

ФИЗИКАЛЫҚ ПРОЦЕССТЕРДІ КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ КУРСЫ

Бұл курс ҚҰУ - нің физика-техникалық факультетінің студенттеріне арналған. Лектор - Хасанов Манас Кабылтайевич.

Тел: +7 (707) 612 66 92
Почта: manas_hasanov@mail.ru

I-бөлім

C-SHARP АҚПАРАТТЫҚ БАҒДАРЛАМАЛАУ ТІЛІ

§2. C-Sharp тілінің Циклдары.

Циклдар.

C-Sharp тілінде циклдар шартты операторлар секілді маңызды рөл атқарады. Егер де бізге әлде бір код бөлігінің қайталанғаны керек болса, онда біз циклдарды қолданамыз. Бұндай жағдай практикада өте жиі кездеседі.

«*while*» **ЦИКЛЫ.**

«*while*» универсалды цикл болып табылады. Ол C# тілінен басқа көптеген бағдарламалау тілдерінде кездеседі. Жалпы жағдайда оның синтаксисін мына түрде жазуға болады:

```
while (шарт) {  
    // "циклдың денесі" деп аталады.  
    // код  
}
```

Мысал. Төмендегі мысал i саның $i < 3$ шарты орындалған кезде шығарады.

```
int i = 0;  
while (i < 3) {  
    Console.WriteLine("i = " + i); // i = 0, i = 1, i = 2 деп шығарады.  
    i++;  
}
```

Цикл денесі ішіндегі кодтың бір рет орындалуын **итерация** деп атайды. Мысалда көрсетілген цикл үш итерация жасайды. Логикалық типке айнала алатын кез келген айнымалы мен өрнек *while* циклының шарты бола алады.

Мысал.

```
int i = 3;  
while (true) {  
    Console.WriteLine("i = " + i); // i = 3, i = 2, i = 1, и.т.б. ....  
    i--; //бұл жерде i айнымалысы кемуде. i-- деген i = i-1 дің қысқаша жазылуы  
}
```

Бұл мысалда шартта `true` тұрғандықтан цикл еш уақытта тоқтамайды. Яғни шексіз жасайтын цикл болып табылады. Ал практикада циклды әлде бір мезетте тоқтату керек болады.

«*break*» операторы арқылы циклды үзу.

Әдетте «*while*» циклі шарт *false* айналған кезде тоқтайды. Алайда, біз оны қалаған кезде «*break*» арқылы тоқтата аламыз.

Мысал.

```
int i = 3;
while (true) {

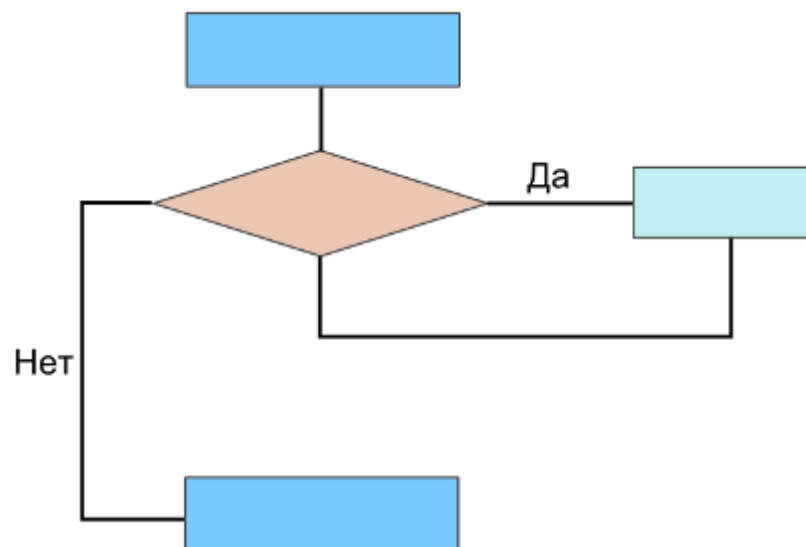
    Console.WriteLine("i = " + i); // i = 3, i = 2, i = 1 шығарады

    i--; // мына жазу арқылы i-дің мәні 1ге кеміп отырады

    if ( i == 0 ) { break; } //мына шарт орындалған кезде циклды үземіз

}
```

«*while*» циклінің блок-схемасы. Әдебиеттерде осындай жазу жиі кездеседі.



