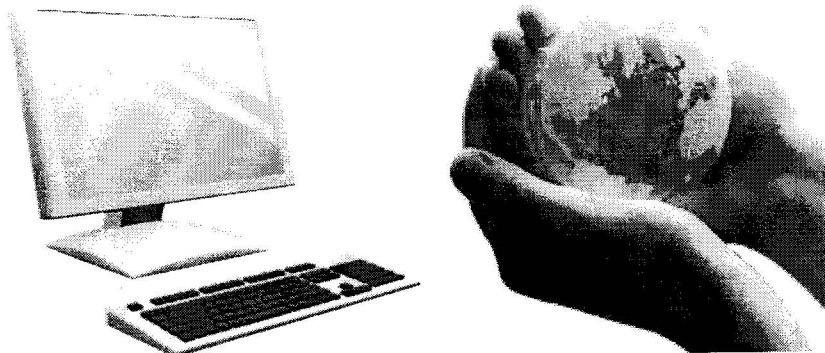


**O'ZBEKISTON ALOQA VA AXBOROTLASHTIRISH AGENTLIGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI  
UNIVERSITETI NUKUS FILIALI**

**INFORMATIKA VA AXBOROT TEXNOLOGIYALARI FAKULTETI**

**Informatika va axborot texnologiyalari  
fanidan**

# **Kurs ishi**



**Bajargan:**

**1<sup>b</sup>-Kasbiy ta'lim Madrimova S.**

**Qabul qilgan:**

**ass. Yadgarov Sh.**

**Nukus-2012**

**Mavzu: Obyektga mo'ljallangan dasturlash tillari.  
C++ Builder dasturi.**

**Reja:**

- I. Kirish**
- II. Asosiy qism:**
  - A)Obyektga mo'ljallangan dasturlash tillari haqida umumiy ma'lumot**
  - B)C++ tili ( alfaviti, o'zgaruvchilari, o'zgarmas (constants)lari arifmetik ifodalari)**
  - C) C++ BUILDER asoslari**
- III. Amaliy ish**
- IV. Xulosa**
- V. Foydalanilgan adabiyotlar**

## Kirish

Bugun inson o'z hayotini kompyuter texnologiyalarisiz tasavvur etolmaydi. Ushbu kompyuterlarga o'rnatiladigan operatsion sistemalar zamonaviy obyektga yo'naltirilgan dasturlash tillari yordamida yoziladi. Hozirgi kunda Respublikamizdagi oliy o'quv yurtlarida «Informatika va axborot texnologiyalari» yo'nalishi talabalariga o'quv rejasiga ko'ra turli xil dasturlash tillarini o'rgatish mo'ljallangan. Hozirgi kunda bir qator obyektga mo'ljallangan dasturlash tillari mavjud bo'lib misol uchun Turbo pascal, SI, Delphi, C++, Java kabi turlari va ularning versiyalari mavjud. Ular orasida C++ tili o'zining imkoniyatlari va qo'llanilashi ko'lamiga qarab yuqori o'rinlarda turadi.

Ushbu kurs ishida C++ algoritmik tili va uning C++BUILDER varsiyasi to'g'risida boshlang'ich ma'lumotlarni berib o'taman, kurs ishining asosiy qismida C++ tilining alfaviti, asosiy va yordamchi operatorlari, arifmetik va mantiqiy ifodalarning C++ tilida kiritilishi hamda oddiy dasturlar ham keltirilgan.

Obyektga mo'ljallangan dasturlash (OMD)tillari bir kunda o'ylab topilgan emas. Uning paydo bo'lishi asosida oddiy dasturlash tillari yotadi. OMD oddiy dasturlash tillari rivojidadagi navbatdagi pog'ona xolos. Vaqt o'tishi bilan qanday uslublar ishlash uchun qulay, qaysinisi noqulay ekanini aniqlash oson bo'lib bordi. OMD eng muvaffaqiyatli, vaqt sinovidan o'tgan uslublarni o'zida mujassam etadi.

Dastlab dasturlash anchayin boshqotirma ixtiro bo'lib, u dasturchilarga dasturlarni kommutatsiya bloki orqali kompyuterning asosiy xotirasiga to'g'ridan-to'g'ri kiritish imkonini berdi. Dasturlar mashina tillarida ikkilik tasavvurda yozilar edi. Dasturlarni mashina tilida yozishda tez-tez xatolarga yo'l qo'yilar, kodni kuzatib borish amalda deyarli mumkin emas edi. Bundan tashqari, mashina kodlaridagi dastur tushunish uchun g'oyat murakkab edi.

Vaqt o'tishi bilan kompyuterlar tobora kengroq qo'llana boshlandi hamda yuqoriroq darajadagi protsedura tillari paydo bo'ldi. Bularning dastlabkisi FORTRAN tili edi. Biroq obyektga mo'ljallangan yondashuv rivojiga asosiy ta'sir keyinroq paydo bo'lgan. Protsedura tillari dasturchiga axborotga ishlov berish dasturini pastroq darajadagi bir nechta protseduraga bo'lib tashlash imkonini beradi.

Obyektga mo'ljallangan dasturlash (OMD) modulli dasturlashdan keyingi mantiqiy pog'onani egallaydi, u modulga nasldan naslga o'tishni va polimorfizmni qo'shadi. Dasturchi OMD dan foydalanar ekan, dasturni bir qator oliy darajali obyektlarga bo'lish yo'li bilan tizimlashtiradi. Har bir obyekt hal qilinayotgan muammoning ma'lum bir tomonini modellashtiradi. OMD endilikda dasturni bajarish jarayonini boshqarish uchun dasturchi diqqatini protseduralarni ketma-ketlikda chaqirib olish ro'yxatini tuzib o'tirishga qaratmaydi. Buning o'rniga obyektlar o'zaro aloqada bo'ladi.

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 hossalari kiritgan va B da UNIX operatsion sistemasining birinchi versiyalarini yozgan. BCPL ham, B ham tipsiz til bo'lgan. Yani o'garuvchilarning ma'lum bir tipi bo'lmagan - har bir o'zgaruvchi kompyuter hotirasida faqat bir bayt yer egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun sonmi, kasrli sonmi yoki harfdekmi, dasturchi vazifasi bo'lgan. 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. Shu sababli C da yozilgan dasturlar kam miqdordagi o'zgarishlar yoki umuman o'zgarishsiz 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 etmoda. Buni ta'minlash uchun obyektli dasturlash g'oyasi ilgari surildi. Huddi 70-chi yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi obyektlar orqali tuzish dasturlash sohasida inqilob qildi.

C++ dan tashqari boshqa ko'p obyektli dasturlashga yo'naltirilgan tillar paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xerox ning Palo Altoda joylashgan ilmiy-qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalk da hamma narsa obyektlarga 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 obyektarning juda boy kutubxonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubxonasidagi tayyor obyekt funksiyalarni qo'llashni o'rganishdir.

## **II. A) Obyektga mo'ljallangan dasturlash tillari.**

Obyektga mo'ljallangan yodashuv, obyektga mo'ljallangan dasturlashning asosiy tayanch iboralari bu obyekt, xabar, aloqadir.

Obyektlar o'rtasidagi aloqa obyektga mo'ljallangan dasturlashning muhim tarkibiy qismidir. Obyektlar o'zaro aloqasining ikkita asosiy usuli mavjud:

Birinchi usul: obyektlar biri ikkinchisidan mustaqil ravishda mavjud bo'ladi. Agar alohida obyektlarga o'zaro aloqa kerak bo'lib qolsa, ular bir-birlariga xabar jo'natadi.

Obyektlar bir-birlari bilan xabarlar yordamida aloqa qiladi. Xabar olgan obyekt ma'lum xatti-harakatlarni bajaradi.

Xabar uzatish bu obyekt holatini o'zgartirish maqsadida uslubni chaqirib olish yoki xulq-atvor modellaridan birini qo'llashning o'zginasidir.

Ikkinchi usul: obyekt tarkibida boshqa obyektlar bo'lishi mumkin. Xuddi OMD da bo'lganidek, obyektlar ham, o'z navbatida agregatsiya yordamida boshqa obyektlardan jamlanishi mumkin. Ushbu obyektlarning har bittasida uslub va belgilarga ega bo'lgan interfeys mavjud bo'ladi.

Xabar — obyektga mo'ljallangan yondashuvning muhim tushunchasi. Xabarlar mexanizmi tufayli obyektlar o'z mustaqilligini saqlab qolishi mumkin. Boshqa biron obyektga xabar jo'natayotgan obyekt uchun xabar olgan obyekt talabdagi xatti-harakatni qanday bajarishi unchalik muhim emas. Unga xatti-harakat bajarilganligining o'zi muhimdir.

