

O'ZBEKISTON RESPUBLIKASI
OLIIY VA MAXSUS TA'LIM VAZIRLIGI
MIRZO ULUG'BEK NOMIDAGI
O'ZBEKISTON MILLIY UNIVERSITETI

Madraximov Sh.F

C++ tilida programmalash asoslari

Toshkent 2009

Annotatsiya

Qoʻllanmada C++ tilining sintaksisi, semantikasi va undagi programmalash texnologiyalari keltirilgan. Ayniqsa, C++ tili asosini tashkil etuvchi berilganlar va ularning turlari, operatorlar, funksiyalar, koʻrsatkichlar va massivlar, hamda berilganlar oqimlari bilan ishlash tushunarli ravishda bayon qilingan va sodda misollarda namunalar keltirilgan.

В предлагаемой пособии приводится синтаксис и семантика языка C++, а также технологии программирования на данном языке. Последовательно и доступно объяснены основные понятия языка C++, такие как данные и их типы, операторы, функции, указатели и массивы, потоки ввода и вывода.

In the offered educational manual the syntax, semantics and programming technologies of C++ are given. The basic concepts of language C++ as data and its types, operators, functions, pointers and files, arrives, input and output data flows are explained in accessible form.

Tuzuvchilar: dotsent Sh.F.Madraximov

Taqrizchilar: OʻzMU dotsenti N.A.Ignatev

Masʼul muharrir: OʻzMU professori M.M.Aripov

Mundarija

Kirish.....	6
1- bob. C++ tili va uning leksik asosi	7
C++ tilidagi programma tuzilishi va uning kompilyasiyasi.....	7
C++ tili alfaviti va leksemalar	9
Identifikatorlar va kalit soʻzlar	9
2- bob. C++ tilida berilganlar va ularning turlari.....	11
Oʻzgarmaslar.....	11
Berilganlar turlari va oʻzgaruvchilar	13
C++ tilining tayanch turlari	14
Turlangan oʻzgarmaslar	16
Sanab oʻtiluvchi tur.....	17
Turni boshqa turga keltirish.....	18
3- bob. Ifodalar va operatorlar.....	20
Arifmetik amallar. Qiymat berish operatori	20
Ifoda tushunchasi	20
Inkrement va dekrement amallari	21
sizeof amali	21
Razryadli mantiqiy amallar	22
CHapga va oʻngga surish amallari	24
Taqqoslash amallari	24
«Vergul» amali	25
Amallarning ustunliklari va bajarilish yoʻnalishlari.....	25
4- bob. Programma bajarilishini boshqarish.....	28
Operator tushunchasi	28
Shart operatorlari	28
if operatori	28
if - else operatori.....	30
?: shart amali.....	33
switch operatori	34
Takrorlash operatorlari	37
for takrorlash operatori.....	38
while takrorlash operatori.....	41
do-while takrorlash operatori	43
break operatori	45
continue operatori	47
goto operatori va nishonlar	48
5-bob. Funksiyalar	51
Funksiya parametrlari va argumentlari.....	52
Kelishuv boʻyicha argumentlar	56
Koʻrinish sohasi. Lokal va global oʻzgaruvchilar	58

:: amali	60
Xotira sinflari	61
Nomlar fazosi	65
Joylashtiriladigan (inline) funksiyalar	67
Rekursiv funksiyalar	68
Qayta yuklanuvchi funksiyalar	71
6-bob. Ko'rsatkichlar va adres oluvchi o'zgaruvchilar	73
Ko'rsatkichlar	73
Ko'rsatkichga boshlang'ich qiymat berish	76
Ko'rsatkich ustida amallar	78
Adresni olish amali	80
Ko'rsatkichlar va adres oluvchi o'zgaruvchilar funksiya parametri sifatida	81
O'zgaruvchan parametrli funksiyalar	83
7-bob. Massivlar	87
Berilganlar massivi tushunchasi	87
Ko'p o'lchamli statik massivlar	90
Ko'p o'lchamli massivlarni initsializatsiyalash	91
Dinamik massivlar bilan ishlash	92
Funksiya va massivlar	95
8-bob. ASCIIZ satrlar va ular ustida amallar	101
Belgi va satrlar	101
Satr uzunligini aniqlash funksiyalari	102
Satrlarni nusxalash	104
Satrlarni ulash	105
Satrlarni solishtirish	108
Satrdagi harflar registrini almashtirish	109
Satrni teskari tartiblash	111
Satrda belgini izlash funksiyalari	112
Satr qismlarini izlash funksiyalari	113
Turlarni o'zgartirish funksiyalari	115
9-bob. string turidagi satrlar	119
Satr qismini boshqa satrga nusxalash funksiyasi	120
Satr qismini boshqa satrga qo'shish funksiyasi	121
Satr qismini boshqa satr ichiga joylashtirish funksiyasi	121
Satr qismini o'chirish funksiyasi	122
Satr qismini almashtirish funksiyasi	122
Satr qismini ajratib olish funksiyasi	123
string turidagi satrni char turiga o'tkazish	123
Satr qismini izlash funksiyalari	123
Satrlarni solishtirish	125

