

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «ПиКЯП»

Отчет по лабораторной работе №4

«Разработка на языке программирования C# »

Выполнил:

студент группы ИУ5-36Б

Рухлин Алексей

Проверил:

преподаватель каф.  
ИУ5

Нардид А. Н.

Москва, 2024 г.

## Описание задания

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

```
using System;
```

```
public abstract class GeometricFigure
```

```
{
```

```
    public virtual double CalculateArea()
```

```
    {
```

```
        return 0;    }
```

```
    public override string ToString()
```

```
    {
```

```
        return $"Тип фигуры: {this.GetType().Name},  
Площадь: {CalculateArea()}";
```

```
    }
```

```
}
```

```
public class Rectangle : GeometricFigure
```

```
{
```

```
    public double Width { get; set; }
```

```
    public double Height { get; set; }
```

```
public Rectangle(double width, double height)
{
    Width = width;
    Height = height;
}

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

public override string ToString()
{
    return $"Прямоугольник: Ширина = {Width},
Высота = {Height}, {base.ToString()}";
}
}
```

```
public class Square : Rectangle
{
    public Square(double side) : base(side, side) { }

    public override string ToString()
    {
        return $"Квадрат: Сторона = {Width},
{base.ToString()}";
    }
}
```

```
public class Circle : GeometricFigure
{

    public double Radius { get; set; }

    public Circle(double radius)
    {
        Radius = radius;
    }
}
```

```
}
```

```
public override double CalculateArea()
{
    return Math.PI * Radius * Radius;
}
```

```
public override string ToString()
{
    return $"Круг: Радиус = {Radius},
{base.ToString()}";
}
}
```

```
public interface IPrint
{
    void Print(); }
}
```

```
public class RectangleWithPrint : Rectangle, IPrint
{
    public RectangleWithPrint(double width, double
height) : base(width, height) { }

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

```
public class SquareWithPrint : Square, IPrint
{
    public SquareWithPrint(double side) : base(side) {
}

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

```
}
```

```
public class CircleWithPrint : Circle, IPrint  
{  
    public CircleWithPrint(double radius) :  
base(radius) { }
```

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

```
class Program  
{  
    static void Main(string[] args)  
    {  
  
        RectangleWithPrint rectangle = new  
RectangleWithPrint(5, 10);
```



```
SquareWithPrint square = new  
SquareWithPrint(4);
```

```
CircleWithPrint circle = new  
CircleWithPrint(7);
```

```
rectangle.Print();  
square.Print();  
circle.Print();  
}  
}
```

---

**Экранные формы с примерами выполнения программы**

**1. Пример 1**

Входные данные (в программе заданы конструктором):

- Прямоугольник: ширина = 5, высота = 10
- Квадрат: сторона = 4
- Круг: радиус = 7

**Вывод программы:**

makefile

Копировать код

Прямоугольник: Ширина = 5, Высота = 10, Тип фигуры:  
RectangleWithPrint, Площадь: 50

Квадрат: Сторона = 4, Тип фигуры: SquareWithPrint,  
Площадь: 16

Круг: Радиус = 7, Тип фигуры: CircleWithPrint, Площадь:  
153.93804002589985

## 2. Пример 2

Изменение данных в программе (другие значения):

- Прямоугольник: ширина = 3, высота = 8
- Квадрат: сторона = 6
- Круг: радиус = 5

### Вывод программы:

makefile

Копировать код

Прямоугольник: Ширина = 3, Высота = 8, Тип фигуры:  
RectangleWithPrint, Площадь: 24

Квадрат: Сторона = 6, Тип фигуры: SquareWithPrint,  
Площадь: 36

Круг: Радиус = 5, Тип фигуры: CircleWithPrint, Площадь:  
78.53981633974483

---

### Вывод

1. В ходе работы была реализована программа, демонстрирующая использование абстрактных классов и интерфейсов.
2. Программа показала корректную работу: расчет площадей геометрических фигур осуществляется верно.
3. Использование интерфейса `IPrint` позволило обеспечить единообразный вывод данных для всех фигур.
4. Работа выполнена в полном соответствии с заданием.