### **Obyektga mo'ljallangan yondashuvning afzalliklari va maqsadlari.**

OMY dasturiy ta'minotni ishlab chiqishda oltita asosiy maqsadga asoslanadi. OMY paradigmasiga muvofiq ishlab chiqilgan dasturiy ta'minot quyidagi xususiyatlarga ega bo'lmog'i lozim:

1. Tabiiylik.
2. Ishonchlilik.
3. Qayta qo'llanish imkoniyati.
4. Kuzatib borishda qulaylik.
5. Takomillashishga qodirlik.
6. Yangi versiyalarni davriy chiqarishning qulayligi.

#### **1.Tabiiylik.**

OMY yordamida tabiiy dasturiy ta'minot yaratiladi. Tabiiy dasturlar tushunarliroq bo'ladi. Dasturlashda «massiv» yoki «xotira sohasi» kabi atamalardan foydalanish o'rniga, yechilayotgan masala mansub bo'lgan soha atamalaridan foydalanish mumkin. Ishlab chiqilayotgan dasturni kompyuter tiliga moslash o'rniga, OMY aniq bir sohaning atamalaridan foydalanish imkonini beradi.

#### **2.Ishonchlilik.**

Yaxshi dasturiy ta'minot boshqa har qanday mahsulotlar, masalan, muzlatkich yoki televizorlar kabi ishonchli bo'lmog'i lozim.

Puxta ishlab chiqilgan va tartib bilan yozilgan obyektga mo'ljallangan dastur

ishonchli bo'ladi. Obyektlarning modulli tabiati dastur qismlaridan birida, uning boshqa qismlariga tegmagan holda, o'zgartishlar amalga oshirish imkonini beradi. Obyekt tushunchasi tufayli, axborotga ushbu axborot kerak bo'lgan shaxslar egalik qiladi, mas'uliyat esa berilgan funksiyalarni bajaruvchilar zimmasiga yuklatiladi.

### **3.Qayta qo'llanish imkoniyati.**

Malakali ishlab chiqilgan obyektga mo'ljallangan sinflarni bemalol takroran ishlatish mumkin. Xuddi modullar kabi, obyektlarni ham turli dasturlarda takroran qo'llash mumkin. Modulli dasturlashdan farqli o'laroq, OMY mavjud obyektlarni kengaytirish uchun vorislikdan, sozlanayotgan kodni yozish uchun esa polimorfizmdan foydalanish imkonini beradi.

### **4.Kuzatib borishda qulaylik.**

Dasturiy mahsulotning ish berish davri uning ishlab chiqilishi bilan tugamaydi. Dasturni ishlatish jarayonida kuzatib borish deb nomlanuvchi tirgak kerak. Dasturga sarflangan 60 foizdan 80 foizgacha vaqt kuzatib borishga ketadi. Ishlab chiqish esa ish berish siklining 20 foizinigina tashkil etadi.

Puxta ishlangan obyektga mo'ljallangan dastur ishlatish uchun qulay bo'ladi. Xatoni bartaraf etish uchun faqat bitta o'ringa to'g'rilash kiritish kifoya qiladi. Chunki ishlatishdagi o'zgarishlar tiniq, boshqa barcha obyektlar takomillashtirish afzalliklaridan avtomatik ravishda foydalana boshlaydi. O'zining tabiiyligi tufayli dastur matni boshqa ishlab chiquvchilar uchun tushunarli bo'lmog'i lozim.

### **5.Takomillashishga qodirlik.**

Foydalanuvchilar dasturni kuzatib borish paytida tez-tez tizimga yangi funksiyalarni qo'shishni iltimos qiladilar. Obyektlar kutubxonasini tuzishning o'zida ham ushbu obyektlarning funksiyalarini kengaytirishga to'g'ri keladi.

Dasturiy ta'minot statik (qotib qolgan) emas. Dasturiy ta'minot foydali bo'lib qolishi uchun uning imkoniyatlarini muttasil kengaytirib borish lozim. OMY da dasturni kengaytirish usullari ko'p. Vorislik, polimorfizm, qayta aniqlash, vakillik hamda ishlab chiqish jarayonida foydalanish mumkin bo'lgan ko'plab boshqa shablonlar shular jumlasidandir.

### **6.Yangi versiyalarni davriy chiqarilishning qulayligi.**

Zamonaviy dasturiy mahsulotning ish berish davri ko'p hollarda haftalar bilan o'lchanadi. OMY tufayli dasturlarni ishlab chiqish davrini qisqartirishga erishildi, chunki dasturlar ancha ishonchli bo'lib bormoqda, kengayishi osonroq hamda takroran qo'llanishi mumkin.

### III. B) C++ tili. Til tuzilishi.

#### 1. Alfavit, identifikator, xizmatchi so'zlar.

Alfavit. C++ alfavitiga quyidagi simvollar kiradi.

- Katta va kichik lotin alfaviti xarflari (A,B,...,Z,a,b,...,z)
- Raqamlar: 0,1,2,3,4,5,6,7,8,9
- Maxsus simvollar: " , { } | [ ] ( ) + - / % \ ; ' . : ? < = > \_ ! & \* # ~ ^

• Ko'rinmaydigan simvollar ("umumlashgan bushliq simvollar"). Leksemalarni o'zaro ajratish uchun ishlatiladigan simvollar (misol uchun bo'shlik, tabulyatsiya, yangi qatorga o'tish belgilari).

Izohlarda, satrlarda va simvolli konstantalarda boshqa literalalar, masalan rus xarflarini ishlatilishi mumkin.

C++ tilida olti hil turdagi leksemalar ishlatiladi: erkin tanlanadigan va ishlatiladigan identifikatorlar, xizmatchi so'zlar, konstantalar( konstanta satrlar), amallar( amallar belgilari), ajratuvchi belgilar.

**Identifikator.** Identifikatorlar lotin xarflari, ostki chiziq belgisi va sonlar ketma ketligidan iborat bo'ladi. Identifikator lotin xarfidan yoki ostki chiziq belgisidan boshlanishi lozim.

Misol uchun:

A1, \_MAX, adress\_01, RIM, rim

Katta va kichik xarflar farklanadi, shuning uchun oxirgi ikki identifikator bir biridan farq qiladi.

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 xarfi, ba'zi kompilyatorlarda 8 ta xarfi inobatga olinadi. Bu holda NUMBER\_OF\_TEST va NUMBER\_OF\_ROOM identifikatorlari bir biridan farq qilmaydi.

**Xizmatchi so'zlar.** Tilda ishlatiluvchi ya'ni dasturchi tomonidan o'zgaruvchilar nomlari sifatida ishlatish mumkin bo'lmagan identifikatorlar xizmatchi so'zlar deyiladi.

**C++ tilida quyidagi xizmatchi so'zlar mavjud:**

|          |          |         |
|----------|----------|---------|
| int      | extern   | else    |
| char     | register | for     |
| float    | typedef  | doʻ     |
| double   | static   | while   |
| struct   | goto     | switch  |
| union    | return   | case    |
| long     | sizeof   | default |
| short    | break    | entry   |
| unsigned | continue |         |
| auto     | if       |         |

## 2. O'zgaruvchilar. (VARIABLES)

O'zgaruvchilar obyekt sifatida. C++ tilining asosiy tushunchalaridan biri nomlangan hotira qismi - obyekt tushunchasidir. Obyektning xususiy holi bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan hotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, hotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikator. O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

### **O'zgaruvchilar tiplari. O'zgaruvchilarning qo'yidagi tiplari mavjuddir:**

char - bitta simvol;  
long char - uzun simvol;  
int - butun son;  
short yoki short int - qisqa butun son;  
long yoki long int - uzun butun son;  
float - haqiqiy son;  
long float yoki double - ikkilangan haqiqiy son;  
long double - uzun ikkilangan haqiqiy son;

Butun sonlar ta'riflanganda ko'rilgan tiplar oldiga unsigned (ishorasiz) ta'rifi qo'shilishi mumkin. Bu ta'rif qo'shilgan butun sonlar ustida amallar mod  $2n$  arifmetikasiga asoslangandir. Bu erda  $n$  soni int tipi hotirada egallovchi razryadlar sonidir. Agar ishorasiz  $k$  soni uzunligi, int soni razryadlar sonidan uzun bo'lsa, bu son qiymati  $k \bmod 2n$  ga teng bo'ladi. Ishorali ya'ni signed tipidagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa unsigned (ishorasiz) tipdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

O'zgaruvchilarni dasturning ixtiyoriy qismida ta'riflash yoki qayta ta'riflash mumkin.

Misol uchun:

```
Int a, b1, ac; yoki  
Int a;  
int b1;  
int ac;
```

O'zgaruvchilar ta'riflanganda ularning qiymatlari aniqlanmagan bo'ladi. Lekin o'zgaruvchilarni ta'riflashda initsializatsiya ya'ni boshlang'ich qiymatlarini ko'rsatish mumkin.

