

1.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {
    static void Main() {
        double a = 0.5;
        Console.Write(Math.Round(Math.Pow(a,2) * 10));
    }
}
```

**на экран будет выведено:**

2.

**Укажите корректные заголовки точки входа программы в C# (укажите все верные ответы):**

- 1) `static public int Main( );`
- 2) `private static void Main(string[] favouriteAnime);`
- 3) `static int main( );`
- 4) `static void Main(string[] args, int bufferSize);`
- 5) `static void Main(string args);`

3.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {
    static void Main() {
        Console.Write(TypeDependentAction((Int16)7));
    }

    static dynamic TypeDependentAction(dynamic d) {
        return d is int ? d : d + d;
    }
}
```

**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.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class Program {     static void Main() {         sbyte a = 94;         sbyte b = 34;         checked {             Console.Write((sbyte)(a + b) - 1);         }     } }</pre> <p><b>на экран будет выведено:</b></p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

5.

<b>Верно, что статический класс (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Может наследоваться от какого-то класса;</li><li>2) Может содержать деструктор;</li><li>3) Может содержать конструктор;</li><li>4) Содержит только статические члены;</li><li>5) Может быть расширен с помощью методов расширений;</li></ul>

6.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class Program {     static void Main() {         int x = 10;         ref int y = ref x;         Console.Write(RefMeth(ref x, ref y) + y);     }      static int RefMeth(ref int a, ref int b) {         a += b;         return a == b ? b-- : --b;     } }</pre> <p><b>на экран будет выведено:</b></p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

7.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {  
    static void Main() {  
        for (int i = 0; i < 5; i++) {  
            Console.Write(i < 2 ? i + 2 : i);  
            if (i < 4) { continue; }  
            Console.Write(~i);  
        }  
    }  
}
```

**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() {  
        float a = 4f;  
        double b = 262.567;  
        decimal c = 7.567m;  
        Console.Write(Math.Pow(a, Math.Round((double)c - (byte)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: +++*

9.

**Укажите номера строк кода, вставка которых в программу вместо пропуска:**

```
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() {  
        < - - - || - - - >  
    }  
}
```

приведёт к выводу в консоль числа 5:

- 1) Console.Write(new B().x);
- 2) Console.Write(new A().x);
- 3) Console.Write(~new A().y);
- 4) Console.Write(++new B().y);
- 5) Console.Write(new B().d);

10.

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

- 1) DateTime – статический класс;
- 2) DateTime имеет конструктор, принимающий все величины времени от годов до миллисекунд (годы, месяцы, дни, часы, минуты, секунды, миллисекунды);
- 3) Объекты DateTime можно сравнивать между собой любым образом;
- 4) DateTime имеет нестатическое свойство Now, возвращающее текущее время;
- 5) DateTime имеет свойства MinValue и MaxValue;

11.

**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:**

12.

В результате выполнения фрагмента программы:
<pre>using System;  class A {     public virtual void Print() {         Console.Write("A");     } }  class B : A {     public override void Print() {         Console.Write("B");         base.Print();     } }  class C : B {     new public void Print() {         Console.Write("C");         base.Print();     } }  class Program {     static void Main() {         A a = new B();         ((C)a).Print();     } }</pre> <p>на экран будет выведено:</p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

13.

В результате выполнения фрагмента программы:
<pre>using System;  class Program {     static void Main() {         Console.WriteLine(Sum(z : 3, 2, x: 0));     }      static int Sum(int x, int y, int z = 10) {         return x + y + z;     } }</pre> <p>на экран будет выведено:</p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

14.

**В результате выполнения фрагмента программы:**

**using System;**

```
class Program {
    static void Main() {
        Exception[] exceptions = { new NullReferenceException(),
                                    new ArgumentNullException(), new ArgumentException() };
        try {
            for (int i = 0; i < exceptions.Length; i++)
                if (exceptions[i] is ArgumentException)
                    throw exceptions[i];
        }
        catch (NullReferenceException ex) when
            (ex is ArgumentException) {
            Console.Write(0);
        }
        catch(ArgumentException ex) when(ex.Message.Contains(" ")) {
            Console.Write(1);
        }
        catch (ArgumentNullException) { Console.Write(2); }
    }
}
```

**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: +++*

15.

**Укажите номера строк кода, вставка которых в программу вместо пропуска:**

**using System;**

```
class Program {
    static void Main() {
        string line = "string";
        < - - - || - - - >
    }
}
```

**приведёт к выводу в консоль числа 4:**

- 1) `Console.Write(line.Substring(0, 3).Length);`
- 2) `Console.Write(line.IndexOf('g'));`
- 3) `Console.Write(line.Substring(2).Length);`
- 4) `Console.Write(line.Replace("s", "").Length);`
- 5) `Console.Write(line.ToCharArray().Length);`

16.

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

- 1) На отлов исключений можно наложить условия их перехвата с помощью ключевого слова where;
- 2) Оператор throw нельзя использовать без аргумента-исключения;
- 3) Оператор throw можно использовать с типами значений;
- 4) Все исключения унаследованы от класса Exception;
- 5) В классе Exception – 4 конструктора, среди которых есть беспараметрический;

17.

**В результате выполнения фрагмента программы:**

**using System;**