Satr xossalarini aniqlash funksiyalari	127
10-bob. Strukturalar va birlashmalar	128
Strukturalar	128
Struktura funksiya argumenti sifatida	131
Strukturalar massivi	132
Strukturalarga ko'rsatkich	133
Dinamik strukturalar	138
Birlashmalar va ular ustida amallar	141
Foydalanuvchi tomonidan aniqlangan berilganlar turi.....	144
Makroslarni aniqlash va joylashtirish.....	145
Makroslarda ishlatiladigan amallar	148
12-bob. O'qish - yozish funksiyalari	150
Fayl tushunchasi	150
Matn va binar fayllar	151
O'qish-yozish oqimlari. Standart oqimlar	152
Belgilarni o'qish-yozish funksiyalari	154
Satrlarni o'qish - yozish funksiyalari	155
Formatli o'qish va yozish funksiyalari	156
Fayldan o'qish-yozish funksiyalari	161
Fayl ko'rsatkichini boshqarish funksiyalari	166
Adabiyotlar	172
Ilovalar	173
1-ilova	173
2-ilova	179
3-ilova	182

Kirish

Ma'lumki, programma mashina kodlarining qandaydir ketma-ketligi bo'lib, aniq bir hisoblash vositasini amal qilishini boshqaradi. Programma ta'minotini yaratish jarayonini osonlash-tirish uchun yuzlab programmalash tillari yaratilgan. Barcha programmalash tillarini ikki toifaga ajratish mumkin:

- quyi darajadagi programmalash tillari;
- yuqori darajadagi programmalash tillari.

Quyi darajadagi programmalash tillariga Assembler turidagi tillar kiradi. Bu tillar nisbatan qisqa va tezkor bajariluvchi kodlarni yaratish imkoniyatini beradi. Lekin, Assembler tilida programma tuzish zahmatli, nisbatan uzoq davom etadigan jarayondir. Bunga qarama-qarshi ravishda yuqori bosqich tillari yaratilganki, ularda tabiiy tilning cheklangan ko'rinishidan foydalangan holda programma tuziladi. Yuqori bosqich tillaridagi operatorlar, berilganlarning turlari, o'zgaruvchilar va programma yozishning turli usullari tilning ifodalash imkoniyati oshiradi va programmani «o'qimishli» bo'lishini ta'minlaydi. Yuqori bosqich tillariga Fortran, PL/1, Prolog, Lisp, Basic, Pascal, C va boshqa tillarni misol keltirish mumkin. Kompyuter arxitekturasini takomillashuvi, kompyuter tarmog'ining rivojlanishi mos ravishda yuqori bosqich tillarini yangi variantlarini yuzaga kelishiga, yangi tillarni paydo bo'lishiga, ayrim tillarni esa yo'qolib ketishiga olib keldi. Hozirda keng tarqalgan tillarga Object Pascal, C++, C#, Php, Java, Asp tillari hisoblanadi. Xususan, S tilining takomillashgan varianti sifatida C++ tilini olishimiz mumkin. 1972 yilda Denis Ritch va Brayan Kernegi tomonidan S tili yaratildi. 1980 yilda Byarn Straustrop S tilining avlodi C++ tilini yaratdiki, unda strukturali va ob'ektga yo'naltirilgan programmalash texnologiyasiga tayangan holda programma yaratish imkoniyati tug'ildi.