Misol uchun:

```
Int I=0;  
Char c='k';
```

Typedef ta'riflovchisi yangi tiplarni kiritishga imkon beradi.

Misol uchun yangi COD tipini kiritish:  
Typedef unsigned char COD;  
COD simbol;

### 3.O'zgarmas (CONSTANTS)lar

Konstanta bu o'zgartirish mumkin bo'lmagan qiymatdir. C++ tilida besh turdagi konstantalar ishlatilishi mumkin: butun sonlar, haqiqiy sonlar, simvollar, sanovchi konstantalar va no'l ko'rsatkich.

#### 1.Ma'lumotlarning butun son turi.

Butun sonlar o'nlik, sakkizlik yoki o'n oltilik sanoq sistemalarida berilishi mumkin.

O'nlik sanoq sistemasida butun sonlar 0-9 raqamlari ketma ketligidan iborat bo'lib, birinchi raqami 0 bo'lishi kerak emas.

Sakkizlik sanoq sistemasida butun sonlar 0 bilan boshlanuvchi 0-7 raqamlaridan iborat ketma ketlikdir.

O'n oltilik sanoq sistemasida butun son 0x eki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F xarflaridan iborat ketma ketlikdir.

Masalan 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, o'n oltilikda 0xF va 0x16 shaklda tasvirlanadi.

Ma'lumolarning uzun butun son turi.

Oxiriga l eki L harflari qo'yilgan o'nlik,sakkizlik yoki o'n oltilik butun son.

Ma'lumotlarning ishorasiz (unsigned) butun son turi:

Ohiriga u yoki U harflari qo'yilgan o'nlik,sakkizlik yoki o'n oltilik oddiy yoki uzun butun son.

#### 2. Ma'lumotlarning haqiqiy son turi:

Olti qismdan iborat bo'lishi mumkin: butun qism, nuqta, kasr qism, yoki E belgisi, o'nlik daraja , F eki f suffikslari.

Masalan : 66. .0 .12 3.14F 1.12e-12

Ma'lumolarning uzun haqiqiy son turi :

Ohiriga L eki l suffikslari qo'yilgan haqiqiy son.

Masalan: 2E+6L;

#### 3. Simvolli konstanta.

Bittalik qavslarga olingan bitta yoki ikkita simvol. Misol uchun 'x','\*','\012','\0','n'- bitta simvolli konstanta; 'dd','\n\t','\x07\x07' ikki simvolli konstantalar.

'\ ' simvolidan boshlangan simvollar eskeyp simvollar deyiladi.Simvolli konstanta qiymati simvolning kompyuterda qabul qilingan sonli kodiga tengdir.

## ESC (eskeyp) simvollar jadvali:

| Yozilishi | Ichki kodi | Simvoli (nomi)      | Ma'nosi                 |
|-----------|------------|---------------------|-------------------------|
| \a        | 0x07       | bel (audible bell)  | Tovush signali          |
| \b        | 0x08       | Bs (bascpase)       | Bir qadam qaytish       |
| \f        | 0x0C       | Ff (form feed)      | Sahifani qaytarish      |
| \n        | 0x0A       | lf (line feed)      | Qatorni o'tkazish       |
| \r        | 0x0D       | Cr(carriage return) | Karetkani qaytarish     |
| \t        | 0x09       | Ht (horizontal tab) | Gorizonta tabulyatsi    |
| \v        | 0x0B       | Vt (vertical tab)   | Vertikal tabulyatsi     |
| \\        | 0x5C       | \ (backslash)       | Teskari chiziq          |
| \'        | 0x27       | ' (single out)      | Apostrif (oddiy qavs)   |
| \"        | 0x22       | " (double quote)    | Ikkilik qavs            |
| \?        | 0x3F       | ? (question mark)   | Savol Belgisi           |
| \000      | 000        | Любой(octalnumber)  | Simvol sakkizlik kodi   |
| \xhh      | 0xhh       | Любой (hex number)  | Simvol o'n oltilik kodi |
| \v        | 0x0B       | Vt (vertical tab)   | Vertikal tabulyatsi     |
| \\        | 0x5C       | \ (backslash)       | Teskari chiziq          |
| \'        | 0x27       | ' (single out)      | Apostrif (oddiy qavs)   |
| \"        | 0x22       | " (double quote)    | Ikkilik qavs            |
| \?        | 0x3F       | ? (question mark)   | Savol Belgisi           |
| \000      | 000        | Любой(octalnumber)  | Simvol sakkizlik kodi   |
| \xhh      | 0xhh       | Любой (hex number)  | Simvol o'n oltilik kodi |

## Satrlı konstanta.

Satrlı konstantalar C++ tili konstantalariga kirmaydi, balki leksemalari alohida tipi hisoblanadi. Shuning uchun adabiyotda satrlı konstantalar satrlı leksemalar deb ham ataladi.

Satrlı konstanta bu ikkilik qavslarga olingan ihtiyoriy simvollar ketma ketligidir. Misol uchun " Men satrlı konstantaman".

Satrlar orasiga eskeyp simvollar ham kirishi mumkin. Bu simvollar oldiga \ belgisi qo'yiladi. Misol uchun :

"\n Bu satr \n uch katorga \n joylashadi".

Satr simvollari hotirada ketma-ket joylashtiriladi va har bir satrlı konstanta ohiriga avtomatik ravishda kompilyator tomonidan '\0' simvoli qo'shiladi. Shunday satrning hotiradagi hajmi simvollar soni+1 baytga tengdir.

Ketma-ket kelgan va bo'shlik, tabulyatsiya yoki satr ohiri belgisi bilan ajratilgan satrlar kompilyatsiya davrida bitta satrga aylantiriladi.

Misol uchun:

"Salom" "Toshkent "

satrlari bitta satr deb qaraladi.

"Salom Toshkent"

Bu qoidaga bir necha qatorga yozilgan satrlar ham bo'ysinadi. Misol uchun :

"O'zbekistonga "

"bahor "

"keldi"

qatorlari bitta qatorga mos:

"O'zbekistonga bahor keldi"

Agar satrda '\ ' belgisi uchrasa va bu belgidan so'ng to '\n' satr ohiri belgisigacha bo'shlik belgisi kelsa bu bo'shlik belgilari '\ ' va '\n' belgisi bilan birga satrdan o'chiriladi. Satrning o'zi keyingi satrda kelgan satr bilan qo'shiladi.

"O'zbekistonga \

" bahor\

" keldi"

qatorlari bitta qatorga mos:

"Uzbekistonga bakhor keldi"

## Sanovchi konstanta.

Sanovchi konstantalar **enum** hizmatchi so'zi yordamida kiritilib, **int** tipidagi sonlarga qulay so'zlarni mos qo'yish uchun ishlatiladi.

Misol uchun:

**enum{one=1,two=2,three=3};**

Agar son qiymatlari ko'rsatilmagan bo'lsa eng chapki so'zga 0 qiymati berilib qolganlariga tartib bo'yicha o'suvchi sonlar mos qo'yiladi:

**Enum{zero,one,two};**

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, two=2;

Konstantalar aralash ko'rinishda kiritilishi ham mumkin:

**Enum(zero,one,for=4,five,seeks}.**

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, for=4;five=5,seeks=6;

Yana bir misol:

Enum BOOLEAN {NO, YES};

Konstantalar qiymatlari:

NO=0, YES=1;

### **Nomlangan konstantalar.**

C ++ tilida o'zgaruvchilardan tashqari nomlangan konstantalar kiritilishi mumkin. Bu konstantalar qiymatlarini dasturda o'zgartirish mumkin emas. Konstantalar nomlari dasturchi tomonidan kiritilgan va hizmatchi so'zlardan farqli bo'lgan identifikatorlar bo'lishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinatsiyasidan iborat identifikatorlar ishlatiladi.

### **Nomlangan konstantalar quyidagi shaklda kiritiladi:**

**Const tip konstanta \_nomi=konstanta \_qiymati.**

Misol uchun:

Const double EULER=2.718282;

Const long M=999999999;

Const R=765;

Ohirgi misolda konstanta tipi ko'rsatilmagan, bu konstanta int tipiga tegishli deb hisoblanadi.

### **Null ko'rsatkich.**

NULL- ko'rsatkich yagona arifmetik bo'lmagan konstantadir. Konkret realizatsiyalarda null ko'rsatkich 0 eki 0L yoki nomlangan konstanta NULL orqali tasvirlanishi mumkin. Shuni aytish lozimki bu konstanta qiymati 0 bo'lishi yoki '0' simvoli kodiga mos kelishi shart emas.

