

Лабораторная работа №3

Цель работы

Приобретение навыков реализации абстрактных классов, виртуальных методов и наследования, а также перегрузки операторов.

В ходе работы требуется создать абстрактный класс **Shape**, объявить абстрактные метод **GetPerimeter** и **GetArea** для расчета периметра и площади, **protected** свойства **Name** и **Area**, реализовать конструктор, принимающий один параметр – имя создаваемой фигуры (**Name**). После чего требуется создать классы фигур **Triangle**, **Square**, **Circle**, наследуемые от абстрактного класса **Shape**, переопределить методы **GetPerimeter** и **GetArea** и реализовать их для треугольника, квадрата и окружности. В заключение необходимо переопределить стандартные операторы сложения, вычитания, умножения, сравнения для классов, наследуемых от **Shape**.

Упражнение 1. Создание абстрактного класса Shape

Создадим абстрактный класс **Shape**, от которого впоследствии будут наследоваться классы фигур (треугольник, окружность, квадрат)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Shape
{
    public abstract class Shape
    {
        //Свойства класса для хранения имени, площади и периметра создаваемой фигуры
        protected string _name;
        protected double _area;
        protected double _perimeter;

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }

        public abstract double Area {get; set;}
        public abstract double Perimeter {get; set;}

        public abstract double GetShapePerimeter();
        public abstract double GetPerimeter();

        public Shape(string name)
        {
            Name = name;
        }
    }
}
```

Определим абстрактные методы для расчета площади фигуры и периметра (длины окружности).

Упражнение 2. Создание классов фигур, наследуемых от Shape

Реализуем класс **Triangle**

```
public class Triangle : Shape
{
    private double _sideAb;
    private double _sideBc;
    private double _sideCa;

    public double SideAb...
    public double SideBc...
    public double SideCa...
    public override double Area...
    public override double Perimeter...

    public override double GetShapePerimeter()
    {
        return (SideAb + SideBc + SideCa);
    }

    public override double GetShapeArea()
    {
        double P = GetShapePerimeter();
        double p = P/2;
        return (Math.Pow((p*(p - SideAb)*(p - SideBc)*(p - SideCa)), 0.5));
    }

    public Triangle(string name, double sideAb, double sideBc, double sideCa)
        : base(name)
    {
        SideAb = sideAb;
        SideBc = sideBc;
        SideCa = sideCa;
    }
}
```

Реализуем проверку вводимых значений в конструкции **get{}set{}** , а также алгоритмы расчета периметра и площади перегрузкой объявленных ранее абстрактных методов.

Реализуйте классы Square и Circle самостоятельно.

Упражнение 3. Перегрузка операторов

C# позволяет переопределять поведение стандартных операторов в зависимости от типов операндов.

Перегрузим стандартный оператор суммирования для типов **Shape** таким образом, чтобы результатом суммирования была сумма площадей фигур.

```
public abstract class Shape
{
    //Свойства класса для хранения имени, площади и периметра создаваемой фигуры
    protected string _name;
    protected double _area;
    protected double _perimeter;

    public string Name
    {
        get { return _name; }
        set { _name = value; }
    }
    public abstract double Area {get; set;}
    public abstract double Perimeter {get; set;}

    public abstract double GetShapePerimeter();
    public abstract double GetShapeArea();

    public Shape(string name)
    {
        Name = name;
    }

    public static double operator +(Shape shape1, Shape shape2)
    {
        return (shape1.Area+shape2.Area);
    }
}
```

В результате мы получим возможность складывать напрямую экземпляры классов, наследуемых от **Shape**.

Переопределите самостоятельно операторы вычитания, умножения и деления, а также логические операторы **<**, **>**.

Упражнение 4. Вывод

Запросите у пользователя все необходимые параметры созданных геометрических фигур, выведите их площади и периметры (длину окружности). Выведите результаты операций над фигурами (сложение, вычитание, умножение, сравнение) с помощью переопределенных операторов.