1- bob. C++ tili va uning leksik asosi

C++ tilidagi programma tuzilishi va uning kompilyasiyasi

C++ tilida programma yaratish bir nechta bosqichlardan iborat bo'ladi. Dastlab, matn tahririda (odatda programmalash muhitining tahririda) programma matni teriladi, bu faylning kengaytmasi «.srr» bo'ladi, Keyingi bosqichda programma matn yozilgan fayl kompilyatorga uzatiladi, agarda programmada xatoliklar bo'lmasa, kompilyator «.obj» kengaytmali ob'ekt 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.

Kompilyasiya jarayonining o'zi ham ikkita bosqichdan tashkil topadi. Boshida preprotessor ishlaydi, u matndagi kompilyasiya 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 sintaksik tahlil qiladi. Leksik tahlil jarayonida u matnni leksemalarga ajratish uchun «probel ajratuvchisini» ishlatadi. Probel ajratuvchisiga - probel belgisi (' '), '\t' - tabulyasiya belgisi, '\n' - keyingi qatorga o'tish belgisi, boshqa ajratuvchilar va izohlar hisoblanadi.

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, “*/” belgilar oralig'ida joylashgan 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.

Misol.

```
int main()
{
    // bu qator izoh hisoblanadi
    int a=0; // int d;
```

```

int c;
/* int b=15 */
/* - izoh boshlanishi
a=c;
izoh tugashi */
return 0;
}

```

Programmada d, b o'zgaruvchilar e'lonlari inobatga olinmaydi va a=c amali bajarilmaydi.

Quyida C++ tilidagi sodda programma matni keltirilgan.

```

#include <iostream.h> // sarlavha faylni qo'shish
int main ()           // bosh funksiya tavsifi
{                     // blok boshlanishi
    cout << "Salom Olam!\n"; // satrni chop etish
    return 0;         // funksiya qaytaradigan
    qiymat
}                     // blok tugashi

```

Programma bajarilishi natijasida ekranga "Salom Olam!" satri chop etiladi.

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. 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:

```
sout << <ifoda>;
```

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

Misol:

```
int Yosh;
cout<<"Yoshingizni kiriting_" ;
cin>>Yosh;
```

Butun turdagi Yosh o'zgaruvchisi kiritilgan qiymatni o'zlash-tiradi. 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.

C++ tili alfaviti va leksemalar

C++ tili alfaviti va leksemalariga quyidagilar kiradi:

- katta va kichik lotin alfaviti harflari;
- raqamlar - 0,1,2,3,4,5,6,7,8,9;
- maxsus belgilar:“ { } [] () + - / % \ ; ' : ? < = > _ ! & ~ # ^ . * ”

Alfavit belgilaridan tilning leksemalari shakllantiriladi: identifikatorlar; kalit (xizmatchi yoki zahiralangan) so'zlar; o'zgarmlar; amallar belgilanishlari; ajratuvchilar.

Identifikatorlar va kalit so'zlar

Programmalash tilining muhim tayanch tushunchalaridan biri - identifikator tushunchasidir. *Identifikator* deganda katta va kichik lotin harflari, raqamlar va tag chiziq ('_') belgilaridan tashkil topgan va

raqamdan boshlanmaydigan belgilar ketma-ketligi tushuniladi. Identifikatorlarda harflarning registrlari (katta yoki kichikligi) hisobga olinadi. Masalan, RUN, run, Run - bu har xil identifikatorlardir. Identifikator uzunligiga chegara qo'yilma-gan, lekin ular kompilyator tomonidan faqat boshidagi 32 belgisi bilan farqlanadi.

Identifikatorlar kalit so'zlar, o'zgaruvchilar, funksiyalar, nishonlar va boshqa ob'ektlarni nomlashda ishlatiladi.

C++ tilining kalit so'zlariga quyidagilar kiradi:

asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed, sizeof, static, struct, switch, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.

Yuqorida keltirilgan identifikatorlarni boshqa maqsadda ishlatish mumkin emas.