Quyidagi jadvalda konstantalar chegaralari va mos tiplari ko'rsatilgan:

| Ma'lumotlar turi | Hajm,<br>bit | Qiymatlar chegarasi | Tip vazifasi                               |
|------------------|--------------|---------------------|--------------------------------------------|
| Unsigned char    | 8            | 0...255             | Char                                       |
| Char             | 8            | -128...127          | Kichik butun sonlar va ASII kodlar         |
| Enum             | 16           | -32768...32767      | Butun sonlar tartiblangan katori           |
| Unsigned int     | 16           | 0...65535           | Katta butun sonlar                         |
| Short int        | 16           | -32768...32767      | Kichik butun sonlar, tsikllarni boshqarish |
| Int              | 16           | -32768...32767      | Kichik butun sonlar, tsikllarni boshqarish |
| Unsigned long    | 32           | 0...4294967295      | Astronomik masofalar                       |
| Long             | 32           | ...2147483647       | Katta sonlar                               |
| Float            | 32           | 3.4E-32...3.4E+38   | Ilmiy hisoblar (7 raqam)                   |
| Double           | 64           | 1.7E-308...1.7E+308 | Ilmiy hisoblar(15 raqam)                   |
| Long double      | 80           | 3.4E-4932...        | 1.1E+4932<br>Moliyaviy hisoblar (19 raqam) |

## 4. C++ DA AMALLAR

### C++ DA ARIFMETIK AMALLAR

Ko'p programmalar ijro davomida arifmetik amallarni bajaradi. C++ dagi amallar quyidagi jadvalda berilgan. Ular ikkita operand bilan ishlatildi.

| C++ dagi amal<br>Arifmetik operator | Algebraik ifoda | C++ dagi ifodasi: |      |
|-------------------------------------|-----------------|-------------------|------|
| Qo'shish                            | +               | h+19              | h+19 |
| Ayirish                             | -               | f-u               | f-u  |
| Ko'paytirish                        | *               | s1                | s*1  |
| Bo'lish                             | /               | v/d,              | v/d  |
| Modul olish                         | %               | k mod 4           | k%4  |

Bularning ba'zi birlarinig hususiyatlarini ko'rib chiqaylik. Butun sonli bo'lishda, yani bo'luvchi ham, bo'linuvchi ham butun son bo'lganda, javob butun son bo'ladi. Javob yahlitlanmaydi, kasr qismi tashlanib yuborilib, butun qismining o'zi qoladi.

Modul operatori (%) butun songa bo'lishdan kelib chiqadigan qoldiqni beradi.  $x\%y$  ifodasi  $x$  ni  $y$  ga bo'lgandan keyin chiqadigan qoldiqni beradi. Demak,  $7\%4$  bizga 3 javobini beradi. % operatori faqat butun sonlar bilan ishlaydi. Vergulli (real) sonlar bilan ishlash uchun "matn.n" kutubxonasidagi fmod funksiyasini qo'llash kerak.

C++ da qavslarning ma'nosi huddi algebradagidekdir. Undan tashqari boshqa boshqa algebraik ifodalarning ketma-ketligi ham odatdagidek. Oldin ko'paytirish, bo'lish va modul olish operatorlari ijro etiladi. Agar bir necha operator ketma-ket kelsa, ular chapdan o'nga qarab ishlanadi. Bu operatorlardan keyin esa qo'shish va ayirish ijro etiladi.

Misol keltiraylik.  $k = m * 5 + 7 \% n / (9 + x);$

Birinchi bo'lib  $m * 5$  hisoblanadi. Keyin  $7 \% n$  topiladi va qoldiq  $(9 + x)$  ga bo'linadi. Chiqqan javob esa  $m * 5$  ning javobiga qo'shiladi. Qisqasini aytsak, amallar matematikadagi kabi. Lekin biz o'qishni osonlashtirish uchun va hato qilish ehtimolini kamaytirish maqsadida qavslarni kengroq ishlatishimiz mumkin. Yuqoridagi misolimiz quyidagi ko'rinishga ega bo'ladi.

$k = (m * 5) + ((7 \% n) / (9 + x));$

## Amallar jadvali:

Arifmetik amallar  
Razryadli amallar

Nisbat amallari

Mantiqiy amallar

+ qo'shish

& va

= = teng

&& va

- bo'lish

| yoki

!= teng emas

|| yoki

\* ko'paytirish

^ inkor

> katta

! inkor

/ bo'lish

<< chapga surish

>= katta yoki teng

% modul olish

>> o'ngga surish

< kichik

- unar minus

~ inkor

<= kichik yoki teng

+ unar plyus

++ oshirish

-- kamaytirish

Amallar jadvali (davomi)

Imlo amallar

Qiymat berish va shartli amallar

Tipli amallar

Adresli amallar

() - doirali qavs

= - oddiy qiymat berish

(tip) - tipni o'zgartirish

& - adresni aniqlash

[] - kavadrat qavs

op= - murakkab qiymat berish

sizeof- hajmni hisoblash

\* - adres bo'yicha qiymat aniqlash yoki joylash

, - vergul

? - shartli amal

## Arifmetik amallar.

Amallar odatda unar ya'ni bitta operandga qo'llaniladigan amallarga va binar ya'ni ikki operandga qo'llaniladigan amallarga ajratiladi.

Binar amallar additiv ya'ni + qo'shuv va - ayirish amallariga, hamda multiplikativ ya'ni \* ko'paytirish, / bo'lish va % modul olish amallariga ajratiladi.

Additiv amallarining ustivorligi multiplikativ amallarining ustivorligidan pastroqdir.

Butun sonni butun songa bo'lganda natija butun songacha yahlitlanadi. Misol uchun  $20/3=6$ ;  $(-20)/3=-6$ ;  $20/(-3)=-6$ .

Modul amali butun sonni butun songa bo'lishdan hosil bo'ladigan qoldikka tengdir.

Agar modul amali musbat operandlarga qo'llanilsa, natija ham musbat bo'ladi, aks holda natija ishorasi kompilyatorga bog'likdir.

Binar arifmetik amallar bajarilganda tiplarni keltirish quyidagi qoidalar asosida amalga oshiriladi:

short va char tiplari int tipiga keltiriladi;

Agar operandlar biri long tipiga tegishli bo'lsa ikkinchi operand ham long tipiga keltiriladi va natija ham long tipiga tegishli buladi;

Agar operandlar biri float tipiga tegishli bulsa ikkinchi operand kham float tipiga keltiriladi va natija ham float tipiga tegishli buladi;

Agar operandlar biri double tipiga tegishli bo'lsa ikkinchi operand ham double tipiga keltiriladi va natija ham double tipiga tegishli buladi;

Agar operandlar biri long double tipiga tegishli bo'lsa ikkinchi operand ham long double tipiga keltiriladi va natija ham long double tipiga tegishli bo'ladi;

Unar amallarga ishorani o'zgartiruvchi unar minus - va unar + amallari kiradi. Bundan tashqari ++ va -- amallari ham unar amallarga kiradi.

++ unar amali qiymatni 1 ga oshirishni ko'rsatadi. Amalni prefiks ya'ni ++i ko'rinishda ishlatish oldin o'zgaruvchi qiymatini oshirib so'ngra foydalanish lozimligini, postfiks ya'ni i++ ko'rinishda ishlatish oldin o'zgaruvchi qiymatidan foydalanib so'ngra oshirish kerakligini ko'rsatadi. Misol uchun i qiymati 2 ga teng bo'lsin, u holda 3+(++i) ifoda qiymati 6 ga, 3+i++ ifoda qiymati 5 ga teng bo'ladi. Ikkala holda ham i qiymati 3 ga teng bo'ladi.

-- unar amali qiymatni 1 ga kamaytirishni ko'rsatadi. Bu amal ham prefiks va postfiks ko'rinishda ishlatilishi mumkin. Bu ikki amalni faqat o'zgaruvchilarga qo'llash mumkindir.

Unar amallarning ustivorligi binar amallardan yuqoridir.

Razryadli amallar. Razryadli amallar natijasi butun sonlarni ikkilik ko'rinishlarining har bir razryadiga mos mantiqiy amallarni qo'llashdan hosil bo'ladi. Masalan 5 kodi 101 ga teng va 6 kodi 110 ga teng.

$6 \& 5$  qiymati 4 ga ya'ni 100 ga teng.

$6 | 5$  qiymati 7 ga ya'ni 111 ga teng.

$6 \wedge 5$  qiymati 3 ga ya'ni 011 ga teng.

$\sim 6$  qiymati 4 ga ya'ni 010 ga teng.

Bu misollarda amallar ustivorligi oshib borishi tartibida berilgandir.

Bu amallardan tashqari  $M \ll N$  chapga razryadli siljitish va  $M \gg N$  ungga razryadli siljitish amallari qo'llaniladi. Siljitish M butun sonning razryadli ko'rinishiga qo'llaniladi. N nechta pozitsiyaga siljitish kerakligini ko'rsatadi.

