**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35б |  | преподаватель каф. ИУ5 |
| Хижняков Вадим |  |  |
| Подпись и дата: |  | Подпись и дата: |

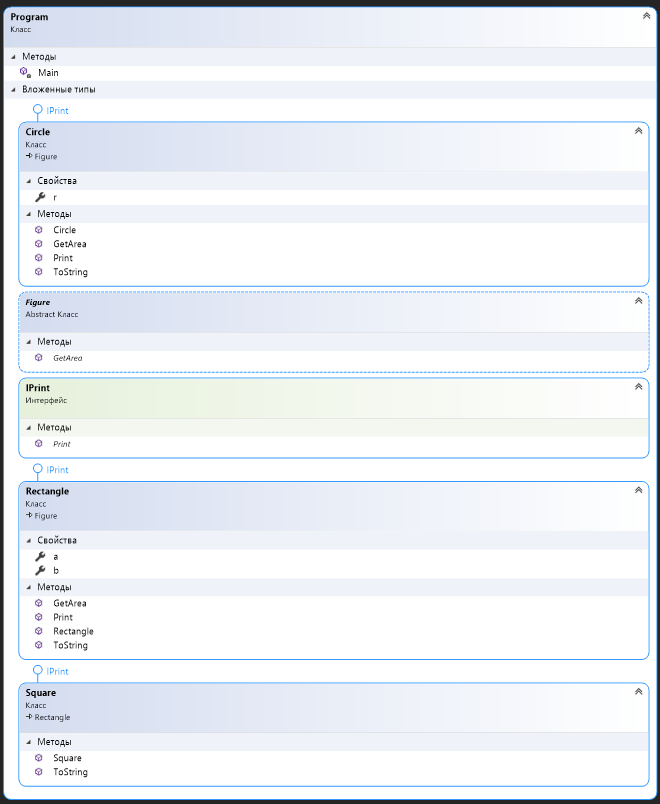
Москва, 2020 г.

# 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().

# Диаграмма классов



# Текст программы

using System;

namespace Sem3Lab2\_Figures

{

class Program

{

abstract class Figure

{

public abstract double GetArea();

}

interface IPrint

{

public void Print();

}

class Rectangle : Figure, IPrint

{

public double a { get; set; }

public double b { get; set; }

public Rectangle(double a, double b)

{

this.a = a;

this.b = b;

}

public override double GetArea()

{

return a \* b;

}

public override string ToString()

{

return "Lenght: " + a.ToString() + " Width: " + b.ToString() + " Area: " + this.GetArea().ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

class Square : Rectangle, IPrint

{

public Square(double a) : base(a, a) { }

public override string ToString()

{

return "Side: " + a.ToString() + " Area: " + this.GetArea().ToString();

}

}

class Circle : Figure, IPrint

{

public double r { get; set; }

public Circle(double r)

{

this.r = r;

}

public override double GetArea()

{

return Math.PI \* Math.Pow(r, 2);

}

public override string ToString()

{

return "Radius: " + r.ToString() + " Area: " + this.GetArea().ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

static void Main(string[] args)

{

Rectangle rec = new Rectangle(4, 5);

rec.Print();

Square sqr = new Square(6);

sqr.Print();

Circle crcl = new Circle(9);

crcl.Print();

}

}

}

# Результаты

