

**O'ZBEKISTON RESPUBLIKASI  
OLIY VA O'RTA-MAXSUS TA'LIM VAZIRLIGI**

**SAMARQAND DAVLAT UNIVERSITETI**

**MEXANIKA-MATEMATIKA FAKULTETI**

**AMALIY MATEMATIKA VA INFORMATIKA BO'LIMI**

**5110700-INFORMATIKA O'QITISH METODIKASI YO'NALISHI**

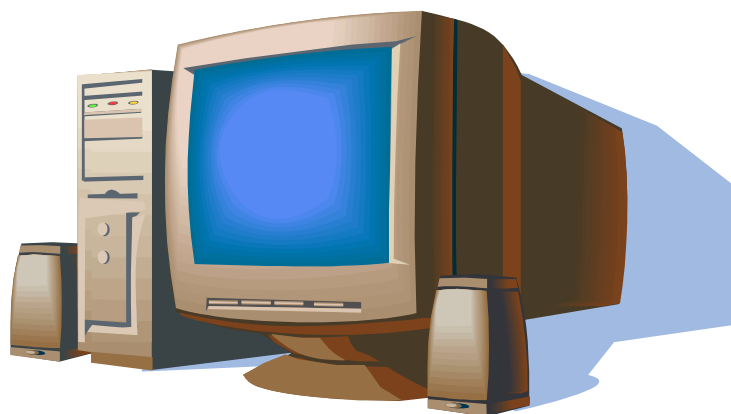
**206-guruh talabasi Achilova Umidaning**

**ALGORITMLASH VA DASTURLASH TILLARI FANIDAN**

**C++ TILIDA TARMOQLANUVCHI JARAYONLAR VA ULARNI  
DASTURLASH MAVZUSIDAN**

# **KURS ISHI**

---



**Kurs ishi rahbari: INATOV A.**

**SAMARQAND-2016**

**MUNDARIJA:**

## **KIRISH**

### **I. C++ dasturlash tilida arifmetik amallarni bajaruvchi dasturlar ustida nazariy malumotlar**

*1.1.C++ dasturlash tili haqida tushuncha.*

*1.2. Matematik funksiyalar va standart kutubxonalar*

### **II. C++ dasturlash tilida arifmetik amallarni bajaruvchi dasturiy ta'minot loyihasi**

*2.1.Tarmoqlanuvchi jarayonlarga oid dastur tuzish.*

### **III. Xulosa**

### **IV. Foydalanilgan adabiyotlar**

## KIRISH

Birinchi elektron hisoblash mashinalari paydo bo'lishi bilan dasturlash tillari evolyutsiyasi boshlanadi. Dastlabki kompyuterlar ikkinchi jahonurushi vaqtida artilleriya snaryadlarining harakat traektoriyasini hisob kitob qilish maqsadida qurilgan edi. Oldin dasturchilar eng sodda mashina tilini o'zida ifodalovchi kompyuter komandalari bilan ishlaganlar. Bu komandalar nol va birlardan tashkil topgan uzun qatorlardan iborat bo'lar edi. Keyinchalik, insonlar uchun tushunarli bo'lgan mashina komandalarini o'zida saqllovchi (masalan, ADD va MOV komandalari) assembler tili yaratildi. Shu vaqtlarda BASIC va COBOL singari yuqori sathli tillar ham paydo bo'ldiki, bu tillar tufayli so'z va gaplarning mantiqiy konstruksiyasidan foydalanib dasturlash imkoniyati yaratildi. Bu komandalarni mashina tiliga interpretatorlar va kompilyatorlar ko'chirar edi. Interpretator dasturni o'qish jarayonida uning komandalarini ketma - ket mashina tiliga o'tkazadi. Kompilyator esa yaxlit programma kodini biror bir oraliq forma - ob'ekt fayliga o'tkazadi. Bu bosqich kompilyatsiya bosqichi deyiladi. Bundan so'ng kompilyator ob'ektli faylni bajariluvchi faylga aylantiradigan kompanovka dasturini chaqiradi. Interpretatorlar bilan ishlash osonroq, chunki dastur komandalari qanday ketma - ketlikda yozilgan bo'lsa shu tarzda bajariladi. Bu esa dastur bajarilishini nazorat qilishni osonlashtiradi. Kompilyator esa kompilyatsiya va kompanovka kabi qo'shimcha bosqichlardan iborat bo'lganligi uchun ulardan hosil bo'ladigan bajariluvchi faylni tahlil qilish va o'zgartirish imkoniyati mavjud emas.

## **1-§. C++ dasturlash tilida arifmetik amallarni bajaruvchi dasturlar ustida nazariy malumotlar**

### **1.1 C++ dasturlash tili haqida tushuncha**

C++ funksiya va ob'ektlarning 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 ob'ekt funksiyalarni qo'llashni o'rganishdir.

#### ***Dasturiy zaxiralarga talablar***

C++ bu yuqori sifatga ega bo'lgan dasturlar yaratish vositasidir. C++ni yuklash uchun kamida 4MB tezkor xotiraga ega bo'lgan 386 kompyuter kerak bo'ladi. Lekin normal ishlash uchun kamida 8MB tezkor xotiraga ega bo'lgan 66MHz li 486DX kompyuteri to'g'ri keladi.. C++da yaratilgan uncha katta bo'lmagan dasturlar ixtiyoriy kompyuterda ishlay oladi. Boshqacha qilib aytganda ularga C++ muhiti uchun zarur bo'lgan tezkor xotira va protsessor tezligi talab qilinmaydi.

#### ***Dasturlash muxiti tuzilishi***

C++ dasturlash muhitining tashqi ko'rinishi Windowsda ko'rish mumkin bo'lgan boshqa dasturlash muhitlari ko'rinishidan farq qiladi. Masalan, Borland Pascal for Windows 7.0, Word for Windows, Program Manager – bularning barchasi MDI dasturlari hisoblanadi va C++ o'xshamaydi. MDI (Multiple Document Interface) – bitta asosiy oynaning ichida bir nechta quyi tipdagi oynalarni boshqarishni bildiradi.

C++ muhiti esa, Single Document Interface (SDI) deb nomlanuvchi boshqa xususiyatga amal qiladi va alohida joylashgan bir nechta oynadan iborat.

