



ПРОГРАММИРОВАНИЕ
НА ЯЗЫКЕ C

Dərs №4

C dilində proqramlaşdırma

Mündəricat

1. for strukturu	3
2. Praktiki nümunələr	10
3. Ev tapşırığı	13

1. for strukturu

Keçən dərsdə siz dövr anlayışı ilə və C proqramlaşdırma dilində dövrləri əks etdirən bəzi strukturlar ilə: while və do while operatorları ilə tanış oldunuz. İndi isə for operatorunun daha bir növünə baxaq. Bu operator nəzəri olaraq while operatorunun tam analoqudur. Lakin praktik olaraq daha rahat idarəsi olan dövr qurmağa imkan verir.

for strukturunun ümumi sintaksisi və iş prinsipi.

```
for (dəyişənin inisializasiyası; şərtlərin yoxlanılması; dəyişənin dəyişdirilməsi)
{
    əməliyyat;
}
```

Dövrün yerinə yetirilmə prinsipi:

1. Dəyişənin inisializasiyası.
2. Şərtin yoxlanılması.
3. Əgər şərt doğrudursa, əməliyyatın yerinə yetirilməsi.
4. Əgər şərt yanlışdırsa, dövrdən sonra gələn operatorun yerinə yetirilməsi.
5. Əgər şərt doğru idisə - idarəedici dəyişənin dəyişdirilməsi.
6. Şərtin yoxlanılması. Sonra 3-cü və ya 4-cü bənd.



İstifadə nümunəsi

Bizə tanış və sadə bir nümunəyə baxaq: 1-dən 5-dək (5 də daxil olmaq şərti ilə) ədədləri dövrün köməyi ilə ekrana verin. Bunu for operatoru vasitəsilə edək.

```
#include <iostream>
void main()
{
    for(int i=1;i<=5;i++)
    {
        cout<<i<<' ';
    }
}
```

Nümunənin şərh.

1. Dövrün daxilində 1-ə bərabər dəyişən elan edilir.

Bu, elə idarəedən dəyişən olacaqdır.

2. Sonra, $i \leq 5$ şərtinin köməyi ilə bu dəyişənin qiyməti yoxlanılır;

3. Əgər şərt doğrudursa (i dəyişəni 6 qiymətini almayana qədər belə olacaq) i-nin qiyməti (`cout<<i;`) ekrana verilməsi və i idarə edən dəyişənin qiymətinin 1-ə (`i++`) dəyişməsi yerinə yetirilir. Sonra yenidən şərt yoxlanılır.

2. Əgər şərt yanlışdırsa (yəni i-nin qiyməti 6-ya bərabər oldu), onda program dövrün bağlanan fiqurlu mötərizəsindən sonra gələn sətərə keçir.

Qeyd: Nəzərə alın ki, birinci addım olan - dəyişənin yaradılması və inisializasiyası yalnız bir dəfə yerinə yetirilir.

for sintaksisinin bəzi xüsusiyyətləri:

for operatorun işinin sadəliyinə baxmayaraq o, yazılış formalarına malikdir.

İdarəedici dəyişənin inisializasiyası

1. Dəyişənin inisializasiyası və yaradılması dövr daxilində baş verir.

```
for(int x=1;x<=100;x++)
{
    cout<<x<<' ';
}
```

2. Dəyişənin yaradılması dövrə qədər, inisializasiyası isə dövr daxilində baş verir.

```
int x;
for(x=1;x<=100;x++)
{
    cout<<x<<' ';
}
```

3. Dəyişənin inisializasiyası və yaradılması dövrə qədər baş verir.

```
int x=1;
for(;x<=100;x++)
{
    cout<<x<<' ';
}
```

Hər üç nümunə funksionaldır və eynigüclüdür.

İdarəedici dəyişənin dəyişdirilməsi.

İdarəedici dəyişənin dəyişdirilməsi prosesini while və do while əməllərində olduğu kimi dövrün gövdəsinin daxilində keçirtmək olar.

```
for(int x=1;x<=100;)
{
    cout<<x<<' ';
    x++;
}
```

Şərt.

Həmçinin, strukturun şərtini buraxmaq olar, lakin bu halda şərt doğru sayılacaq (şərt üzərinə heç bir tələb qoyulmadığı üçün). Beləliklə biz həmişə doğru olan şərt və bunun nəticəsində isə **SONSUZ DÖVR** alırıq.

```
for(int x=1; ;x++)
{
    cout<<x<<' ';
}
```

Qeyd: Əgər bilmək istəyirsinizsə, şərti buraxaraq sonsuz dövrdən necə yaxa qutarmaq olar, dərsin o biri bölməsini oxumağı məsləhət görürük.

