

## Лекция 6 ОПЕРАТОР ЦИКЛА FOR

### 6.1 Формат записи оператора цикла for

Оператор цикла **for** применяется в программах, когда заранее известно количество циклических операций.

Формат записи оператора **for**:

for(выражение1;выражение2;выражение3)

{ операторы; }, где

выражение1 – задает начальное значение управляющей переменной цикла; выражение2 – определяет конечное значение управляющей переменной цикла; выражение3 – задает шаг изменения управляющей переменной цикла.

Например: for (i = 0; i <= 10; i++) {операторы цикла}

Графическое представление цикла for изображено на рисунке 6.1

Начальное  
значение  
управляющей  
переменной.  
Проверка  
конечного  
значения  
управляющей  
переменной.  
Операторы  
цикла.  
Изменение шага  
управляющей  
переменной.

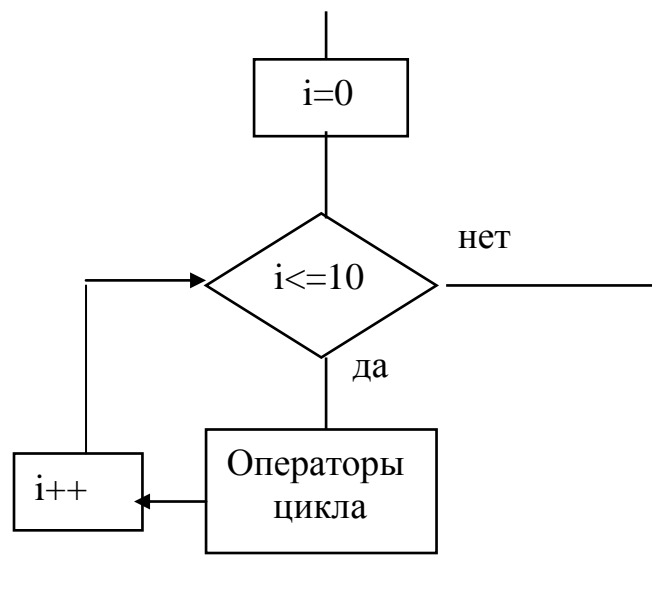


Рисунок 6.1 – Графическое представление оператора for.

Рассмотрим еще примеры записи цикла for:

1) for (int i = 0; i <= 10; i++) {операторы цикла}

В данном примере записан оператор цикла for, в котором управляющая переменная целого типа объявляется внутри цикла и изменяется от 0 до 10 (включительно) с шагом 1.

2) for ( i = 10; i >= 0; i--) {операторы цикла}

В этом примере объявленная ранее переменная i используется в качестве управляющей переменной, которая изменяется от 10 до 0 (включительно) с шагом минус 1.

3) for ( x = 0; x <= 1; x = x + 0.1) {операторы цикла }

В данном примере в качестве управляющей переменной используется переменная вещественного типа x, которая изменяется от 0 до 1 с шагом 0.1.

В языке C# имеется специальный цикл (foreach), специально предназначенный для работы с данными массивов, коллекций и других форм объединения множества данных. Формат записи и работа этого цикла будет рассмотрена в теме массивы.

## 6.2 Использование оператора цикла for в программе

Для демонстрации работы оператора цикла for рассмотрим решение учебной задачи 6.1.

Задача 6.1 Напечатать 10 случайных целых чисел в диапазоне от минус 50 до 50. Найти и напечатать наибольшее и наименьшее числа.

Разработку алгоритма решения задачи не обязательно заканчивать изображением его структурной схемы. Алгоритм может быть представлен и в словесной форме – подробное описание всех шагов решения задачи.

Сначала нам необходимо объявить две переменные max и min, в которые мы будем записывать соответственно максимальное и минимальное значения из 10 случайно сформированных чисел.

Поскольку мы знаем диапазон случайных чисел, то переменной max необходимо присвоить число меньше или равное минимальному значению диапазона, например, минус 100, а переменной min число больше или равное максимальному значению диапазона. Естественно, лучше всего переменным max и min присвоить первое сформированное случайное число, но по условию задачи их нужно формировать 10 и, конечно лучше это делать в цикле, а нам необходимо задать некоторые значения до цикла. Если не использовать операции присваивания, то по умолчанию этим переменным будут присвоены нулевые значения, но число 0 является частью заданного диапазона и возможен вариант (все 10 сформированных чисел больше или меньше 0), когда нулевые значения могут исказить результат работы программы.

Итак, необходимо сформировать 10 случайных чисел, поэтому целесообразно использовать цикл for с управляющей переменной целого типа, изменяющейся от 1 до 10 и с шагом равным единице.