Agar siz C++ga o'xshash SDI dasturlardan foydalansangaz, boshqa dasturlar oynalari uni yopib qo'ymasligi uchun ularni pastga tushirib qo'yish kerak. Agar boshqa dasturga o'tish zaruriyati tug'ilib qoladigan bo'lsa, C++ dasturi bosh oynasidagi kichraytirish tugmasini bosish kifoya.

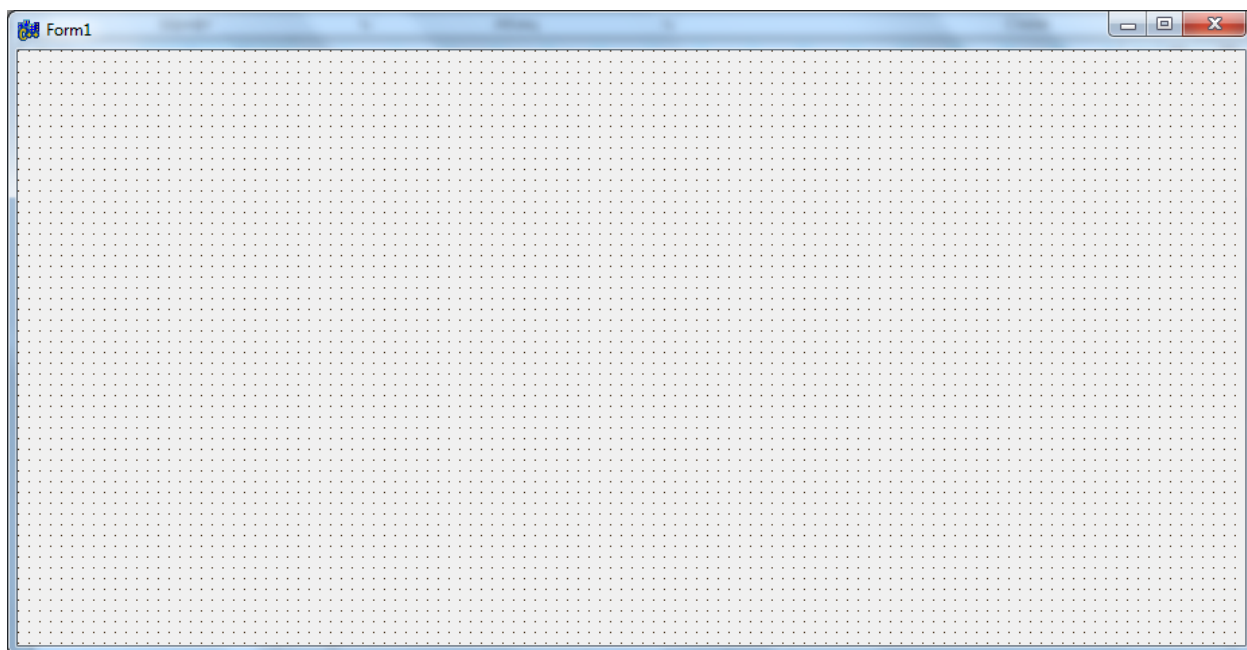
### ***Dasturlash muxitining asosiy tashkil etuvchilar.***

Quyida C++ning asosiy tashkil etuvchilari keltirilgan:

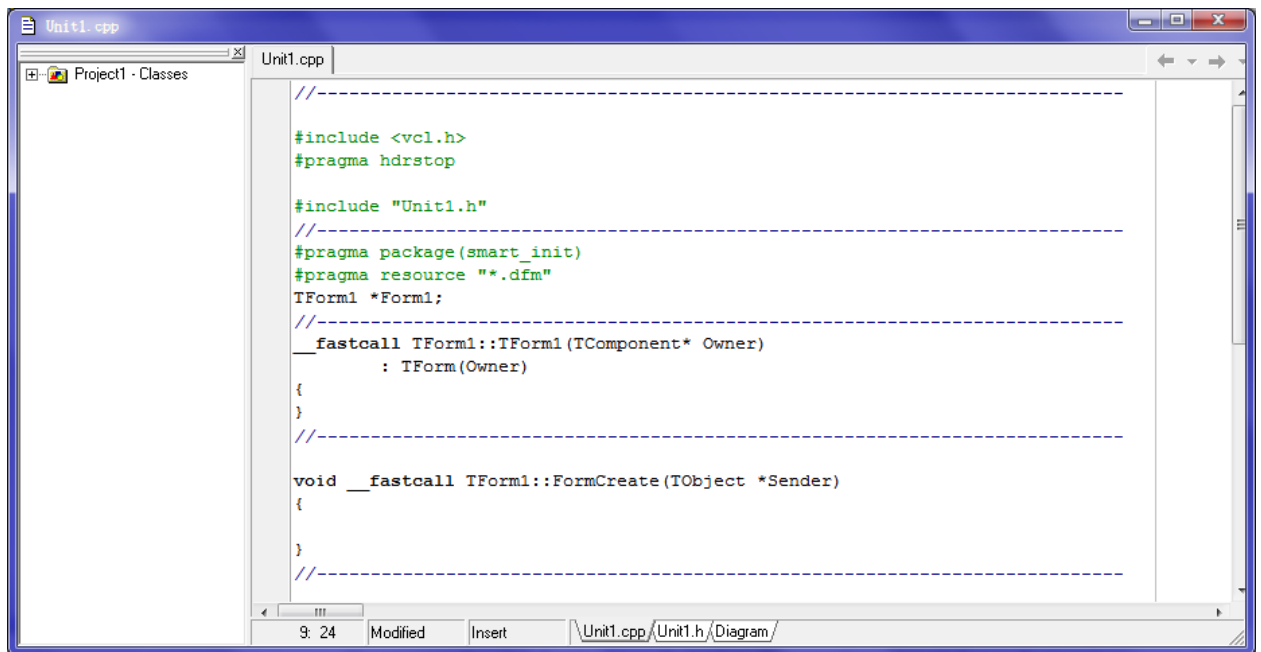
1. Formalar dizayneri (FormDesigner)
2. Dastur matni muharriri (Editor Window)
3. Komponentlar to'plami (Component Palette)
4. Ob'ektlar inspektori (Object Inspector)
5. Yordam tizimi (On-line help)

C++ning bulardan tashqari sizga dasturlash muhiti va dasturni sozlash uchun xizmat qiluvchi tizim menyusi, uskunalar chizgichi kabi boshqa vositalari ham bor.

