

# Операторы цикла

***Операторы цикла позволяют выполнить участок программы требуемое число раз***

- ***Цикл for***
- ***Циклы while***

# Цикл *for* (перебор)

- Оператор цикла *for* имеет следующий вид:

**for** (инициализация; условие; изменение)  
оператор

- Оператор, стоящий после закрывающей скобки, **задает тело цикла**. В большинстве случаев телом цикла является **блок**. Сколько раз будет выполняться тело цикла, зависит от трех управляющих элементов, заданных в скобках.
    - **Инициализация** задает начальное значение некоторой переменной, называемой управляющей переменной цикла.
    - **Условие** задает условие выполнения цикла, при обращении в FALSE цикл завершается.
    - **Изменение** описывает, как меняется переменная цикла в каждой итерации выполнения.
- Если условие цикла истинно, то выполняется тело цикла, затем изменяются значения переменной цикла и снова проверяется условие. Как только условие становится ложным, цикл завершает свою работу.

- Например, для вывода на экран значений целых чисел от 1 до 10 можно использовать следующий цикл:

```
for (int i = 1; i <= 10; i++)  
    Console.Write("{0} ", i);
```

- *Переменная цикла часто объявляется непосредственно в инициализации и соответственно являются локальной в цикле переменной, так что после завершения цикла она перестанет существовать.*

**Пример.** Вычислить сумму чисел от 1 до 100.

```
int s = 0;
    for (int i = 1; i <= 100; i++)
        s = s + i;
    Console.WriteLine(s);
```

Выражение инициализации присваивает переменной  $i$  значение 1.

Выражение цикла прибавляет 1 к переменной  $i$ , а в теле цикла происходит суммирование переменных  $s$  и  $i$ .

Цикл завершает свою работу когда переменная  $i$  достигнет значения 101. То есть последним числом, которое будет прибавлено к переменной  $s$ , будет 100.

В конце работы оператора **for** переменная  $s$  будет содержать число 5050 – сумму натуральных чисел от 1 до 100.

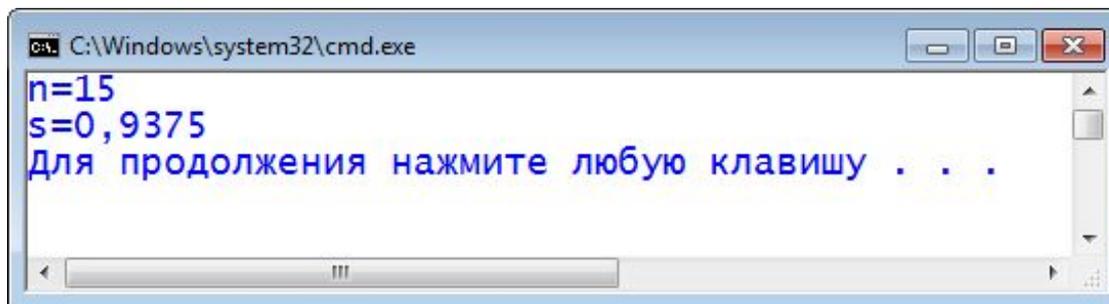
```
int s = 0;
for (int i = 1; i >= 100; i++) s = s + i; // ни разу!
```

**Пример.** Для заданного натурального  $n$  вычислить сумму

$$S = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \dots + \frac{1}{n \cdot (n + 1)}$$

$$S = \sum_{i=1}^n \frac{1}{i \cdot (i + 1)}$$

```
Console.Write("n=");  
string w = Console.ReadLine();  
int n = Convert.ToInt32(w);  
double s = 0;  
for (i = 1; i <= n; i++)  
    s = s + 1.0 / i / (i + 1); // s += 1.0 / i / (i + 1)  
Console.WriteLine("s={0}",s);
```



```
C:\Windows\system32\cmd.exe  
n=15  
s=0,9375  
Для продолжения нажмите любую клавишу . . .
```

**Пример.** Распечатать таблицу значений функции  $y=y(t)$ , для аргумента, изменяющегося в заданных пределах (от -3 до 14) с заданным шагом (1,8). Значения, большие 100 выводить без дробной части.

$$y = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < 10 \\ x^2, & x \geq 10 \end{cases}$$

```
double Xn=-3, Xk=14, dX=1.8, y;
Console.WriteLine("-----");
Console.WriteLine("|      X      |      Y      |");
Console.WriteLine("-----");
for ( double x = Xn; x <= Xk; x += dX)
{
    if ( x < 0 ) y = 0;
    else if ( x < 10 ) y = x;
    else          y = x * x;
    if (y>100) Console.WriteLine("| {0,9:F2} | {1,9:F0} |", x, y);
    else Console.WriteLine("| {0,9:F2} | {1,9:F2} |", x, y);
}
Console.WriteLine("-----");
```

```
C:\Windows\system32\cmd.exe
```

X	Y
-3,00	0,00
-1,20	0,00
0,60	0,60
2,40	2,40
4,20	4,20
6,00	6,00
7,80	7,80
9,60	9,60
11,40	129
13,20	174

Для продолжения нажмите любую клавишу .

## Цикл ***while*** с предусловием.

Оператор **while** используется для организации циклического выполнения *тела цикла, пока **выполняется** некоторое условие.*

**while** (<выражение>) <тело оператора>;

В конструкции ***while*** вычисляется **выражение**. Если его значение true, то выполняется тело оператора и выражение вычисляется снова. Этот цикл продолжается до тех пор, пока значение выражения не станет false, после чего выполнение программы продолжается с места после тела оператора.

**Пример.** Вычислить сумму чисел от 1 до 100.

```
int s = 0;
int i = 100;
while(i > 0)
{
    s = s + i; i--;
}
```

В теле цикла к сумме  $s$  (изначально = 0) прибавляется слагаемое  $i$  (изначально = 100) и это слагаемое уменьшается на 1. Цикл закончит свое выполнение, когда слагаемое  $i$  станет равным нулю и не будет выполняться условие  $i > 0$ .

## Оператор **do-while** – цикл с постусловием.

Структура **do...while** используется для организации циклического выполнения оператора или операторов, называемых *телом цикла*, до тех пор, пока *не окажется нарушенным* некоторое условие.

**do <тело оператора> while (<выражение>);**

Тело оператора **do** выполняется один или несколько раз до тех пор, пока выражение **<выражение>** не станет **ложным**. Вначале выполняется **<тело оператора>**, затем вычисляется **<выражение>**.

Если выражение **ложно**, то оператор **do** завершается и управление передается следующему оператору в программе.

Если выражение **истинно**, то тело оператора выполняется снова и снова проверяется выражение.

**Пример. ВЫЧИСЛИТЬ** сумму чисел от 1 до 100.

```
int s=0, i = 100;  
do  
{  
    s = s + i;  
    i--;  
} while(i > 0);
```

## Прерывание цикла:

### операторы **break**, **continue**, **return**.

- Оператор **break** прерывает выполнение тела любого цикла **for**, **do**, **while** и передает управление следующему за циклом выполняемому оператору.
- Для прерывания циклов, расположенных в функциях, можно воспользоваться оператором **return**. В отличие от оператора **break**, оператор **return** прервет не только выполнение цикла, но и выполнение той функции, в которой расположен цикл.
- Оператор **continue** прерывает выполнение текущей итерации цикла и передает управление на следующую итерацию.