

Московский государственный технический
университет имени Н.Э. Баумана

Парадигмы и конструкции языков программирования
Отчёт по лабораторной работе №2

Работу выполнил
Студент группы ИУ5-35Б
Шиленок Д.А.

2023 г.

Задание

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.
3. Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».
4. Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.
5. Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».
6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод `Object.ToString()`, который возвращает в виде строки основные параметры фигуры и ее площадь.
7. Разработать интерфейс `IPrint`. Интерфейс содержит метод `Print()`, который не принимает параметров и возвращает `void`. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса `IPrint`. Переопределяемый метод `Print()` выводит на консоль информацию, возвращаемую переопределенным методом `ToString()`.

Текст программы

Shape.cs

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

namespace lab_2
{
    interface IPrint
    {
        void Print();
    }

    abstract class Shape : IComparable, IPrint
    {
        public double Area { get; set; }
        public abstract double CalculateArea();
        public int CompareTo(object? obj)
        {
            if (obj == null) return 1;

            if (obj is Shape otherShape)
                return this.Area.CompareTo(otherShape.Area);
            else
                throw new ArgumentException("Object is not a Shape");
        }

        public abstract void Print();
    }

    class Rectangle : Shape
    {
        private double _width;
        private double _height;
        public double Width
        {
            get { return this._width; }
            set { this._width = (value > 0 ? value : 0); Area = CalculateArea(); }
        }
        public double Height
        {
            get { return this._height; }
            set { this._height = (value > 0 ? value : 0); Area = CalculateArea(); }
        }
        public Rectangle(double width, double height)
        {
            Width = width;
            Height = height;
        }
    }
}
```

```

    public override double CalculateArea()
    {
        return Width * Height;
    }

    public override string ToString()
    {
        return $"Rectangle: Width = {Width}, Height = {Height}, Area = {Area}";
    }

    public override void Print()
    {
        Console.WriteLine(this.ToString());
    }
}

class Square : Rectangle
{
    public Square(double width) : base(width, width)
    {
    }
    public override string ToString()
    {
        return $"Square: Width = {Width}, Area = {Area}";
    }
}

class Circle : Shape
{
    private double _radius;
    public double Radius
    {
        get { return this._radius; }
        set { this._radius = (value > 0 ? value : 0); Area = CalculateArea(); }
    }
    public Circle(double radius)
    {
        Radius = radius;
    }
    public override double CalculateArea()
    {
        return Math.PI * Radius * Radius;
    }
    public override string ToString()
    {
        return $"Circle: Radius = {Radius}, Area = {Area}";
    }
    public override void Print()
    {

```

```

        Console.WriteLine(this.ToString());
    }

}
}

```

Program.cs

```

namespace lab_2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Rectangle rectangle1 = new Rectangle(10, 2);
            Rectangle rectangle2 = new Rectangle(5, 8);
            Square square = new Square(4);
            Circle circle = new Circle(1);
            ArrayList arrayList = new ArrayList();
            arrayList.Add(rectangle1);
            arrayList.Add(rectangle2);
            arrayList.Add(square);
            arrayList.Add(circle);
            foreach (Shape shape in arrayList) { shape.Print(); }
        }
    }
}

```

Вывод программы

```

Rectangle: Width = 10, Height = 2, Area = 20
Rectangle: Width = 5, Height = 8, Area = 40
Square: Width = 4, Area = 16
Circle: Radius = 1, Area = 3,141592653589793

```