```
class A {  
    protected virtual void Meth() {  
        Console.Write(B.x * 2);  
        new B().Meth();  
    }  
}
```

```
class B : A {  
    static internal int x = 0;  
    protected override void Meth() {  
        if (x++ == 3) return;  
        Console.Write(x * 3);  
        base.Meth();  
    }  
}
```

```
class C : B {  
    static void Main() {  
        new C().Meth();  
    }  
}
```

**на экран будет выведено:**

18.

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

- 1) При использовании потока внутри блока using можно не использовать методы flush() и close() для корректного завершения работы с данным потоком;
- 2) FileMode.Create генерирует исключение, если в указанном каталоге существует одноимённый файл;
- 3) Юникод – 16 битная кодировка;
- 4) Метод Seek() класса StreamWriter осуществляет перемещение потока на указанную позицию;
- 5) Все классы работы с файлами унаследованы от абстрактного класса MarshalByRefObject;



19.

<b>Выберите верные утверждения (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Оператор объединения с null является право-ориентированным (выполняется справа налево);</li><li>2) Оператор * одновременно является унарным и мультипликативным;</li><li>3) Тернарный оператор имеет более высокий приоритет, чем оператор присваивания =;</li><li>4) Запрещается одновременно использовать префиксный и постфиксный декремент/инкремент;</li><li>5) Оператор сравнение с последующим приравниванием имеет наименьший приоритет;</li></ul>

20.

<b>Укажите корректные объявления индексатора (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) public ref int this[ref int a] { get =&gt; ref a; }</li><li>2) private int this[int x, int y] { set { x = y; } }</li><li>3) protected internal dynamic this[dynamic a] { get { return x + 5; } }</li><li>4) public double @this { get =&gt; @this + 5.5; }</li><li>5) public void this[int z] { set { z = value * 50; } }</li></ul>

21.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class Program {     static void Main() {         int[][] args = new int[2][];         args[0] = new int[2];         args[1] = new int[3];         for (int i = 0; i &lt; args.Length; i++) {             for (int j = 0; j &lt; args[i].Length; j++) {                 args[i][j] = (i + j) * 2;                 Console.Write(args[i][j]);             }         }     } }</pre> <p>на экран будет выведено:</p> <p>Примечание:</p> <p>Если возникнет ошибка компиляции, введите: ***</p> <p>Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---</p> <p>Если возникнет ошибка исполнения или исключение, введите: +++</p>

22.

<b>Выберите верные утверждения (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Автоматически реализуемое свойство может иметь только аксессор set;</li><li>2) Свойства могут быть виртуальными, но не могут иметь перегрузок в классе;</li><li>3) Возможно объявление protected свойства с internal get;</li><li>4) Аксессоры свойств могут быть вызваны явно;</li><li>5) Свойства могут быть статическими;</li></ul>

23.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {  
    static void Main() {  
        string line = "-asda-ds-";  
        Console.Write(line.Trim('-'));  
    }  
}
```

**на экран будет выведено:**

24.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {  
    static void Main() {  
        int[] args = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };  
        int[][] arr = new int[5][];  
        int s = 0;  
        for (int i = 0; i < arr.Length; i++) {  
            arr[i] = args;  
            for (int j = 0; j < arr.Length; j++) {  
                s += arr[i][j];  
            }  
            Array.Reverse(arr[i]);  
        }  
        Console.Write(s);  
    }  
}
```

**на экран будет выведено:**

25.

**В результате выполнения фрагмента программы:**

`using System;`

