

1. Как может выглядеть заголовок метода Main?

1. `public static int Main (string[] args)`
2. `public static int Main ()`
3. `public void Main (string[] args)`
4. `static int Main (string[] args)`
5. `static void Main ()`
6. `public int Main (string[] args)`
7. `public static int Main (int n, string[] args)`

2. Пусть описаны структура и класс:

```
struct S { int x,y; }  
class C { int x,y; }
```

Какой вариант использования их выдаст синтаксическую ошибку?

1.

```
class Class1  
{  
    static void Main()  
    {  
        C c = new C();  
        S s = new S();  
    }  
}
```

2.

```
class Class1  
{  
    static void Main()  
    {  
        C c;  
        S s = new S();  
    }  
}
```

3.

```
class Class1  
{  
    static void Main()  
    {  
        C c;  
        S s;  
    }  
}
```

4.

```
class Class1  
{  
    static void Main()  
    {  
        C c = new C();  
        S s;  
    }  
}
```

3. Чему будет равняться значение переменной b?
Считаем, что класс C описан.

```
ObjTest c1 = new C();
```

```
ObjTest c2 = c1;
```

```
object o = c2;
```

```
bool b = o.Equals(c1);
```

1. True
2. False
3. Синтаксическая ошибка
4. Ошибка времени выполнения

4. Отметьте только совместимые с CLS типы данных C#

1. sbyte
2. byte
3. short
4. ushort
5. int
6. uint
7. long
8. ulong

5. Какие описание класса правильные?

1.

```
class Class1
{ int n = 1;
  static void Main()
  {   int n;
      Console.WriteLine(n);
  }
}
```

2.

```
class Class1
{ int n = 1;
  static void Main()
  {   int n = 2;
      Console.WriteLine(n);
  }
}
```

3.

```
class Class1
{ int n = 1;
  static void Main()
  {   int k = 2;
      Console.WriteLine(n);
  }
}
```

4.

```
class Class1
{ int n = 1;
  static void Main()
  {   int k;
      Console.WriteLine(n);
  }
}
```

6. Что выведет на экран данная программа?

```
class Class1
{
    static void Main()
    {
        for (int i=0; i<10; i++);
        Console.WriteLine(i);
    }
}
```

1. Ничего не выведет – синтаксическая ошибка
2. Ничего не выведет – ошибка времени выполнения
3. 0
4. 9
5. 10
6. Числа от 0 до 9
7. Числа от 0 до 10

7. Что выведет на экран данная программа?

```
class C
{ public static double f(double x) { return x*x; }
  public static float f(float x) { return 0.5f*x; }
}
```

```
class Class1
{ static void Main()
  { Console.WriteLine(C.f(2.0));
  }
}
```

1. Ничего не выведет – синтаксическая ошибка
2. Ничего не выведет – ошибка времени выполнения
3. 4.0
4. 1.0
5. C.f

8. Что выведет на экран данная программа?

```
class C { public static void p(ref int x) { x = -x; }
        }

class Class1 {    static void Main()
                {    int n = 1;
                    C.p(n);
                    Console.WriteLine(n);
                }
            }
```

1. Синтаксическая ошибка в описании функции **p** класса **C**
2. Синтаксическая ошибка при вызове функции
3. Ошибка времени выполнения
4. 1
5. -1

9. Что выведет на экран данная программа?

```
class Class1
{   public static void Add(int x, int y, out int ans)
    { ans = x + y; }

    static void Main()
    { int x;
      Add(2,3,out x);
      Console.WriteLine(x);
    }
}
```

1. Синтаксическая ошибка в описании функции **Add**
2. Синтаксическая ошибка при вызове функции **Add**
3. Неинициализированная переменная x в Main
4. 5
5. Ошибка времени выполнения

10. Что выведет на экран данная программа?

```
class Class1 {    public static void p(params object [ ] L)
                {
                    foreach (object o in L)
                        Console.WriteLine(o);
                }

                static void Main()
                { p(1,"abc");
                }
            }
```

1. Синтаксическая ошибка в описании функции **p**
2. Синтаксическая ошибка при вызове функции **p**
3. 1
abc
4. System.Int32
System.String
5. System.object
System.object

11. Какие из описаний массива являются недопустимыми?

