlar a'zolari) ning talaffuz qilinishida namoyon bo'ladi. Yuqorida aytilganlarning hammasi gapiradigan soatlarning dasturiy amalga oshirilishiga to'liq taalluqli.

Sinflarni ishlab chiqish: Sinflar ma'lum maqsadlarga erishish uchun ishlab chiqiladi. Odatda dasturchi mavhum g'oyadan boshlaydi va u asta-sekin loyihani ishlab chiqish jarayonida turli detallar bilan to'ldirib boriladi. Ba'zida bir- biriga g'oyat o'xshash bo'lgan bir necha sinfni ishlab chiqish bilan ish tugallanadi. Sinflarda kodlarni bu kabi takrorlash (dubllashtirish) dan qochish uchun bu

sinflarni ikki qismga bo'lish kerak, ya'ni umumiy qismni ota sinfida aniqlab, farqlanadiganlarini hosila sinfda qoldirish kerak.

Sinfdan foydalanishdan oldin u e'lon qilinishi kerak. Odatda, amaliyotchi dasturchi tayyor bazaviy sinflardan foydalanadi, bundan tashqari u barcha spetsifikatsiyalarni va ichki ishlash yo'llarini bilishi mutlaqo shart emas. Biroq, C++ bazaviy sinfdan foydalanishingiz uchun qanday ma'lumotlar a'zolari va metodlarga kira olishingiz mumkinligini (C++ Builder komponentasi qo'llansa, taqdim etilayotgan xususiyatlar va voqealarni ham) albatta bilishingiz lozim.

Bazaviy sinfni e'lon qilish: C++ Builder sizga o'z xususiyatlari, ma'lumotlari, metodlari va voqealari nomlarini inkapsulatsiyalaydigan bazaviy sinfni e'lon qilish imkonini beradi. O'zlarining bevosita vazifalarini bajarishdan tashqari, obyektli metodlar sinf xususiyatlari va ma'lumotlari qiymatlariga kirish uchun ma'lum imtiyozlarga ham ega bo'ladilar.

Sinf ichidagi har bir e'lon qilish, sinf nomi qaysi seksiyada paydo bo'lishiga qarab, bu nomlarga kirish imtiyozlarini aniqlaydi. Har bir seksiya quyidagi kalit-so'zlarning biridan boshlanadi: private, protected, public. Bazaviy sinfni e'lon qilishning umumlashma sintaksisi quyidagi ko'rinishga ega:

class className

private:

<privat ma'lumotlar a'zolari> <privat konstruktorlar> <privat metodlar>

protected:

<himoyalangan ma'lumotlar a'zolari> <himoyalangan konstruktorlar>

<himoyalangan metodlar>

public:

<ommaviy xususiyatlar> <ommaviy ma'lumotlar a'zolari> <ommaviy konstruktorlar> <ommaviy destruktor> <ommaviy metodlar>

Shunday qilib, C++da bazaviy sinfni e'lon qilish quyidagicha kirish huquqlari va tegishli ko'rishlik sohasini taqdim etadi:

Privat private nomlar faqat ushbu sinf metodlariga ruxsat etilgan eng cheklangan kirish huquqiga ega. Hosila sinflar uchun bazaviy sinflarning privat metodlariga kirish taqiqlangan.

Himoyalangan protected nomlar ushbu sinf va undan hosil bo'lgan sinflar metodlariga ruxsat etilgan kirish huquqiga ega.

Ommaviy public nomlar barcha sinflar va ularning obyektlari metodlariga ruxsat etilgan cheksiz kirish huquqiga ega.

Quyidagi qoidalar sinf e'lon qilinishining turli seksiyalarining hosil bo'lishida qo'llaniladi:

- Seksiyalar har qanday tartibda ham paydo bo'lishi, ularning nomlari esa takroran uchrayverishi mumkin.
- Agar seksiya nomlanmagan bo'lsa, sinf nomlarining keyingi e'lonlarini kompilyator privat deb hisoblaydi. Bu yerda sinf va tuzilmaning e'lon qilinishida farq yuzaga kelyapti: tuzilma yashirin holda ommaviy deb olib qaraladi.
- Agar siz haqiqatan ham ma'lumotlar a'zolariga har qayerdan kirishni ruxsat etmoqchi bo'lmasangiz, imkon darajasida ularni ommaviy seksiyaga joylashtirmang. Faqat hosila sinflar metodlariga kirish huquqini berish uchun ularni odatda himoyalangan deb e'lon qiladilar.
- Metodlardan ma'lumotlar xususiyatlari va a'zolarini tanlash, tekshirish va qiymatlarini o'rnatish uchun foydalaning.
- Konstruktorlar va destruktorlar maxsus funksiyalar bo'lib, qiymatni qaytarmaydilar va o'z sinfining nomiga ega bo'ladilar. Konstruktor berilgan sinf obyektini quradi, destruktor esa uni olib tashlaydi.
- C++ning bittadan ortiq yo'riqnomasiga ega bo'lgan metodlarni (konstruktorlar va destruktorlar kabi) sinfdan tashqarida deb e'lon qilish tavsiya etiladi.

Navbatdagi misolda bazaviy sinfnig e'lon qilinishini biror bir aniq taxmin bilan to'ldirishga harakat qilingani ko'rsatilgan. Shuni ta'kidlab o'tish lozimki, C++ Builder sinfi komponentlari uchun Count xususiyatini himoyalangan

seksiyada e'lon qilish xos bo'lsa, FCount ma'lumotlar a'zosiga yozuvni amalga oshiradigan SetCount metodini privat seksiyada e'lon qilish xos.

Class Tpoint {private:

int FCount; //Ma'lumotlarning privat a'zosi void _fastcall SetCount(int Value);

protected:

_property int Count=//Himoyalangan xususiyat {read=Fcount, write=SetCount};

double x;//Himoyalangan ma'lumotlar a'zosi double y;//Himoyalangan ma'lumotlar a'zosi public:

Tpoint(double xVal, double yVal);//Konstruktor double getX(); double getY();

Metodlarning e'lon qilinishi va aniqlanishi turli fayllarda saqlanadi. Misollar shuni ko'rsatadiki, metodlar sinfdan tashqarida aniqlanganda ularning nomlarini kvalifikatsiya qilish (ixtisoslashtirish) kerak. Metodning ko'rimlilik sohasini aniqlaydigan uning bunday kvalifikatsiya sintaksisi quyidagi ko'rinishga ega:

<Sinf nomi>::<metod nomi>

Siz sinfni e'lon qilganingizdan keyin uning nomidan ushbu sinf obyektini e'lon qilishda identifikat.or turi sifatida foydalanish mumkin. Masalan:

TPoint MyPoint.*

Sinf metodlarini sinfdan tashqarida aniqlash: Funksiya prototipi ichida joylashtirilgan employee sinfini ko'rib chiqamiz. Funksiyaning o'zi esa sinfdan tashqarida aniqlangan.

Navbatdagi CLASSFUN.CPP dasturi show_employee funksiyasining ta'rifini sinfdan tashqarida joylashtiradi va bunda sinf nomini ko'rsatish uchun global ruxsat operatoridan foydalanadi:

#include <iostream.h>

#include <string.h> class employee {

public: char name [64]; long employee_id; float salary; void

show_employee(void);

};

void employee::show_employee(void)

{

```

cout << "Ism: " << name << endl;
cout << "Tartib raqami:" << employee_id << endl;
cout << "Maosh:" << salary << endl;
};
void main(void)
{
employee worker, boss; strcpy(worker.name, "John Doe"); worker.employee_id =
12345; worker.salary = 25000; strcpy(boss.name, "Happy Jamsa");
boss.employee_id = 101; boss.salary = 1011012.00; worker.show_employee();
boss.show_employee();
}

```

1.4. Konstruktorlar va destruktorglar

Nomlaridan ko‘rinib turganidek, konstruktor — bu metod bo‘lib, u o‘z xotirasida ushbu sinf obyektini quradi, destruktor esa — bu obyektini olib tashlaydigan metod. Konstruktorlar va destruktorglar boshqa obyektli metodlardan quyidagi xususiyatlariga ko‘ra farqlanadi:

- O‘z sinfi nomi bilan bir xil bo‘lgan nomga ega.
- Qaytariladigan qiymatga ega emas.
- Garchi hosila sinf bazaviy sinflarning konstruktorlari va destruktorglarini chaqira olsa-da, konstruktor va destruktorglarning o‘zlari vorislik qilolmaydi.