Chapga N pozitsiyaga surish bu operand qiymatini ikkining N chi daraasiga kupaytirishga mos keladi. Misol uchun  $5 \ll 2 = 20$ . Bu amalning bitli kurinishi:  $101 \ll 2 = 10100$ .

Agar operand musbat bulsa N poziciyaga ungga surish chap operandni ikkining N chi darajasiga bo'lib kasr qismini tashlab yuborishga mosdir. Misol uchun  $5 \gg 2 = 1$ . Bu amalning bitli kurinishi  $101 \gg 2 = 001 = 1$ . Agarda operand qiymati manfiy bulsa ikki variant mavjuddir: arifmetik siljitishda bushatilayotgan razryadlar ishora razryadi qiymati bilan to'ldiriladi, mantiqiy siljitishda bushatilayotgan razryadlar nullar bilan tuldiriladi.

Razryadli surish amallarining ustivorligi o'zaro teng, razryadli inkor amalidan past, qolgan razryadli amallardan yuqoridir. Razryadli inkor amali unar qolgan amallar binar amallarga kiradi.

Nisbat amallari. Nisbat amallari qiymatlari 1 ga teng agar nisbat bajarilsa va aksincha 0 ga tengdir. Nisbat amallari arifmetik tipdagi operandlarga yoki ko'rsatkichlarga qo'llaniladi.

Misollar:

$1!=0$  qiymati 1 ga teng;

$1==0$  qiymati 0 ga teng;

$3>=3$  qiymati 1 ga teng;

$3>3$  qiymati 0 ga teng;

$2<=2$  qiymati 1 ga teng;

$2<2$  qiymati 0 ga teng;

Katta  $>$ , kichik  $<$ , katta eki teng  $>=$ , kichik eki teng  $<=$  amallarining ustivorligi bir hildir.

Teng  $==$  va teng emas  $!=$  amallarining ustivorligi uzaro teng va qolgan amallardan pastdir.

### **Mantiqiy amallar.**

C ++ tilida mantiqiy tip yo'kdir. Shuning uchun mantiqiy amallarni butun sonlarga qo'llanadi. Bu amallarning natijalari quyidagicha aniqlanadi:

$x||y$  amali 1 ga teng agar  $x>0$  eki  $y>0$  bo'lsa, aksincha 0 ga teng

$x\&\&y$  amali 1 ga teng agar  $x>0$  va  $y>0$  bo'lsa, aksincha 0 ga teng

$!x$  amali 1 ga teng agar  $x>0$  bo'lsa, aksincha 0 ga teng

Bu misollarda amallar ustivorligi oshib borish tartibida berilgandir.

Inkor  $!$  amali unar qolganlari binar amallardir.

Bu amallardan tashqari quyidagi amallar ham mavjuddir:

Qiymat berish amali. Qiymat berish amali = binar amal bo'lib chap operandi odatda o'zgaruvchi o'ng operandi odatda ifodaga teng bo'ladi. Misol uchun  $Z=4.7+3.34$

Bu qiymati 8.04 ga teng ifodadir. Bu qiymat  $Z$  o'zgaruvchiga ham beriladi.

Bu ifoda ohiriga nuqta vergul ; belgisi quyilganda operatorga aylanadi.

$Z=4.7+3.34$

Bitta ifodada bir necha qiymat berish amallari qo'llanilishi mumkin. Misol uchun:

$C=y=f=4.2+2.8;$

Bundan tashqari C ++ tili da murakkab qiymat berish amali mavjud bo'lib, umumiy ko'rinishi quyidagichadir:

O'zgaruvchi\_nomi amal= ifoda;

Bu erda amal quyidagi amallardan biri  $*,/, \%, +, -, \&, ^, |, <<, >>.$

Misol uchun:

$X+=4$  ifoda  $x=x+4$  ifodaga ekvivalentdir;

$X*=a$  ifoda  $x=x*a$  ifodaga ekvivalentdir;

$X/=a+b$  ifoda  $x=x/(a+b)$  ifodaga ekvivalentdir;

$X>>=4$  ifoda  $x=x>>4$  ifodaga ekvivalentdir;

Imlo belgilari amal sifatida. C ++ tilida ba'zi bir imlo belgilari ham amal sifatida ishlatilishi mumkin. Bu belgilardan oddiy () va kvadrat [] qavslardir. Oddiy qavslar binar amal deb qaralib ifodalarda yoki funktsiyaga muroaat qilishda foydalaniladi. Funktsiyaga murojaat qilish qo'yjidagi shaklda amalga oshiriladi:

<funktsiya nomi> (<argumentlar ruyhati>). Misol uchun  $\sin(x)$  eki  $\max(a,b)$ .

Kvadrat qavslardan massivlarga murojaat qilishda foydalaniladi. Bu murojaat quyidagicha amalga oshiriladi:

<massiv nomi>[<indeks>]. Misol uchun  $a[5]$  eki  $b[n][m]$ .

Vergul simvolini ajratuvchi belgi deb ham qarash mumkin amal sifatida ham qarash mumkin. Vergul bilan ajratilgan amallar ketma-ketligi bir amal deb qaralib, chapdan o'ngga hisoblanadi va ohirgi ifoda qiymati natija deb qaraladi. Misol uchun:  $d=4, d+2$  amali natijasi 8 ga teng.

Shartli amal. Shartli amal ternar amal deyiladi va uchta operanddan iborat bo'ladi: <1-ifoda>?<2-ifoda>:<3-ifoda>

Shartli amal bajarilganda avval 1- ifoda hisoblanadi. Agar 1-ifoda qiymati 0 dan farqli bo'lsa 2- ifoda hisoblanadi va qiymati natija sifatida qabul qilinadi, aks holda 3-ifoda hisoblanadi va qiymati natija sifatida qabul qilinadi.

Misol uchun modulni hisoblash:  $x<0?-x:x$  yoki ikkita son kichigini hisoblash  $a<b?a:b$

Shuni aytish lozimki shartli ifodadan har qanday ifoda sifatida foydalanish mumkin. Agar F FLOAT tipga, a N - INT tipga tegishli bo'lsa ,

$(N > 0) ? F : N$

ifoda N musbat eki manfiyligidan qat'iy nazar DOUBLE tipga tegishli bo'ladi.

Shartli ifodada birinchi ifodani qavsga olish shart emas.

Tiplar bilan ishlovchi amallar. Tiplarni o'zgartirish amali quyidagi ko'rinishga ega:

(tip\_nomi) operand; Bu amal operandlar qiymatini ko'rsatilgan tipga keltirish uchun ishlatiladi. Operand sifatida kostanta, o'zgaruvchi yoki qavslarga olinga ifoda kelishi mumkin. Misol uchun (long)6 amali konstanta qiymatini o'zgartirmagan holda operativ hotirada egallagan baytlar sonini oshiradi. Bu misolda konstanta tipi o'zgarmagan bo'lsa, (double) 6 eki (float) 6 amali konstanta ichki ko'rinishini ham o'zgartiradi. Katta butun sonlar hakikiy tipga keltirilganda sonning aniqligi yuqolishi mumkin.

sizeof amali operand sifatida ko'rsatilgan ob'ektning baytlarda hotiradagi hajmini hisoblash uchun ishlatiladi. Bu amalning ikki ko'rinishi mavjud:

sizeof ifoda sizeof (tip) Misol uchun:

Sizeof 3.14=8

Sizeof 3.14f=4

Sizeof 3.14L=10

Sizeof(char)=1

Sizeof(double)=8.

### Amallar ustivorligi:

| Rang | Amallar                                            | Yo'nalish      |
|------|----------------------------------------------------|----------------|
| 1    | () [] -> :: .                                      | Chapdan o'ngga |
| 2    | ! ~ + - ++ -- & * (tip) sizeof<br>new delete tip() | O'ngdan chapga |
| 3    | . * ->*                                            | Chapdan o'ngga |
| 4    | * / % (multiplikativ binar amallar)                | Chapdan o'ngga |
| 5    | + - (additiv binar amallar)                        | Chapdan o'ngga |
| 6    | << >>                                              | Chapdan o'ngga |
| 7    | < <= >= >                                          | Chapdan o'ngga |
| 8    | = !=                                               | Chapdan o'ngga |
| 9    | &                                                  | Chapdan o'ngga |
| 10   | ^                                                  | Chapdan o'ngga |
| 11   |                                                    | Chapdan o'ngga |
| 12   | &&                                                 | Chapdan o'ngga |
| 13   |                                                    | Chapdan o'ngga |
| 14   | ?:(shartli amal)                                   | Chapdan o'ngga |
| 15   | = *= /= %= += -= &= ^=  =<br><<= >>=               | Chapdan o'ngga |
| 16   | , (vergul amali)                                   | Chapdan o'ngga |

## II. C) C++BUILDER ASOSLARI

### Dastlabki tanishuv

C++obyektga mo'ljallangan dasturlash tilining C++BUILDER versiyasining boshqa versiyalaridan varqi shundaki u ishlab chiqishning integratsiyalashgan muhiti Komponentalar Palitrasini birlashtiradi. Shakllar Muharriri, Kod Muharriri, Obyektlar Noziri, Obyektlar Xazinasini — bular hammasi kod va zaxiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlaridir.

