

1.

**Выберите номера строк, которые приведут к выводу в консоль числа 5:**  
**using System;**

```
class MyClass {  
    private int x = 4;  
    protected int y = 4;  
    internal int z = 5;  
  
    public int ShowX() {  
        return x++;  
    }  
}  
  
class Program {  
    static void Main() {  
        MyClass obj = new MyClass();  
        Console.WriteLine(obj.x + 1);           //1  
        Console.WriteLine(++obj.y);             //2  
        Console.WriteLine(obj.z);               //3  
        Console.WriteLine(obj.ShowX());         //4  
        Console.WriteLine(new MyClass().z);     //5  
    }  
}
```

- 1) 1
- 2) 2
- 3) 3
- 4) 4
- 5) 5

2.

**Выберите допустимые модификаторы для конструктора (укажите все верные ответы):**

- 1) static
- 2) extern
- 3) public
- 4) private
- 5) abstract

3.

**В результате выполнения фрагмента программы:**

```
using System;

class A {
    public A(byte arg) {
        Console.Write(arg * 2);
    }

    public A(uint arg) {
        Console.Write(arg * 2);
    }

    public A(double arg) {
        Console.Write((int)arg);
    }

    public A(string arg) {
        Console.Write(arg.Substring(3, 3));
    }

    public A(int arg1, int arg2, string format) {
        Console.Write((arg1 + arg2).ToString() +
format.Replace('a', 'b'));
    }
}

class Program {
    static void Main() {
        A obj;
        obj = new A(150);
        obj = new A("313312");
        obj = new A(3, 5, "format");
    }
}
```

**на экран будет выведено:**

4.

**Верно, что статический конструктор (укажите все верные ответы):**

- 1) Может вызваться только один раз в течение работы всей программы.
- 2) Вызывается всегда при обращении к статическому члену класса с данным конструктором.
- 3) Может быть только в статическом классе.
- 4) Может принимать аргументы.
- 5) Не может иметь модификаторов доступа.

5.

Укажите строки кода, вставка которых вместо пропуска в программу:

```
class Class {  
    public int x;  
    int y;  
    private int z;  
    public Class(int x, int y = 5) {  
        y = 5;  
    }  
}
```

```
class Program {  
    static void Main() {  
        <- - - || - - ->  
    }  
}
```

НЕ приведет к ошибке компиляции

- 1) Class obj = new Class { x = 5 };
- 2) Class obj = new Class(10) { z = 5 };
- 3) Class obj = new Class(x: 4) { x = 5 };
- 4) Class obj = new Class(y: 10, 20);
- 5) Class obj = new Class(x: 5, 10);

6.

В результате выполнения фрагмента программы:

```
using System;  
  
class Class {  
    public static readonly int x = 5;  
    public Class(int x) {  
        x = 10;  
    }  
}  
  
class Program {  
    static void Main() {  
        Class obj = new Class(20);  
        Console.WriteLine(Class.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: +++*

7.

**В результате выполнения фрагмента программы:**

```
using System;

class Class {
    public int x;
    public int y;
    public int z;
    public Class() : this(10) {
        x = y = 10;
    }

    public Class(int x) : this(5, 10, 15) {
        x = z = x;
    }

    public Class(int x, int y) {
        y = z = 2 * x;
    }

    public Class(int x, int y, int z) : this(1, 2) {
        x = y = z = x = y;
    }
}

class Program {
    static void Main() {
        Class obj = new Class();
        Console.Write(obj.x + obj.y + obj.z);
    }
}
```

**на экран будет выведено:**

8.

**Свойство, в отличие от индексатора (укажите все верные ответы):**

- 1) Может иметь имя.
- 2) Имеет тип возвращаемого значения.
- 3) Может быть перегружено.
- 4) Имеет аксессоры get и set.
- 5) Может быть инициализированным в объявлении.

9.

**Выберите верные утверждения (укажите все верные ответы):**

- 1) Модификатором partial могут быть помечены свойства.
- 2) Метод, помеченный модификатором partial, не может иметь модификаторы доступа.
- 3) partial-методы могут быть объявлены только в partial-классе или partial-структуре.
- 4) Методы, помеченные модификатором partial, могут возвращать только void-значение.
- 5) Методы, помеченные модификатором partial, могут не иметь тела метода.

10.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class Class {  
    public static readonly int x;  
    public Class(int x) {  
        Class.x = x * 3;  
    }  
}
```

```
class Program {  
    static void Main() {  
        Class obj = new Class(10);  
        Console.Write(Class.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: +++*

1	35
2	1234
3	3003128formbt
4	125
5	3
6	5
7	30
8	5
9	2345
10	***