



**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Э. БАУМАНА**

*Факультет «Информатика и системы управления»*

*Кафедра «Системы обработки информации и управления (ИУ5)»*

*Дисциплина «Базовые компоненты Интернет-технологий»*

***Отчет по лабораторной работе № 2:***

***«Работа с классами»***

*Выполнила: Журавлева Полина Валерьевна*

*Группа: ИУ5-31Б*

*Преподаватель: Гапанюк Юрий Евгеньевич*

---

*Дата: 01.10.2018*

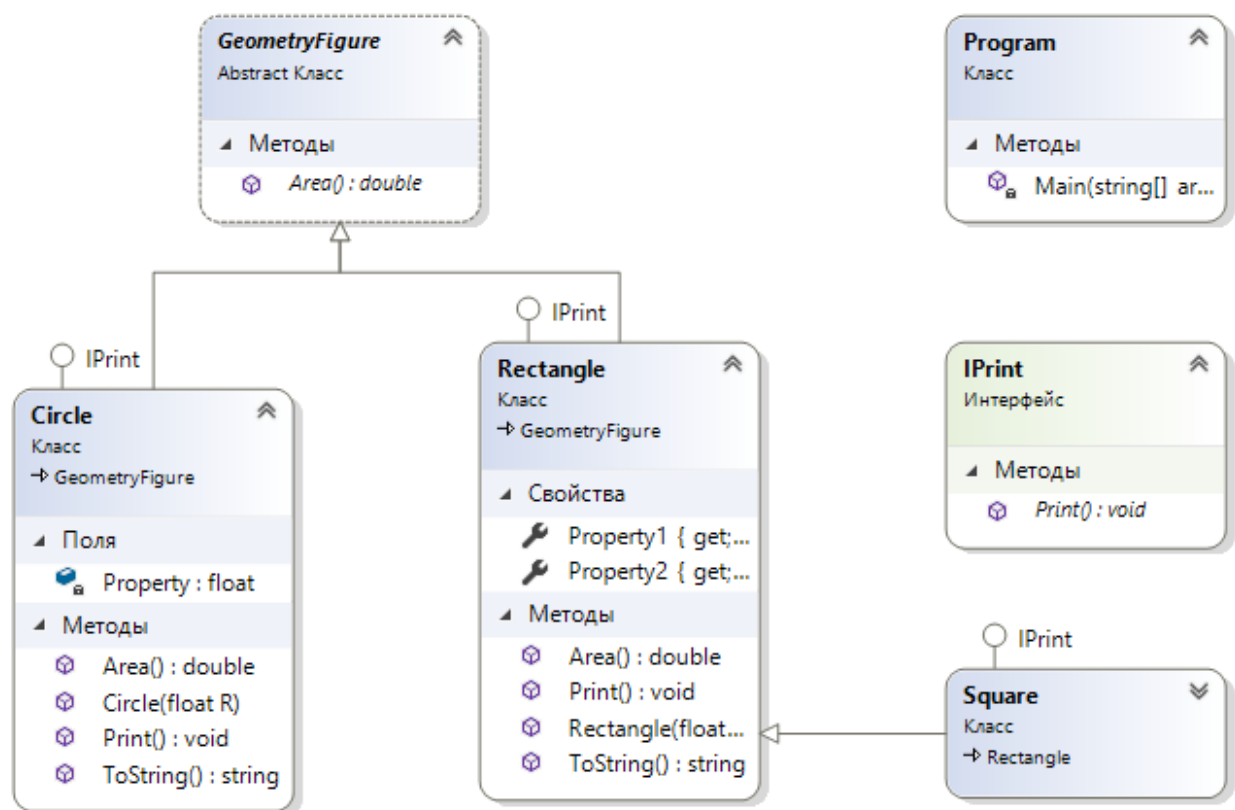
*Подпись:*

### **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()`.

## Диаграмма классов:



## Текст программы:

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

namespace ConsoleApp2
{
    abstract class GeometryFigure
    {
        public abstract double Area();
    }

    interface IPrint
    {
        void Print();
    }

    class Rectangle : GeometryFigure, IPrint
    {
        public float Property1 { get; set; }
        public float Property2 { get; set; }

        public Rectangle(float Weight, float Height)
        {
            Property1 = Weight;
            Property2 = Height;
        }

        public override double Area()
        {
            return Property1 * Property2;
        }

        public void Print()
        {
            Console.WriteLine("Rectangle:\n" + ToString());
        }

        public override string ToString()
        {
            return "Width = " + Property1.ToString() + "\nHeight = " +
Property2.ToString() + "\nS = " + Area().ToString() + "\n";
        }
    }

    class Square : Rectangle, IPrint
    {
        private float side;

        public Square(float s) : base(s, s) { side = s; }

        public new void Print()
        {
            Console.WriteLine("Square:\n" + ToString());
        }

        public override string ToString()
        {
            return "Side = " + side.ToString() + "\nS = " + Area().ToString() + "\n";
        }
    }
}
```

```

class Circle : GeometryFigure, IPrint
{
    private readonly float Property;
    public Circle(float R)
    {
        Property = R;
    }
    public override double Area()
    {
        return Math.PI * Math.Pow(Property, 2);
    }

    public void Print()
    {
        Console.WriteLine("Circle:\n" + ToString());
    }

    public override string ToString()
    {
        return "R = " + Property.ToString() + "\nS = " + Area().ToString() + "\n";
    }
}

class Program
{
    static void Main(string[] args)
    {
        Rectangle Rect = new Rectangle(6, 8);
        Rect.Print();
        Rect.Property1 = 3;
        Rect.Property2 = 4;
        Rect.Print();

        Square Sq = new Square(12);
        Sq.Print();

        Circle Cir = new Circle(9);
        Cir.Print();

        Console.ReadKey();
    }
}

```

### Примеры выполнения программы:

```
Rectangle:
Width = 6
Height = 8
S = 48

Rectangle:
Width = 3
Height = 4
S = 12

Square:
Side = 12
S = 144

Circle:
R = 9
S = 254,469004940773
```

```
Rectangle:
Width = 6
Height = 8
S = 48

Rectangle:
Width = 50
Height = 2
S = 100

Square:
Side = 15
S = 225

Circle:
R = 8
S = 201,061929829747
```

```
Rectangle:
Width = 6
Height = 8
S = 48

Rectangle:
Width = 10
Height = 10
S = 100

Square:
Side = 10
S = 100

Circle:
R = 10
S = 314,159265358979
```