Agar boshqacha e‘lon qilinmagan bo‘lsa, kompilyator tomonidan avtomatik tarzda public sifatida generatsiya qilinadi.

Sinf obyektlarining yaratilishi va yo‘q qilinishini tegishli tarzda kafolatlash uchun kompilyator tomonidan chaqirib olinadi. Agar obyekt dinamik xotiraning ajratilishi va yo‘q qilinishini talab qilsa, new va delete operatorlariga noaniq murojaatga ega bo‘lishi mumkin. Quyida konstruktorlar va destruktorglar e‘lonining umumlashma sintaksisini namoyish qiluvchi misol keltiramiz:

```

class className {public:
//class Name ma'lumotlarining boshqa a'zolari;

```

```

className (<parametrlar ro‘yxati;-) //Dalillarga ega konstruktor
className(const className&);//Nusxa ko‘chirish konstruktori
//Boshqa konstruktorlar ~
className();//Destruktor
//Boshqa metodlar
};

```

Sinf soni cheklanmagan konstruktorlarga ega bo‘lishi, shu jumladan, konstruktorlarga umuman ega bo‘lmasligi mumkin. Konstruktorlar virtual deb e‘lon qilinishi mumkin emas. Hamma konstruktorlarni himoyalangan seksiyaga joylashtirmang hamda yashirin argumentlar qiymatidan foydalanib, ularning sonini kamaytirishga intiling.

Ko‘p hollarda sinf obyektlarini initsiallashtirish (nomlash) ning bir nechta usullariga ega bo‘lish yaxshi natija beradi. Bunga bir nechta konstruktor vositasida erishish mumkin. Masalan:

```

class date {
int month, day, year; public:
// ...
date(int, int, int); //yilning oyi, kuni date(char*); //satr ko‘rinishidagi sana
date(int); //bugungi kun, oy, yil
date(); //yashirin sana: bugungisi
};

```

Ortiqcha yuklangan funksiyalar qanday qoidalarga amal qilsa, konstruktorlar ham parametrlar turlariga nisbatan xuddi shunday qoidalarga amal qiladilar. Agar konstruktorlar o‘z parametrlari turlari bo‘yicha ancha-muncha farq qilsa, kompilyator har gal foydalanganda to‘g‘ri parametrni tanlab olishi mumkin:

```

date today(4);
date july4("1983-yil, 4-iyul");
date guy("5-noyabr")
date now; //yashirish initsiallashtiradi (nomlanadi)

```

Konstruktorlarning uch turi mavjud:

Yashirin konstruktor parametrlarga ega emas. Agar sinf bitta ham konstruktorga ega bo'lsa, kompilyator avtomatik tarzda bitta yashirin konstruktor yaratadiki, u o'z sinfiga mansub obyektни yaratishda xotirani shunchaki ajratib beradi.

Date holatida har bir parametr uchun «yashirin qabul qilish: today» (bugun) sifatida talqin qilinadigan yashiringan qiymatni berish mumkin:

```
class date {
    int month, day, year; public:
    // ...
    date(int d =0, int m =0, int y =0);
    date(char*);// satr vositasida berilgan sana
};
date::date(int d, int m, int y)
{
    day = d ? d : today.day; month = m ? m : today.month; year = y ? y : today.year;
    // yo'l qo'yiladigan sana ekanini tekshirish // ...
}
```

Dalillarga ega konstruktor obyektning yaratilish paytida uni initsiallashtirish (nomlash) ga, ya'ni turli funksiyalarni chaqirib olishga, dinamik xotirani ajratish, o'zgaruvchilarga dastlabki qiymatlarni berish va h.k. ga imkon beradi.

Nusxa ko'chirish konstruktori berilgan sinf obyektlarini yaratish uchun ma'lumotlarni ushbu sinfning mavjud bo'lgan boshqa obyektidan nusxa ko'chirish yo'lidan borishga mo'ljallangan. Bunday konstruktorlar ma'lumotlarning dinamik tuzilmalarini modellashtiradigan obyektlar nusxasini yaratishda ayniqsa maqsadga muvofiqdir. Biroq yashirin holda kompilyator yuzaki nusxalash konstruktorlari (shallow copy constructors) deb ataluvchi faqat ma'lumotlar a'zolaridan nusxa oladigan konstruktorlarni yaratadi. Shuning uchun, agar biror bir ma'lumotlar a'zolari ko'rsatkichlarga ega bo'lsa, ma'lumotlarning o'zidan nusxa ko'chirilmaydi. «Chuqur» nusxalashni konstruktor kodiga joriy qilish uchun tegishli yo'riqnomalarni kiritish kerak. Misol:

```
class string {
```

```

char *Str; int size; public:
string(string&); // nusxa ko'chirish konstruktorelari
};
string::string(string& right) // dinamik o'zgaruvchilar va resurslar { // nusxalarini
yaratadi
s = new char[right->size]; strcpy(Str, right->Str);
}

```

Sinf faqat bitta ommaviy destruktorni e'lon qilishi mumkin. Uning o'z sinfi bilan bir xil bo'lgan nomi oldidan ~ (tilda) belgisi turishi kerak. Destruktor parametrlarga ega emas hamda virtual deb e'lon qilinishi mumkin. Agar sinf destruktore e'loniga ega bo'lmasa, kompilyator avtomatik ravishda uni yaratadi.

Odatda tegishli konstruktorelari bajaragan operatsiyalarga teskari operatsiyalarni destruktorelari bajaradi. Agar siz fayl sinfi obyektini yaratgan bo'lsangiz, bu holda destruktorelari bu faylning yopilishi ehtimoldan xoli emas. Agar sinf konstruktorelari ma'lumotlar massivi uchun dinamik xotirani ajratsa (new operatori yordamida), bu holda destruktorelari ajratilgan xotirani bo'shatishi (delete operatori yordamida) ehtimoldan xoli emas va h.k.

date obyekti konstruktorelari va destruktorelari misol:

```

class date { int *day, *month, *year
public:
date(int d, int m, int y)
{
day=new int; month=new int; year=new int;
*day= d ? d : 1;
*month = m ? m : 1;
*year = y ? y : 1;
}
~date()
{
delete day; delete month; delete year;
}
}

```

```
}};
```

this ko'rsatkichi

Obyekt uchun chaqirilgan funksiya (a'zo) da ma'lumotlar (obyekt a'zolari) ga bevosita iqtibos qilish mumkin. Masalan:

```
class x { int m; public:  
int readm() { return m; }  
}; x aa; x bb; void f()  
{  
int a = aa.readm(); int b = bb.readm();  
// ...  
}
```

readm() a'zosining birinchi chaqirilishida m aa.m ga tegishli bo'ladi, ikkinchi chaqirilishida esa bb.m ga tegishli bo'ladi.

x* this;

sifatida yashirin tavsiflangan hamda shunday nomlangan (initsiallashtirilgan) ki, u obyektни ko'rsatadi (funksiya—a'zo mana shu obyekt uchun chaqirilgan). X sinfini ekvivalent ravishda quyidagicha tavsiflash mumkin:

```
class x {  
int m;  
public:  
int readm() {return this->m;}  
};
```

A'zolarga iqtibos qilganda this dan foydalanish ortiqcha. This asosan funksiya-a'zolari yozishda qo'llanadi (bu funksiya-a'zolar bevosita ko'rsatkichlar ustida ish olib boradi).

Konstantali funksiya-a'zolar va konstantali obyektlar: Biror bir funksiya yoki sinfga sinf obyektining shaxsiy qismiga kirish ruxsati talab qilinsa, ba'zida kirish huquqi qoidalaridan istisno tariqasida chetlanish talab qilinadi. Bu esa sinfning o'zi o'z obyektlariga «chetdan» kirish huquqlarini belgilaydi degan tamoyilga mos keladi. Kirishni nazorat qilish vositalariga funksiya-a'zolarining

konstantali (const) deb e'lon qilinishi kiradi. Bu holda ular o'zlari birga chaqirilayotgan joriy obyekt qiymatlarini o'zgartirish huquqiga ega emaslar. Shunda bunday funksiyaning sarlavhasi quyidagi ko'rinishga ega bo'ladi:

```
void dat::put() const {...}
```

Konstanta obyektlarini ham xuddi shunday aniqlash mumkin:

```
const class a{...} value;
```

II BOB. C VA C++ TILIDA MAVJUD KUTUBXONALAR RO'YXATI

C++ tili kutubxonasi xuddi C tili kutubxonasi kabi aniqlanadi. Tashkil etilayotgan sarlavha fayli quyidagicha farqlarga ega[3]:

- Har bir sarlavha fayli xuddi C tilidagi kabi o'z nomiga ega ammo «C» oldi qo'shimchasiz va hech qanday kengaytmaga ega emas. Masalan, C tilidagi <cstdlib> sarlavha fayli C++ tilida stdlib.h bilan ekvivalent;

- Kutubxonalarning har bir elementi *std namespace* nomlar fazosida aniqlanadi;

Ammo C tili bilan muvofiqlikni saqlash maqsadida sarlavha fayli "name.h" ko'rinishida bo'ladi.

