

1.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
delegate void Del();
```

```
class Program {
```

```
    static void Meth1() => Console.Write(1);
```

```
    static void Meth2() => Console.Write(2);
```

```
    static void Meth3() => Console.Write(3);
```

```
    static void Main() {
```

```
        Del d = Meth2;
```

```
        Del d2 = Meth3;
```

```
        d += Meth1;
```

```
        d += Meth2;
```

```
        Del d3 = (Del)Delegate.Combine(d, d2);
```

```
        d3 -= Meth2;
```

```
        d3();
```

```
    }
```

```
}
```

на экран будет выведено:

2.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
delegate void Del();
```

```
class Program {
```

```
    static void Main() {
```

```
        int x = 5;
```

```
        Del d = delegate {
```

```
            int y = 10;
```

```
            for (int i = 0; i < 3; i++) {
```

```
                Console.Write(x * i + y);
```

```
            }
```

```
        };
```

```
        d();
```

```
    }
```

```
}
```

на экран будет выведено:

3.

**Верно, что в сигнатуру делегата входит (укажите все верные ответы):**

- 1) Тип возвращаемого значения.
- 2) Типы входных параметров.
- 3) Имена входных параметров.
- 4) Модификаторы ref, out, params.
- 5) Модификатор доступа.

4.

<b>Делегат допускает следующие модификаторы (укажите все верные ответы):</b>
1) public. 2) static. 3) abstract. 4) void. 5) private.

5.

<b>Укажите строки кода, вставка которых вместо пропуска в программу:</b> <b>using System;</b>  <b>delegate void Del(ref int x, out int y, params int[] args);</b> <b>class Program {</b> <b>static void Main() {</b> <b>int x = 5;</b> <b>int y;</b> <b>&lt; - - -    - - - &gt;</b> <b>}</b> <b>}</b> <b>НЕ приведет к ошибке компиляции</b>
1) <b>Del d = (ref int z, out int t, int[] args) =&gt; {t = 5;};</b> 2) <b>Del d = null;</b> 3) <b>Del d = delegate (ref int x, out int y, params int[] args) { };</b> 4) <b>Del d = delegate { };</b> 5) <b>Del d = new Del();</b>

6.

<b>Допустимы следующие лямбда-выражения (укажите все верные ответы):</b>
1) <b>() =&gt; { return; };</b> 2) <b>(int x) =&gt; x + x;</b> 3) <b>int x =&gt; x + x;</b> 4) <b>(int x) =&gt; return x;</b> 5) <b>x =&gt; 2 * x;</b>

7.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
delegate void Del(ref int x, params int[] args);  
class Program {  
    static void Main() {  
        int x = 5;  
        Del d = delegate (ref int y, params int[] args) {  
            for (int i = 0; i < args.Length; i++) {  
                Console.Write(args[i] + x * i);  
            }  
        };  
        d(ref x, 1, 4, 10);  
    }  
}
```

**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;
```

```
delegate int Del(params int[] args);  
class Program {  
    static void Main() {  
        Del d = (int[] x) => {  
            int res = 0;  
            for (int i = 0; i < x.Length; i++) {  
                res += x[i] - i;  
            }  
            return res;  
        };  
        d += (int[] args) => args.Length - 5;  
        d -= (int[] args) => args.Length - 5;  
        Console.Write(d(1, 3, 5, 6, 8, 10));  
    }  
}
```

**на экран будет выведено:**

9.

**В результате выполнения фрагмента программы:**

```
using System;

class Program {
    delegate void Message();

    static void Main(string[] args) {
        Message mes;
        mes = GoodMorning;
        mes += GoodEvening;
        Hi(mes);
    }
    private static void GoodMorning() {
        Console.WriteLine("2");
    }
    private static void GoodEvening() {
        Console.WriteLine("1");
    }
    private static void Hi(Message mes) {
        for (int i = 0; i < 3; i++) {
            mes?.Invoke();
        }
    }
}
```

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;

delegate string Sum(int number);
class Program {
    static Sum SomeVar() {
        string s = "";
        Sum del = delegate (int number) {
            for (int i = 0; i < number; i++)
                s += i.ToString();
            return s;
        };
        return del;
    }
    static void Main() {
        Sum del1 = SomeVar();
        for (int i = 0; i <= 3; i++) {
            Console.WriteLine(del1(i));
        }
    }
}
```

на экран будет выведено:

1	213
2	101520
3	124
4	145
5	12
6	125
7	***
8	1
9	212121
10	0001001012