

**Московский государственный технический**

**университет им. Н.Э. Баумана.**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5. Курс «Базовые компоненты интернет технологий»**

**Отчет по лабораторной работе № 2**

**“Работа с классами”**

Выполнил:

студент группы ИУ5-33

Терентьев Владислав

Москва, 2019 г.

## 1. Постановка задачи

Разработать программу, реализующую работу с классами.

Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.

Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».

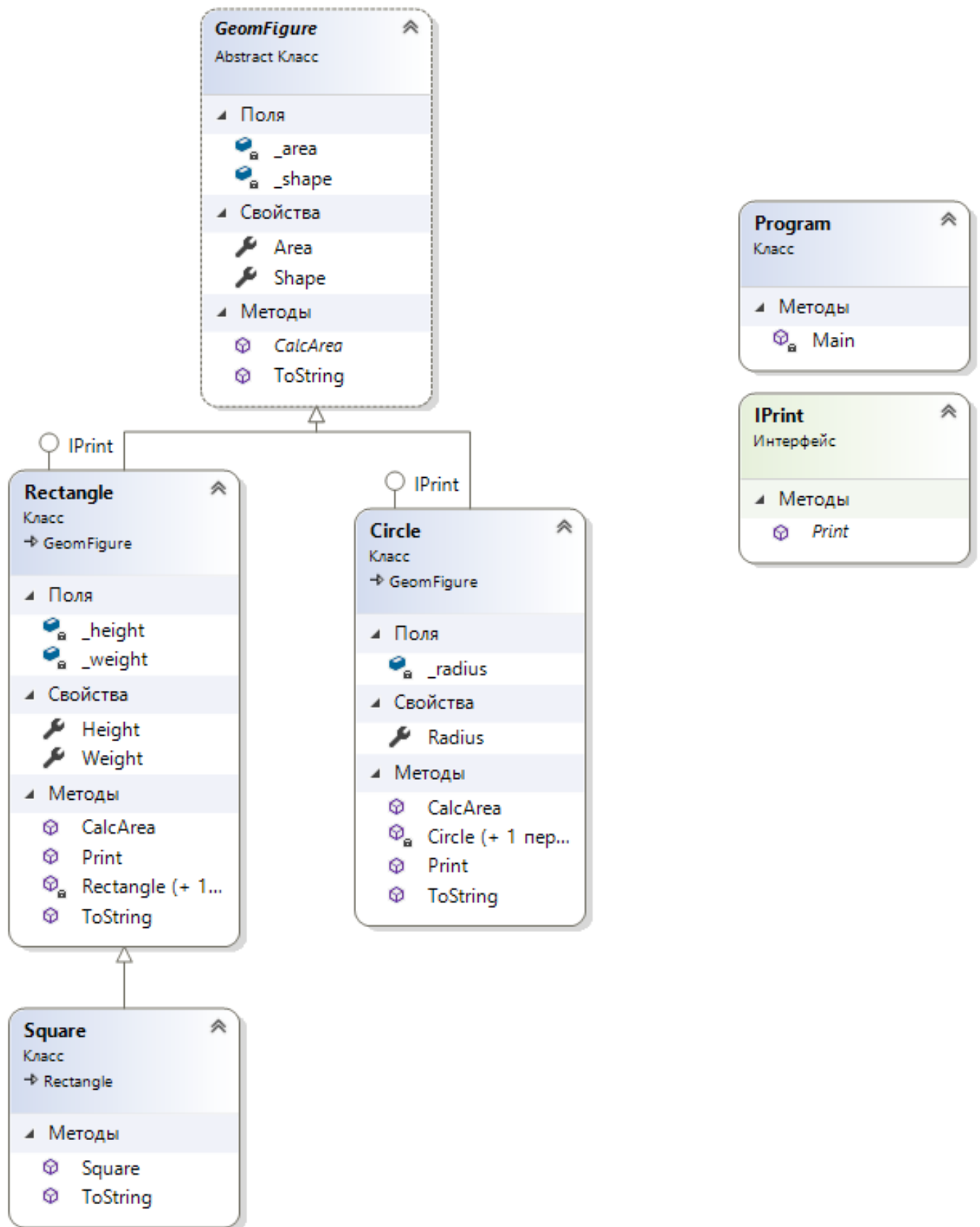
Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.

Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».

Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод `Object.ToString()`, который возвращает в виде строки основные параметры фигуры и ее площадь.

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()`.

## 2. Диаграмма классов



### 3. Текст программы

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

namespace laba2
{
```

```

class Program
{
    static void Main(string[] args)
    {
        Console.Title = "Терентьев Владислав ИУ5-33";
        Rectangle a = new Rectangle(4, 8);
        a.Print();
        Square b = new Square(5.5);
        b.Print();
        Circle c = new Circle(3);
        c.Print();
        Console.ReadKey();
    }
}

abstract class GeomFigure
{
    string _shape;
    double _area;

    public string Shape
    {
        get { return _shape; }
        protected set { _shape = value; }
    }

    public double Area
    {
        get { return _area; }
        protected set { _area = value; }
    }

    public abstract double CalcArea();

    public override string ToString() { return _shape + " площадью " +
_area.ToString(); }
}

interface IPrint { void Print(); }

class Rectangle : GeomFigure, IPrint
{
    double _height = 0;
    double _weight = 0;