■ Komponentalar Palitrasi ilovalarni ko'rishda taklif qilinadigan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat.

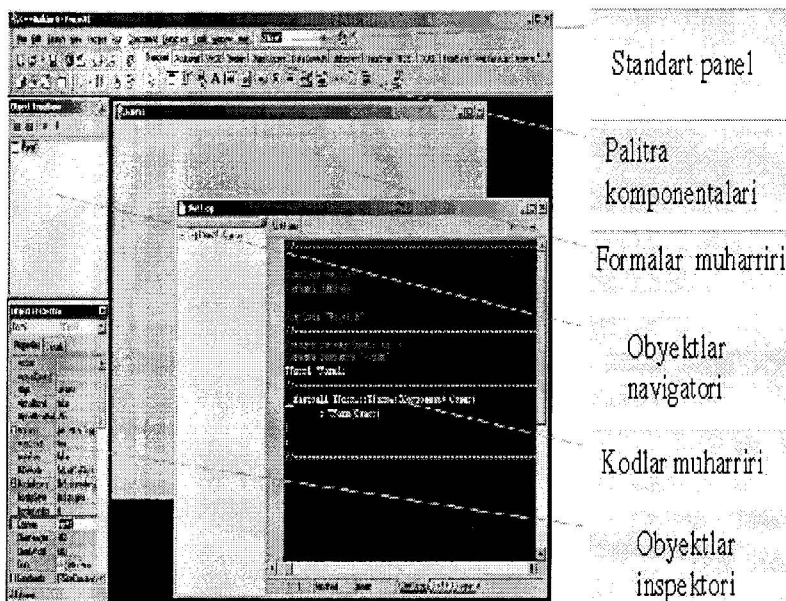
□ Shakllar Muharriri dasturning foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan.

□ Kod Muharriri dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun mo'ljallangan.

□ Obyektlar Noziri qotib qolgan chigal dasturlash zaruratisiz obyektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni

o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan obyektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi.

■ Obyektlar Xazinasini ma'lumotlarning shakl va modullari kabi obyektarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi.



Ishlab chiqish muhitining tuzilishi.

C++ Builder ilovalarni qurishning vizual metodikasini Komponentalar Palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir

komponenta (masalan, tugmacha) bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi.

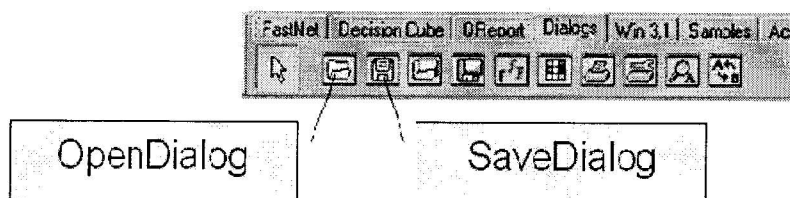
## Komponentalar palitrasi

C++ Builder 32 razryadli takomillashtirilgan Vizual Komponentalar Kutubxonasi VCL (Visual Component Library) bilan birgalikda yetkazib beriladi. Bu kutubxona eng murakkab ilovalarni ko'rish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari Palitralar komponentalarining instrumental Panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi.

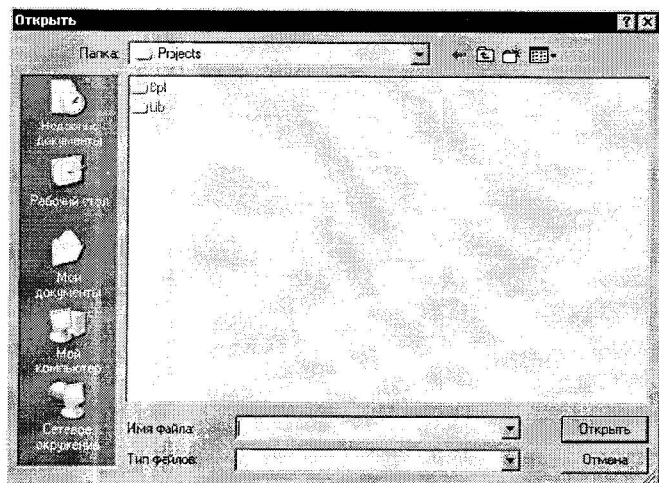
Kutubxona Windows va Windows 95 operatsion tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys obyektlarining to'liq inkapsulatsiyalanishini o'z ichiga oladi.

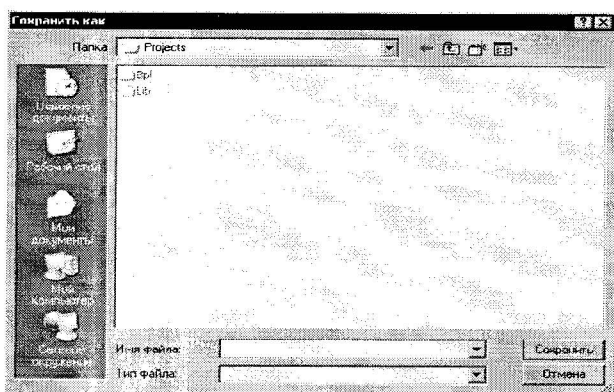
## Fayllarni izlashning dialogli (ikkita individning muloqoti) darchalari

Windows operatsyon tizimida fayllarni qidirish uchun ularni saqlash va ochishning universal dialogli darchalari ko'zda tutilgan bo'lib, ulardan foydalanishda Dialogs qo'shimcha ilovadan tegishli komponentalarni shaklda joylashtirish kerak:



Bu komponentalarni shaklda joylashtirgach, fayllar bilan ishlashning standart dialogli darchalarini chaqirib olish mumkin:





Misol: «Matn muharriri»

Dastur tavsifi

Vazifa: matnli fayllarni o'zgartirish va yaratishga qodir bo'lgan dasturni yaratish. Fayllarni diskdan ochish va kiritilgan o'zgarishlarni saqlash imkoniyatini ta'minlash.

Fayllarni qidirish, shuningdek, faylni saqlash joyini tanlash uchun standart dialoglardan hamda fayllarni ochish/saqlashdan foydalanish.

Fayl matnini Memo maydonida aks ettirish.

Muammolar

ifstream va ofstream sinflari obykti yaratilishda va fayl bilan assotsiatsiya qilinishda uzatilayotgan fayl nomidan belgilar massivi sifatida foydalanadi, standart dialoglar esa «satr» AnsiString turidagi qiymatlarni qaytaradi. Ya'ni ifstream yoki ofstream turdagi obyektga dialogli darcha qaytarayotgan qiymatning to'g'ridan-to'g'ri uzatilishi mumkin emas.

Bu muammoni hal qilish uchun satrni belgilar massiviga o'zgartirib yuborish protsedurasini yaratish tavsiya qilinadi.

Zarur bo'lgan bilimlar

Ushbu dasturni ishlab chiqish uchun ishlab chiqish muhitini standart komponentlari bilan ishlashni bilish lozim — muloqot oynalari fayllarni qidirish uchun mo'ljallangan. Bundan tashqari, fayllarni diskda matn holatida o'qish va saqlashni bilish lozim.

Shuningdek, fayllar matnini diskdan o'qib olish va diskda saqlashni bilish kerak.

Yechim

Ushbu dasturni yaratishda shaklga ikkita tugmani joylashtirish kerak. Ular mos ravishda fayllarni ochish va yopish uchun mo'ljallangan. Shuningdek, tegishli dialogli darchalarni ham joylashtirish kerak. Voqealar qayta ishlatgichlariga dialogli darcha chaqirishi (SaveDialog1->Execute) ni joylashtirish lozim. Dialogli darchalar voqealarining qayta ishlatgichi OnCanClosega fayllar bilan ishlashni amalga oshiradigan dasturiy kodni joylashtirish kerak.

Fayllar bilan ishlash dialogli darchaning OnCanClose voqeasi yuzaga kelganda, tegishli dialogli darchaning FileName xususiyatida tanlangan fayl nomi bo'ladi.

ifstream sinfi obyektining satriga yozilgan fayl haqidagi axborotni uzatish uchun satrni belgilar massivida qayta o'zgartirish kerak. Buning hammadan oson yo'li —

massivning birinchi elementiga iqtibosni uzatadigan protsedurani yaratish. Bu protsedura muntazam ravishda, elementma-element, satrdan belgilarni olgan holda massivni to'ldirishi kerak. Bu protsedura yordamida barcha zarur qayta o'zgarishlarni osongina amalga oshirish mumkin.

