

```

using System;

namespace OOP_12_2
{
    abstract class Shape
    {
        public abstract double Area();
    }

    class Triangle:Shape
    {
        double a, b, c;
        public Triangle(double a, double b, double c)
        {
            this.a = a;
            this.b = b;
            this.c = c;
        }
        public override double Area()
        {
            double p = (a + b + c) / 2;
            return Math.Sqrt(p * (p - a) * (p - b) * (p - c));
        }
    }

    class Rectangle:Shape
    {
        double a, b;
        public Rectangle(double a, double b)
        {
            this.a = a;
            this.b = b;
        }
        public override double Area()
        {
            return a * b;
        }
    }

    class Circle : Shape
    {
        double r;
        public Circle(double r)
        {
            this.r = r;
        }
        public override double Area()
        {
            return Math.PI * r * r;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Circle circle = new Circle(2);
            Rectangle rectangle = new Rectangle(3, 4);
            Triangle triangle = new Triangle(3, 4, 5);
        }
    }
}

```

```

        Console.WriteLine("{0:###.###}", circle.Area());
        Console.WriteLine("{0:###.###}", rectangle.Area());
        Console.WriteLine("{0:###.###}", triangle.Area());

        Shape[] shapes = { circle, rectangle, triangle };
        foreach (Shape s in shapes)
        {
            Console.WriteLine("{0:###.###}", s.Area());
        }
    }
}

```

```
using System;
```

```
namespace OOP_12_3
```

```
{
    abstract class Shape
    {
        public string Name { get; set; }
        public void Info()
        {
            Console.WriteLine("Це {0} і його площа становить {1:###.###}", Name, Area());
        }
        public abstract double Area();
    }
}

```

```
class Triangle : Shape
```

```
{
    double a, b, c;
    public Triangle(double a, double b, double c)
    {
        Name = "трикутник";
        this.a = a;
        this.b = b;
        this.c = c;
    }
    public override double Area()
    {
        double p = (a + b + c) / 2;
        return Math.Sqrt(p * (p - a) * (p - b) * (p - c));
    }
}

```

```
class Rectangle : Shape
```

```
{
    double a, b;
    public Rectangle(double a, double b)
    {
        Name = "прямокутник";
        this.a = a;
        this.b = b;
    }
}

```

```

        public override double Area()
        {
            return a * b;
        }
    }
    class Circle : Shape
    {
        double r;
        public Circle(double r)
        {
            Name = "кpyr";
            this.r = r;
        }
        public override double Area()
        {
            return Math.PI * r * r;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Circle circle = new Circle(2);
            Rectangle rectangle = new Rectangle(3, 4);
            Triangle triangle = new Triangle(3, 4, 5);

            Console.WriteLine("{0:###}", circle.Area());
            Console.WriteLine("{0:###}", rectangle.Area());
            Console.WriteLine("{0:###}", triangle.Area());

            Shape[] shapes = { circle, rectangle, triangle };
            foreach (Shape s in shapes)
            {
                s.Info();
            }
        }
    }
}

using System;

namespace SampleInterface
{
    interface IFlyable
    {
        void Fly();
    }

    class Rocket : IFlyable
    {
        public void Fly()
        {
            Console.WriteLine("Ракета літає, бо має реактивний двигун");
        }
    }
}

```

```

class Eagle : IFlyable
{
    public void Fly()
    {
        Console.WriteLine("Орел літає, бо має крила");
    }
}

class Balloon : IFlyable
{
    public void Fly()
    {
        Console.WriteLine("Повітряна куля літає, бо легша за повітря");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Eagle eagle = new Eagle();
        Rocket rocket = new Rocket();
        Balloon balloon = new Balloon();

        eagle.Fly();
        rocket.Fly();
        balloon.Fly();
    }
}
}

```

```

using System;

```

```

namespace SampleInterface
{
    interface IFlyable
    {
        void Fly();
    }

    interface ISwimmable
    {
        void Swimm();
    }

    class Rocket : IFlyable
    {
        public void Fly()
        {
            Console.WriteLine("Ракета літає, бо має реактивний двигун");
        }
    }

    class Sailboat : ISwimmable
    {
        public void Swimm()
        {

```

```

        Console.WriteLine("Парусне судно плаває, оскільки має вітрило");
    }
}

class Duck : IFlyable, ISwimmable
{
    public void Fly()
    {
        Console.WriteLine("Качка літає, бо має крила");
    }
    public void Swimm()
    {
        Console.WriteLine("Качка плаває, тому що має перпонки на лапках");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Rocket rocket = new Rocket();
        rocket.Fly();

        Sailboat sailboat = new Sailboat();
        sailboat.Swimm();

        Duck duck = new Duck();
        duck.Fly();
        duck.Swimm();
    }
}

```

```
using System;
```

```

namespace SampleInterface4
{
    interface ISwimmable
    {
        void ToSwimm();
    }

    interface IDiveable : ISwimmable
    {
        void ToDive();
    }

    class Swimmer : ISwimmable
    {
        public void ToSwimm()
        {
            Console.WriteLine("Плавець може добре плавати");
        }
    }
}

