

1.

**Интерфейс может содержать в себе (укажите все верные ответы):**

- 1) Методы.
- 2) Свойства.
- 3) Поля.
- 4) События.
- 5) Индексаторы.

2.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
interface IInterface {  
    int Division(int args);  
}
```

```
interface IInterface2 {  
    int Division(int args);  
}
```

```
class A : IInterface, IInterface2 {  
    public int Division(int args) => args % 5;  
    int IInterface.Division(int args) => args % 3;  
}
```

```
class Program {  
    static void Main() {  
        IInterface a = new A();  
        Console.WriteLine(a.Division(38));  
    }  
}
```

**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: +++*

3.

**В результате выполнения фрагмента программы:**

```
using System;

interface IInterface {
    int Meth(int args);
}

interface IInterface2 {
    int Meth(IInterface ii);
}

class A : IInterface {
    public int Meth(int args) => args * 3;
}

class B : A, IInterface2 {
    public int Meth(IInterface ii) {
        return ii.Meth(30);
    }
}

class Program {
    static void Main() {
        A a = new A();
        B b = new B();
        Console.Write(b.Meth(a));
    }
}
```

**на экран будет выведено:**

4.

**В результате выполнения фрагмента программы:**

```
using System;
```

```
public class Type1 { }
public class Type2 : Type1 { }
public class Type3 : Type2 { }

public class Program {
    public static Type3 MyMethod(Type1 t) {
        switch (t.ToString()) {
            case "Type1":
                Console.Write(1);
                break;
            case "Type2":
                Console.Write(2);
                break;
            case "Type3":
                Console.Write(3);
                break;
        }
        return t as Type3 ?? new Type3();
    }
    static void Main() {
        Func<Type1, Type3> f1 = MyMethod;
        Func<Type2, Type2> f2 = f1;
        Console.Write(MyMethod(new Type3()));
        f1(new Type1());
        f2(new Type2());
    }
}
```

**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.

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

- 1) Ковариантность позволяет приводить базовый тип к производному.
- 2) Контравариантность позволяет приводить производный тип к базовому.
- 3) Все типы, которые возможно обобщить, могут содержать в себе ковариантность или контравариантность.
- 4) Если тип не является ковариантным или контравариантным, то такой тип называется инвариантным.
- 5) Ковариантные типы могут быть только типом возвращаемого значения, а контравариантные типы могут быть только типов входного параметра.

6.

**Выберите делегаты из .NET (укажите все верные ответы):**

- 1) Action.
- 2) Function.
- 3) Working.
- 4) Func.
- 5) Comparasion.

7.

**В результате выполнения фрагмента программы:**

```
using System;

interface IInterface<T> {
    T GetT(T arg);
}

class A<T1> : IInterface<T1>
    where T1 : struct {
    public T1 GetT(T1 arg) => arg;
}

class B<T2> : A<T2> {
    public int GetT(int arg) => arg;
}

class Program {
    static void Main() {
        A<int> obj = new B<int>();
        Console.Write(obj.GetT(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.

**Объявлен класс MyClass<T>. Укажите допустимые ограничения для T, не вызывающие ошибки компиляции (считаем, что требуемые пространства имён присутствуют) (укажите все верные ответы):**

- 1) where T : class, struct, new();
- 2) where T : Random, IComparable, new();
- 3) where T : struct, Int32, new();
- 4) where T : new(), class;
- 5) where T : class, IDisposable, IEnumerable, IEnumerator<T>, new();

9.

**Обобщённым может быть (укажите все верные ответы):**

- 1) Интерфейс.
- 2) Структура.
- 3) Поле.
- 4) Класс.
- 5) Делегат.

10.

**В результате выполнения фрагмента программы:**

```
using System;

class MyClass<T> {
    public static T X = default(T);
    public static void Meth() {
        Console.Write(X.ToString());
    }
}

class Program {
    static void Main() {
        Console.Write( (MyClass<int>.X++ + 5) *
(MyClass<int>.X++ + 3));
        MyClass<uint>.Meth();
    }
}

на экран будет выведено:
```

1	1245
2	2
3	90
4	3Type312
5	45
6	14
7	***
8	25
9	1245
10	200