C++da dasturchilar ko'p vaqtlarini Formalar dizayneri va Dastur matni muharriri orasida o'tish bilan o'tkazadilar. Siz dasturlashni boshlashdan oldin shu ikki muhim elementni ajrata olishingizga ishonch hosil qiling. Formalar dizayneri 1-rasmda, Dastur matni muharriri 2-rasmda keltirilgan.



1-rasm. Formalar dizayneri.



2-rasm. Dastur matni muharriri

C++dagi Formalar dizaynerining tuzilishi shu qadar oddiyki, dasturning tashqi ko‘rinishini yaratish bolalar o‘yiniga o‘xshab ketadi. Formalar dizayneri dastlab oddiy bo‘sh oynadan iborat bo‘ladi va siz uni keyinchalik komponentlar to‘plamidan olingan ob‘ektlar bilan to‘ldirishingiz mumkin.

Formalar dizaynerining muhimligiga qaramasdan, dasturchilarning ko‘p vaqtlari Dastur matni muharririda kechadi. Chunki bu erda dasturning asosiy kodi (ish bajaruvchi qismi) yoziladi.

Komponentlar to‘plami sizga formalar dizayneriga o‘rnatish va dasturning tashqi ko‘rinishida ishtirok etuvchi ob‘ektlarni tanlash uchun xizmat qiladi. Komponentlar to‘plamidan foydalanish uchun avval kerakli ob‘ekt ustiga sichqonchani bosing va so‘ngra formalar dizayneriga sichqonchani bosish bilan tanlangan ob‘ektni formaga qo‘yishingiz mumkin.

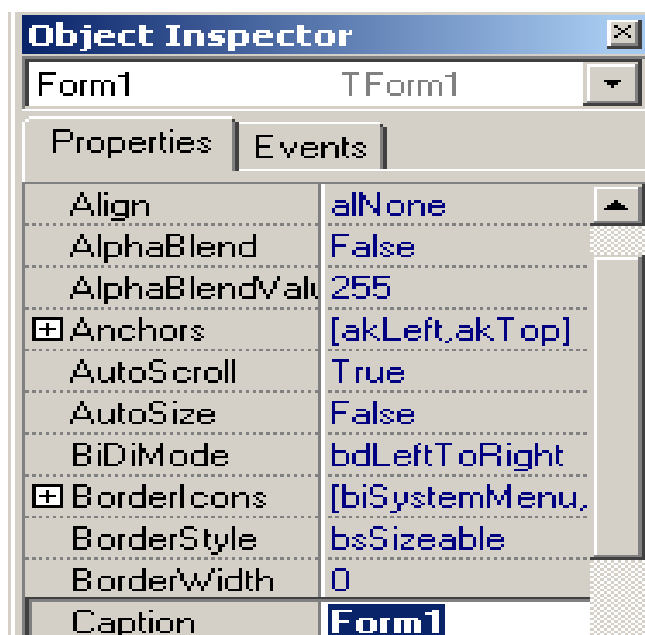


3- rasm. Komponentlar to‘plami.

Komponentlar to'plamida ob'ektlar guruhlar buyicha alohida sahifalarga bo'lingan. To'plamda Standard, Additional, Dialogs va hokazo sahifalar yig'ilgan. Agar siz bu nomlarning ixtiyoriysini bosingiz, shu guruh ob'ektlari joylashgan sahifaga o'tishingiz mumkin.

Masalan, siz formaga TEdit ob'ektini tashladingiz; siz uni bir joydan ikkinchi joyga ko'chirishingiz va uning o'lchamlarini o'zgartirishingiz mumkin. Boshqa ko'pchilik ob'ektlar bilan ham xuddi shunday ishlarni amalga oshirish mumkin. Lekin, dastur bajarilishida formada ko'rinmaydigan (TMenu yoki TdataBase kabi) ob'ektlar o'z ko'rinishini o'zgartirmaydi.

Formalar dizayneridan chap tomonda siz ob'ektlar inspektorini ko'rishingiz mumkin. Bu oynadagi ma'lumotlar formada tanlangan ob'ektning tipiga qarab, o'zgarib borishiga e'tibor bering. Har bir komponent bu ob'ektdir, va uning ko'rinishi hamda harakatini ob'ektlar inspektori yordamida o'zgartirish mumkinligini unutmang!



4-rasm. Ob'ektlar inspektori.

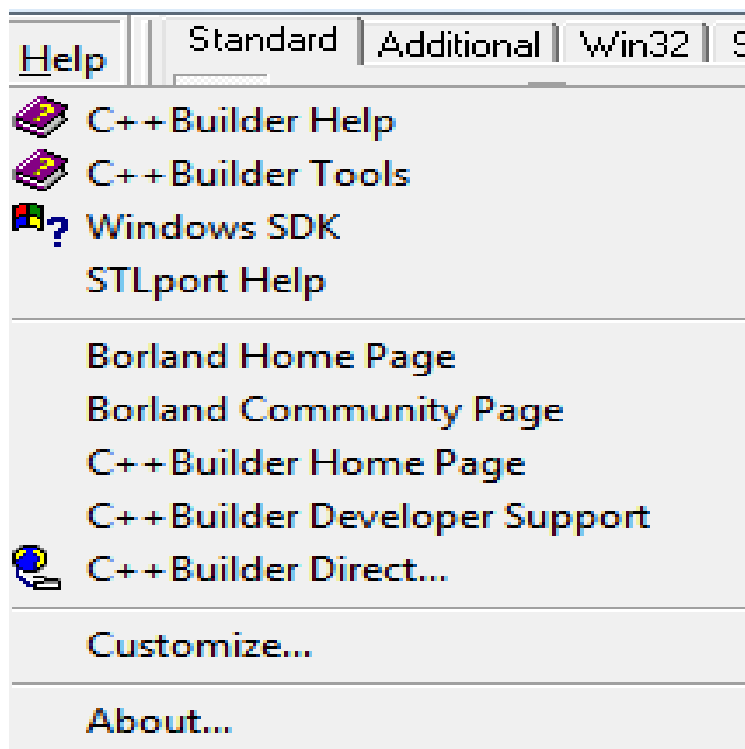
Ob'ektlar inspektori joriy ob'ektning harakatlarini aniqlash uchun ikki sahifadan tashkil topgan. Birinchi sahifa – xususiyatlar ro'yxati bo'lsa, ikkinchisi ob'ektning holatlari ro'yxatidir. Joriy ob'ektning biror tomonini o'zgartirish kerak bo'lsa, siz albatta Ob'ektlar inspektoriga murojaat qilishingiz kerak. Misol uchun

siz TLabel elementining nomini o'zgartirishingiz, yoki uning Caption, Left, Top, Height, va Width kabi xususiyatlari yordamida joylashishini o'zgartirishingiz mumkin.