Yuxarıda yazılanlardan belə bir nəticəyə gələ bilərik: **for strukturunun heç bir hissəsi məcburi deyil.**

Göründüyü kimi, for operatorunun işi sadədir və while operatorunun işi ilə eynidir. Bəs hansını seçməli?! Bu qoyulmuş məsələdən və sizin qərarınızdan asılıdır.

break operatoru

Bəzən dövrün yerinə yetirilməsini suni olaraq dayandırmaq məcburiyyətində qalırıq. Bunun üçün artıq sizə məlum olan break operatorundan istifadə olunur. Bu operator dövrün gövdəsində, dayandırmaq tələb olunan yerdə yerləşməlidir.

Məsələn, məhz bu operatorun köməyi ilə for dövründə şərtin göstərilmədiyi halda biz sonsuz dövr problemini həll edə bilərik.

Nümunəyə baxaq:

```
#include <iostream>
using namespace std;
void main()
{
    for(int x=1; ;x++)
    {
        if(x==4) break;// əgər x 4-ə bərabədirsə, dövrü dayandır
        cout<<x<<' ';
    }
    cout<<"Bye!";
}
```

Nümunənin şərhı

1. Qaydaya əsasən dövrün şərti olmadığı üçün o həmişə doğrudur.
2. x dəyişəninə 1, 2 və 3 qiymətlərində if operatorunun şərti yerinə yetirilməyəcək. Təbii ki, break operatoru if operatorunun gövdəsində olduğu üçün işləməyəcək. Bununla belə, ekrana ardıcıl olaraq 1,2,3 ədədləri çıxacaq.
3. x dəyişəni 4-ə bərabər olanda, program if operatorunun gövdəsinə keçəcək və break əməliyyatı yerinə yetiriləcək. Bu anda dövr dayanacaq, programın yerinə yetirilməsi isə for operatorunun bağlanan fiqurlu mötərizəsindən sonrakı sətirinə keçəcək.
4. Ekranda Bye! Yazısı görünəcək.
5. 4 rəqəmi ekranda heç vaxt görünməyəcək. Belə ki, əgər break operatoru işləməyə başlayıbsa, dövr daxilində ondan aşağıda yerləşən heç bir əməliyyat yerinə yetirilməyəcək.

Qeyd: break operatoru ya dövr daxilində, ya da switch operatorunda istifadə oluna bilər. İstənilən başqa yerdə yerləşməsi kompilyasiya mərhələsində xəraya gətirib çıxara bilər.

continue operatoru

continue operatoru dövrün cari təkrarının dayandırılmasını və növbəti addıma keçidi reallaşdırır. Bəzi hallarda belə əməllər zəruri olur. Əgər continue operatoru yerinə yetirilsə, dövrün növündən asılı olaraq aşağıdakılar baş verə bilər:

while və do while dövrləri addımın yerinə yetirilməsini dayandırır və şərtin yoxlanılmasına keçirlər.

Həmçinin for dövrü də addımın yerinə yetirilməsini dayandırır. Amma o əvvəlcə idarəedici dəyişənin dəyişdirilməsinə daha sonra isə şərtin yoxlanılmasına keçir.

Nümunəyə baxaq: ekrana 0-dan 25-dək bütün tam tək ədədləri çıxarın. Layihənin adı: Odd.

```
#include <iostream>
using namespace std;
void main()
{
    for(int i=0;i<26;i++)

        {
            if(i%2==0)// əgər ədəd 2-yə qalıqsız bölünürsə
            {
                continue;// dövrün təkrarını saxlamaq və i++ -a keçmək
            }
            cout<<i<<<"\n";
        }
}
```

Nümunənin şərh

1. Dövr öz fəaliyyətini 0-dan başlayır və 25-ədək təkrarlanır.
2. Dövrün daxilində belə bir şərt nəzərə alınır: əgər i ədədi cütdürsə, dövrün (continue;) cari addımını dayandırmaq və i++ strukturuna keçmək lazımdır.

3. Fəaliyyətə başlamış continue operatorundan aşağıda yerləşənlər cari addımda artıq yerinə yetirilməyəcəklər.

4. Əgər if şərti yerinə yetirilmirsə, deməli i ədədi təkdir. Bu halda, if operatoruna baxılmır, həmin rəqəm isə ekrana verilir.

Dərsin nəzəri hissəsi ilə tanış olduqdan sonra növbəti bölməyə keçə bilərik. Burada biz bir neçə praktiki məsələlərə baxacağıq.

2. Praktiki nümunələr

Nümunə 1

Məsələnin qoyuluşu

Divar saati hər saatda zəng vurur. 12 saat ərzində saatin vurduğu zəng zərblərini saya bilən program yazın. Lahiyanın adı Time –dir.