Bulardan tashqari C++ tilidagi o'zgarishlar quyidagicha

- `wchar_t`, `char16_t`, `char32_t` va `bool` turlari C++ tilining fundamental turlari hisoblanadi. Shuning uchun ham C da bular aniqlanmagan. Bulardan tashqari sarlavhada yana bir qancha kichik aniqlanishlar mavjud bo'lib bulardan biri <iso 646.h> sarlavxa faylidir. U C++ tilining kalit so'zlaridan biri hisoblanadi.

- Quyidagi funksiyalarning parametrlarida bir qator o'zgarishlar mavjud: `strchr`, `strpbrk`, `strrchr`, `memchr`, `strstr`

- <cstdlib> sarlavxa faylida aniqlangan funksiyalarga C++ tilida qo'shimchalar kiritilganligi kuzatiladi.

C++ tili versiyalarida ayrim funksiyalar mazmun jihatidan aniqlangan parametrlarga qo'shimcha turlarga ega masalan, `float` va `long double` kabi. C++ 11 1990 yilda C tiliga qo'shilgan. Bu jarayon xalqaro C tilini standartlashtirish uyushmasida amalga oshirilgan. Ammo 2011 yildagi C tilining ISO standarti C++ tili bilan mos kelmaydi. C va C++ tilida mavjud kutubxonalar (Sarlavxa fayllari):

- <cassert> (`assert.h`)-C tilining kutubxonasi

<cctype> (ctype.h)- funksiyalar bilan muloqot o'rnatish kutubxonasi
<errno> (errno.h)- C tilida xatoliklar
<ctype> (ctype.h)- Suzuvchi vergulli sonlar uchun maxsus turlar
<iso646> (iso646.h) xalqaro standartlashtirish uyushmasi operatorlarni to'g'ri yozilishi bo'yicha 656 ta muqobil tekshiruvlarni amalga oshiruvchi kutubxona
<climits> (limits.h)- tarkibiy turlar o'lchami
<locale> (locale.h)- C tilining mahalliyashtirish kutubxonasi
<cmath> (math.h)- C tilining matematik funksiyalar kutubxonasi
<setjmp> (setjmp.h)- Mahalliy bo'lmagan o'tishlar
<signal> (signal.h)- C tilining Signallarga murojat qiluvchi kutubxonasi
<stdarg> (stdarg.h)- o'zgaruvchi argumentlarga murojaat
<stddef> (stddef.h)- C tilining standart aniqlanganlari
<stdio> (stdio.h)- C tilining kiritish-chiqarish amallarini bajaruvchi kutubxona
<stdlib> (stdlib.h)- C tilining umumiy standart utilitalari kutubxonasi
<string> (string.h)- C tilining satrlar bilan ishlovchi kutubxonasi
<time> (time.h)- C tilining vaqt bilan ishlovchi kutubxonasi
Xalqaro standartlashtirish uyushmasi tomonidan C++ tilining ISO C-90 versiyasiga yana 2 ta qo'shimcha <wchar> va <wctype> nomli kutubxonalar qushdi.

2.1. Ma'lumotlarni kiritish va chiqarish va akslantirish funksiyalari bilan ishlovchi kutubxonalar

<stdio> (stdio.h) kutubxonasi

C++ tilida qiymatlarni kiritish chiqarish uchun C tilidan o'tgan stdio.h kutubxonasidan foydalaniladi. Bu kutubxonada mavjud funksiyalar klaviatura, printer, terminal kabi qurilmalarni fayl oqimlari sifatida qabul qiladi. Oqim – bu o'zaro bir jinsli sodir bo'luvchi xususiyatga ega jarayondir. Oqimlar <stdio> kutubxonasida fayl ko'rsatgichlari kabi qayta ishlanadi. Fayl ko'rsatgichlari – bu

oqimda unikal aniqlanib, bajariladigan amal uning parametri sifatida beriladi. Bu kutubxonada 3 ta standart oqim mavjud; *stdin, stdout* va *stderr* lar. Bu oqimlar kutubxona chaqirilishi bilan avtomatik tarzda aniqlanadi [4,10].

Oqim xususiyatlar – Oqimlar bir qator xususiyatlarga ega ya’ni ularda qanday funksiyalardan foydalanishmoqda va ular orqali ma’lumotlarni chiqarish qanday amalga oshirilmoqda. Oqimlarning bunday xususiyatlarining ko’pchiligi “fopen” funksiyasi bilan aniqlangan:

Ma’lumotlarni o’qish – yozish uchun ruxsat: Bu holat oqimdan o’qish yoki fizik xotiraga ma’lumotlarni yozish uchun qo’llaniladi.

Matn/Binar: Matnlar oqimi – bu har biri yakunlanuvchi satr deb tushuniladi. Matn holati satrlarni o’z xarakteriga ko’ra o’qish yozishni amalga oshiradi. Binar – xarakterli oqim esa fizik xotiradan o’qish yozishni to’g’ridan – to’g’ri amalga oshiradi.

Orientatsiya: Faylni ochishda oqim hech qanday orientatsiyaga ega bo’lmaydi. Bajarilayotgan amalga bog’liq ravishda chiqariluvchi miqdorlar keng formatli bitlar yoki baytlar ko’rinishida bo’ladi.

Indikatorlar: Oqimning joriy holatini aniqlovchi va bir qator chiqarish amallarini bajaruvchi ichki aniqlangan ichki indikatorlar mavjud bo’lib, ular quyidagilar:

Xatolik indikator. Bu indikator oqimga bog’liq bo’lgan xatoliklar ro’y berganda ishlaydi. Bu indekator “ferror” funksiyasi hisoblanadi.

Fayl oxirini ko’rsatuvchi indekator. Agar fayl oxirini ko’rsatuvchi indekator o’rnatilgan bo’lsa, u oxirgi o’qilgan ma’lumotni nazorat qiladi. U “feof” funksiyasi hisoblanadi.

Holat indekatori. Bu oqim uchun ma’lumotlarni kiritish chiqarish indekatori hisoblanadi. Uning qiymatini “ftall” funksiya yordamida olish mumkin.

Funksiyalar:

<i>Fayl ustida amal bajaruvchi funksiyalar:</i>	
remove	faylni qayta yuklash.

rename	faylni qayta nomlash.
Tmpnam	vaqtinchalik faylni nomini qo'yish.
Tmpfile	vaqtinchalik fayl ochish.
<i>Faylga bog'lanishni amalga oshiruvchi funksiyalar:</i>	
fclose	faylni yopish
fflush	<i>flush</i> oqimi.
fopen	faylni ochish.
freopen	faylni turli usullarda ochish.
setbuf	oqimdagi ma'lumotlar buferi.
setvbuf	bufferini o'zgartirish oqimi.
<i>Kiritish chiqarish formati:</i>	
fprintf	formatlangan ma'lumotlarni oqimga yozish.
fscanf	formatlangan ma'lumotlarni oqimga qo'shish.
printf	stdout fayliga formatlangan ma'lumotlarni chiqarish.
scanf	stdin faylidan formatlangan ma'lumotlarni o'qish.
snprintf	ma'lum o'lchamga ega ma'lumotni bufferga yozish.
sprintf	formatlangan ma'lumotlarni satrga yozish.
sscanf	formatlangan ma'lumotlarni satrdan o'qish.
vfprintf	o'zgaruvchidagi formatlangan ma'lumotlarni oqimga yozish.
vscanf	oqimdan formatlangan ma'lumotlarni o'zgaruvchiga o'qish.
vprintf	o'zgaruvchidan stdout fayliga formatlangan ma'lumotlarni chiqarish.
vscanf	<i>stdin</i> faylidan formatlangan ma'lumotlarni o'zgaruvchidan o'qish.
vsprintf	o'zgaruvchidagi formatlangan ma'lumotlarni ma'lum o'lchamga ega bufferga chiqaradi.
vsprintf	o'zgaruvchidagi formatlangan ma'lumotlarni satrga chiqarish.