Protessor registrlarini belgilash uchun quyidagi so'zlar ishlatiladi:

_AH, _AL, _AX, _EAX, _BH, _BL, _BX, _EBX, _CL, _CH, _CX, _ECX, _DH, _DL, _DX, _EDX, _CS, _ESP, _EBP, _FS, _GS, _DI, _EDI, _SI, _ESI, _BP, _SP, _DS, _ES, _SS, _FLAGS.

Bulardan tashqari «_» (ikkita tagchiziq) belgilaridan boshlangan identifikatorlar kutubxonalar uchun zahiralangan. Shu sababli '_' va «_» belgilarni identifikatorning birinchi belgisi sifatida ishlatmagan ma'qul. Identifikator belgilar orasida probel ishlatish mumkin emas, zarur bo'lganda uning o'rniga '_' ishlatish mumkin: Cilindr_radiusi, ailana_diametiri.

2- bob. C++ tilida berilganlar va ularning turlari

O'zgarmlar

O'zgarmlar (literal) - bu fiksirlangan sonni, satrni va belgini ifodalovchi leksemdir.

O'zgarmlar beshta guruhga bo'linadi - *butun, haqiqiy (suzuvchi nuqtali), sanab o'tiluvchi, belgi (literli) va satr («string», literli satr)*.

Kompilyator o'zgarmlarni leksema sifatida aniqlaydi, unga xotiradan joy ajratadi, ko'rinishi va qiymatiga (turiga) qarab mos guruhlariga bo'ladi.

Butun o'zgarmlar. Butun o'zgarmlar quyidagi formatlarda bo'ladi:

- o'nlik son;
- sakkizlik son;
- o'n oltilik son.

O'nlik o'zgarmlar 0 raqamidan farqli raqamdan boshlanuvchi raqamlar ketma-ketligi va 0 hisoblanadi: `0; 123; 7987; 11`.

Manfiy o'zgarmlar - bu ishorasiz o'zgarmlar bo'lib, unga faqat ishorani o'zgartirish amali qo'llanilgan deb hisoblanadi.

Sakkizlik o'zgarmlar 0 raqamidan boshlanuvchi sakkizlik sanoq sistemasi (0,1,...,7) raqamlaridan tashkil topgan raqamlar ketma-ketligi:

`023; 0777; 0`.

O'n oltilik o'zgarmlar `0x` yoki `0X` belgilaridan boshlanadigan o'n oltilik sanoq sistemasi raqamlaridan iborat ketma-ketlik hisoblanadi:

`0x1A; 0X9F2D; 0x23`.

Harf belgilar ixtiyoriy registrlarda berilishi mumkin.

Kompilyator sonning qiymatiga qarab unga mos turni belgilaydi. Agar tilda belgilangan turlar programma tuzuvchini qanoatlantirmasa, u oshkor ravishda turni ko'rsatishi mumkin. Buning uchun butun o'zgarmlar raqamlari oxiriga, probelsiz `l` yoki `L` (long), `u` yoki `U` (unsigned) yoziladi. Zarur hollarda bitta o'zgarmlar uchun bu belgilarning ikkitasini ham ishlatish mumkin:

`451u, 012U1, 0xA2L`.

Haqiqiy o'zgarmlar. Haqiqiy o'zgarmlar - suzuvchi nuqtali son bo'lib, u ikki xil formatda berilishi mumkin:

- o'nlik fiksirlangan nuqtali formatda. Bu ko'rinishda son nuqta orqali ajratilgan butun va kasr qismlar ko'rinishida bo'ladi. Sonning butun yoki kasr qismi bo'lmasligi mumkin, lekin nuqta albatta bo'lishi kerak. Fiksirlangan nuqtali o'zgarmlarga misollar: `24.56; 13.0; 66.; .87;`
- eksponensial shaklda haqiqiy o'zgarmlar 6 qismdan iborat bo'ladi:

- 1) butun qismi (o'nli butun son);
- 2) o'nli kasr nuqta belgisi;
- 3) kasr qismi (o'nlik ishorasiz o'zgarmas);
- 4) eksponenta belgisi 'e' yoki 'E';
- 5) o'n darajasi ko'rsatkichi (o'nli butun son);
- 6) qo'shimcha belgisi ('F' yoki 'f', 'L' yoki 'l').