Бұл жерде ашық тіктөртбұрыш арқылы программаның негізгі коды белгіленген. Ромб арқылы циклдың шарты көрсетілген. Күлгін тіктөртбұрыш арқылы циклдың денесі көрсетілген.

Фигурлы жақша туралы.

Егер циклдың денесі тек бір ғана жазудан (өрнектен, операциядан) тұрса, онда фигурлы жақшаны жазбаса болады.

Мысал.

```
int i = 0;
while (i < 5) i++;

Console.WriteLine("i = " + i); //5
```

«do...while» ЦИКЛЫ

«do...while» циклі арқылы циклдың орындалуын тексеретін шартты цикл денесінен кейін орналастыруға болады. Яғни бұл цикл «while» қарағанда бірінше орындалады содан соң тексереді. Жалпы жағдайда оның синтаксисін мына түрде жазуға болады:

```
do {  
    // "циклдың денесі"  
} while ( условие );
```

Бұл жерде цикл алдымен орындалады, содан соң шартта *true*, туындауына тексереді. Егер шартта *true* туындаса онда циклдың денесіндегі кодты тағы да орындайды.

Мысал.

```
int i = 3;  
do {  
    Console.WriteLine("i = " + i); // 3, 2, 1 шығарады да циклдан шығып кетеді  
    i--;  
} while (i>=0)
```

«do...while» циклі, цикл бір рет болса да орындалу қажет болған жағдайларда қолданылады. Алайда, практикада сіздер *while(...)* {...} конструкциясын көбірек кездестіретін боласыздар.

Келесі итерацияға секіру: «continue»

Кей кезде циклдың денесі ішінде әлде бір шарт орындалған кезде төмендегі код бөлігін орындамай келесі итерацияға секіру қажеттілігі туындайды. Бұл қажеттілік «continue» арқылы шешіледі.

Мысал. Бұл жерде цикл тек тақ сандарды ғана шығарады.

```
int i = 0;  
do {  
    if (i % 2 == 0) continue;  
    Console.WriteLine("i = " + i); //мына код i жұп болса орындалмайды  
    i++; //мына код i жұп болса орындалмайды  
} while (i<=10)
```

Айнымалыны циклдың ішінде жариялау.

Цикл ішінде жарияланған айнымалылар тек осы цикл ішінде ғана өмір сүреді. Циклдан шыққаннан кейін олар жойылып кетеді де біз оларды қолдана алмай қаламыз.

Мысал.

```
int i = 0;
do {
    if (i % 2 == 0) continue;

    int j = i + 1;

    Console.WriteLine("i = " + i); // шығарады 1, 3, 5, 7, 9
    Console.WriteLine("j = " + j); // шығарады 2, 4, 6, 8, 10

    i++;
} while (i<=10)

// Мына жерде j айнымалысын қолдана алмаймыз. Оны компилятор жойып жіберген
```

«for» циклы.

«for» циклы практикада ең көп кездесетін цикл болып табылады. Оның синтаксисы мынадай:

```
for (бастапқы айнымалының мәні; цикл үзетін шарт; қадам) {

    // ... цикл денесі ...

}
```

Түсінікті болу үшін мысал қарастырайық.

Мысал.

```
for (int i = 0; i < 5; i++) {

    Console.WriteLine("i = " + i); // 0, 1, 2, 3, 4 шығарады i = 5 болғанда үзіледі.

}
```

Енді мынадай кестені қарастырайық:

бастапқы айнымалының мәні	$i = 0$	i -дың осы мәнімен циклды бастайды.
шарт	$i < 5$	Әр итерация сайын осы шартты тексереді. Егер шартта <i>false</i> , туындаса онда цикл үзіледі.
қадам	$i++$	Шартты тексерер алдында i -мәңін өзгертеді.
цикл денесі	<code>Console.WriteLine("i = " + i);</code>	Егер шарт <i>true</i> қабылдаса, онда қайта қайта орындалады.

