

1.

В результате выполнения фрагмента программы:

```
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 C();
        B b = new C();
        C c = new C();
        a.Print();
        b.Print();
        c.Print();
    }
}
```

на экран будет выведено:

2.

В результате выполнения фрагмента программы:

```
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();  
    }  
}
```

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;

class A {
    public A() {
        Console.WriteLine("A");
    }

    public void Print() {
        Console.WriteLine("APrint");
    }
}

class B : A{
    public B() {
        Console.WriteLine("B");
    }

    public override void Print() {
        Console.WriteLine("BPrint");
    }
}

class Program {
    static void Main() {
        A obj = new B();
        obj.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: +++

4.

В результате выполнения фрагмента программы:
<pre>using System; class Program { private enum MyEnum { arg1 = 1, arg2, arg3 = 5, arg4 = 8 } static void Main() { Console.WriteLine(MyEnum.arg3.HasFlag(MyEnum.arg1)); } }</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>

5.

Выберите виды отношения между классами (укажите все верные ответы):
<ul style="list-style-type: none">1) Независимость.2) Вложение.3) Включение.4) Агрегация.5) Декомпозиция.

6.

<p>В данном фрагменте кода использованы следующие виды отношений между классами:</p> <pre>class A { B b; public A() { b = new B(new A()); } } class B : A { A a; public B(A a) { this.a = a; } }</pre>
<ul style="list-style-type: none">1) Вложение.2) Агрегация.3) Композиция.4) Наследование.5) Независимость.

7.

В результате выполнения фрагмента программы:

```
using System;

enum MyEnum {
    a = 5,
    b, c,
    d, e,
    f = 13
}

class Program {
    static void Main() {
        Console.Write((int)MyEnum.c + (int)MyEnum.e);
    }
}
```

на экран будет выведено:

8.

В результате выполнения фрагмента программы:

```
using System;

enum MyEnum {
    a = 5,
    b, c,
    d, e,
    f = 13
}

enum MyEnum2 {
    a, b,
    c = 6, d,
    e, f
}

class Program {
    static void Main() {
        Console.Write(MyEnum.e < MyEnum2.e);
    }
}
```

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.

Выберите все верные утверждения:

- 1) В перечислениях первый член имеет умалчиваемое значение, равное 1
- 2) Базовый тип перечислений по умолчанию: **uint**.
- 3) Члены перечисления могут содержать в себе одинаковые члены по числовому значению.
- 4) Вместо точек с запятой в перечислениях используются запятые.
- 5) Умалчиваемое значение для членов перечисления – на 1 больше предыдущего.

10.

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:

1	BABACBA
2	+++
3	***
4	True
5	1234
6	234
7	16
8	***
9	345
10	326496