vsscanf	satrdan formatlangan ma'lumotlarni o'zgaruvchiga o'qish.
<i>Belgilarni kiritish chiqarish xarakteri:</i>	
fgetc	oqimdan belgini o'qish.
fgets	oqimdan satrni o'qish.
fputc	belgini oqimga chiqarish.
fputs	satrni oqimga chiqarish.
getc	oqimdan belgini o'qish.
putc	oqimga belgini chiqarish.
getchar	<i>stdin</i> faylidan belgini o'qish.
gets	<i>stdin</i> faylidan satrni o'qish.
putchar	<i>stdout</i> fayliga belgini chiqarish.
Puts	<i>stdout</i> fayliga satrni chiqarish.
ungetc	oqimdan belgini yo'qotish.
<i>Ma'lumotlarni bevosita kiritish chiqarish:</i>	
fread	ma'lumotlar blokini oqimdan o'qish.
fwrite	ma'lumotlar blokini oqimga chiqarish.
<i>Faylning joylashuvi:</i>	
fgetpos	oqimning joriy holatini aniqlash.
fseek	oqimning holatini aniqlovchi indekator.
fsetpos	oqimlar to'plarining joylashuvini aniqlovchi indekator.
ftell	oqimning joriy holatini aniqlash.
rewind	oqimlarning boshlang'ich holatini aniqlash.
<i>Xatolarga ishlov berish:</i>	
clearerr	xatolik indekatorini o'chirish.
feof	fayl oxiri indekatorini tekshirish.
ferror	xatolik indekatorini tekshirish.
perror	bosmalashdagi xatolik xabari.
<i>Makroaniqlovlar:</i>	
BUFSIZ	buffer o'lchami

EOF	fayl oxiri
FILENAME MAX	bir vaqtda ochilish mumkin bo'lgan oqimlarning miqdori.
FOPEN MAX	bir vaqtda ochilish mumkin bo'lgan oqimlarning maksimal miqdori.
L tmpnam	vaqtinchalik fayl nomining minimal uzunligi.
NULL	bo'sh ko'rsatkich.
TMP_MAX	vaqtinchalik fayl nomining qo'shimcha aniqlovchilar.
<i>Turlari:</i>	
FILE	oqimni boshqarish uchun ma'lumotlarni saqlovchi obyekt.
fpos_t	faylning joylashuv chegarasini aniqlovchi ma'lumotlarni saqlovchi obyekt.

<cstdlib> (stdlib.h) kutubxonasi

Bu kutubxona bir qator xotirani dinamik boshqaruvni amalga oshiruvchi, tasodifiy sonlarni hosil qilish, dasturlash muhitining bog'lovchilari, butun sonlarning arifmetikasi akslantirish, saralash va izlash funksiyalarini o'zida saqlaydi [3,10].

<i>Funksiyalar:</i>	
atof	satr ko'rinishidagi sonni haqiqiy son formatiga o'tkazish.
atoi	satr ko'rinishidagi sonni butun son formatiga o'tkazish.
atol	satr ko'rinishidagi sonni uzun butun son formatiga o'tkazish.
atoll	satr ko'rinishidagi sonni ikkilangan uzun son formatiga o'tkazish.
strtod	satr ko'rinishidagi sonni ikkilangan haqiqiy son formatiga o'tkazish.
strtof	satr ko'rinishidagi sonni haqiqiy son formatiga o'tkazish.
strtol	satr ko'rinishidagi sonni uzun butun son formatiga o'tkazish.
strtold	satr ko'rinishidagi sonni uzun haqiqiy son formatiga

	o'tkazish.
strtoll	satr ko'rinishidagi sonni ikkilangan butun son formatiga o'tkazish.
strtoul	satr ko'rinishidagi sonni ishorasiz uzun butun son formatiga o'tkazish.
strtoull	satr ko'rinishidagi sonni ishorasiz ikkilangan uzun butun son formatiga o'tkazish.
rand	tasodifiy sonlar generator.
srand	tasodifiy sonlar generatorini boshqarish.
calloc	dinamik massiv uchun xotiradan joy ajratadi.
<i>Muhim funksiyalar:</i>	
abort	joriy jarayonni to'xtatish.
atexit	(ma'lumotlarni chiqarish) joriy holatdan chiqib ketish.
at_quick_exit	joriy holatni favqulotda tark etish funksiyasi.
exit	chaqirilgan jarayonni tugatish.
getenv	muhim qaytaruvchi qiymatni olish.
quick_exit	chaqirilgan jarayonni favqulotda tark etish.
system	sistema buyruqlarini bajarish.
Exit	chaqirilgan jarayonni tugatish.
<i>Qidiruv va saralash funksiyalari:</i>	
B search	massivda <i>binar</i> qidiruv.
qsort	massiv elementlarini saralash.
<i>Butun arifmetika funksiyalari:</i>	
abs	absolyut qiymat.
div	butunga bo'lish.
labs	uzun butun sonning absolyut qiymati.
ldiv	uzun butun sonlarni butunga bo'lish.
llabs	ikkililik uzun butun sonlarning absolyut qiymati.
lldiv	uzun butun sonlarni butunga bo'lish.

<i>Multibayt satrlar:</i>	
mbstowcs	multibayt satrlarni multi belgilarga aylantiradi.
<i>Multibayt belgilar:</i>	
mblen	multibayt belgilarning uzuzligini qaytarish.
mbtowc	multibayt belgini keng formatli multibayt belgiga o'tkazadi.
wctomb	keng formatli multibayt belgini multibayt belgiga o'tkazadi.
westombs	multibayt belgilarni multibayt satrlarga almashtiradi.
<i>Makro o'zgarmaslar:</i>	
EXIT FAILURE	muvafaqqiyatsiz tugallanuvchi kod.
EXIT SUCCESS	muvafaqqiyatli tugallanuvchi kod.
MB_CUR_MAX	multibayt belgilarni maksimal o'lchami.
NULL	bo'sh ko'rsatkich.
RAND_MAX	<i>rand</i> funksiyasi qaytaradigan makimal qiymat.
<i>Tiplar:</i>	
div_t	<i>div</i> amali qaytaradigan qiymatlarni saqlaydi.
ldiv_t	<i>ldiv</i> amali qaytaradigan qiymatlarni saqlaydi.
lldiv_t	<i>lldiv</i> amali qaytaradigan qiymatlarni saqlaydi.
size_t	ishorasiz butun tur.

2.2. Satrlar va simvollar bilan ishlovchi kutubxonalar

<cctype> (ctype.h) kutubxonasi

Bu kutubxona tabaqalashtiruvchi va individual xarakterga ega bir qator funksiyalarni o'zida tasvirlaydi.

Funksiyalar-bu funksiyalar int turidagi qiymatlar qabul qiladi va int turidagi qiymatlarni yoki mantiqiy turga mos keluvchi qiymatlardan birini qaytaradi: qaytarilgan 0 qiymat "false" qiymatini 0 dan farqli ixtiyoriy qiymat esa

“true” qiymatini bildiradi. 2 ta to’plamga oid funksiyalar mavjud: funksiya parametrlari 2 ta kategoriyadagi chegaralarni qo’yadi:

isalnum	Belgining alifbo harfi yoki raqam ekanligini tekshiradi.
isalpha	Belgining alifbo harflaridan biri ekanligini tekshiradi.
isblank	Belgining son ekanligini tekshiradi.
iscntrl	Belgining haraktorli belgilaridan biri ekanligini tekshiradi.
isdigit	Belgining o’nlik sanoq sistemasidagi raqam ekanligini bildiradi.
isgraph	Belgining grafik tasvir haraktoriga ega ekanligini bildiradi.
islower	Belgining kichik harf ekanligini tekshiradi.
isprini	Belgining bosmalanuvchanlik haraktorini tekshiradi.
ispunct	Belgining punkitnatsiya belgilaridan biri ekanligini tekshiradi.
isspace	Belgining bo’sh joy ekanligini tekshiradi.
isupper	Belgining katta harf ekanligini tekshiradi.
isxdigit	Belgining o’n oltilik sanoq tizimidagi son ekanligini tekshiradi.
Satr harflarning registrlarini almashtiradigan funksiyalar:	
tolower	Bosh harflarni kichik harflarga almashtiradi.
toupper	Kichik harflarni bosh harflarga almashtiradi.

Birinchi turdagi funksiyalar ASCII jadvalidagi dastlabki 127 ta belgilari bilan ishlaydi.