Eksponensial shakldagi o'zgarmas sonlarga misollar: $1e2$; $5e+3$; $.25e4$; $31.4e-1$.

Belgi o'zgarmaslar. Belgi o'zgarmaslar qo'shtirmoq ('','-apostroflar) ichiga olingan alohida belgilardan tashkil topadi va u char kalit so'zi bilan aniqlanadi. Belgi o'zgarmas uchun xotirada bir bayt joy ajratiladi va unda butun son ko'rinishidagi belgining ASCII kodi joylashadi. Quyidagilar belgi o'zgarmaslarga misol bo'ladi: 'e', 'e', '7', 'z', 'w', '+', 'sh', '*', 'a', 's'.

1.1-jadval. C++ tilida escape -belgilar jadvali

Escape belgilari	Ichki kod (16 son)	Nomi	Amal
\\	0x5C	\	Teskari yon chiziqni chop etish
\'	0x27	'	Apostrofni chop etish
\"	0x22	"	Qo'shtirmoqni chop etish
\?	0x3F	?	So'roq belgisi
\a	0x07	bel	Tovush signalini berish
\b	0x08	bs	Kursorni 1 belgi o'rniga orqaga qaytarish
\f	0x0C	ff	Sahifani o'tkazish
\n	0x0A	lf	Qatorni o'tkazish
\r	0x0D	cr	Kursorni ayni qatorning boshiga qaytarish
\t	0x09	ht	Navbatdagi tabulyasiya joyiga o'tish
\v	0x0D	vt	Vertikal tabulyasiya (pastga)
\000	000		Cakkizlik kodi
\xNN	0xNN		Belgi o'n oltilik kodi bilan berilgan

Ayrim belgi o'zgarmaslar '\ ' belgisidan boshlanadi, bu belgi birinchidan, grafik ko'rinishga ega bo'lmagan o'zgarmaslarni belgilaydi, ikkinchidan, maxsus vazifalar yuklangan belgilar - apostrof belgisi('), savol belgisini('?'), teskari yon chiziq belgisini ('\') va ikkita qo'shtirmoq belgisini('') chop qilish uchun ishlatiladi. Undan tashqari, bu belgi orqali belgini ko'rinishini emas, balki oshkor ravishda uning ASCII kodini sakkizlik yoki o'n oltilik shaklda yozish mumkin. Bunday belgidan boshlangan belgilar escape ketma-ketliklar deyiladi (1.1-jadval).

C++ tilida qo'shimcha ravishda wide harfli o'zgarmaslar va ko'p belgili o'zgarmaslar aniqlangan.

wide harfli o'zgarmlar turi milliy kodlarni belgilash uchun kiritilgan bo'lib, u wchar_t kalit so'zi bilan beriladi, hamda xotirada 2 bayt joy egallaydi. Bu o'zgarmlar L belgisidan boshlanadi:

```
L'\013\022', L'cc'
```

Ko'p belgili o'zgarmlar turi int bo'lib, u to'rtta belgidan iborat bo'lishi mumkin:

```
'abc', '\001\002\003\004'.
```

Satr o'zgarmlar. Ikkita qo'shtirnoq (“,”) ichiga olingan belgilar ketma-ketligi *satr o'zgarmlar* deyiladi:

```
"Bu satr o'zgarmlar va uning nomi string\n"
```

Satr ichida escape ketma-ketligi ham ishlatilishi mumkin, faqat bu ketma-ketlik apostroflar yoziladi.

Probel bilan ajratib yozilgan satrlar kompilyator tomonidan yagona satrga ulanadi (konkantenatsiya):

```
"Satr - bu belgilar massivi" /* bu satr keyingi
satrga ko'shiladi */ ", uning turi char[]";
```

Bu yozuv

```
"Satr - bu belgilar massivi, uning turi char[]";
```

yozuvi bilan ekvivalent hisoblanadi.