1. `string[] z = new string[100];`
2. `string[] z;`
`z = new string[10];`
3. `string[] z = { "abc", "abcd", "abcde" };`
4. `string[3] z = { "abc", "abcd", "abcde" };`
5. `string[] z = string[3] { "abc", "abcd", "abcde" };`

12. Что выведет на экран данная программа?

```
class Class1 { static void Main()
    {
        string s = "abc";
        string t = s.ToUpper();

        Console.WriteLine(s);
        Console.WriteLine(t);
    }
}
```

1. В программе есть синтаксические ошибки
2. abc
abc
3. abc
ABC
4. ABC
ABC
5. ABC
abc

13. Что выведет на экран данная программа?

```
class A { public A() { Console.Write("0"); }  
        public A(int n) : this()  
        { Console.Write("1");  
        }  
    }  
  
class B { static void Main()  
        { A a = new A(1);  
        }  
    }
```

1. Синтаксическая ошибка в описании одного из конструкторов
2. Синтаксическая ошибка в описании объекта **a**
3. 0
4. 1
5. 01
6. 10

14. Что выведет на экран данная программа?

```
class A { public A() { Console.Write("0"); }  
        public A(int n) : this()  
        { Console.Write("1");  
        }  
    }  
  
class B : A { public B(int n) { Console.Write("2"); }  
    }  
  
class C { static void Main()  
        {  
            B b = new B(1);  
        }  
    }
```

1. 0
2. 1
3. 2
4. 02
5. 012
6. 20
7. 210

15. Что выведет на экран данная программа?

```
class A { public A() { Console.Write("0"); }  
        public A(int n) : this()  
        { Console.Write("1");  
        }  
    }  
  
class B : A { public B(int n) : base(n) { Console.Write("2"); }  
            }  
  
class C { static void Main()  
        {  
            B b = new B(1);  
        }  
    }
```

1. 0
2. 1
3. 2
4. 02
5. 012
6. 20
7. 210

16. Что выведет на экран данная программа?

```
class A
{
    public static int n;
    static A() { Console.Write("A");
                n=1;
            }
}

class Class1
{
    static void Main() { Console.WriteLine(A.n);
    }
}
```

1. 0
2. 1
3. A0
4. A1
5. Ошибка компиляции
6. Ошибка времени выполнения

17. В каких местах программы можно задать значение обычных полей «только для чтения»?

1. Инициализация непосредственно при описании
2. В `private` методе класса
3. В статическом конструкторе
4. В обычном конструкторе
5. В статическом методе класса
6. В любом методе класса

18. Пусть имеется следующее описание

```
interface IA { void p(); }  
class C : IA { public void p() { Console.WriteLine("P"); } }
```

Какие способы вызова функции p() являются допустимыми?

1. C c = new C(); c.p();
2. C c = new C(); (IA) c.p();
3. C c = new C(); ((IA) c).p();
4. C c = new C(); c. (IA) p();

19. Пусть имеется класс A, который **не** поддерживает интерфейс IB. Что произойдет в следующем коде?

```
A a = new A();
```

```
IB b = a as IB;
```

1. Синтаксическая ошибка во второй строке
2. Произойдет исключение при выполнении программы
3. Переменная b будет иметь значение null
4. Объект b будет иметь все свойства класса A и поддерживать интерфейс IB
5. Объект b будет ссылкой на объект a

20. Пусть в классе A есть массив строк s.
Как правильно организовать индексатор в классе A,
возвращающий эти строки?

1. `public string this[int i] { get { return s[i];} }`
2. `public string A[int i] { get { return s[i]; } }`
3. `public string this[int i] { return s[i]; }`
4. `public string A[int i] { return s[i]; }`

21. Как правильно должен выглядеть заголовок перегруженного оператора + класса A?

