

Практическая работа №8

Составление программ циклической структуры. Оператор цикла `while`.

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

Оператор цикла `while` предназначен для организации циклического процесса, в котором выполнение каждой следующей итерации определяется на основе истинности некоторого условия. Оператор цикла `while` еще называют оператором цикла с предусловием.

Общая форма оператора цикла `while` следующая:

`while` (*условие*) *оператор*;

здесь

- *условие* – некоторое условие согласно синтаксису языка C#. Инструкция *оператор* выполняется до тех пор, пока значение *условие* = `true`. Как только значение *условие* становится равным `false`, то циклический процесс прекращается и выполняются следующие после `while` операторы;

- *оператор* – один или несколько операторов. Если в цикле `while` нужно выполнить несколько операторов одновременно, то эти операторы берутся в фигурные скобки { }.

Тело цикла (*оператор*) выполняется, пока значение *условие* есть истинным (`true`). Оператор цикла должен быть организован таким образом, чтобы в конечном счете значение условия стало равно `false`. Иначе, программа «зависнет», так как выйдет бесконечный цикл.

Общая форма оператора цикла `while`, в котором выполняются несколько (два и более) операторов имеет следующий вид

```
while (условие)  
{  
    оператор1;  
    оператор2;  
    ...  
    операторN;  
}
```

Понятие вложенных циклов `while`

Цикл `while` может быть вложенным в другие конструкции. Этими конструкциями могут быть:

- циклы `for`, `while`, `do...while`;
- оператор условного перехода `if`;
- оператор выбора `switch`.

Количество уровней вложения, в которых может использоваться цикл `while`, есть неограниченным.

Схема оператора *while*

На рисунке 1 изображена схема оператора *while*. Как видно из схемы, сначала происходит проверка условия. Если значение условия равно **True**, то выполняется тело цикла. Выход из циклического процесса происходит в случае, если значение условия становится равно **False**.

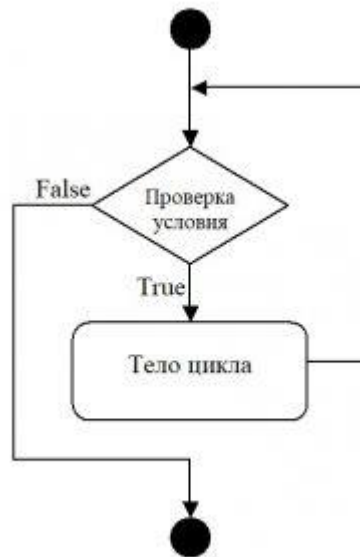


Рисунок 1. Схема оператора *while*

Примеры решения задач с оператором цикла *while*

Вывод результатов вычисления в цикле

Условие задачи. Получить таблицу температур по Цельсию t_c от -50 до +50 градусов а также их эквивалентов по шкале Фаренгейта t_f , используя соотношение

$$t_f = \frac{9}{5}t_c + 32$$

Решение. Текст программы решения данной задачи для приложения типа **Console Application**, с использованием цикла *while*, следующий

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

namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            // шкала температур Цельсий => Фаренгейт
            int tc; // текущее значение температуры по Цельсию
            double tf; // значение температуры по Фаренгейту
```

```

tc = -50;

Console.WriteLine("Шкала температур: Цельсий -
Фаренгейт");

while (tc <= 50)
{
    tf = 9.0 / 5.0 * tc + 32;
    tc++;
    Console.WriteLine("{0} C => {1} F", tc, tf);
}
}
}

```

Пример цикла **while**, вложенным в оператор выбора **switch**. В зависимости от значения переменной **a**, выбирается соответствующий вариант нахождения суммы:

- если **a=1**, то вычисляется **s = 5+10+...+100**;
- если **a=2**, то вычисляется **s = 5+10+...+20**;
- если **a=3**, то вычисляется **s = 5+10**;
- иначе вычисляется **s=100**.

Программный код приложения типа **Console Application**, решающий данную задачу следующий:

```

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

namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            int a;
            int s = 0;
            int i;

            a = 7;
            // выбор алгоритма решения задачи в зависимости от значения a
            switch (a)
            {
                case 1:
                    i = 0;
                    while (i < 100)
                    {
                        i = i + 5;
                        s = s + i;
                    }
                    break;
                case 2:

```

```

        i = 0;
        while (i < 20)
        {
            i = i + 5;
            s = s + i;
        }
        break;
    case 3:
        i = 0;
        while (i < 10)
        {
            i = i + 5;
            s = s + i;
        }
        break;
    default:
        s = 100;
        break;
    }
    Console.WriteLine("s = {0}", s);
}
}
}

```

В результате выполнения вышеприведенного кода, на экран будет выведен результат
s = 100

Выполнить задание: Написать программу вывода на экран следующей информации с использованием цикла while:

1. Выведите последовательность 3 5 7 9 ... 21 (от 3 до 21 с шагом = 2).
2. Выведите последовательность 15 12 9 6 3 0 (от 15 до 0 с шагом = -3).
3. Вычислите умножение 2-значных четных чисел в интервале [10;20] ($10 * 12 * 14 * 16 * 18 * 20$).
4. Вводятся 5 вещественных чисел. Вычислите сложение введенных чисел.
5. Найти среднее арифметическое всех целых чисел от 1 до 1000.
6. Найти среднее арифметическое всех целых чисел от 100 до b (значение b вводится с клавиатуры; $b \geq 100$);
7. Найти среднее арифметическое всех целых чисел от a до 200 (значение a вводится с клавиатуры; $a \leq 200$);
8. Найти среднее арифметическое всех целых чисел от a до b (значения a и b вводятся с клавиатуры; $b \geq a$);
9. Найти сумму квадратов всех целых чисел от a до 50 (значение a вводится с клавиатуры; $0 \leq a \leq 50$);
10. Найти сумму квадратов всех целых чисел от 1 до n (значение n вводится с клавиатуры; $1 \leq n \leq 100$);
11. Найти сумму квадратов всех целых чисел от a до b (значения a и b вводятся с клавиатуры; $b \geq a$);

12. Дано натуральное число n . Напечатать разложение этого числа на простые множители. Реализовать вариант каждый простой множитель должен быть напечатан один раз;

13. Дано натуральное число n . Напечатать разложение этого числа на простые множители. Реализовать вариант каждый простой множитель должен быть напечатан столько раз сколько раз он входит в разложение.

Контрольные вопросы:

1. Что нужно сделать, чтобы циклически выполнялись несколько операторов?
2. Когда проверяется истинность выражения в операторе цикла `repeat`?
3. Верно ли, что истинность выражения в цикле `repeat` является условием окончания цикла?
4. Верно ли, что цикл с параметром применяется в тех случаях, когда заранее известно число повторений?
5. Могут ли внутренний и внешний циклы быть циклами разных видов?
6. Когда проверяется истинность выражения в цикле `while`?