    Rectangle() { Shape = "Прямоугольник"; }

    public override double CalcArea() { return _height * _weight; }

    public double Height
    {
        get { return _height; }
        private set
        {
            if (value < 0) { throw new Exception("Высота не может быть
отрицательной"); }
            else { _height = value; }
        }
    }

    public double Weight
    {
        get { return _weight; }
        private set

```

```

        {
            if (value < 0) { throw new Exception("Ширина не может быть отрицательной"); }
            else { _weight = value; }
        }
    }

    public Rectangle(double h, double w) : this()
    {
        Height = h;
        Weight = w;
        Area = CalcArea();
    }

    public override string ToString() { return base.ToString() + ", высотой " + _height + " и шириной " + _weight; }

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

class Square : Rectangle
{
    public Square(double leng) : base(leng, leng) { Shape = "Квадрат"; }

    public override string ToString() { return base.ToString() + " (стороной " + Height + ")"; }
}

class Circle : GeomFigure, IPrint
{
    double _radius = 0;

    Circle() { Shape = "Окружность"; }

    public override double CalcArea() { return Math.PI * _radius * _radius; }

    public double Radius
    {
        get { return _radius; }
        private set
        {
            if (value < 0) { throw new Exception("Радиус не может быть отрицательным"); }
            else { _radius = value; }
        }
    }

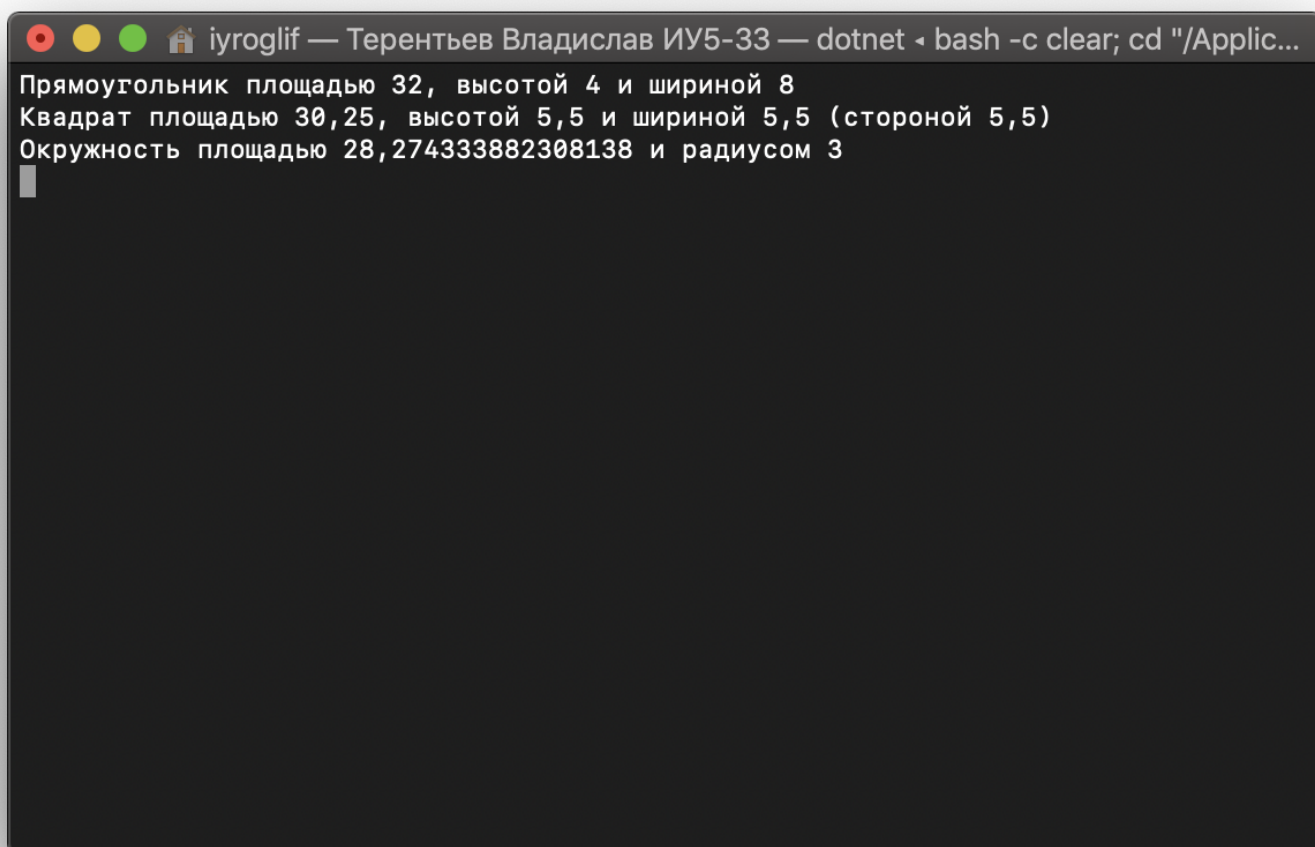
    public Circle(double r) : this()
    {
        Radius = r;
        Area = CalcArea();
    }

    public override string ToString() { return base.ToString() + " и радиусом " + _radius; }

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

```

#### 4. Анализ результатов



```
iyroglif — Терентьев Владислав ИУ5-33 — dotnet ◀ bash -c clear; cd "/Applic...  
Прямоугольник площадью 32, высотой 4 и шириной 8  
Квадрат площадью 30,25, высотой 5,5 и шириной 5,5 (стороной 5,5)  
Окружность площадью 28,274333882308138 и радиусом 3
```

#### 5. Ссылка на репозиторий

<https://github.com/iYroglif/newlabs>