1. `A operator + (A a1, A a2) { ... }`
2. `public A operator + (A a1, A a2) { ... }`
3. `static A operator + (A a1, A a2) { ... }`
4. `public static A operator + (A a1, A a2) { ... }`

22. Пусть определены два класса A и C следующим образом

```
class A { public delegate void D();  
    public void Q(D x) { x(); }  
}
```

```
class C { public static void W() { Console.WriteLine("W"); }  
}
```

А в функции Main создан объект A a = new A();
Какой вызов метода Q допустим?

1. A.Q(new D(C.W));
2. a.Q(new D(C.W));
3. A.Q(new A.D(C.W));
4. a.Q(new A.D(C.W));
5. a.Q(C.W);
6. a.Q((A.D)(C.W));
7. a.Q(new C.W);

23. Пусть определены два класса A и C следующим образом

```
class A { public delegate void D();  
    public void Q(D x) { x(); }  
}
```

```
class C { public static void W() { Console.WriteLine("W"); }  
}
```

А в функции Main создан объект A a = new A();
Какой вызов метода Q допустим?

1. A.D d = new A.D(C.W);
a.Q(d);

4. D d = D(C.W);
a.Q(d);

2. A.D d = A.D(C.W);
a.Q(d);

5. A.D d = new A.D(C.W);
a.Q(new d);

3. A.D d = A.D(C.W);
a.Q(new d);

6. A.D d = new C.W;
a.Q(d);

24. Пусть определен класс A

```
class A { public delegate void E(string s);  
        public static event E Evn;  
}
```

Мы хотим добавить в него метод p, который бы вызывал событие Evn. Какой вариант метода правильный?

1. `public void p(bool b) { if (b) E("Buh!"); }`
2. `public void p(bool b) { if (b) Evn("Buh!"); }`
3. `public void p(bool b) { if (b) new Evn("Buh!"); }`
4. `public void p(bool b) { if (b) new E("Buh!"); }`

25. Пусть определен класс A и класс C

```
class A { public delegate void E(string s);  
        public static event E Evn;  
        public void p(bool b) { if (b) Evn("Buh!"); }  
    }
```

```
class C { public static void S(string s) { Console.WriteLine(s); } }
```

В методе Main создан объект A `a = new A();` Как подключить прослушивание события?

1. `A.Evn += new A.E(C.S);`

2. `a.Evn += new A.E(C.S);`

3. `A.Evn += A.E(C.S);`

4. `a.Evn += A.E(C.S)`

5. `A.Evn += C.S;`

6. `a.Evn += C.S;`

7. `A.Evn += new C.S;`

8. `a.Evn += new C.S;`

26. Что произойдет если при возникновении события не установлен обработчик события?

1. Ничего не произойдет
2. Событие будет обработчиком по умолчанию
3. Произойдет исключение

27. Как запустить вторичный поток?

Под `c.proc` подразумевается метод некоторого класса

1. `Thread t = new Thread(new ThreadStart(c.proc));`
`t.Start();`
2. `Thread t = new Thread(c.proc);`
`t.Start();`
3. `ThreadStart t = new ThreadStart(c.proc);`
`t.Start();`
4. `ThreadStart t = new ThreadStart(new Thread(c.proc));`
`t.Start();`

28. Какие варианты получения объекта Type допустимы, если объект a создан следующим образом
`A a = new A();` ?

1. `Type T = a.GetType();`
2. `Type T = A.GetType();`
3. `Type T = typeof(a);`
4. `Type T = typeof(A);`
5. `Type T = Type.GetType(A);`
6. `Type T = Type.GetType("A");`

29. Что выведет следующий фрагмент кода?

```
A a = new A();  
Type T = a.GetType();  
Console.WriteLine(T.BaseType);
```

1. A
2. object.A
3. System.Object
4. System.A
5. System.Object.A

30. Какие из перечисленных классов используются для ввода/вывода информации в память, а не в файл?

1. FileStream
2. StreamReader
3. MemoryStream
4. BufferedStream
5. StringReader
6. TextReader
7. BinaryReader
8. StringWriter