Ob'ektlar inspektori oynasining quyi qismidagi sahifa nomlaridan ob'ektlarning xususiyatlari va holatlari oynasiga o'tish mumkin. Holatlar sahifasi dastur matni muharriri bilan bevosita bog'liq bo'lib, undagi holatlarning ixtiyoriysiga sichqonchani ikki marta bossak, ob'ektning shu holatiga tegishli dastur sarlavhasi muharrirda paydo bo'ladi va sizga shu holatda bajariladigan ishlarni dastur ko'rinishida yozish imkoniyati beriladi. C++ bu imkoniyatini keyinoq to'liq ko'rib o'tamiz.

### ***C++dasturlash muxitining yordam tizimi***

Muhiti C++ning oxirgi muhim qismi bu yordam tizimidir (on-line help). Bu vositaga murojaat qilish uchun avval menyu satridan Help so'ngra Contents buyug'ini tanlash kerak. Ekranda 5-rasmda ko'rsatilgan - **ixtiyoriy ma'lumotni tez topish yordam tizimi** namoyon bo'ladi.



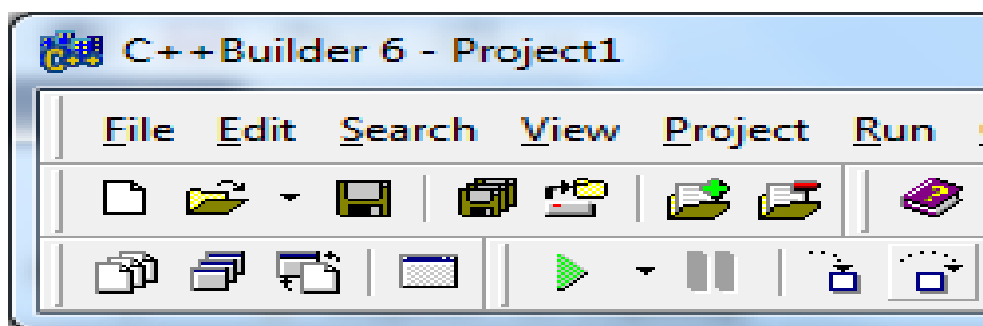
5-rasm. C++ yordam tizimi

Yordam tizimi joriy holatga bog‘liq hisoblanadi, ya’ni siz F1 tugmasini bosishingiz bilan joriy holatga mos ma’lumotni olishingiz mumkin. Masalan, Ob’ektlar inspektorida biror xususiyatni tanlab, F1 tugmasini bossak – joriy xususiyatning qo‘llanilishi haqida ma’lumotga ega bo‘lamiz. Qo‘shimcha elementlar

Bu bo‘limda biz asosan dasturlash uchun yordamchi hisoblangan uchta qo‘shimcha elementlar haqida gapiramiz:

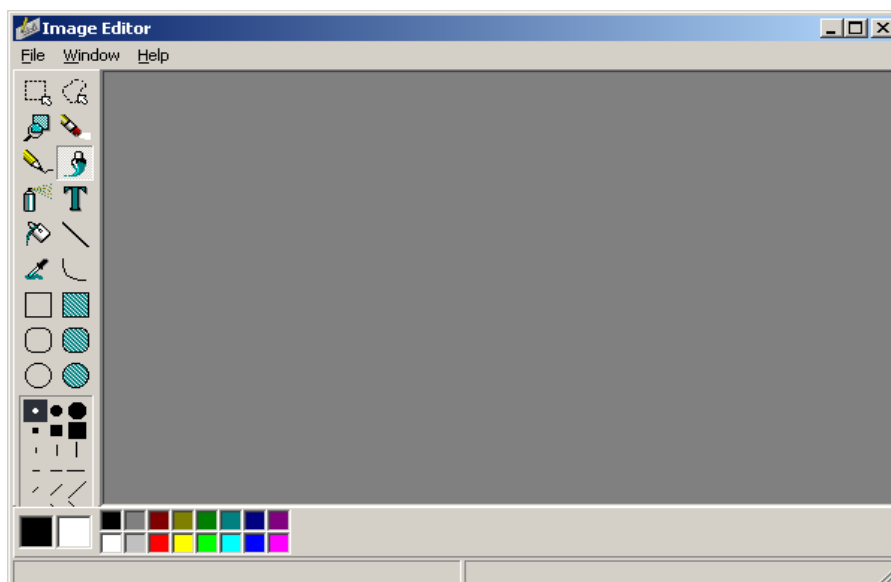
- Menyu (Menu System)
- Tezkor murojaat tugmalari (SpeedBar)
- Rasmlar muharriri (Image Editor)

Menyu C++ning eng tezkor va moslashuvchan vositasidir, chunki, undagi buyruqlarni tezkor tugmalar yordamida ham bajarish mumkin. Menyuning qulayligi shundaki, bu erda tugmachalar va belgilardan farqli ravishda aniq so‘zlar ishlatiladi. SpeedBar bevosita menyuning quyi qismida, komponentlar to‘plamidan chap tomonda joylashadi.



6-rasm: Tezkor murojaat tugmalari (SpeedBar)

7-rasmda keltirilgan Rasmlar muharriri Windowsdagi Paint dasturi kabi ishlaydi. Siz bu ob’ektga Tools menyusning Image Editor buyrug‘ini tanlash bilan murojaat qilishingiz mumkin.



7-rasm. Rasmlar muharriri

Endi dasturchilar C++ da ko'p qo'llaydigan elementlar bilan tanishib o'tamiz. Instrumental vositalar yuqorida ko'rib o'tilganlardan tashqari C++ tarkibida yana 5 ta vosita mavjud. Ular quyidagilar:

- Ichki dastur nazoratchisi
- Tashqi dastur nazoratchisi
- Buyruq satri kompilyatori
- WinSight
- WinSpector

Bu vositalarning alohida guruhga bulinganligi ularning boshqalaridan ko'ra kamroq amamiyatga molik ekanligida emas, balki, ular dasturlash jarayonida ancha mavhum texnik vazifalarni bajarishidadir.