```
class Program {  
    static void Main() {  
        string row = "0123456789";  
        row = new string(row.ToCharArray(), 4, 3);  
        Console.WriteLine(row);  
    }  
}
```

**на экран будет выведено:**

26.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class A {     public int x;     public A(int x) {         this.x = x;     } }  class Program {     static void Main() {         A a = new A() { x = 10 };         a.x++;         Console.Write(a.x);     } }</pre> <p><b>на экран будет выведено:</b></p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

27.

<b>Верно, что деструкторы в языке C# (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Могут содержаться в статическом классе;</li><li>2) Могут быть перегружены;</li><li>3) Не могут быть явно вызваны;</li><li>4) Могут содержать модификаторы;</li><li>5) Могут принимать консольные параметры;</li></ul>

28.

<b>Выберите верные утверждения (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Модификатор <code>out</code> указывается как при объявлении метода, так и при его вызове;</li><li>2) Возможно объявление поля с модификатором <code>ref</code>;</li><li>3) Метод с типом возвращаемого значения, помеченным модификатором <code>ref</code> не может сохранить значение в переменную без данного модификатора;</li><li>4) <code>ref local</code> переменные могут быть инициализированы константами;</li><li>5) Переменной с модификатором <code>out</code> в теле метода обязательно должно быть присвоено значение до выхода из метода;</li></ul>

29.

<b>Выберите верные утверждения (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Методы расширений могут быть объявлены только в абстрактном классе;</li><li>2) Методы расширений могут дополнять только нестатические классы;</li><li>3) Для создания методов расширений необходимо передать в качестве параметра имя расширяемого метода с ключевым словом <code>this</code>, причём этот параметр должен стоять последним в списке параметров.</li><li>4) Методы расширений могут быть виртуальными;</li><li>5) Методы расширений не могут быть приватными;</li></ul>

30.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class A {     public A() {         Console.Write("Pen");     }     static A() {         Console.Write("Pineapple");     } }  class B : A {     public B() {         Console.Write("Apple");     }     static B() {         Console.Write("Goose");     } }  class Program {     static void Main() {         A a = new A();         B b = new B();     } }</pre> <b>на экран будет выведено:</b>

31.

<b>Комментарии верно использованы в следующих строках (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) /* Console.WriteLine(); */</li><li>2) Console.Write("//Comment");</li><li>3) /Text Comment</li><li>4) /* Comment</li><li>5) this */</li></ul>

32.

<b>Модификатором override может быть помечен(о) (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) Метод;</li><li>2) Свойство;</li><li>3) Индексатор;</li><li>4) Переменная;</li><li>5) Деструктор;</li></ul>

**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:**

34.

**В результате выполнения фрагмента программы:**

```
using System;

class Program {
    static void Main() {
        try {
            Console.Write("Try");
            Print();
        }
        catch (StackOverflowException) {
            Console.Write("Catch");
        }
        finally {
            Console.Write("Finally");
        }
    }

    public static void Print() {
        Console.Write("Exception");
        Print();
    }
}
```

35.

**В результате выполнения фрагмента программы:**

```
using System;

class Program {
    static void Main() {
        int turns = 10;
        Meth(turns);
    }

    static string Meth(int c)
    {
        return c < 3 ? (c % 2).ToString() : Meth(c / 2 + 1) + "0";
    }
}
```

**на экран будет выведено:**

36.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class Program {     static void Main() {         for(int i = 5; i &gt;= 0; i--) {             if(!(bool)i)                 Console.Write(100);             Console.Write(i);         }     } }</pre> <p>на экран будет выведено:</p> <p><i>Примечание:</i> Если возникнет ошибка компиляции, введите: *** Если ошибок и исключений нет, но на экран не выведется ничего, введите: --- Если возникнет ошибка исполнения или исключение, введите: +++</p>

37.

<b>Выберите пары методов, имеющие различную сигнатуру (укажите все верные ответы):</b>
<ul style="list-style-type: none"><li>1) void Meth(ref int a); - void Meth(out int a);</li><li>2) int Meth(); - public int Meth();</li><li>3) void Meth(int a); - void Meth(double a);</li><li>4) int Meth(int a); - int Meth(int a, int b);</li><li>5) void Meth(int[] a); - void Meth(params int[] a);</li></ul>

38.

<b>В результате выполнения фрагмента программы:</b>
<pre>using System;  class Program {     static int x = 8;     static int y = 2;      static void Main() {         Console.Write(Meth(x--, ++y));     }      static int Meth(int a, int b) {         if (x++ == ++b + x % y + 1) return b;         return a;     } }</pre> <p>на экран будет выведено:</p>

39.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class Program {  
    static void Main() {  
        int x;  
        ref int y = x;  
        for(int i = 0; i < y; i++)  
            Console.Write(x + y);  
    }  
}
```

**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: +++*

40.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
class Program {  
    static void Main() {  
        string s = "12345";  
        int arg;  
        try {  
            int.Parse(string.Concat(s[0] + s[1]));  
        }  
        catch (FormatException) {  
            Console.Write("Format" + s);  
        }  
        Console.Write("String" + s);  
    }  
}
```

**на экран будет выведено:**



1	2
2	12
3	14
4	+++
5	134
6	39
7	232345
8	16
9	34
10	235
11	1213121
12	+++
13	***
14	1
15	3
16	45
17	326496
18	135
19	1234
20	23
21	02246
22	25
23	asda-ds
24	100
25	456
26	***
27	3
28	15
29	2
30	PineapplePenGoosePenApple
31	1
32	123
33	3003128formbt
34	+++
35	00000
36	***
37	34
38	4
39	***
40	String12345

