Московский государственный технический университет имени Н.Э. Баумана (МГТУ им. Н.Э. Баумана)

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

ПО ДИСЦИПЛИНЕ «БАЗОВЫЕ КОМПОНЕНТЫ ИНТЕРНЕТ ТЕХНОЛОГИЙ»

Выполнил(а): Белов Ф.А.

Проверил(а): Гапанюк Ю.Е.

Москва, 2017

**Формулировка задачи:**

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».

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().

**Код**

using System;

using System.Collections;

using System.Linq;

using System.Text;

namespace \_3sem\_2lab

{

class MainClass

{

public static void Main(string[] args)

{

Rectangle rect = new Rectangle(3, 5);

Square sq = new Square(3);

Circle circ = new Circle(3);

rect.Print();

sq.Print();

circ.Print();

Console.ReadLine();

}

}

interface IPrint

{

void Print();

}

abstract class Figure

{

public abstract double Area();

}

class Rectangle: Figure, IPrint

{

private double \_width;

private double \_height;

public Rectangle(double w, double h)

{

width = w;

height = h;

}

public double width

{

get { return \_width; }

private set { \_width = value; }

}

public double height

{

get { return \_height; }

private set { \_height = value; }

}

public override double Area() {

return width \* height;

}

public override string ToString()

{

return string.Format("[Rectangle: width = {0}, height = {1}, area = {2}]", width, height, Area());

}

public void Print()

{

Console.WriteLine(this);

}

}

class Square: Rectangle, IPrint

{

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

public override string ToString()

{

return string.Format("[Square: length = {0}, area = {1}]", height, Area());

}

}

class Circle: Figure, IPrint

{

private double \_radius;

public Circle (double r) {

radius = r;

}

public double radius

{

get { return \_radius; }

private set { \_radius = value; }

}

public override double Area()

{

return Math.PI \* radius \* radius;

}

public override string ToString()

{

return string.Format("[Circle: radius = {0}, area = {1}]", radius, Area());

}

public void Print()

{

Console.WriteLine(this);

}

}

}