ASCII Qiymati	Belgilar	<u>iscntrl</u>	<u>isblank</u>	<u>isspace</u>	<u>isupper</u>	<u>islower</u>	<u>isalpha</u>	<u>isdigit</u>	<u>isxdigit</u>	<u>isalnum</u>	<u>ispunct</u>	<u>isgraph</u>	<u>isprint</u>
0x00 .. 0x08	NUL, (other control codes)	x											
0x09	tab ('\t')	x	x	x									
0x0A .. 0x0D	(white-space control codes: '\f', '\v', '\n', '\r')	x		x									
0x0E .. 0x1F	(other control codes)	x											
0x20	space (' ')		x	x									x
0x21 .. 0x2F	!"#\$%&'()*+,-./										x	x	x
0x30 .. 0x39	0123456789							x	x	x		x	x

0x3a .. 0x40	;<=>?@											X	X	X
0x41 .. 0x46	ABCDEF				X		X		X	X			X	X
0x47 .. 0x5A	GHIJKLMNOPQRSTUVWXYZ				X		X			X			X	X
0x5B .. 0x60	[\]^_`											X	X	X
0x61 .. 0x66	abcdef					X	X		X	X			X	X
0x67 .. 0x7A	Ghijklmnopqrstuvwxyz					X	X			X			X	X
0x7B .. 0x7E	{ } ~											X	X	X
0x7F	(DEL)	X												

<cstring> (string.h) kutubxonasi

Bu sarlavha faylida C tilidan vorislik bilan o'tgan satrlar va massivlar bilan ishlovchi bir qator funksiyalar majud. Quyida ularning tavsifini keltiramiz[10]:

<i>Nusxalash funksiyalari:</i>	
memcpy	xotiradagi blok nusxasini olish.
memmove	xotiraga blokni yuklash.
Strcpy	satrni nusxalash.
Strncpy	satrdagi belgi nusxasini olish.
<i>Satrlarni ulash funksiyalari:</i>	
strcat	satrlarni ulash.
strncat	satrga belgilarni qo'shish.
<i>Taqqoslash funksiyalari:</i>	
memcpy	xotiraning ikkita blokini taqqoslash.
stremmp	ikkita satrlarni taqqoslash.
strcoll	ikkita satrni taqqoslashda belgi o'rinlaridan foydalanish.
strncmp	ikkita satrlardagi belgilarni taqqoslash.
strxfrm	satrni belgilar to'plamiga o'tkazish.
<i>Izlash funksiyalari:</i>	
memchr	ko'rsatilgan belgini xotira blokidan izlash.
strchr	satrdan ko'rsatilgan belgining (birinchisining) joylashuv o'rnini

	aniqlash.
strncspn	satr va belgi orasidagi munosabatni qaytaradi.
strpbrk	satrdan belgilarni izlash.
strrchr	satrdan ko'rsatilgan belgining (oxirgisining) joylashuv o'rnini aniqlash.
strspn	ko'rsatilga belgini satrga joylashtiradi.
strstr	qism satrni izlash.
<i>Boshqa funksiyalari:</i>	
memset	xotiraga blokni to'ldirish.
strerror	satrdagi mavjud xatolikning ko'rsatkichini qaytaradi.
strlen	satr uzunligini qaytardi.
<i>Makroaniqlovlar:</i>	
NULL	bo'sh ko'rsatkich.
<i>Tiplar:</i>	
size_t	ishorasiz butun tur.

<wchar> (wchar.h) kutubxonasi.

Bu kutubxona C tilidagi uzun satrlar bilan ishlovchi bir qator funksiyalarni o'zida saqlaydi. Quyida funksiyalar tavsifini keltiramiz:

fgetwc	oqimdan ko'p sonli belgilarni o'qish.
fgetws	oqimdan uzun satrni o'qish.
fputwc	oqimga ko'p sonli belgilarni yozish.
fputws	oqimga uzun satrni o'qish.
fwide	oqimni oriyentasiya qilish.
fwprintf	oqimga sana formatidagi ma'lumotlarni yozish.
fwscanf	oqimdan formatlangan ma'lumotlarni o'qish.
getwc	oqimdan ko'p sonli belgilarni o'qish.
getwchar	stdin (klaviatura) faylidan sonli belgilarni o'qish.

putwc	oqimga ko'p sonli belgilarni yozish
putwchar	stdout fayliga (monitorga) belgilar massivini yozish.
swprintf	formatlangan ma'lumotlarni yozish uchun satrga chiqarish.
swscanf	formatlangan ma'lumotlarni satrdan o'qish.
ungetwc	oqimdan belgilar massivini yo'qotish.
vfwprintf	o'zgaruvchilar ro'yxatidagi formatlangan ma'lumotlarni o'lchamli buferga yozish.
vfwscanf	oqimdan formatlangan ma'lumotlarni o'zgaruvchilar ro'yxatiga o'qish.
vswscanf	formatlangan ma'lumotlarni uzun satrdan o'zgaruvchilar ro'yxatiga
vwprintf	o'zgaruvchilar ro'yxatidagi formatlangan ma'lumotlarni stdout (monitor) fayliga chiqarish.
wscanf	stdin faylidan formatlangan ma'lumotlarni o'qish.
<i>Umumiy utilitalar:</i>	
wcstod	uzun satrni <i>double</i> turiga o'tkazish.
wcstof	uzun satrni <i>float</i> turiga o'tkazish.
wcstold	uzun satrni <i>long double</i> turiga o'tkazish.
wcstol	uzun satrni <i>long integer</i> turiga o'tkazish.
wcstoll	uzun satrni <i>long long integer</i> turiga o'tkazish.
wcstoul	uzun satrni <i>unsigned long integer</i> turiga o'tkazish.
wcstoull	uzun satrni <i>unsigned long long integer</i> turiga o'tkazish.
<i>Belgi va satrlarni akslantirish funksiyalari:</i>	
btowc	kichik baytli belgini katta baytli belgiga o'tkazish.
mbrlen	multibayt belgi uzunligini aniqlash.
mbrtowc	multibayt belgi ketma ketligini massivga akslantirish.
mbsinit	konvertasita jarayonidagi cheklanishni aniqlash.
mbsrtowes	multibayt satrni satrlar massiviga akslantirish.
wctomb	belgilar massivini belgilarning multibayt ketma ketligiga

	akslantirish.
wctob	belgilar massivini kichik baytli belgilarga akslantirish.
wcstromb	uzun belgilar satrini multibaytli satrga akslantirish.
<i>Satrlar bilan ishlovchi funksiyalar:</i>	
wcscat	uzun satrlarni ulash.
wcschr	uzun satrda uchraydigan (birinchi) belgining joylashuv o'rnini aniqlaydi.
wcscmp	ikki marta satrni taqqoslash.
wcscoll	belgilarning joylashuv o'rniga nisbatan ikkita uzun satrni taqqoslash.
wcscopy	uzun satrni nusxalash.
wcscspn	uzun satrda ko'rsatilgan belgi uchraguncha bo'lgan uzunlikni qaytaradi.
wcslen	uzun satrning uzuzligini qaytaradi.
wcsncat	uzun satrga belgini qo'shish.
wcsncmp	ikkita uzun satrlarning belgilarini taqqoslash.
wcsncpy	uzun satrdadan belgilarini nusxalash.
wcspbrk	uzun satrdadan belgilarini izlash
wcsrchr	uzun satrdan oxirgi uchragan ko'rsatilgan belgini izlash.
wcsspn	uzun satrdan berilgan belgigacha bo'lgan belgilar to'plamini qaytaradi.
wcsstr	berilgan uzun satrlardan ko'rsatilgan qism satrini izlash.
wcsxfrm	uzun satrni joylashuv o'rniga nisbatan akslantirish.
wmemchr	belgini belgilar massivi blokidan izlash.
wmemcmp	ikkita belgilar massivi blokini taqqoslash.
wmemcpy	belgilar massivi blokini nusxalash.
wmemmove	belgilar massivi blokini yuklash.
wmemset	belgilar massivini to'ldirish.
<i>Vaqt bilan bog'liq funksiyalar:</i>	

wcsftime	uzun satr ko'rinishidagi vaqt formati.
<i>Turlar:</i>	
mbstate_t	multibayt akslantiruvchi turlar.
size_t	ishorasiz butun tur.
struct_tm	vaqt strukturasi.
wchar_t	belgilar massivi.
wint_t	uzun <i>int</i> turi.
<i>Makroo'zgarmaslar:</i>	
NULL	bo'sh ko'rsatkich.
WCHAR MAX	<i>wchar t</i> turning maksimal qiymati.
WCHAR MIN	<i>wchar t</i> turning minimal qiymati.
WEOF	uzun fayl oxiri.