C++da kuchli dasturchi bo'lish uchun siz C++ dastur nazoratchisi qanday tuzilganligini bilishingiz kerak. Nazoratchi sizga yozilgan dasturni qadamba-qadam tekshirish, bir martadadastur matnining bir qatorini bajarish va dasturdagi uzgaruvchilar qiymatlarini alohida ko'rish oynasi (Watch)da kuzatish imkonini beradi.

Yuqorida sanab o'tilgan besh elementdan eng muhimi bo'lgan ichki dastur nazoratchisi

dastur matni muharriri oynasida ishlaydi. Tashqi dastur nazoratchisining ichkididan farqi u ancha tez va kuchliligidadir. Lekin uni ishlatish uchun avval C++ muhitidan chiqish kerak.

Endi Kompilyatorlar haqida gaplashaylik. Tashqi kompilyator DCC.EXE deb nomlanadi, u asosan dasturni tashqi dastur nazoratchisi yordamida o'tqazishdan oldin ishlatiladi. Lekin ko'pchilik dasturchilar o'z dasturlarini C++ muhitining o'zida nazoradan o'tkazadilar.

WinSight va WinSpector Windows muhitida kuchli dasturchilar uchun qiziqarli hisoblanadi. Bu vositalar ikkilamchi hisoblanadi va faqat tor texnik maqsadlarda ishlatiladi.

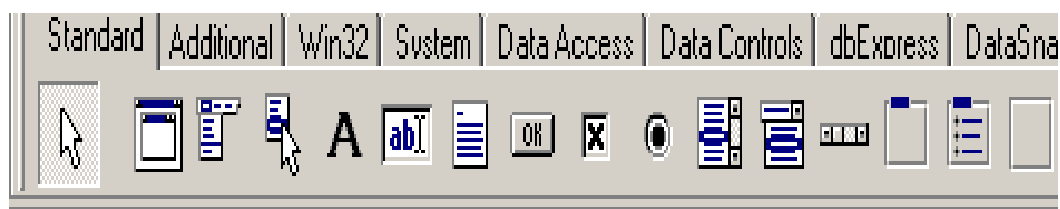
Bu ikki vositadan WinSight ancha foydaliroq. Uning asosiy vazifasi Windows tomonidan yuborilayotgan xabarlardan sizni voqif qilib turishdir. C++ malakasiz foydalanuvchilardan bu tizimning murakkab qismlarini yashirishga mo'ljallangan bo'lsada, Windows holatlarni boshqarish mumkin bo'lgan tizim sanaladi. C++ sizga Windowsning hamma xabarlariga javob berish imkoniyatini beradi. Natijada, malakali foydalanuvchilarga WinSight zarur bo'lib qoladi.

WinSpector EHMning joriy holatini matnli fayllarga saqlab qo'yish uchun xizmat qiladi; va siz bu fayllar yordamida dasturning qaysi qismi ishlayotganligini va nimalarni o'zgartirish kerakligini bilib olishingiz mumkin. Standart komponentlar

### ***Komponentlar to'plami***

C++ dasturlash muhiti bilan tanishishni davom ettirar ekanmiz, komponentlar to'plami haqida gapirishga to'g'ri keladi.

Komponentlar to'plamining birinchi sahifasida 14ta ob'ektlar joylashgan bo'lib, hamma dasturchilar albatta ushbu ob'ektlardan foydalanishgan. Chunki, juda ham kamchilik tugma, ro'yxat, kiritish oynalari va hozirgi kuzlardan foydalanmasdan dastur tuza oladi. Bu ob'ektlarning barchasi Windowsdagi sichqoncha va muloqot oynasi kabi muhimdir.



## 8-rasm. Standart Komponentlar to'plami

Har bir sahifadagi komponentlar to'plami va joylashishini o'zgartirish mumkin. Demak, siz mavjud komponentlarni o'rnini o'zgartirish bilan birgalikda ularning tarkibiga yangilarini qo'shishingiz mumkin. C++ning standart komponentlari va ularni ishlatish haqidagi qisqacha izoh quyida keltirilgan. Bu komponentlarni o'rganayotgan vaqtda qo'l ostingizda kompyuter bo'lgani ma'qul, chunki birvarakayiga ularning ishlash prinsipi bilan ham tanishishingiz mumkin.

-  **TMainMenu** – o'z dasturingizga bosh menyu qo'shish imkonini beradi. **TmainMenu**ni formaga qo'ygan vaqtda u kichkina belgi ko'rinishini oladi; bunday belgilarni "ko'rinmas komponent" deb ataydilar, chunki, dastur bajarilish vaqtida ular ekranda ko'rinmaydi. Menyuni yaratish uch qadamdan iborat: (1) TMainMenu formaga o'rnatish, (2) ob'ektlar inspektorining Items xususiyati yordamida Menyu dizaynerini chaqirish, (3) Menyu Dizaynerida menyu qismlarini kiritish.

-  **TPopupMenu** yordamchi menyuni yaratish. Bu menyu sichqonchani o'ng tugmasi bosilganda ko'rinadi.

-  **TLabel** matnlarni ekranda namoyish qilish uchun qo'llanadi. Agar siz ob'ektlar inspektorining Font xususiyatiga sichkonchani ikki marta bossangiz, **Label**ning shrifti, rangi, harflar rangi va o'lchamlarini o'zgartirishingiz mumkin.

-  **TEdit** – Windowsdagi standart ma'lumot kiritish oynasi. U qisqa matnlarni namoyish qilishi va dastur bajarilish vaqtida foydalanuvchiga o'z ma'lumotlarini kiritish imkonini beradi.

-  **TMemo** – TEditning boshqacha ko'rinishi. Katta matnlar bilan ishlashni ko'zda tutadi. TMemo matn qismlarini qatordan qatorga ko'chirishi, buferda saqlashi, oxirgi amallardan voz kechishi, umuman olganda oddiy muxarrir amallarini bajara oladi. TMemo 32Kb matn hajmi bilan chegaralanadi, bu taxminan 10-20 sahifa matnga tengdir.