Fayl ichidagini Memo1 maydoniga yozish uchun ifstream sinfining getline() funksiyasi yordamida satrlarni izchil o'qib borish hamda ularni Memo1 maydoniga, bu maydonning tarmoq obykti Line(Memo1->Lines->Add(satr);) ga tegishli Add() funksiyasi yordamida yozib qo'yish kerak.

Faylda axborotni saqlash uchun unga Memo1 obykti satrlarini belgima-belgi yozib qo'yish kerak, bunda yangi satrni satr oxiri belgisi (\n) bilan boshlash kerak.

### Shakl

Formada ko'rinib turadigan komponentalardan tashqari, shuningdek, Dialogs - SaveDialog va LoadDialog qo'shimcha ilovalardan olingan komponentalar ham mavjuddir.

### C++ Builder da Grafika.

C++Builder dasturi Windows GDI funksiyalarini turli darajalarda inkapsulatsiyalaydi. Bu o'rinda bir usul muhim bo'lib, uning vositasida grafik komponentalar o'z tasvirlarini monitor ekranida taqdim etadi. GDI funksiyasi to'g'ridan-to'g'ri chaqirilganda, ushbu grafik komponentalarga qurilma konteksti deskriptori (device context handle) ni uzatish kerak. Bu deskriptor siz tanlab olgan rassomchilik ashyolari — perolar, mo'yqalamlar, shriftlarni chiqarib beradi. Grafik tasvirlar bilan ishlash tugagach, siz qurilma kontekstini dastlabki holatga keltirib qo'yishga majbursiz va shundan keyingina undan ozod bo'lishingiz mumkin.

Shu darajada detallashtirilgan grafika bilan ishlashga sizni majbur qilish o'rniga, C++Builder grafik komponentalarning Canvas (Kanva — Asos) xususiyati vositasida sodda va tugal interfeysni taklif qiladi. Bu xususiyat qurilmaning to'g'ri kontekstini nomlaydi (initsiallashtiradi) hamda siz rasm chizishni to'xtatgan kerakli vaqtda uni ozod qiladi. Asos pero, mo'yqalam va shrift tavsiflari nomidan ish ko'radigan berilgan xususiyatlarga ega.

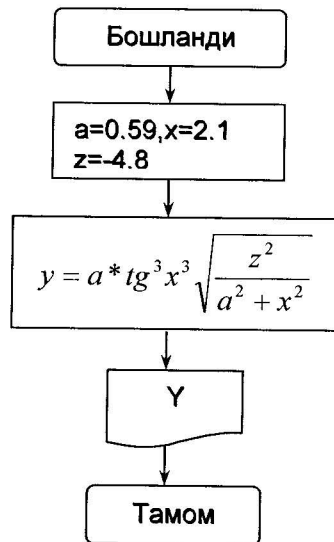
Grafik komponentalar bilan ishlashda foydalanuvchi amalga oshirishi lozim bo'lgan yagona ish — bu qo'llanayotgan rasm chizish ashyolarining tavsiflarini aniqlash. Ashyolarni yaratish, tanlash va ozod qilishda sizdan tizim zaxiralarini kuzatib borish talab qilinmaydi.

#### IV. Amaliy ish

Quyidagi hisoblash fo'rmulalari yordamida Y ni hisoblang,

Bunda:  $a=0.59$ ,  $z=-4,8$ ,  $x=2$ .  $y = a * tg^3 x^3 \sqrt{\frac{z^2}{a^2 + x^2}}$

Algoritm blok – sxemasi



Algoritmning Si tilidagi dasturi:

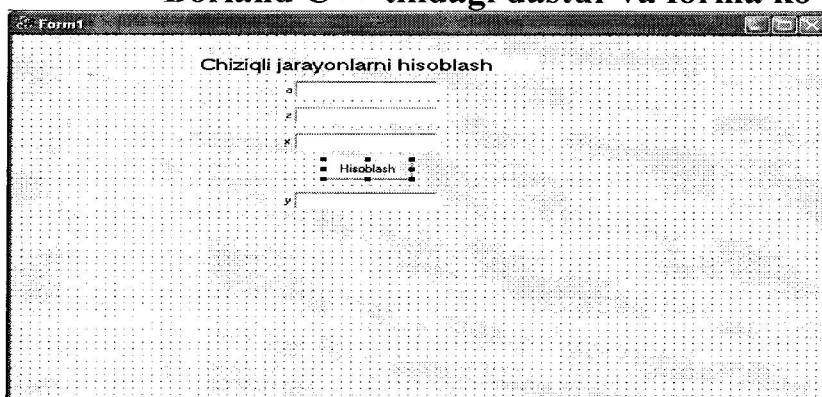
```
#include <stdio.h>
#include <iostream.h>
#include <math.h>
main()
{
    float x,z,a,k,r;
    a=0.59;
    z=-4,8;
    x=2.1;

    r=x*x*x;
    k=tan(r);
    y=a*k*k*k+sqrt(z*z/(a*a+x*x));
    printf("y=%f",y);
}
```

Natija:

Y=-1.008810

## Borland C++ tilidagi dastur va forma ko`rinishi



```
//-----

#include <vcl.h>
#include <math.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)
{
    float y,x,z,a,k,r;
    a=StrToFloat(Edit1->Text);
    z=StrToFloat(Edit1->Text);
    x=StrToFloat(Edit1->Text);
    r=x*x*x;
    k=tan(r);
    y=a*k*k*k+sqrt(z*z/(a*a+x*x));
    Edit4->Text=FloatToStr(y);
}
//-----
```

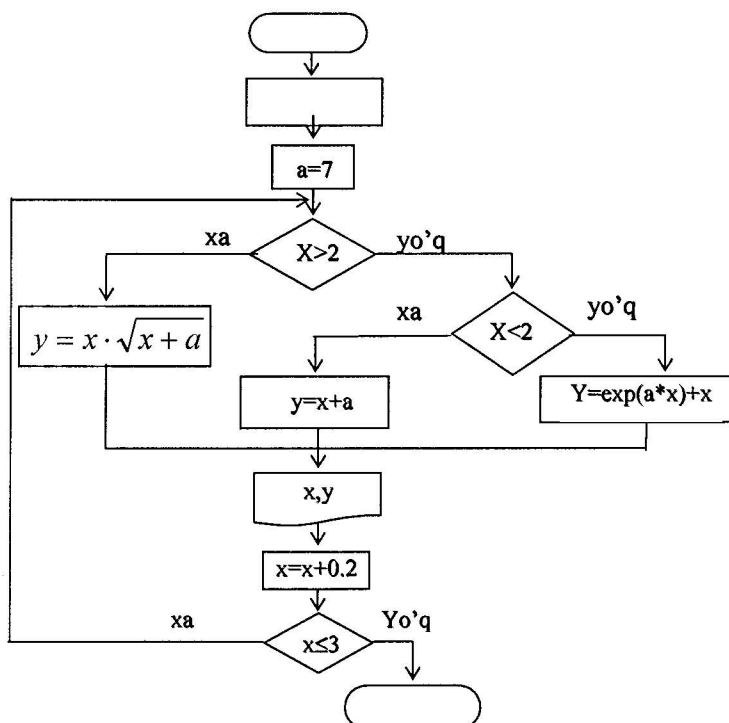
## Natija

Quyidagi tarmoqli funktsiyani argumentning turli qiymatlarida hisoblash algoritmi uchun blok-sxema tuzib, uning dasturini yozish talab qilinadi:

$$y = \begin{cases} x \cdot \sqrt{x+a} & \text{agar } x > 2, \\ x+a & \text{agar } x < 2, \\ e^{ax} + x & \text{agar } x = 2, \end{cases}$$

Bu yerda  $x \in [1; 3]$   $x$  ning o'zgarish qadami  $h_x = 0.2$ ,  $a = 7$ .