Мысал.

```
for (int i = 4; i > 0; i--) {  
    Console.WriteLine("i = " + i); // 4, 3, 2, 1 шығарады  
}
```

1. Бірінші итерация алдында `int` типті, аты `i` болатын айнымалы жарияланып оған 4 деген сан меншіктелген.
2. `i > 0` шарты тексеріледі. Егер шарт қанағаттандырылса(`true` қабылдаса), онда компилятор цикл денесі ішіндегі кодты орындайды. Ал егер де шарт қанағаттандырылмаса, онда цикл үзіледі.
3. Цикл денесі ішіндегі код орындалған соң `i`-дың мәні бірге азайады `i--`.
4. Содан кейін `i > 0` шарты тексеріледі де, шарт қанағаттандырылса цикл қайта орындалады.

Бұл жерде `i` айнымалысы цикл ішінде жарияланған болып табылады. Енді ары қарай мысалдар қарастырайық:

Есеп 2.1.

`a` деген бүтін сан және `n` деген натурал сан (оң бүтін сан) берілген a^n есептеп табыңыз.

Шешім.

```
using System;  
namespace App  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            //a - деген бүтін сан.  
            int a = 2;  
            //n - деген натурал сан.  
            uint n = 8;  
  
            //0 мен n арасында өзгертін i деген бүтін айнымалы еңгіземіз.  
            int i = 1;  
            //b деген бүтін айнымалы еңгіземіз.және b=a^n.  
            int b = 1;  
  
            while (i<=n) {  
                b = b * a;  
                i++;  
            }  
  
            Console.WriteLine("a^n = " + b);  
  
            Console.Read();  
        }  
    }  
}
```

Екінші шешім.

```
using System;
namespace App
{
    class Program
    {
        static void Main(string[] args)
        {
            //a - деген бүтін сан.
            int a = 2;
            //n - деген натурал сан.
            uint n = 8;

            //0 мен n арасында өзгертін i деген бүтін айнымалы еңгіземіз.
            int i = 1;
            //b деген бүтін айнымалы еңгіземіз.және b=a^n.
            int b = 1;

            while (i>0) {
                b = b * a;
                i--;
            }

            Console.WriteLine("a^n = " + b);

            Console.Read();
        }
    }
}
```

Есеп 2.2.

Арифметикалық прогрессияның алғашқы 100 мүшесінің қосындысын табыңыз.
 $a_1 = 1$ и $d = 0.5$. Арифметикалық прогрессияның алғашқы n мүшесінің қосындысы мына формула арқылы есептеледі.

$$S_{n=100} = \frac{2a_1 + (n-1)d}{2}n = \frac{2 + (99)0.5}{2}100 = 2575$$

Алайда біз тура әдіспен есептейтін боламыз.

Шешім.

```
static void Main(string[] args)
{
    float a_1 = 1;
    float d = 0.5f;
    float s = 0;

    for( int i = 1; i <=100; i++)
    {
        float a_i = a_1 + d*(i-1);
        s = s + a_i;
        Console.WriteLine("S_" + i + "]" = " + s);
    }

    Console.WriteLine("S_n = " + s);

    Console.Read();
}
```

Студент өз бетімен орындайтын тапсырма. Есеп №1.

Геометриялық прогрессияның алғашқы 50 мүшесінің қосындысын табыңыз.
 $a_1 = 1$ и $q = 0.5$. Тура әдіспен есептеңіз.

Студент өз бетімен орындайтын тапсырма. Есеп №2.

Гармоникалық қатардың алғашқы 500 мүшесінің қосындысын табыңыз.

$$\sum_{k=1}^{500} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{500}$$