<cwctype>(wctype.h) kutubxonasi

Bu kutubxona belgilar massivi uchun sinflovchi va akslantiruvchi qator funksiyalarni o'zida saqlaydi. Quyida funksiyalar tavsifini keltiramiz.

Katta xajmdagi ma'lumotlar to'planishi uchun C tizimida ASC kodlari jadvali sinflashtirilgan. Ular aniqlangan toifalar bo'yicha bir qancha hususiyatlarni aniqlaydi[3,10].

<i>Funksiyalar:</i>	
iswalnum	belgilar massivini alifbo harflari va raqamlaridan iboratligini tekshiradi.
iswalpha	belgilar massivini alifbo harflaridan iboratligini tekshiradi.
iswblank	belgilar massivini belgining son ekanligini tekshiradi
iswcntrl	belgilar massivini belgining haraktorli belgilaridan biri ekanligini tekshiradi

iswdigit	belgilar massivini belgining o'nlik sanoq sistemasidagi raqam ekanligini bildiradi
iswgraph	belgilar massivini belgining grafik tasvir harakteriga ega ekanligini bildiradi
iswlower	belgilar massivini belgining kichik harf ekanligini tekshiradi
iswprini	belgilar massivini belgining bosmalanuvchanlik haraktorini tekshiradi
iswpunct	belgilar massivini belgining punkitnatsiya belgilaridan biri ekanligini tekshiradi
iswspace	belgilar massivini belgining bo'sh joy ekanligini tekshiradi
iswupper	belgilar massivini belgining katta harf ekanligini tekshiradi
iswxdigit	belgilar massivini belgining o'n oltilik sanoq tizimidagi son ekanligini tekshiradi

Bundan tashqari satr xarflarining registrarlarni almashtiradigan ikkita funksiya xam mavjud:

towlower	belgilar massivini bosh harflarni kichik harflarga almashtiradi.
towupper	belgilar massivini kichik harflarni bosh harflarga almashtiradi

Kengaytirilgan sinflash va akslantirish funksiyasi:

iswctype	belgilar massivining kengaytirilgan formatda ekanligini tekshirish.
towctrans	akslantirishdan foydalanib konvertasiya qilish.
wctrans	belgining akslantirilgan qiymatini qaytarish.
wctype	belgining kamaytirilgan formatini qaytaradi.
Turlar:	
wctrans_t	belgilar massivini akslantirish turi.
wctype_t	belgining kamaytirilgan formatini qaytaradi.
wint_t	belgilar massivining butun turi.
O'zgarmaslar:	

WEOF	Uzun fayl oxiri.
-------------	------------------

2.3. Matematik funksiyalar bilan ishlovchi kutubxonalar

C++ tilidan foydalanuvchilar bu kutubxonadan juda keng foydalanadilar. `math.h` kutubxonasi turli matematik funksiyalar, makrokonstantalar va matematik turlar bilan boyitilgan. Quyida biz ularning tavsifini keltiramiz:

***math.h* kutubxonasining funksiyalari**

<i>Funksiya</i>	<i>Bajaradigan amali</i>
<i>Trigonometrik funksiyalar:</i>	
cos	Radianga teng bo'lgan burchakning kosinusini qaytaradi
sin	Radianga teng bo'lgan burchakning sinusini qaytaradi
tan	Radianga teng bo'lgan burchakning tangensini qaytaradi
acos	Radianda berilgan argumentning arkkosinus qiymatini qaytaradi
asin	Radianda berilgan argumentning arksinus qiymatini qaytaradi
atan	Radianda berilgan argumentning arktangens qiymatini qaytaradi
atan2	Radianda berilgan x/y argumentning arktangens qiymatini qaytaradi
<i>Giperbolik funksiyalar:</i>	
cosh	Radianda berilgan burchakning giperbolik kosinusini qaytaradi
sinh	Radianda berilgan burchakning giperbolik sinusini qaytaradi
tanh	Radianda berilgan burchakning giperbolik tangensini qaytaradi
acosh	Radianda berilgan burchakning giperbolik arkkosinusini qaytaradi
asinh	Radianda berilgan burchakning giperbolik arksinusini qaytaradi

atanh	Radianda berilgan burchakning giperbolik arktangensini qaytaradi
<i>Logarifmik funksiyalar</i>	
exp	e^x qiymatni qaytaradi
frexp	Sonning mantissasini va darajasining qiymatini ajratib, mantissa qiymatini qaytaradi va darajasini ko'rsatilgan adresga joylashtiradi
ldexp(x)	$x * 2^{\text{exp}}$ qiymatni qaytaradi
log	Sonning natural logarifmini qaytaradi
log10	Sonning 10 asosli logarifmini qaytaradi
modf	Haqiqiy sonning butun va haqiqiy qismlarini ajratish
exp2	Funksiyaning ikkilik ko'rsatkichini aniqlash
expm1	-1 ko'rsatkich bilan hisoblah
ilogb	Ikkita predmetdan iborat nabordagi butun son logarifmi
log1p	Logarifmni +1 ko'rsatkich bilan hisoblash
log2	Ikkilik logarifmni hisoblash
logb	Logarifmni suzuvchu vergulli son asosida hisoblash
<i>Darajaga ko'tarish funksiyalari</i>	
pow	Darajaga ko'taradi
pow10	10 sonining darajasini hisoblaydi
sqrt	Sondan kvadrat ildiz chiqaradi
cbrt	Sondan kub ildiz chiqaradi
hypot	to'g'ri burchakli uchburchakning katetlari bo'yicha gipotenuzasini hisoblaydi
<i>Xatolik funksiyalari va gamma funksiyalari</i>	
erf	Xatolik funksiyasini hisoblash
erfc	Qo'shimcha xatolik funksiyasini hisoblash
tgamma	Gamma funksiyani hisoblash
lgamma	Ro'yxatdan o'tkazuvchi gamma funksiyani hisoblash

<i>Yaxlitlash va qoldiqni hisoblash funksiyalari:</i>	
ceil	haqiqiy qiymatni unga eng yaqin katta butun songacha aylantiradi va uni haqiqiy ko'rinishini qaytaradi
floor	haqiqiy qiymatni eng yaqin kichik songa aylantiradi va uni haqiqiy son ko'rinishida qaytaradi
fmod	Ikkita sonni bo'lish natijasidagi qoldiqni qaytaradi. % amaliga o'xshagan, faqat haqiqiy son qaytaradi
trunc	Sonni yaxlitlash
round	Sonni yaqin butun songa yaxlitlash
lround	Uzun sonni yaxlitlash
llround	Ikkilik uzun sonni yaxlitlash
rint	Butun songa yaxlitlash
lrint	Uzun butun songa yaxlitlash
llrint	Ikkilik uzun butun songa yaxlitlash
nearbyint	Butun songa yaqin bo'lgan qiymat bilan yaxlitlah
remainder	Qoldiqni hisoblash
remquo	Qoldiqni va faktorni hisoblash
<i>Absolyut qiymatni hisoblash funksiyalari</i>	
abs	Sonning absolyut qiymatini qaytaradi
fabs	Haqiqiy sonning absolyut qiymatini qaytaradi
labs	Uzun butun sonning absolyut qiymatini qaytaradi
<i>Turlar</i>	
<u>double t</u>	Haqiqiy sonlar turi
<u>float t</u>	Haqiqiy sonlar turi

2.4. Vaqt va sana bilan bog'liq kutubxonalar

<ctime> (time.h) kutubxonasi

Bu kutubxonaga vaqt va sanaga bog'liq bo'lgan ma'lumotlarni olish va ularni boshqarish funksiyalariga ega. C++ tilining <time.h> kutubxonasida quyidagi turlar mavjud: *clock_t*, *time_t* va *tm*. *clock_t* va *time_t* turlari yil taqvimini bo'yicha vaqt va sananing butun turdagi qiymatlari majmuasini saqlaydi. *tm* strukturasi esa bir qator komponentalarga ajraladi va quyidagicha aniqlanadi:

```

struct tm {
    int tm_sec;           /*sekundlar, 0-59*/
    int tm_min;          /*minutlar, 0-59*/
    int tm_hour;         /*soatlar, 0-23*/
    int tm_day;          /*oy kunlari, 1-31*/
    int tm_mon;          /*oylar, 0-11 */
    int tm_year;         /*yillar, 1900 yildan boshlanadi*/
    int tm_wday;         /*hafta kunlari, 0-6*/
    int tm_yday;         /*yil kunlari, 0-365*/
    int tm_isdst;        /*yozgi vaqt indekatori*/
}

```

Bu strukturadagi yozgi vaqt indekatori yozgi sanalar uchun noldan farqli, aks holda nol qiymatiga ega bo'ladi. Bundan tashqari <time.h> kutubxonasida CLOCKS_PER_SEC makrosi aniqlangan bo'lib, tizim soatining sekundlardagi taktilar sonini qaytaradi.

