**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №2

«Работа с классами»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Корчевский Александр |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

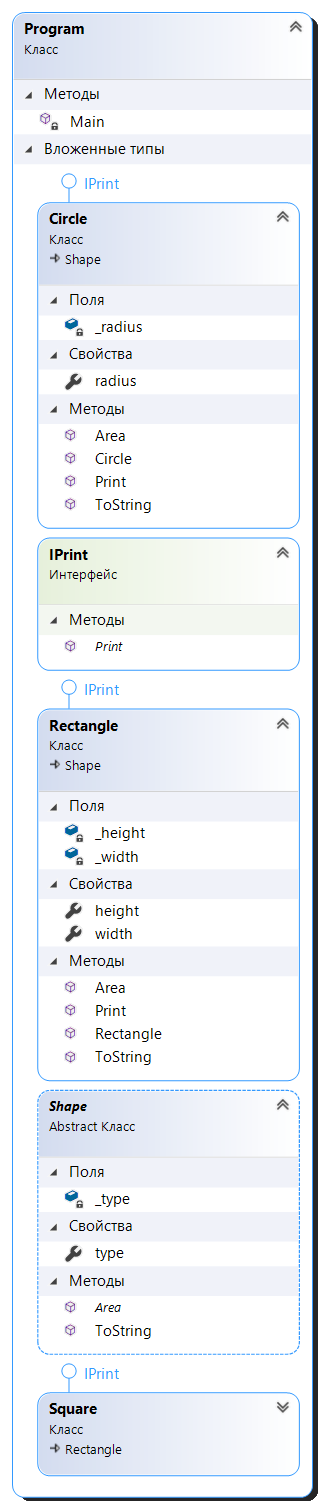
Москва, 2020 г.

**Условие 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().

**Диаграмма классов:**



**Текст программы:**

using System;

namespace lab02

{

class Program

{

interface IPrint

{

void Print();

}

abstract class Shape

{

private string \_type;

public string type

{

get => \_type;

protected set

{

\_type = value;

}

}

public abstract double Area();

public override string ToString()

{

return type + " площадью " + Area().ToString();

}

}

class Rectangle : Shape, IPrint

{

private double \_width;

private double \_height;

public double width

{

get => \_width;

set

{

if (value < 0)

{

Console.WriteLine("Width can not be < 0, it is set to 0.");

\_width = 0;

}

else

\_width = value;

}

}

public double height

{

get => \_height;

set

{

if (value < 0)

{

Console.WriteLine("Height can not be < 0, it is set to 0.");

\_height = 0;

}

else

\_height = value;

}

}

public override double Area() => width \* height;

public Rectangle(double Width, double Height)

{

width = Width;

height = Height;

type = "Прямоугольник";

}

public override string ToString()

{

return base.ToString() + "\nВысота - " + height + "\nШирина - " + width + "\n";

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

class Square : Rectangle, IPrint

{

public Square(double Length) : base (Length, Length)

{

type = "Квадрат";

}

public override string ToString()

{

return type + " площадью " + Area().ToString() + "\nДлина - " + width + "\n";

}

}

class Circle : Shape, IPrint

{

private double \_radius;

public double radius { get; set; }

public override double Area() => Math.PI \* radius \* radius;

public override string ToString()

{

return base.ToString() + "\nРадиус - " + radius + "\n";

}

public void Print()

{

Console.WriteLine(ToString());

}

public Circle(double Radius) { radius = Radius; type = "Круг"; }

}

static void Main(string[] args)

{

Rectangle a = new Rectangle(2, 3);

a.Print();

Square b = new Square(4);

b.Print();

Circle c = new Circle(0.564);

c.Print();

}

}

}

**Результаты:**