- **TButton** dastur bajarilash vaqtida tugma bosilishi bilan biror amal bajarilishini ko'zda tutadi. Delphida hamma narsa oddiy. TButtonni formga ko'yib, unga ikki marta sichqon tugmasini bossak, biz dastur matni muharririda tugma bosilish holatiga dastur tuzishimiz mumkin bo'lgan oynaga o'tamiz. Endi dastur matnini bajariladigan buyruqlar kodi bilan to'ldirish kerak. Masalan:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
MessageDlg('Ishlaringiz yaxshimi?',mtConfirmation,mbYesNoCancel,0);
```

```
end;
```



- **TCheckBox** chap tomonida kichkina darchasi bo'lgan matn satrini akslantiradi. Darchaga uning belgilanganligini bildiruvchi belgi quyish mumkin. Masalan, Worddagi Pechat muloqot oynasining Pechat v fayl bo'limida aynan shu ob'ekt ishlatilgan.



- **TRadioButton** bir nechta holatlardan birini tanlash imkonini beradi. Masalan, Worddagi Pechat muloqot oynasining chop qilinishi kerak bo'lgan sahifalar oralag'ini tanlash bo'limida aynan shu ob'ekt ishlatilgan.



- **TListBox** ro'yxatli ma'lumotlarni namoyish qilishga mo'ljallangan masalan Worddagi Otkryt buyrug'i yordamida ochiladigan muloqot oynasidagi fayllar va kataloglar ro'yxati aynan shu ob'ektga joylashtirilgan.



- **TComboBox** tashqi ko'rinishidan ListBoxga o'xshab ketadi, ular orasidagi farq ComboBoxning yuqorisida ma'lumot kiritish mumkin bo'lgan maydoni ham bor. ComboBoxning ko'plab turlari mavjud, ko'p ishlatiladigani esa, ro'yxati pastga ochiladigan (drop-down combo box)dir.



- **TScrollbar** – o'tkazish yo'lakchasi. Ko'pincha tahrirlanadigan yoki ko'rinadigan ma'lumotlar ekran chegaralaridan chiqib ketganda paydo bo'ladi.

-  **TGroupBox** Windowsga formadagi ob'ektlarning qanday joylashganligini bildirish uchun qullash mumkin.

-  **TPanel** – TGroupBoxga o'xshash ob'ekt bo'lib, bezash uchun ishlatiladi.

## 1.2 Matematik funksiyalar va standart kutubxonalar

Kutubxonaning standart matematik funksiyalari ko'pgina amallarni bajarishga imkon beradi. Biz bu kutubxona misolida funksiyalar bilan ishlashni ko'rib chiqamiz.

Masalan bizning dasturimizda quyidagi satr bor bo'lsin:

```
double = k;
```

```
int m = 123;
```

```
k = cin(m);
```

kompilyator ushbu satrni ko'rganida, standart kutubxonadan cin funksiyasini chaqiradi. Kirish qiymati sifatida m ni berdik, yani funksiya hisoblagan qiymat k ga berildi. Funksiya agumentlari o'zgarmas sonlar (konstanta) o'zgaruvchilar, ifodalar va boshqa mos keluvchi qiymat hisoblaydigan funksiyalar bo'lishi mumkin. Masalan:

```
int g = 49, k = 100;
```

```
cout << "4900 ning ildizi -> "<< sqrt( g * k );
```

Ekranda:

```
4900 ning ildizi -> 70;
```

Matematik funksiyalar aksariyat hollarda double turidagi qiymat hisoblaydi. Kiruvchi argumentning turi sifatida esa double ga keltirilishi mumkin bo'lgan tur beriladi. Bu funksiyalarni ishlatish uchun math.h (yangi ko'rinishda cmath) e'lon faylini include bilan asosiy dastur tanasiga kiritish kerak. Quyida matematik funksiyalar kutubxonasining bazi bir a'zolarini beraylik, x va y o'zgaruvchilari double turiga ega.

| Funksiya  | Aniqlanishi                                                    | Misol                                               |
|-----------|----------------------------------------------------------------|-----------------------------------------------------|
| ceil(x)   | x ni x dan katta yoki unga teng bo'lgan                        | ceil(12.6)=13.0                                     |
|           | eng kichik butun songacha yahlitlaydi                          | ceil(-2.4)=-2.0                                     |
| cos(x)    | x ning trigonometrik kosinusi (x radianda)                     | cos(0.0)= 1.0                                       |
| exp(x)    | e ning x chi darajasi (eskponetsial f-ya)                      | exp(1.0)=2.71828<br>exp(2.0)=7.38906                |
| fabs(x)   | x ning absolyut qiymati                                        | x>0=>abs(x)=x<br>x=0=>abs(x)= 0.0<br>x<0=>abs(x)=-x |
| floor(x)  | x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi | floor(4.8)=4.0<br>floor(-15.9)=-6.0                 |
| fmod(x,y) | x/y ning qoldig'ini kasr son turida beradi                     | fmod(7.3,1.7)=0.5                                   |
| log(x)    | x ning natural lagorifmi (e asosiga ko'ra)                     | log(2.718282)=1.0                                   |
| log10(x)  | x ning 10 asosiga ko'ra lagorifmi                              | log10(1000.0)=3.0                                   |
| pow(x,y)  | x ning y chi darajasini beradi                                 | pow(3,4)=81.0<br>pow(16,0.25)=2                     |
| sin(x)    | x ning trigonometrik sinusi (x radianda)                       | sin(0.0)=0.0                                        |
| sqrt(x)   | x ning kvadrat ildizi                                          | sqrt(625.0)=25.0                                    |
| tan(x)    | x ning trigonometrik tangensi (x radianda)                     | tan(0.0)=0                                          |

## 2-§. C++ dasturlash tilida arifmetik amallarni bajaruvchi dasturiy ta'minot loyihasi

### 2.1 Tarmoqlanuvchi jarayonlarga oid dastur tuzish.

**if** shartli operator hisoblashlar jarayonini ikki yo'nalishga tarmoqlash uchun ishlatiladi. Operatorning tuzulishi sxemasi 1.5-rasmda keltirilgan. Operatorning formati:

```
if (ifoda) operator_1; [else operator_2;]
```

Avval arifmetik tip yoki ko'rsatkich tipga ega ifoda hisoblanadi. Agar u nolga teng bo'lmasa (true qiymatga ega) birinchi operator, aks holda – ikkinchisi bajariladi. Shundan so'ng boshqaruv shartli operatoridan keyin keluvchi operatorga uzatiladi.

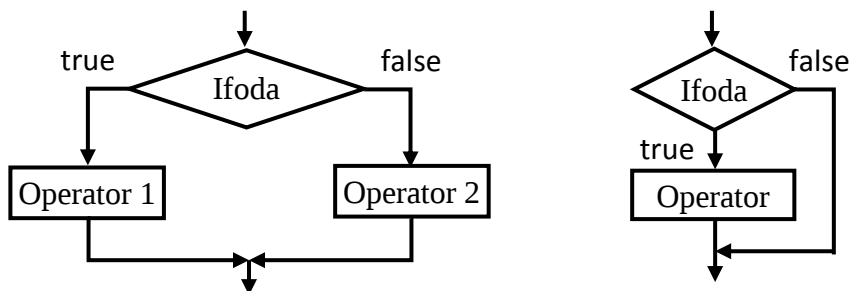
Tarmoqlaridan biri bo'lmasligi mumkin, false ibora bilan birgalikda ikkinchi tarmoqni tashlab yuborish mantiqli hisoblanadi.

Agar qandaydir tarmoqda bir nechta operatorlarni bajarish talab etilsa, ularni blokka kiritish zarur, aks holda kompyuter tarmoqlanish qayerda tugashini

tushunmasligi mumkin. Blok ixtiyoriy operatorni, jumladan, tavsiflashlar va boshqa shartli operatorlarni o'z ichiga olishi mukin (lekin faqat tavsiflashlardan iborat bo'la olmaydi).

Blokda tavsiflangan o'zgaruvchi blokdan tashqarida mavjud emasligini e'tiborga olish lozim.

### 1.5-rasm. Shartli operatorning tuzilish sxemasi



Misollar:

if (a<0) b = 1; // 1

if (a<b && (a>d | | a==0)) b++; else {b \*= a; a = 0;} // 2

if (a<b) {if (a<c) m = a; else m = c;} else {if (b<c) m = b; else m = c;} // 3

if (a++) b++; // 4

if (b>a) max = b; else max = a; //5

1-misolda else tarmog'i yo'q. Bunday konstruktsiya "operatorni o'tkazishi" deb ataladi, chunki qiymat berish shartning bajarilmasligiga bog'liq bajariladi yoki o'tkazib yuboriladi.

Agar bir nechta sartni tekshirish talab qilinsa mantiqiy amallar ishorolari bilan birlashtiriladi. Masalan, 2-misoldagi ifoda rost bo'ladi, agar bir vaqtning a<b shart va qavslardagi shartlardan biri bajarilsa. Agar ichki qavslar tashlab yuborilsa, dastlab VA mantiqiy amali, so'ngra YOKI mantiqiy amali bajariladi.

3-misoldagi operator uchta o'zgaruvchidan eng kichik qiymatni hisoblaydi. Katta qavslar bu holda shart emas, chunki kompilyator else qismini eng yaqin if ga o'tkazadi.

4-misolda. if operatorida ifodalar sifatida ko'proq munosabatlar amalidan foydalanilsada, bu shart emasligini ko'rsatadi.

5-misoldagi konstruktsiya sodda shartli amal ko'rinishda yozish ko'rgazmaliroq (bu yerda: max = (b > a) ? b: a;).

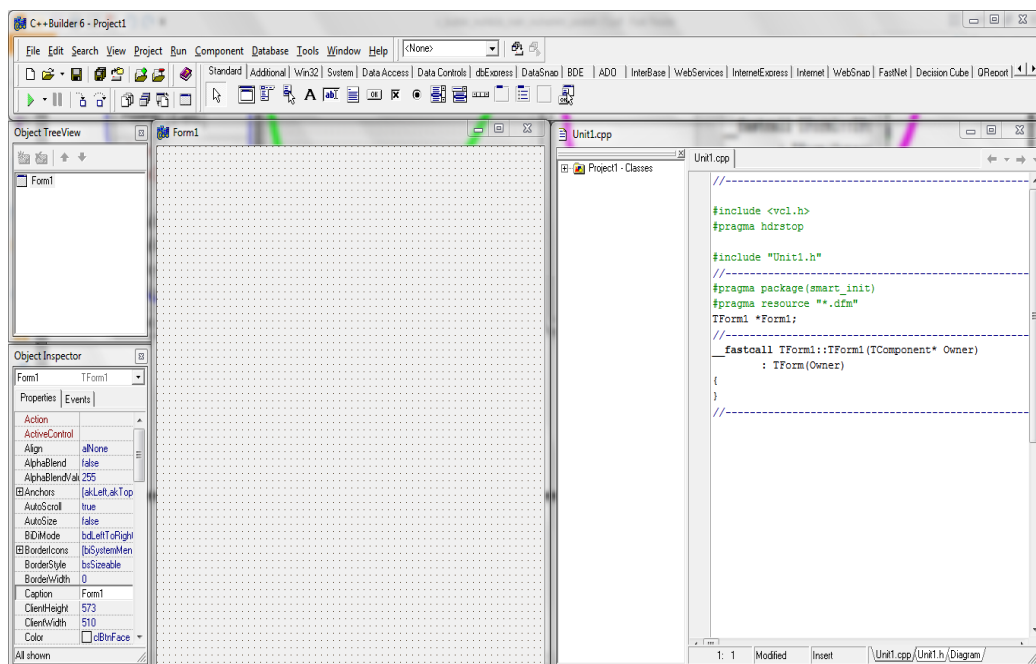
## Masalaning qo'yilishi:

Soliha tug'ilgan kuniga  $n$  ta do'stini chaqirishga qaror qildi. Uning do'stlari mevalarni yoqtirganliklari uchun  $u$  do'stlari uchun  $m$  ta apelsin xarid qilishga qaror qildi. U har bir apelsinni mehmonlar o'rtasida teng taqsimlash maqsadida bir xil sondagi bo'laklarga ajratmoqchi (Solihaning o'zi apelsin yemaydi). Yuqoridagi shartlarning bajarilishi uchun har bir apelsinni ajratish lozim bo'lgan minimal sondagi bo'laklarni hisoblovchi dasturni yozing.

*Kiruvchi ma'lumotlar:* Kiruvchi fayl ikkita musbat butun son  $n$  va  $m$  dan tashkil topgan ( $1 \leq n, m \leq 10$ ).

*Chiquvchi ma'lumotlar:* Chiquvchi faylga masalaning javobini chiqaring.

**Dastur tuzish muhiti:** Dasturni C++ Builder muhitida tuzishni ko'rib chiqamiz. Biz ishlaydigan muhitning boshlang'ich ko'rinishida bosh Menyu, Forma, Unit qismi, obyektlar sxemasini, ko'rsatish qismi va obyekt tahrilagich bo'limchalaridan iborat bo'ladi. Ya'ni quyidagi oynalar bo'ladi:

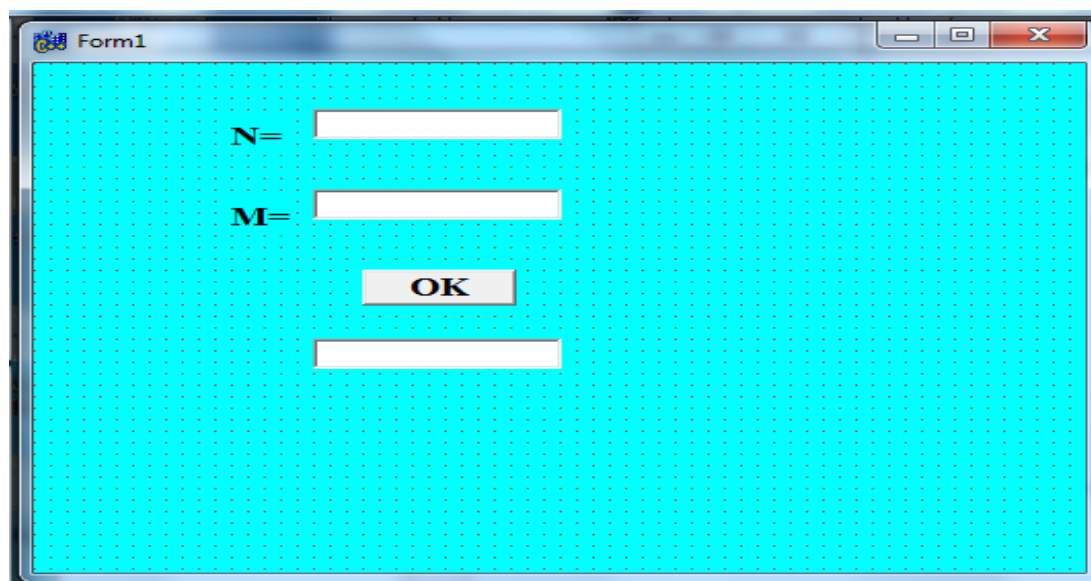


**Dasturni tuzish jarayoni.**

Dastlab C++ Builder dasturlash tilini ishga tayyorlab olamiz. Bu dasturni uchun quyidagi ketma- ketlikda C++ Builder 6 dasturiga kiramiz: **Пуск ->Вспомогательные программы -> Borland C++ Builder 6 -> C++ Builder 6 .**

### Masalaning yechilishi:

Bu dasturni tayyorlash uchun “Form1” ga “**Standart**” asboblardan 2 ta Label, 3 ta Edit va Button olamiz. Shunda Form oynasi quyidagicha ko’rinishga keladi:



Endi “**Button1**” ning **onClick** xossasiga quyidagi kodni kiritamiz:

```
#include <vcl.h>
```

```
#pragma hdrstop
```

```
#include "Unit1.h"
```

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

```
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
```

```
TForm1 *Form1;
```

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

```

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}

//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int n,m,qold;

    float d;

    n=StrToInt(Edit1->Text);
    m=StrToInt(Edit2->Text);
    qold=m%n;
    if (qold==0)
        Edit3->Text='1';
    else
        {double d=n/qold;
        Edit3->Text=FloatToStr(d);
        }
}

//-----

```

Bu dasturni ishlatish uchun dasturning “**Run**” menyusining “Run” buyrug’ini yoki klaviaturadan **F9** klavishini bosib ishga tushirishimiz mumkin. Shunda bizga dasturning tayyor quyidagi oynasi hosil bo’ladi.

Form1

N=

M=

OK

Bu dasturni ishlatish uchun ikkita son kiritib “OK” tugmasini bosib natija olishimiz mumkin. Masalan :

Form1

N=

M=

OK

## Xulosa.

Xulosa qilib aytganda ushbu “**kurs ishi**” tayyorlash davomida men shunday xulosaga keldim, masalalar yechishni bajaruvchi dastur yaratish uchun C++ Builder muhiti haqida yetarli ma'lumotga ega bo'lishimiz kerak. Formaga matnlar bilan ishlashda kerak bo'ladigan komponentalarni joylashimiz va har bir komponentaning tegishli kodlarini kiritib chiqishni o'rgandik. Dasturni kompilyatsiya qilishdan avval unga tegishli barcha fayllarni saqlab qo'yishni ham o'rgandik. Har bir dastur bir emas bir nechta fayllardan iborat bo'ladi. Ularni saqlab qo'yayotganda har bir dasturni alohida kataloglarda maxsus nom ostida saqlash maqsadga muvofiqligi haqida ko'nikmaga ega bo'ldik. Chunki dasturni boshqa joyga ko'chirishga to'g'ri kelib qolgan vaqtda dasturlarning o'xshash fayllarini adashtirib yuborish mumkinligi, bunday holatda dasturda xatolik aniqlanib dastur ishlamay qolishi, dasturni to'liq saqlash uchun faqat save ni emas balki save all ni bosib hamma fayllarni saqlashni o'rgandik.

Biz bu dasturni tuzib obektlar yaratishni va ular orqali shunga o'xshash yana boshqa obektlar yaratdik bu dasturda ishlashimga yana bir sabab boshqa obektlarga mo'ljallanmagan dasturlash tillarida faqat masalani matematik dasturi tuzib natija olinadi bunda yani C++ Builderda esa boshqa imkoniyatlar va kampanentalar bilan ham ishlashni o'rgandik.

## **Foydalanilgan adabiyotlar**

1. Q.Abdurahimov C++ Dasturlash asoslari 2014
2. Ашарина Н.А. Основы программирования на языках Си, C++. Учебный курс. М.: 2002 г.
3. Подбельский В.В. Язык C++ М.: Финансы и статистика, 1996.
4. [www.tammi.uz](http://www.tammi.uz)
5. [www.ziyonet.uz](http://www.ziyonet.uz)
6. [www.google.uz](http://www.google.uz) qidiruv sayti