Quyida bu kutubxonaning standart funksiyalari tavsifini keltiramiz:

<i>Funksiya</i>	<i>Bajaradigan amali</i>
clock	soat dasturi
difftime	ikkita turli vaqtlar oralig'ini qaytaradi.
mktime	vaqtni <i>tm</i> strukturasiidan <i>time_t</i> turiga o'tkazadi.
time	joriy vaqtni qaytaradi.
<i>Akslantirish funksiyalari:</i>	
asctime	vaqtni <i>tm</i> strukturasiidan satr turiga o'tkazadi.

ctime	<i>time_t</i> turidagi qiymatni satr turiga o'tkazadi.
gmtime	vaqtning <i>tm_t</i> turini xuddi <i>UTS</i> ko'rinishida kabi <i>tm</i> strukturasi ga o'tadi.
localtime	vaqtning <i>tm_t</i> turidan mahalliy vaqt ko'rinishidagi <i>tm</i> satrga o'tadi.
strftime	vaqt formatini satr ko'rinishida ifodalaydi.
<i>Makro o'zgarmlar:</i>	
CLOCKS PER SEC	soatni sekundlarda ifodalash.
NULL	bo'sh ko'rsatkich.
<i>Turlari:</i>	
clock_t	soat turi
size_t	ishorasiz butun tur.
time_t	vaqt turi
struct_tm	vaqt strukturasi.

III BOB. DASTURIY VOSITALAR QURISHDA C++ TILI KUTUBXONALARIDAN FOYDALANISH

3.1. Belgilar va satrlar bilan ishlovchi kutubxonalardan foydalanish

Ushbu paragrafda C va C++ dasturlash tillarida mavjud `<string.h>` va `<ctype.h>` (C tilida `<cstring>` va `<cctype>`) kutubxonalarining imkoniyatlarini yoritish ya'ni belgi va satrlarga bog'liq masalalar uchun dasturlar qurishda bu kutubxonalardan foydalanish qaraladi. Bizga ma'lumki C++ tilida tuzilgan har qanday dasturda foydalaniladigan kutubxonalar `#include` kalit so'zi bilan chaqirilishi shart. Shundagina chaqirilgan kutubxonalar imkoniyatlaridan foydalanish mumkin bo'ladi.

C++ tilida belgi va satrlarga turli ishlov berish jarayonlari muhim qo'llanmalarga ega bo'lganligi sababli ularni batafsil o'rganish ahamiyatli hisoblanadi. Keltirilgan kutubxonalarda mavjud funksiyalardan foydalanib berilgan satrdagi harflar, raqamlarga bog'liq masalalar uchun dasturlar tuzish imkoniyati tug'iladi. Ko'p hollarda satrlar bilan ishlash uchun `<string.h>` kutubxonasidan foydalanish ham maqsadga muvofiq. Har ikkala kutubxonaga tegishli bo'lgan bir qator funksiyalar mavjud. Endi bu funksiyalar yordamida dasturlanadigan masalalardan bir misol keltiramiz.

Masala. Berilgan satrdagi kichik va katta lotin harflarining va raqamlarning miqdorini hisoblab beruvchi dastur qurisin.

```
#pragma hdrstop
```

```
#include <condefs.h>
```

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

```
#include <ctype.h>
```

```
#include <string.h>
```

```
//-----
```

```

#pragma argsused

int main(int argc, char **argv)
{
    string s;

    int l,i,n1=0,n2=0,n3=0;

    cin>>s;

    l=s.length();

    for (i=0;i<l;i++)
    {
        if(isdigit(s[i])) n1+=1;

        if (islower(s[i])) n2+=1;

        if (isupper(s[i])) n3+=1;
    }

    cout<<"Satrdagi raqamlar miqdori "<<n1<<" ta\n" ;

    cout<<"Satrdagi kichik harflar soni"<<n2<<" ta\n";

    cout<<"Satrdagi katta harflar soni"<<n3<<" ta";

    cin>>"\n";

    return 0;

}

```

Ushbu dastur bajarilgandan so'ng quyidagi natijalar olindi:

Salom maktab!

Satrdagi raqamlar miqdori 0 ta

Satrdagi kichik harflar miqdori 10 ta

Satrdagi katta harflar miqdori 1 ta

Yuqorida keltirilgan kabi masalalarni yechish matnlarga ishlov berish jarayonini yanada takomillashtirish hamda axborotlar himoyasi sohasida mavjud dolzarb masalalarni hal etishda o'zining muhim tatbiqlariga ega.

3.2. Dasturlashda math.h kutubxonasining imkoniyatlaridan foydalanish

C++ tilining math.h kutubxonasi yordamida matematik masalalarni dasturlash jarayonini ko'rib chiqamiz.

Bu kutubxonadagi mavjud matematik funksiyalarni quyidagi to'rt guruhga bo'lishimiz mumkin:

- Trigonometrik funksiyalar
- Giperbolik funksiyalar
- Eksponensial va logarifmik funksiyalar
- Boshqa turdagi matematik funksiyalar.

math.h kutubxonasidan foydalanishni namoyish etish uchun quyidagi dasturni keltiramiz.

Masala. n musbat butun va a haqiqiy sonlari berilgan bo'lsa, y ning qiymati quyidagicha hisoblansin:

$$y = \begin{cases} a^2, & \text{agar } n = 1 \text{ bo'lsa;} \\ th(a), & \text{agar } n = 2 \text{ bo'lsa;} \\ \arccos(a), & \text{agar } n = 3 \text{ bo'lsa;} \\ \lg(a), & \text{agar } n = 4 \text{ bo'lsa;} \\ 0, & \text{boshqa hollarda.} \end{cases}$$

```
#pragma hdrstop
```

```
#include <condefs.h>
```

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



```

#include <math.h>

//-----

#pragma argsused

int main(int argc, char **argv)
{
int n;

float a,x;

cin>>a>>n;

switch (n)
{
case 1: x=pow(a,2); break;

case 2: x=tanh(a); break;

case 3: x=acos(a); break;

case 4: x=log10(a); break;

default: x=0;

}

cout<<x;

cin>>"\n";

return 0;

}

```

Ushbu dastur bajarilgandan so'ng quyidagi natijalar olindi:

Misol 1:	Misol 2:	Misol 3:
67 4	45 1	34 3
1.82607	2025	acos: DOMAIN error
		nan

Yuqorida keltirilgan dastur yordamida ko'rsatilgan funksiyaga mos qiymatni hisoblash imkoniyati tug'iladi. Dasturlash tilidagi bunday imkoniyat aniq va tabiiy fanlarning turli yo'nalishlarida o'z sohasiga bog'liq funksiyalarni hisoblash uchun foydalanuvchilarga o'zining funksiyalar kutubxonasini hosil qilish imkoniyatini beradi.

3.3. Dasturiy vositalarda vaqt va sanaga bog'liq funksiyalarni qo'llash

Bir qator dasturiy vositalar qurishda vaqt va sanaga bog'liq ma'lumotlar bilan ishlashga to'g'ri keladi.

Kutubxona imkoniyatlarininamoyish etish maqsadida quyida keltirilgan masala uchun dasturiy vasita quramiz.

Masala. Funksiyaning bajarilish vaqtini aniqlovchi dastur tuzilsin.

Algoritmning samarali ishlashini aniqlash uchun unga sarflanadigan mashina vaqtini bilish muhim faktorlardan hisoblanadi. Ma'lumki C++ tilida dasturlar bir necha funksiyalar majmuasidan iborat bo'lganligi sababli yaxlit dasturga emas, balki funksiyalarning bajarilishiga ketadigan vaqtni aniqlash yetarli bo'ladi. Yuqorida qo'yilgan masala shu maqsadda keltirilgan.

