**Московский государственный технический университет**

**им. Н.Э. Баумана**

Кафедра ИУ–5

**Отчет по лабораторной работе №2 по курсу «Базовые компоненты интернет технологий»**

Выполнил:

студент гр. ИУ5-32Б

Деркачев А.В.

\_\_\_\_\_\_\_\_

Проверил:

к.т.н., доцент

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_

**Москва – 2014 г**

1. **Цель выполнения лабораторной работы**

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

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

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

1. **Листинг программы**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{ class Program

{ static void Main(string[] args)

{ pryamoug p1;

Console.WriteLine("Прямоугольник");

Console.Write("Длина: ");

double dl1 = Convert.ToDouble(Console.ReadLine());

Console.Write("Ширина: ");

double si1 = Convert.ToDouble(Console.ReadLine());

p1 = new pryamoug(dl1, si1);

p1.print();

Console.ReadKey();

quad q1;

Console.WriteLine("Квадрат");

Console.Write("Длина: ");

double s1 = Convert.ToDouble(Console.ReadLine());

q1 = new quad(s1);

q1.print();

Console.ReadKey();

krug k1;

Console.WriteLine("Круг");

Console.Write("Радиус: ");

double r1 = Convert.ToDouble(Console.ReadLine());

k1 = new krug(r1);

k1.print();

Console.ReadKey();}}}

GeomFig.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{ public abstract class GeomFig

{ public double s;

public virtual double S()

{ return s; }

public override string ToString()

{return base.ToString() + "";}}}

IPrint.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{ interface IPrint

{void print();}}

krug.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{class krug: GeomFig, IPrint

{private double r;

public double R

{ get { return r; } set { r = value; } }

public krug(double r)

{ this.r = r; }

public override double S()

{s = 3.14\*r\*r;

return base.S();}

public override string ToString()

{return base.ToString()+ "\n" + "Радиус круга: " + r + "\n"+ "Площадь круга" + S() +"\n";}

public void print()

{ Console.WriteLine(this.ToString()); }}}

pryamoug.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{class pryamoug:GeomFig,IPrint

{ public double shir;

public double dlin;

public pryamoug(double dlin, double shir)

{ this.dlin = dlin;

this.shir = shir;}

public override double S()

{ s = dlin \* shir;

return base.S();}

public override string ToString()

{return base.ToString() + "\n" + "Длина прямоугольника: " + dlin + "\n" + "Ширина прямоугольника: " + shir + "\n" + "Площадь прямоугольника: " + S() + "\n"; }

public void print()

{ Console.WriteLine(this.ToString()); }}}

quad.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lr2FT

{ class quad: pryamoug//, IPrint //pryamoug

{ public quad(double s): base(s, s)

{ }

public override string ToString()

{return base.ToString() + "\n" + "Длина квадрата: " + dlin + "\n" + "Площадь квадрата" + S() + "\n";}

public void print()

{ Console.WriteLine(this.ToString()); }}}

1. **Результаты работы программы**



**4. Диаграмма классов**