**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 ИУ5-33 |  |  |
| Миронова Александра |  |  |
| Подпись и дата:  16.10.20 |  | Подпись и дата: |

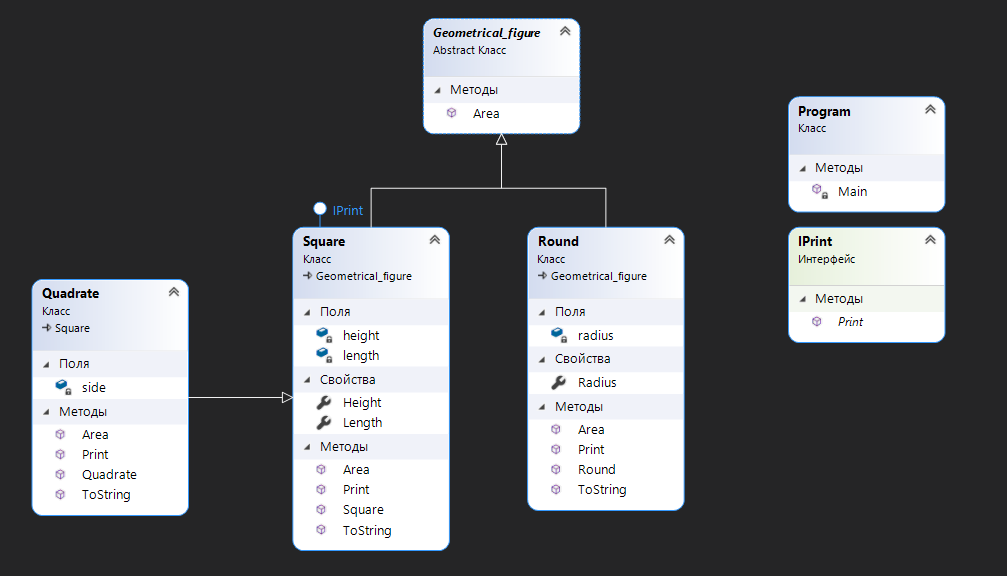
Москва, 2020 г.

**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:**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Geometrical\_fugures\_1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Program is done by Mironova Alexandra Romanovna group IU5-33b");

Console.WriteLine("\n");

Console.WriteLine("Square");

Square MySquare = new Square(5, 4);

MySquare.Print();

Console.WriteLine("\n");

Console.WriteLine("Quadrate");

Quadrate MyQuadrate = new Quadrate(3);

MyQuadrate.Print();

Console.WriteLine("\n");

Console.WriteLine("Round");

Round MyRound = new Round(4);

MyRound.Print();

Console.ReadKey();

}

}

////////////////////////////////////-

interface IPrint

{

void Print();

}

abstract class Geometrical\_figure

{

public virtual double Area()

{

return 0;

}

}

////////////////////////////////////-

class Square : Geometrical\_figure, IPrint

{

public Square(double length, double height)

{

this.length = length;

this.height = height;

}

private double length;

private double height;

override public double Area()

{

return length\* height;

}

public override string ToString()

{

return "Height: "+ height.ToString() + " Length: "+ length.ToString() + " Square area = " + this.Area().ToString();

}

public new void Print()

{

Console.WriteLine(this.ToString());

}

public double Length

{

get

{

return length;

}

/\* set

{

this.length = value;

}\*/

}

public double Height

{

get

{

return height;

}

/\* set

{

this.height = value;

}\*/

}

}

////////////////////////////////////-

class Quadrate : Square

{

public Quadrate(double side): base (side, side)

{

this.side = side;

}

private double side;

override public double Area()

{

return side \* side;

}

override public string ToString()

{

return "Side: " + side.ToString() + " Quadrate area = " + this.Area().ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

////////////////////////////////////-

class Round : Geometrical\_figure

{

public Round(double radius)

{

this.radius = radius;

}

private double radius;

override public double Area()

{

return radius \* radius \* Math.PI;

}

override public string ToString()

{

return "Radius: " + radius.ToString() + "Round area = " + this.Area().ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

public double Radius

{

get

{

return radius;

}

}

}

}

**Экранные формы с примерами выполнения программы:**