Qo'yilgan masalaga oydinlik kiritish uchun N musbat butun sonning barcha butun bo'luvchilarini aniqlash masalasiga tuzilgan 2 xil algoritmdagi funksiyani taqqoslaymiz.

```
#pragma hdrstop
```

```
#include <condefs.h>
```

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

```

#include <time.h>

//-----

#pragma argsused

int Buluvchi_1(unsigned int k)

{ int m=0;

for (int i=1;i<=k;i++)

if (k%i==0) m+=1;

return m;

}

int Buluvchi_2(unsigned int k)

{ unsigned int i,m=1, n=0;

if (k%2) m=2;

for (i=1;i<=k/2;i=i+m)

if (k%i==0) n+=1;

return n+1;

}

int main(int argc, char **argv)

{ time_t t,t0,t1,t2;

unsigned int N,i,m;

cin>>N;

t0=time(&t);

m=Buluvchi_1(N); t1=time(&t);

```

```

cout<<"N sonining bo'luvchilari soni "<<m<<" ta ";

cout<<"1-algoritmning bajarilishiga ketgan vaqt: "<<t1-t0<<"\n";

m=Buluvchi_2(N); t2=time(&t);

cout<<"N sonining bo'luvchilari soni "<<m<<" ta";

cout<<"2-algoritmning bajarilishiga ketgan vaqt "<<t2-t1;

cin>>"\n";

    return 0;

}

```

Ushbu dastur bajarilgandan so'ng quyidagi natijalar olindi:

345

N sonining bo'luvchilari soni 8 ta 1-algoritmning bajarilishiga ketgan vaqt: 73

N sonining bo'luvchilari soni 8 ta 1-algoritmning bajarilishiga ketgan vaqt: 22

Keltirilgan dasturdagi funksiyalarning bajarilishiga ketgan vaqt bo'luvchilari miqdorini aniqlash lozim bo'lgan son yetarlicha katta bo'lganda 2 xil ko'rinishga ega bo'ladi. 2-funksiya 1-funksiyaga qaraganda ancha kam vaqt sarflaydi. Chunki unda bajariladigan amallar soni kam. Keltirilgan masala ancha sodda bo'lganligi sababli uni oq quti usuliga asoslanib tekshirish mumkin.

Masala. Shoshqaltosh o'yini mashhur o'yinlardan biridir. O'yin qoidasi quyidagicha: Ishtirokchi 2 ta toshni tashlaydi. Har bir tosh 6 ta tomondan iborat va ular 1,2,3,4,5 va 6 qiymatlaridan biriga ega. Toshlar tushgandan so'ng ularning yuqori qismlarining qiymatlari yig'indisi hisoblanadi. Agar birinchi otishdan so'ng yig'indi 7 yoki 11 ga teng bo'lsa, ishtirokchi yutgan hisoblanadi. Agar birinchi otishda yig'indi 2,3 yoki 12 ga teng bo'lsa, ishtirokchi yutqazgan hisoblanadi. Agar birinchi otishda yig'indi 4,5,6,8,9 va 10 qiymatlaridan biriga teng bo'lsa, bu qiymat "Ochko" hisoblanadi. Yutish uchun ishtirokchi toshlarni o'zi yiqqan

ochkolari qiymatiga teng son chiqmaguncha tashlaydi. Agar bu otishlar vaqtida toshlar yig'indisi 7 ga teng holat sodir bo'lsa, ishtirokchi yutqazadi.

Quyida ushbu o'yinning dasturiy vositasini keltiramiz:

```
#include <vcl.h>

#pragma hdrstop

#include<iostream.h>

#include<stdlib.h>

#include<time.h>

//-----

#pragma argsused

int rollDiece(void);

int main(int argc, char* argv[])

{

enum Status {CONTINUE,WON,LOST };

int sum, myPoint;

Status gameStatus;

srand(time(NULL));

sum=rollDiece();

switch(sum) {

case 7,11: gameStatus=WON; break;

case 2,3,12: gameStatus=LOST; break;
```

```

default: gameStatus=CONTINUE;

myPoint=sum;

cout<<"Ochkolar: "<<myPoint<<endl;

break;

}

while (gameStatus==CONTINUE) {

sum=rollDiece();

if (sum==myPoint)

    gameStatus=WON;

    else if (sum==7)

        gameStatus=LOST;

    }

if (gameStatus==WON)

cout<<"Ishtirokchi yutdi"<<endl;

else cout<<"Ishtirokchi yutqazdi"<<endl;

cin>>"\n";

return 0;

}

int rollDiece(void){

int diel,diel2,worksum;

diel=1+rand()%6;

diel2=1+rand()%6;

```

```

worksum=d1el+d1el2;

cout<<"Ishtirokchi tashladi "<<d1el<<"+ "<<d1el2<<" = "<<worksum<<endl;

return worksum;

}

```

Ushbu dastur bajarilishi natijasida quyidagi natijalar olindi:

Ishtirokchi tashladi $6 + 2 = 8$

Ochkolar: 8

Ishtirokchi tashladi $3 + 2 = 5$

Ishtirokchi tashladi $4 + 2 = 6$

Ishtirokchi tashladi $2 + 3 = 5$

Ishtirokchi tashladi $6 + 6 = 12$

Ishtirokchi tashladi $1 + 6 = 7$

Ishtirokchi yutqazdi

.....

Ishtirokchi tashladi $3 + 1 = 4$

Ochkolar: 4

Ishtirokchi tashladi $3 + 1 = 4$

Ishtirokchi yutdi

.....

Ishtirokchi tashladi $6 + 5 = 11$

Ishtirokchi yutdi

.....

Bu kabi murakkab algoritmlar asosida quriladigan masalalar uchun algoritmning bajarilish vaqtini hisoblash ya'ni qora quti usulini qo'llash yaxshi samara beradigan usuldir. C++ tilining `<time.h>` kutubxonasidan foydalanish vaqtga bog'liq bo'lgan yuqoridagi kabi masalalarni samarali hal etish imkoniyatini beradi.

XULOSA

Ushbu bitiruv ishida qaralgan C++ tilining kutubxonalarini va ularning programma tuzishdagi ahamiyati mavzusi ostida qo'yilgan masalalarni o'rganish bo'yicha quyidagilar bajarildi:

1. C++ dasturlash tilining imkoniyatlari o'rganib chiqilib, ishga aloqador va ishga muhim deb hisoblanganlari ish tarkibiga kiritildi.
2. C++ tilining bir qator kutubxonalarini va ularda mavjud imkoniyatlar batafsil o'rganib chiqildi va ish tarkibiga kiritildi.
3. Amaliy dasturlar qurish yo'li bilan C++ tilining <ctype.h>, <string.h>, <math.h> va <time.h> kutubxonalaridan samarali foydalanish yo'llari ko'rsatib berildi.

Ushbu ishda keltirilgan ma'lumotlardan C++ tilining kutubxonalariga bog'liq imkoniyatlarini qo'llashni yo'lga qo'yishni, ya'ni kengroq o'rganishni istovchilarning foydalanishlari yaxshi natija beradi deb hisoblaymiz.

FOYDALANILGAN ADABIYOTLAR VA INTERNET RESURSLARI

1. I. A. Karimov “Yuksak ma’naviyat yengilmas kuch.” T. Ma’naviyat, 2010 y. 173 s.
2. I. A. Karimov “O’zbekiston mustaqillikka erishish ostonasida” T.O’zbekiston 2011 y. 432 s.
3. Г.Шилдт –“Полный справочник по С++” – М-2006., 801 стр.
- 4.Р.Седжвик – “Фундаментальные алгоритмы на С++” – М 2001., 687 стр.
5. Ш.Ф.Мадрахимов, С.М.Гайназаров “С++ тилида программалаш асослари” – Тошкент-2009 й., 196
6. М.Э.Абрамян “Электронный задачник по программированию” Ростов - на - Дону 2005 г. 182 стр.
7. А.А.Халджигитов, Sh.F.Madraximov, U.E.Adambayev, E.A Eshboyev, Informatika va dasturlash. T.:O‘zMU, 2005 y, -148 s.
8. Н.Н.Непейвода- Стили и методы программирования. Интернет университет информационных технологий. INTUIT.ru, 2005 г., 320 стр.
9. О.М.Shukurov, E.A.Eshboyev, B.H.Shovaliyev – “Delphi va C++ algoritmik tillarida dasturlash” – Qarshi-2012 y., 228 s.

[10.www.cplusplus.com](http://www.cplusplus.com)

[11.www.acm.timus.ru](http://www.acm.timus.ru)

[12.www.codeforces.ru](http://www.codeforces.ru)

[13.www.cyberforum.ru](http://www.cyberforum.ru)

[14.www.cybern.ru](http://www.cybern.ru)

[15.www.delphisources.ru](http://www.delphisources.ru)

[16.ziyonet.uz](http://ziyonet.uz)