Отчет по лабораторной работе 2 по курсу

«Базовые компоненты интернет технологий (БКИТ)»

Выполнил

Студент группы ИУ5-33

Зинченко Ярослав Игоревич

Москва 2017

**Требования:**

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 программы на языке C#:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2

{

class Program

{

static double InputVal(string prompt)

{

double a = 0;

do

Console.Write(prompt);

while (!double.TryParse(Console.ReadLine(), out a));

return a;

}

public static class STATE

{

public const String Rectangle = "1";

public const String Square = "2";

public const String Circle = "3";

}

static string Menu()

{

Console.WriteLine("Площадь какой фигуры необходимо посчитать? \n");

Console.WriteLine("1) Прямоугольник;");

Console.WriteLine("2) Квадрат;");

Console.WriteLine("3) Окружность;");

Console.WriteLine("e) Выход; \n");

return Console.ReadLine();

}

static void Main(string[] args)

{

bool work = true;

IPrint obj;

double a1, b1;

while (work)

{

switch (Menu())

{

case STATE.Rectangle:

a1 = InputVal("Введите высоту прямоугольника:");

b1 = InputVal("Введите ширину прямоугольника:");

obj = new Rectangle(a1, b1);

obj.Print();

break;

case STATE.Square:

a1 = InputVal("Введите высоту квадрата:");

obj = new Quadrate(a1);

obj.Print();

break;

case STATE.Circle:

a1 = InputVal("Введите радиус окружности:");

obj = new Circle(a1);

obj.Print();

break;

default:

work = false;

break;

}

Console.WriteLine("Нажмите любую клавишу для продолжения ...");

Console.ReadKey();

Console.Clear();

}

}

}

interface IPrint

{

void Print();

}

abstract class GeometricFigure

{

public GeometricFigure() { }

public virtual double Square()

{

return 0;

}

public abstract override string ToString();

}

class Rectangle : GeometricFigure, IPrint

{

public Rectangle(double height1, double width1)

{

\_height = height1;

\_width = width1;

}

private double \_height = 0;

public double height

{

get { return \_height; }

set { \_height = value; }

}

private double \_width = 0;

public double width

{

get { return \_width; }

set { \_width = value; }

}

public override double Square()

{

return \_width \* \_height;

}

public override string ToString()

{

return "Rectangle: " + width.ToString() + "x" + height.ToString() + ", S = " + Square().ToString();

}

public void Print()

{

Console.WriteLine(this);

}

}

class Quadrate : Rectangle

{

public Quadrate(double height1) : base(height1, height1) { }

public override double Square()

{

return height \* height;

}

public override string ToString()

{

return "Square: " + height.ToString() + "x" + height.ToString() + ", S = " + Square().ToString();

}

}

class Circle : GeometricFigure, IPrint

{

public Circle(double radius)

{

\_radius = radius;

}

private double \_radius = 0;

public double radius

{

get { return \_radius; }

set { \_radius = value; }

}

public override double Square()

{

return Math.PI \* \_radius \* \_radius;

}

public override string ToString()

{

return "Circle: " + radius.ToString() + ", S = " + Square().ToString();

}

public void Print()

{

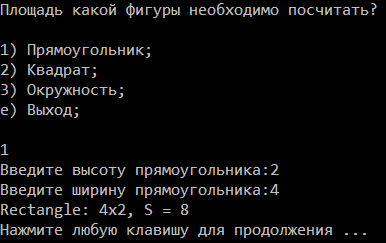
Console.WriteLine(this);

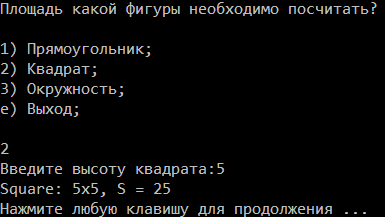
}

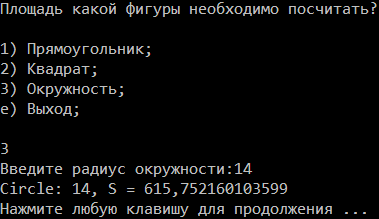
}

}

**Пример:**

****

****

****

**Диаграмма классов:**

****