Тело цикла должно включать формирование случайного числа a в заданном диапазоне, его печать и поочередное сравнение этого числа со значениями max и min. Если a больше max, то в max необходимо записать значение a. Если a меньше min, то в min необходимо записать a (первое сформированное случайное число всегда будет больше max и меньше min).

По окончании 10 циклических операций, необходимо напечатать значения переменных max и min.

Исходный код программы:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
```

```

static void Main()
{
    int i, j, k, a, max, min;
    max = -100; min = 100; j = 0; k = 0;
    Random rnd = new Random();
    for(i=1;i<=10;i++)
    {
        a=rnd.Next() % 101 - 50;
        Console.Write("  {0}",a.ToString());
        if (a>max) {max = a; j = i;}
        if (a<min) {min = a; k = i;}
    }
    Console.WriteLine();
    Console.WriteLine("max={0} НОМЕР={1}",max,j);
    Console.WriteLine("min={0} НОМЕР={1}",min,k);
    Console.WriteLine("Для продолжения нажмите клавишу
Enter");
    Console.ReadLine();
}
}

```

Работа программы:

-33 -26 -10 -16 -44 40 -50 -35 -45 -38

max=40 НОМЕР=6

min=-50 НОМЕР=7

Для продолжения нажмите клавишу Enter

В дополнение к разработанному алгоритму мы ввели еще две переменные **j** и **k**, в которые запоминали соответственно номера максимального и минимального чисел в последовательности сформированных чисел.

В некоторых языках программирования изменять значение управляющей переменной в теле цикла категорически запрещено. В языке **C#** это разрешается. Например, фрагмент программы:

```

for(i=0;i<=10;i++)
{
    Console.Write("  {0}",i.ToString());
    if (i == 3) i = 6;
}
Console.WriteLine();

```

напечатает 0 1 2 3 7 8 9 10

В записи оператора цикла **for** можно объявлять управляющую переменную. Например:

```
for (int j=1;j<=10;j++) { операторы; }
```

Объявив переменную **j** внутри цикла **for**, мы автоматически определяем «время жизни» этой переменной, равное времени работы этого цикла. Т.е.

при запуске цикла `for` в памяти компьютера выделяется место под значения управляющей переменной `j`, а по окончании цикла это место памяти будет освобождаться – помечаться как свободное. Этот способ объявления локальной переменной применяется при создании программ, одним из требований, к которым является минимальный объем занимаемой памяти компьютера.

Оператор цикла `for` позволяет использовать несколько управляющих переменных, например,

```
for ( i=1, j=10; i<j; i + +, j - -)
```

Можно превратить цикл `for` в цикл с условием, например,

```
for ( i=0; (char) Console.Read() != 'n'; i++)
```

Эта запись цикла `for` связывает окончание работы цикла не с управляющей переменной, а с вводом символа `'n'`.

В языке `C#` возможны некоторые экзотические формы записи цикла `for`, например,

`for (i=0; i<=10) { . . . ; i++ }` – шаг изменения управляющей переменной вынесен из описания заголовка цикла в его тело;

`for ( ; ; )` – бесконечный цикл и т.д.

В рассматриваемых учебных программах основным требованием является наглядность алгоритма работы программы, что часто противоречит требованию ее минимального объема, поэтому все экзотические формы записи цикла **for** мы не рассматриваем.

## Оператор цикла for. Общая форма. Примеры решения задач

### 1. Назначение оператора цикла for. Общая форма

Цикл for предназначен для организации циклического процесса. С помощью цикла for можно организовывать циклический процесс любого типа, в котором:

- количество итераций цикла заведомо известно;
- количество итераций цикла неизвестно и определяется на основе выполнения некоторого условия.

Общая форма оператора цикла for

```
for (инициализация; условие; выражение)
{
    // некоторые инструкции, операторы
    // ...
}
```

Где:

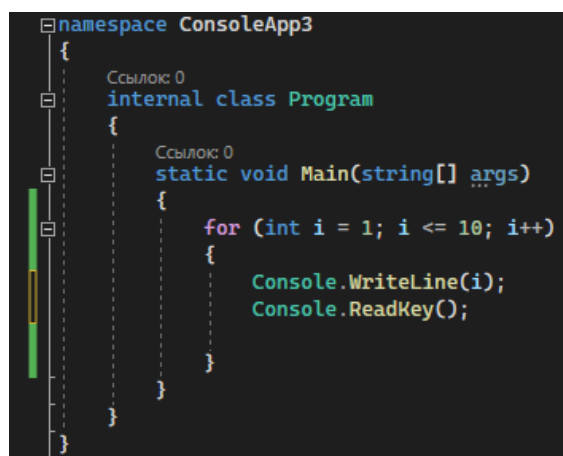
✓ инициализация – выражение, инициализирующее переменную-счетчик. Эта переменная определяет количество итераций, которые должны выполняться в цикле;

✓ условие – логическое выражение, которое определяет необходимость выполнения следующей итерации цикла. Если условие = true, то выполняется следующая итерация цикла. Если значение условие = false, то происходит прекращение выполнения цикла и переход к следующему оператору, который следует за оператором for;

✓ выражение – некоторое выражение, изменяющее значение переменной-счетчика. Необязательно выражение может изменять значение переменной-счетчика. Также переменная-счетчик может изменять свое значение в теле цикла.

### Разновидности цикла for. Примеры

**Пример 1.** Вывести все числа от 1 до 10.