```

```

class Diver : IDiveable
{
    public void ToDive()
    {
        Console.WriteLine("Дайвер може добре пірнати");
    }

    public void ToSwimm()
    {
        Console.WriteLine("Дайвер також може добре плавати");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Swimmer swimmer = new Swimmer();
        swimmer.ToSwimm();

        Console.WriteLine();
        Diver diver = new Diver();
        diver.ToDive();
        diver.ToSwimm();
    }
}

```

```

using System;

namespace OOP_18
{
    class Student : IComparable<Student>
    {
        public string Name { get; set; }
        public int Mark { get; set; }
        public Student(string name, int mark)
        {
            Name = name;
            Mark = mark;
        }
        public int CompareTo(Student otherStudent)
        {
            if (Mark > otherStudent.Mark) return 1;
            else
            if (Mark < otherStudent.Mark) return -1;
            else
                return 0;
        }
    }
}

```

```

class Program
{
    static void Main(string[] args)
    {
        Student[] list = new Student[5];

        list[0] = new Student("Яковенко", 80);
        list[1] = new Student("Петренко", 79);
        list[2] = new Student("Антонова", 85);
        list[3] = new Student("Федоренко", 75);
        list[4] = new Student("Богданов", 94);

        Console.WriteLine("Початковий список студентів");
        foreach (Student x in list)
        {
            Console.WriteLine("{0,-10} {1}", x.Name, x.Mark);
        }
        Console.WriteLine("Відсортований список студентів");
        Array.Sort(list);
        foreach (Student x in list)
        {
            Console.WriteLine("{0,-10} {1}", x.Name, x.Mark);
        }
    }
}
}

```

```

using System;

namespace OOP_AbstractClassAndInterface
{
    // Абстрактний клас Птах

    abstract class Bird
    {
        public string Name { get; set; }
        public double Weight { get; set; }
        public abstract void AboutMe();
    }

    interface IFlyable
    {
        void Fly();
    }

    interface IRunnable
    {
        void Run();
    }
}

```

```

interface ISwimable
{
    void Swim();
}

// Страус
class Ostrich : Bird, IRunnable
{
    public Ostrich(string name, double weight)
    {
        Name = name;
        Weight = weight;
    }
    public override void AboutMe()
    {
        Console.WriteLine($"Я страус {Name} і моя вага {Weight} кг.");
        Run();
    }

    public void Run()
    {
        Console.WriteLine("Страус бігає, розвиваючи швидкість до 60-70 км/год");
    }
}

// Пінгвін
class Penguin : Bird, IRunnable, ISwimable
{
    public Penguin(string name, double weight)
    {
        Name = name;
        Weight = weight;
    }
    public override void AboutMe()
    {
        Console.WriteLine($"Я пінгвін {Name} і моя вага {Weight} кг.");
        Run();
        Swim();
    }

    public void Run()
    {
        Console.WriteLine("На суші пінгвіни повільно ходять, перевалюючись з боку на
бік");
    }

    public void Swim()
    {
        Console.WriteLine("Пінгвіни чудово плавають і пірнають");
    }
}

// Курка
class Hen : Bird, IFlyable, IRunnable
{
    public Hen(string name, double weight)

```



```

    {
        Name = name;
        Weight = weight;
    }
    public override void AboutMe()
    {
        Console.WriteLine($"Я курка {Name} і моя вага {Weight} кг.");
        Fly();
        Run();
    }

    public void Fly()
    {
        Console.WriteLine("Курка летить погано і недалеко");
    }

    public void Run()
    {
        Console.WriteLine("Курка досить швидко бігає");
    }
}

// Качка
class Duck : Bird, ISwimable, IFlyable, IRunnable
{
    public Duck(string name, double weight)
    {
        Name = name;
        Weight = weight;
    }
    public override void AboutMe()
    {
        Console.WriteLine($"Я качка {Name} і моя вага {Weight} кг.");
        Fly();
        Run();
        Swim();
    }

    public void Fly()
    {
        Console.WriteLine("Дикі качки добре літають");
    }

    public void Run()
    {
        Console.WriteLine("Качка може бігати, але незграбно перевалюючись");
    }

    public void Swim()
    {
        Console.WriteLine("Качка вміє плавати і навіть пірнати");
    }
}

```

```

class Program
{
    static void Main(string[] args)
    {
        Duck duck = new Duck("Доналд", 4);
        Hen hen = new Hen("Ліфї", 3);
        Ostrich ostrich = new Ostrich("Роккі", 150);
        Penguin penguin = new Penguin("Ріко", 20);

        Bird[] birds = { duck, hen, ostrich, penguin };
        foreach (Bird bird in birds)
        {
            bird.AboutMe();
        }

        ISwimable[] swimables = { duck, penguin };
        Console.WriteLine("Птахи, які можуть плавати:");
        foreach (var bird in swimables)
        {
            bird.Swim();
        }
    }
}

```