Kodun reallaşdırılması

```
#include <iostream>
using namespace std;
void main(){
    int sum=0;
    for(int bom=1; bom<=12;bom++){
        sum+=bom;// zəng zərblərinin sayının cəmlənməsi
    }

    // saat 78 dəfə zəng vurdu.
    cout<<" Hours have punched "<<sum<<"times.\n\n";
}
```

Kodun şərhı

1. Əvvəldən 0-a bərabər sum dəyişəni elan edilir.
2. Dövr 3 stuktur vasitəsilə qurulur int bom=1; — başlanğıc inisializasiya, bom<=12; — şərt, bom++ — idarəedici dəyişənin dəyişdirilməsi.
3. Dövrün gövdəsinin daxilində idarəedici dəyişənin ümumi cəmə əlavə edilməsi yolu ilə zərblərin sayı toplanılır.
4. i dəyişəni 13 qiymətini aldıqda dövr dayanır və nəticə ekrana verilir.

Nümunə 2

Məsələnin qoyuluşu:

İstifadəçi klaviatura vasitəsilə ardıcıl olaraq tam ədədlər daxil edir. Elə ki o, klaviaturadan 0 rəqəmini daxil etdi, əvvəlcə daxil etdiyi bütün tam ədədlərin cəmi ekrana çıxarılmalıdır. Lahiyənin adı Amount-dur.

Kodun reallaşdırılması:

```
#include <iostream>
using namespace std;
void main(){
    int digit, sum=0;

    for(;;){ // sonsuz dövrün yerinə yetirilməsi

        cout<<"Enter digit:";
        cin>>digit; // ədədin daxil edilməsi
        if(digit==0)// 0 daxil edilsə
            break;//dövrün dayandırılması
        sum+=digit;// cəmləmə
    }

    // nəticənin göstərilməsi
    cout<<"Sum of digits"<<sum<<"\n\n";
}
```

Kodun şərhı:

1. Proqramda şərti sonsuz dövr istifadə edilmişdir. Yəni, dövrün dayandırılması süni yolla (break) baş verir.
2. Dövrün hər bir təkrarında istifadəçi ədəd daxil edir.
3. Yoxlama həyata keçirilir, əgər 0 ədədi daxil edilmişdirsə, dövrü dayandırmaq lazımdır, əgər daxil edilən ədəd 0 deyilsə, ədədi ümumi cəmə əlavə etmək lazımdır.
4. break operatoru işini qurtardıqdan və dövr öz fəaliyyətini dayandırdıqdan sonra klaviatura vasitəsi ilə daxil edilən bütün ədədlərin cəmi ekranda görünəcək.

Nümunə 3

Məsələnin qoyuluşu.

Elə bir program yazın ki, klaviaturadan daxil edilən ədədin bölənlərini göstərsin. Lahiyənin adı Number-dir.

Kodun reallaşdırılması.

```
#include <iostream>
using namespace std;
void main(){
    int digit;
    cout<<"Enter digit:";
    cin>>digit;

    // dövr 2-dən daxil edilən ədədə qədər olan bütün ədədləri yoxlayır
    for(int i=2;i<digit;i++){

        // əgər ədəd i-nin cari qiymətinə
        // qalıqsız bölünmürsə
        // verilmiş addımı dayandırmaqlı
        // və növbəti addıma keçməli

        if(digit%i!=0)
            continue;

        // i-ni ekranda göstərməli
        cout<<i<<"\n";
    }
}
```

Kodun şərhı:

1. İstifadəçi analiz üçün ədəd daxil edir.
2. 2-dən verilmiş ədədə qədər olan ədədləri dövr ardıcıl olaraq seçir.
3. Yoxlama aparılır: əgər verilmiş ədəd cari ədədə qalıqsız bölünmürsə dövrün bu addımını dayandırmaq və i++ (continue) hissəsinə keçmək lazımdır.
4. Əgər verilmiş ədəd cari ədədə qalıqsız bölünürsə, ekranda cari ədəd göstərilir.

3. Ev tapşırığı

İkinci dərsdə siz ədədi rəqəmlərinə ayırmağı öyrəndiniz. Bugünkü ev tapşırığı məhz bu prinsipə əsaslanır, lakin dövrlərdən də istifadə etmək lazım gələcək.

1. İstifadəçi klaviaturadan ədəd daxil edir— proqram göstərməlidir ki, bu ədəddə neçə rəqəm var. Ədəd bir dəyişənə mənimsədir.

Qeyd: Məsələn, istifadəçi 12345 ədədini daxil etmişdir. Ekranda verilməlidir ki, bu ədəddə 5 rəqəm var.

2. İstifadəçi klaviaturadan ədəd daxil edir. Bu ədədi tərs çevirmək və ekrana vermək lazımdır.

Qeyd: Məsələn, istifadəçi 12345 ədədini daxil etmişdir. Ədəd ekranda çevrilmiş vəziyyətdə — 54321 şəklində göstərilməlidir.

3. İstifadəçi klaviaturadan ədəd daxil edir. Bu ədədin rəqəmləri cəmini ekrana vermək lazımdır.

Qeyd: Məsələn, istifadəçi 12345 ədədini daxil edir. Ekrana verilməlidir ki, bu ədədin rəqəmləri cəmi 15-dir.