```
namespace ConsoleApp3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine(i);
                Console.ReadKey();
            }
        }
    }
}
```

**Пример 2.** Вывести таблицу умножения для определенного числа (например, для 5).

```
namespace ConsoleApp3
{
    Ссылка: 0
    internal class Program
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            int number = 5;
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine($"{number} x {i} = {number * i}");
            }
            Console.ReadLine();
        }
    }
}
```

**Пример 3.** В примере опущена часть, которая инициализирует переменную-счетчик. Вычисляется сумма.

$$S = 2 + 4 + 8 + \dots + 100$$

```
namespace ConsoleApp3
{
    Ссылка: 0
    internal class Program
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            // вычислить сумму 2+4+8+...+100
            int sum = 0;
            int i = 0;

            // нет части инициализации переменной-счетчика
            for (; i <= 100; i += 2)
                sum += i;
            Console.WriteLine("Ответ " + sum);
            Console.ReadLine();
            // sum = 2550
        }
    }
}
```

**Пример 4.** Вывести все четные числа от 1 до 50.

```
namespace ConsoleApp3
{
    Ссылка: 0
    internal class Program
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            for (int i = 2; i <= 50; i += 2)
            {
                Console.WriteLine(i);
            }
            Console.ReadLine();
        }
    }
}
```

**Пример 5.** В примере цикл for не содержит инициализации и условия. Задано натуральное число n. Определить максимальную цифру этого числа.

```
Ссылка: 0
static void Main(string[] args)
{
    int n;
    int t, d;
    int max;

    // ввод n
    Console.Write("n = ");
    n = Convert.ToInt32(Console.ReadLine());

    t = n;
    max = 0; // максимальная цифра

    for (; t > 0;) // цикл не содержит инициализации и прироста счетчика
    {
        d = t % 10;
        if (max < d) max = d;
        t = t / 10;
    }

    Console.WriteLine("Max = {0}", max);
    Console.ReadKey();
}
```

**Пример 6.** Найти сумму всех целых чисел от 100 до 200. Фрагмент программы, которая решает данную задачу с использованием цикла for следующий:

```
namespace ConsoleApp3
{
    Ссылка: 0
    internal class Program
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            int sum;
            int i;
            sum = 0;
            for (i = 100; i <= 200; i++)
                sum += i;
            Console.WriteLine("Max = {0}", sum);
            Console.ReadKey();
        }
    }
}
```

### Задание по программированию: Ряд - 1

Даны два целых числа A и B (при этом  $A \leq B$ ). Выведите все числа от A до B включительно.

#### Формат ввода

Вводятся два целых числа.

#### Формат вывода

Выведите ответ на задачу.

```
1  Тест 1
2  Входные данные:
3  1
4  10
5
6  Вывод программы:
7  1 2 3 4 5 6 7 8 9 10
8
9
10
11 Тест 2
12 Входные данные:
13 -3
14 14
15
16 Вывод программы:
17 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
18
```

Рисунок 1 Пример

### Задание по программированию: Ряд - 2

Даны два целых числа A и B. Выведите все числа от A до B включительно, в порядке возрастания, если  $A < B$ , или в порядке убывания в противном случае.

#### Формат ввода

Вводятся два целых числа.

#### Формат вывода

Выведите ответ на задачу.

```
1  Тест 1
2  Входные данные:
3  1
4  10
5
6  Вывод программы:
7  1 2 3 4 5 6 7 8 9 10
8
9
10
11 Тест 2
12 Входные данные:
13 10
14 1
15
16 Вывод программы:
17 10 9 8 7 6 5 4 3 2 1
18
```

Рисунок 2 Пример



Дополнительные задания:

1. Вывести все четные числа от 1 до 20.
2. Вывести сумму всех чисел от 1 до 50.
3. Проверить, является ли число простым.
4. Вычислить факториал числа
5. Найти сумму всех нечетных чисел от 1 до 30.
6. Вывести числа от 20 до 1 в обратном порядке.
7. Вывести числа от 1 до 100, заменяя числа, которые содержат цифру 7, на "Boom".
8. Найти сумму чисел от 1 до 100, которые делятся на 3 или на 5.