Uzun satrni bir nechta qatorga yozish mumkin va buning uchun qator oxirida '\n' belgisi qo'yiladi:

```
"Kompilyator har bir satr uchun kompyuter xotirasida\
satr uzunligiga teng sondagi baytlardagi alohida \
xotira ajratadi va bitta - 0 qiymatli bayt qo'shadi";
```

Yuqoridagi uchta qatorda yozilgan satr keltirilgan. Teskari yon chiziq ('\n') belgisi keyingi qatorda yozilgan belgilar ketma-ketligini yuqoridagi satrga qo'shish kerakligini bildiradi. Agar qo'shiladigan satr boshlanishida probellar bo'lsa, ular ham satr tarkibiga kiradi.

Satr xotirada joylashganda uning oxiriga '\0' (0 kodli belgi) qo'shiladi va bu belgi satr tugaganligini bildiradi. Shu sababli satr uzunligi, uning «haqiqiy» qiymatidan bittaga ko'p bo'ladi.

Berilganlar turlari va o'zgaruvchilar

Programma bajarilishi paytida qandaydir berilganlarni saqlab turish uchun o'zgaruvchilar va o'zgarmlardan foydalaniladi. *O'zgaruvchi* - programma ob'ekti bo'lib, xotiradagi bir nechta yacheykalarni egallaydi va berilganlarni saqlash uchun xizmat qiladi. *O'zgaruvchi* nomga, o'lchamga va boshqa atributlarga - ko'rinish sohasi, amal qilish vaqti va boshqa

xususiyatlarga ega bo'ladi. O'zgaruvchilarni ishlatish uchun ular albatta e'lon qilinishi kerak. E'lon natijasida o'zgaruvchi uchun xotiradan qandaydir soha zahiralanadi, soha o'lchami esa o'zgaruvchining konkret turiga bog'liq bo'ladi. Shuni qayd etish zarurki, bitta turga turli apparat platformalarda turlicha joy ajratilishi mumkin.

O'zgaruvchi e'loni uning turini aniqlovchi kalit so'zi bilan boshlanadi va '=' belgisi orqali boshlang'ich qiymat beriladi (shart emas). Bitta kalit so'z bilan bir nechta o'zgaruvchilarni e'lon qilish mumkin. Buning uchun o'zgaruvchilar bir-biridan ';' belgisi bilan ajratiladi. E'lonlar ';' belgisi bilan tugaydi. O'zgaruvchi nomi 255 belgidan oshmasligi kerak.

C++ tilining tayanch turlari

C++ tilining tayanch turlari, ularning baytlardagi o'lchamlari va qiymatlarining chegaralari 1.2-jadvalda keltirilgan.

Butun son turlari. Butun son qiymatlarni qabul qiladigan o'zgaruvchilar int (butun), short (qisqa) va long (uzun) kalit so'zlar bilan aniqlanadi. O'zgaruvchi qiymatlari ishorali bo'lishi yoki unsigned kalit so'zi bilan ishorasiz son sifatida qaralishi mumkin (1-ilovaga qarang).

Belgi turi. Belgi turidagi o'zgaruvchilar char kalit so'zi bilan beriladi va ular o'zida belgining ASCII kodini saqlaydi. Belgi turidagi qiymatlar nisbatan murakkab bo'lgan tuzilmalar - satrlar, belgilar massivlari va hakozalearni hosil qilishda ishlatiladi (2-ilovaga qarang).

1.2-jadval. C++ tilining tayanch turlari

Tur nomi	Baytlardagi o'lchami	Qiymat chegarasi
bool	1	true yoki false
unsigned short int	2	0..65535
short int	2	-32768..32767
unsigned long int	4	0..42949667295
long int	4	-2147483648..2147483647
int (16 razryadli)	2	-32768..32767
int (32 razryadli)	4	-2147483648..2147483647
unsigned int (16 razryadli)	2	0..65535
unsigned int (32 razryadli)	4	0..42949667295
unsigned char	1	0..255
char	1	-128..127
float	4	1.2E-38..3.4E38
double	8	2.2E-308..1.8E308
long double (32 razryadli)	10	3.4e-4932..-3.4e4932
void	2 yoki 4	-



Lituz.com

**To'liq qismini
Shu tugmani
bosish orqali
sotib oling!**