

## Практическая работа №9

### Составление программ циклической структуры. Оператор цикла `for`.

Цель работы: научиться создавать, вводить в компьютер, выполнять и исправлять простейшие программы с циклическим оператором.

Норма времени: 2 часа.

Оборудование: персональный компьютер, Visual Studio 2019.

#### Ход работы

Алгоритм называется циклическим, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных.

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.

Если оператор цикла `for` применяется для одного оператора, то в общей форме фигурные скобки можно опустить

`for` (инициализация; условие; выражение)

*оператор*;

здесь **оператор** – один оператор или одна инструкция.

**Пример 1.** В примере опущена часть, которая инициализирует переменную-счетчик. Вычисляется сумма

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

Фрагмент кода, решающий данную задачу следующий:

```

// вычислить сумму 2+4+8+...+100
int sum = 0;
int i = 0;

// нет части инициализации переменной-счетчика
for (; i <= 100; i += 2)
    sum += i;
// sum = 2550

```

**Пример 2.** В примере опущена часть, которая проверяет условие выполнения следующей итерации. Пример вычисляет сумму элементов массива **A**.

```

// вычислить сумму элементов массива A
double[] A = { 2.5, 1.2, 0.8, 3.3, 4.2 };
double sum = 0;
int i;

// нет части, которая проверяет условие выполнения цикла
for (i = 0; ; i++)
{
    if (i == A.Length) break;
    sum += A[i];
}
// sum = 12.0

```

**Пример 3.** В примере опущена часть выражения, которая изменяет переменную-счетчик. Дано вещественное число **a** и натуральное число **n**. Вычислить:

$$a \times (a+1) \times \dots \times (a+n-1)$$

Фрагмент кода, решающий данную задачу

```

// вычислить произведение a*(a+1)*...*(a+n-1)
int mult;
int i;
int n, a;

n = 5;
a = 4;
mult = 1;

// отсутствует часть прироста переменной-счетчика i
for (i = 0; i < n; )
{
    mult = mult * (a + i);
    i++;
}
// mult = 6720

```

**Пример 4.** В примере цикл **for** не содержит инициализации и условия. Задано натуральное число **n**. Определить максимальную цифру этого числа.

Решение данной задачи с использованием цикла **for** (приложение типа **Console Application**)

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;

namespace ConsoleApplication4
{
    class Program
    {
        static void Main(string[] args)
        {
            // определить максимальную цифру числа n
            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);
        }
    }
}

```

Результат выполнения вышеприведенной программы:

n = 283915

Max = 9

**Пример 5.** В примере цикл **for** не содержит условия и выражения. Выход из цикла **for** осуществляется с помощью инструкции **break**.

Задан массив **B** чисел типа **float**. Найти позицию **pos** первого элемента массива, значение которого лежит в диапазоне от -5 до +5.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication4
{
    class Program
    {
        static void Main(string[] args)
        {
            // найти позицию первого вхождения

```

```

float[] B = { 12.4f, -11.9f, 7.5f, 2.3f, 1.8f };
int pos;

for (pos = 0; ; ) // цикл for не содержит условия и
выражения
{
    if (pos == B.Length)
        break;
    if ((B[pos] >= -5) && (B[pos] <= 5))
        break;
    pos++;
}

if (pos == B.Length)
    Console.WriteLine("Искомое элемента нет в массиве.");
else
    Console.WriteLine("pos = {0}", pos);
}
}

```

Результат выполнения программы

pos = 3

**Выполнить задание:** Написать программу вывода на экран следующей информации:

1. целые числа 1, 3, 5, ..., 101 в строчку через пробел;
2. целые числа 10, 12, 14, ..., 80 в обратном порядке в столбик;
3. таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, 3, ..., 100 фунтов (1 фунтов = 453г);
4. таблицу перевода 5, 10, 15, ..., 500 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры);
5. таблицу стоимости для 10, 20, 30, ..., 200 штук товара, при условии, что одна штука товара стоит  $x$  руб (значение  $x$  водится с клавиатуры);
6. таблицу перевода расстояний в дюймах в сантиметры для значений 2, 4, 6, ..., 24 дюймов (1 дюйм = 25.4 мм);
7. кубы всех целых чисел из диапазона от  $A$  до  $B$  ( $A \leq B$ ) в обратном порядке;
8. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на цифру  $X$ ;
9. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на цифру  $X$  или  $Y$ ;
10. все целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на любую четную цифру;
11. только положительные целые числа из диапазона от  $A$  до  $B$  ( $A \leq B$ );
12. все целые четные числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ), оканчивающиеся на цифру  $X$  или  $Y$  в обратном порядке;
13. все целые числа из диапазона от  $A$  до  $B$ , кратные трем ( $A \leq B$ );

14. все трехзначные числа, в записи которых две цифры одинаковые;
15. все четные числа из диапазона от  $A$  до  $B$ , кратные пяти ( $A \leq B$ );
16. только отрицательные четные числа из диапазона от  $A$  до  $B$  ( $A \leq B$ ) в обратном порядке;
17. все двухзначные числа, в записи которых все цифры разные;
18. все двухзначные числа, в которых старшая цифра отличается от младшей не больше чем на 1;
19. все трехзначные числа, которые начинаются и заканчиваются на одну и ту же цифру;
20. все трехзначные числа, в которых хотя бы две цифры повторяются.

### **Контрольные вопросы:**

1. Из каких элементов состоят циклические операторы?
2. Для чего используется оператор For?
3. Что представляет собой «тело цикла»?
4. Что такое «итерация»?
5. Что такое «счетчик цикла»?