

1.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class A {  
    public int x = 10;  
}
```

```
class B : A {  
    public int x = 5;  
}
```

```
class Program {  
    static void Main() {  
        A a = new B();  
        Console.WriteLine(a.x);  
    }  
}
```

**на экран будет выведено:**

2.

**Укажите строки кода, которые вызовут ошибки компиляции или исключение:**

```
class A { }
```

```
class B : A { }
```

```
class C : A { }
```

```
class D : C{ }
```

```
class E : B{ }
```

```
class Program {  
    static void Main() {  
        A a = new B(); //1  
        C c = new E(); //2  
        A e = (A)new E(); //3  
        C d = (A)new D(); //4  
        B b = (B)new A(); //5  
    }  
}
```

1) 1

2) 2

3) 3

4) 4

5) 5

3.

Укажите строки кода, раскомментирование которых:

```
using System;
```

```
class A {  
    protected int x {  
        get => 5;  
    }  
    public int y = 6;  
}
```

```
class B : A {  
    public int x = 6;  
    internal int y = 4;  
    int d = 5;  
}
```

```
class Program {  
    static void Main() {  
        //Console.Write(new B().x);    //1  
        //Console.Write(new A().x);    //2  
        //Console.Write(new A().y--);  //3  
        //Console.Write(++new B().y);  //4  
        //Console.Write(new B().d);    //5  
    }  
}
```

приведёт к выводу в консоль числа 5

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

4.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class C1 {  
    protected void Print() {  
        Console.Write("1");  
    }  
}
```

```
class C2 : C1 {  
    new protected void Print() {  
        base.Print();  
        Console.Write("2");  
        base.Print();  
    }  
}
```

```
class C3 : C2 {  
    new public void Print() {  
        base.Print();  
        Console.Write("3");  
        base.Print();  
    }  
}
```

```
class Program {  
    static void Main() {  
        C3 ob3 = new C3();  
        ob3.Print();  
    }  
}
```

**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: +++*

5.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class A {  
    public A() {  
        Console.Write("A");  
    }  
    static A() {  
        Console.Write("SA");  
    }  
}  
class B : A {  
    public B() {  
        Console.Write("B");  
    }  
    static B() {  
        Console.Write("SB");  
    }  
}  
class Program {  
    static void Main() {  
        B b = new B();  
    }  
}
```

**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: +++*

6.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System; using System.IO;  class Program {     static void Main() {         BinaryWriter bw = new BinaryWriter(new FileStream("out.txt", FileMode.Create));         using (bw) {             for (int i = 0; i &lt; 10; i++) {                 bw.Write(i);                 bw.Seek(3, SeekOrigin.Begin);             }         }         StreamReader sr = new StreamReader(new FileStream("out.txt", FileMode.Open));         using (sr) {             try {                 while (!sr.EndOfStream) {                     Console.Write(sr.Read());                 }             }             catch {                 Console.Write(1);             }         }     } }</pre> <p>на экран будет выведено:</p>

7.

<b>Выберите верные утверждения (укажите все верные ответы):</b>
<ol style="list-style-type: none"><li>1) DateTime – статический класс.</li><li>2) DateTime имеет конструктор, принимающий все величины времени от миллисекунд до годов. (миллисекунды, секунды, минуты, часы, дни, месяцы, годы)</li><li>3) Объекты DateTime можно сравнивать между собой любым образом.</li><li>4) DateTime имеет статическое свойство Now, возвращающее текущее время.</li><li>5) DateTime имеет свойства MinValue и MaxValue.</li></ol>

8.

**В результате выполнения фрагмента программы:**

```
using System;

struct MyStruct {
    public const int X = 15;
    public static int x = 5;

    public MyStruct(int xt) {
        x = xt;
    }
}

class Program {
    static void Main() {
        MyStruct obj = new MyStruct(10);
        Console.Write(MyStruct.X ^ 5 * MyStruct.x + 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: +++*

9.

**В результате выполнения фрагмента программы:**

```
using System;

struct MyStruct {
    public int X;
    public int x {
        get => X;
        set => X = value;
    }

    public MyStruct(int xt, int y) {
        x = xt;
        X = xt;
    }
}

class Program {
    static void Main() {
        MyStruct obj = new MyStruct(10, 20);
        Console.Write(obj.x + obj.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: +++*

10.

<b>Выберите все верные утверждения:</b>
---

- |   |
|---|
| <ul style="list-style-type: none"><li>1) При использовании потока внутри блока <b>using</b> можно не использовать методы <b>flush()</b> и <b>close()</b> для корректного завершения работы с данным потоком.</li><li>2) <b>FileMode.Create</b> генерирует исключение, если в указанном каталоге существует одноимённый файл.</li><li>3) Юникод – 16 битная кодировка.</li><li>4) Метод <b>Seek()</b> класса <b>StreamWriter</b> осуществляет перемещение потока на указанную позицию.</li><li>5) Все классы работы с файлами унаследованы от абстрактного класса <b>MarshalByRefObject</b>.</li></ul> |
|---|

1	10
2	245
3	4
4	1213121
5	SBSAAB
6	0009000
7	2345
8	58
9	***
10	135