

1 .

Выберите верные утверждения (укажите все верные ответы):

- 1) Любой цикл (while, for, foreach, do-while) можно сделать бесконечным задав условие цикла соответствующим образом.
- 2) do-while работает, как минимум, один раз.
- 3) В for возможно задать условие как через целочисленную, так и через вещественную переменную.
- 4) При одном операторе в цикле необязательно ставить операторные скобки.
- 5) Переменная оператора foreach доступна только для чтения.

2 .

Выберите верные утверждения (укажите все верные ответы):

- 1) В операторе switch каждый case должен быть закрыт в конце break или return.
- 2) В switch обязателен блок оператора, если используется 2 и более операторов.
- 3) В switch обязательно использование default.
- 4) В switch допустимо использование массивов в качестве выражения для включения.
- 5) В switch можно писать пустые блоки case.

3 .

В результате выполнения фрагмента программы:

`using System;`

```
class Program {  
    static void Main() {  
        int x = 10;  
        while ((byte)-x > 246) {  
            Console.Write(x);  
            x -= 100;  
        }  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

4 .

Выберите верные утверждения (укажите все верные ответы):

- 1) При использовании модификатора ref, требуется в методе переменной, помеченной этим модификатором, присвоить значение.
- 2) При использовании модификатора out, переменная, помеченная этим модификатором, до передачи её в метод должна быть без значения.
- 3) Переменная, помеченная ref – модификатором должна быть присвоена до передачи её в метод.
- 4) ref и out – модификаторы, для передачи переменных по ссылке на них, а не по значению.
- 5) С помощью модификаторов ref и out можно передавать по ссылке как значимые, так и ссылочные типы.

5.

Выберите пары методов, имеющие различную сигнатуру (укажите все верные ответы):

- 1) void Meth(ref int a); - void Meth(out int a);
- 2) int Meth(); - public int Meth();
- 3) void Meth(int a); - void Meth(double a);
- 4) int Meth(int a); - int Meth(int a, int b);
- 5) void Meth(int[] a); - void Meth(params int[] a);

6.

В результате выполнения фрагмента программы:

```
using System;
```

```
class Program {  
    static void Main() {  
        int x = 2;  
        switch (x) {  
            case 1:  
                Console.Write(13);  
                goto case 3;  
            case 2:  
                Console.Write(21);  
                goto case 1;  
            case 3:  
                Console.Write(31);  
                goto Label;  
        }  
        Console.Write(500);  
        Label:;  
        Console.Write(100);  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

7.

В результате выполнения фрагмента программы:

```
using System;
```

```
class Program {  
    static void Main() {  
        int x = 20;  
        for (int i = 0; i < 10; i++) {  
            if (x < 10) break;  
            Console.Write(x - 5);  
            x -= 10;  
            if (x < 15) continue;  
        }  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

8.

В результате выполнения фрагмента программы:

```
using System;
```

```
class Program {  
    static void Main() {  
        int x = 1, y = 1;  
        Add(ref x, ref y);  
        Mult(out x, out y);  
        Console.Write(x + " " + y);  
    }  
  
    static void Add(ref int x, ref int y) {  
        x += y;  
        y += x;  
    }  
  
    static void Mult(out int x, out int y) {  
        x = 5 * Math.Max(3, 5);  
        y = 5 * x;  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

9.

В результате выполнения фрагмента программы:

```
using System;
```

```
class Program {  
    static void Main() {  
        Console.WriteLine(Sum(z : 3, x : 2, 0));  
    }  
  
    static int Sum(int x, int y, int z = 10) {  
        return x + y + z;  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

10.

В результате выполнения фрагмента программы:

```
using System;
```

```
class Program {  
    static void Main() {  
        int a = 5;  
        Meth(ref a);  
        Console.WriteLine(a + Meth(a));  
    }  
  
    public static int Meth(int a) {  
        return (Int32)(a / Math.Log10(100));  
    }  
  
    public static void Meth(ref int a) {  
        a *= 3;  
    }  
}
```

на экран будет выведено:

Примечание:

*Если возникнет ошибка компиляции, введите: ****

Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---

Если возникнет ошибка исполнения или исключение, введите: +++

1	12345
2	45
3	---
4	345
5	34
6	211331100
7	155
8	25125
9	***
10	22