## Algoritm blok-sxemasi



### Algoritmnining Si tilidagi dasturi:

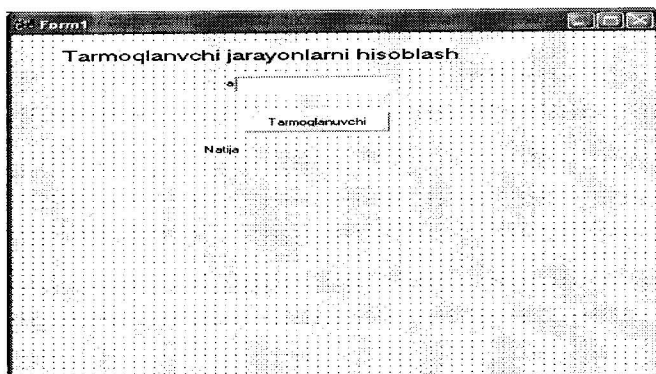
```
#include<math.h>
#include<stdio.h>

main()
{
    float y,x,a;
        x=1;
        a=7;
        Again:
        if(x>2) y=x*sqrt(x+a);
        if(x<2) y=x+a;
        if(x==2) y=exp(a*x)+x;
        printf("x = %f,  y =%f\n",x,y);
        x=x+0.2;
        if (x<=3.2) goto Again;
}
```

#### Natija:

|               |              |
|---------------|--------------|
| x = 1.000000, | y = 8.000000 |
| x = 1.200000, | y = 8.200000 |
| x = 1.400000, | y = 8.400000 |
| x = 1.600000, | y = 8.600000 |
| x = 1.800000, | y = 8.800000 |
| x = 2.000000, | y = 6.000001 |
| x = 2.200000, | y = 6.672931 |
| x = 2.400000, | y = 7.358262 |
| x = 2.600000, | y = 8.055807 |
| x = 2.800000, | y = 8.765388 |
| x = 3.000000, | y = 9.486835 |

### Borland C++ tilidagi dastur va forma ko`rinishi



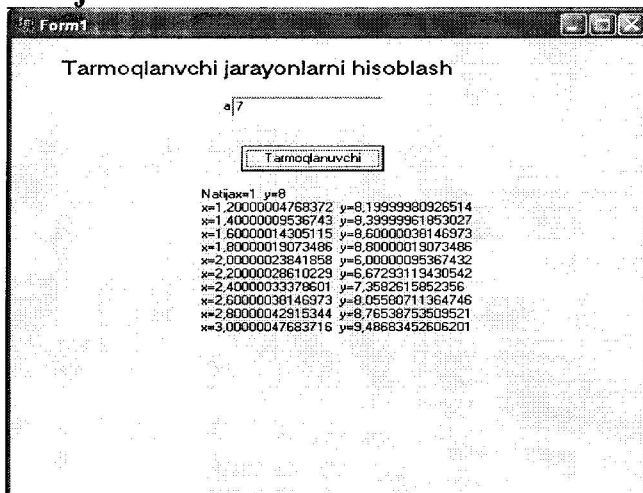
```

//-----
#include <vcl.h>
#include <math.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)
{
float y,x,a;
a=StrToFloat(Edit1->Text);
x=1;
Again:
    if(x>2) y=x*sqrt(x+a);
    if(x<2) y=x+a;
    if(x==2) y=exp(a*x)+x;
Label2->Caption=Label2->Caption+"x="+FloatToStr(x)+" y="+FloatToStr(y)+"\n";
    x=x+0.2;
    if (x<=3.2) goto Again;
}
//-----

```

## Natija



**Bizga quyidagi funksiya berilgan bo'lsin:**

$$y = a \cdot e^{-\sqrt{x}} \cdot \cos bx + c;$$

berilgan funksiyaning  $x$  ning  $x \in [1; 5]$  oraliqdagi qiymatlari uchun  $h_x=0.5$  qadam bilan hisoblashni tashkillash talab qilinadi. Bu yerda  $a=1,5$ ;  $b=2$ ;  $c=-0,75$ .

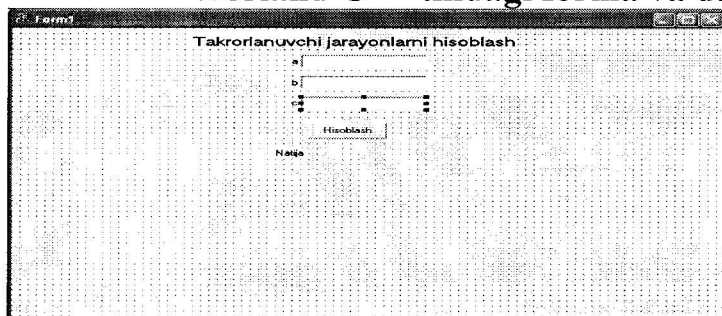
**Algoritmining Si tilidagi dasturi.**

```
#include<stdio.h>
#include<math.h>
main()
{
    float x,a,b,c,y;
    a=1.5;
    b=2;
    c=-0.75;
    for(x=1;x<=5;x+=0.5){
        y=a*exp(-sqrt(x))*cos(b*x)+c;
        printf("\n x = %f, y = %f",x,y);
    }
}
```

Natija:

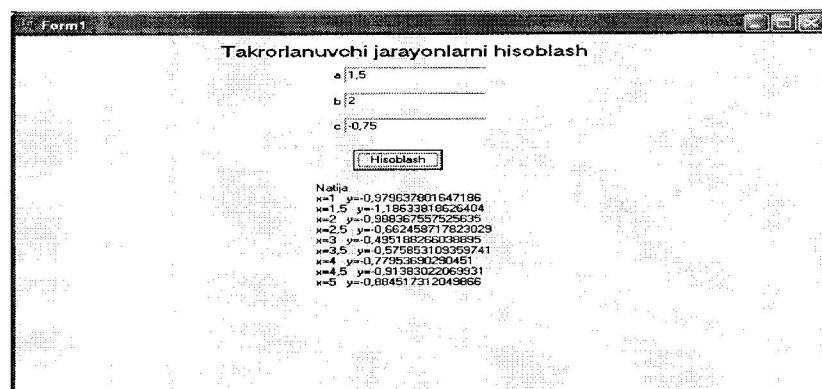
|               |               |
|---------------|---------------|
| x = 1.000000, | y = -0.979638 |
| x = 1.500000, | y = -1.186338 |
| x = 2.000000, | y = -0.988368 |
| x = 2.500000, | y = -0.662459 |
| x = 3.000000, | y = -0.495188 |
| x = 3.500000, | y = -0.575853 |
| x = 4.000000, | y = -0.779537 |
| x = 4.500000, | y = -0.913830 |
| x = 5.000000, | y = -0.884517 |

**Borland C++ tilidagi forma va dastur ko'rinishi**



```
//-----
#include <vcl.h>
#include <math.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)
{
float x,a,b,c,y;
a=StrToFloat(Edit1->Text);
b=StrToFloat(Edit2->Text);
c=StrToFloat(Edit3->Text);
for(x=1;x<=5;x+=0.5)
{
y=a*exp(-sqrt(x))*cos(b*x)+c;
Label5->Caption=Label5->Caption+"\n"+"x="+FloatToStr(x)+" y="+FloatToStr(y);
}
}
//-----
```

## Natija



## IV. Xulosa

Xulosa o'rnida shuni aytish joizki inson bu davrda faqat mehnat qurollarini yaratish bilan chegaralanib qolmay, balki u o'zining aqliy mehnatini yengillashtirish qurollarini ham yaratdi. Bunga oddiy hisob-kitob toshlaridan tortib, hozirgi kunda ham o'z kuchi va qulayligini yo'qotmagan chotlar misol bo'la oladi. Inson eng oddiy hisoblash texnikasini yaratibgina qolmasdan uni yanada takomillashtirdi. Dast avval inson eng sodda dasturlash tillarini yaratdi va ularni rivojlantirib hozirgi zamonaviy dasturlash tillari va obyektga yo'naltirilgan dasturlash tillarini yaratdi. Obyektga yo'naltirilgan dasturlash tillarining asosida B, BCPL, Turbo SI, C, Borland C++ tillari yotadi. Hozirda obyektga mo'ljallangan dasturlash tillari ichida Delphi, C++, tillari ancha ommalashgan. Obyektga yo'naltirilgan dasturlash fani hozirgi kunda axborot texnologiyalari sohasining barcha yo'nalishlarida o'rgatilmoqda.

Shu sababli men obyektga mo'ljallangan dasturlash tillari, C++tili bo'yicha kurs ishi tayyorlash jarayonida oz bo'lsada nazariy va amaliy bilimga ega bo'ldim.

C++tili obyekt ustida turli mantiqiy arifmetik xususiy amallarni bajarishga mo'ljallangan. Men ushbu kurs ishini tayorlash jarayonida ko'p internet saytlari, obyektga mo'ljallangan dasturlash tillari bo'yicha o'quv uslubiy qo'llanmalardan foydalandim.

Men ushbu C++ tilini o'rganuvchi xar bir foydalanuvchiga omad va muvofaqqiyat tilayman.

## V. Foydalanilgan adabiyotlar.

1. "Obyektga mo'ljallangan dasturlash tillari" (o'rta maxsus va oliy ta'lim o'quv yurtlari uchun o'quv qo'llanma).
2. [www.ziynet.uz](http://www.ziynet.uz)
3. WWW.Google.uz
4. C++ tilidan o'quv uslubiy qo'llanma (Toshkent 2010).
5. A.A. Abduqodirov, A.G'. Hayitov, P.P. Shodiyev, "Axborot texnologiyalari"
6. U.Y. Yo'ldoshev, P.p. Boqiyev, "Informatika".