***Отчет по лабораторной работе № 2 по курсу С#***

7

(количество листов)

Студент группы ИУ5-34:

Тюлькина Нина

Подпись Дата

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

Руководитель: Гапанюк Ю.Е.

Подпись Дата

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

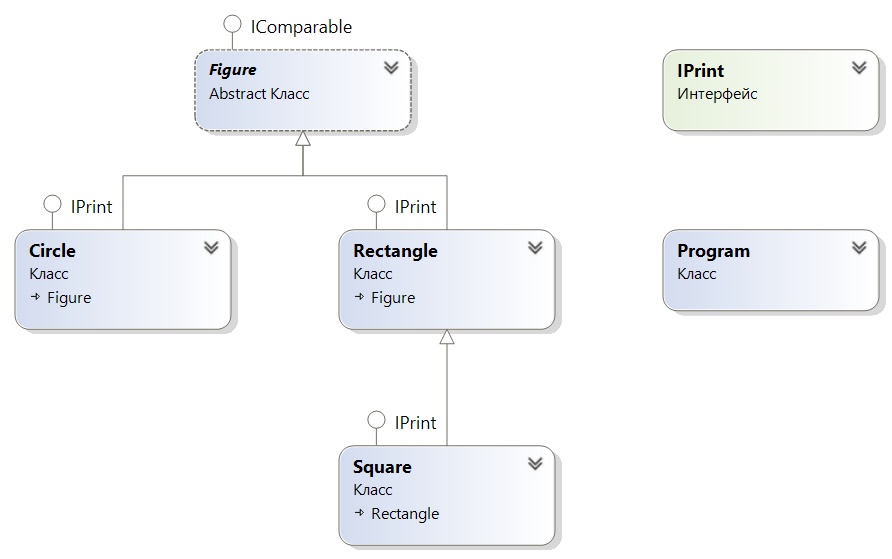
Москва, 2017

*Условие 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 «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;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2

{

class Program

{

static bool check(double value) { //проверка на +

if (value <= 0) {

Console.WriteLine("Вы ввели отрицательное значение ", value);

return false;

}

return true;

}

static int menu() {

string but\_str;

int but\_int;

bool flag;

Console.WriteLine("1)Прямоугольник");

Console.WriteLine("2)Квадрат");

Console.WriteLine("3)Круг");

Console.WriteLine("4)Я хочу выйти");

but\_str = Console.ReadLine();

flag = int.TryParse(but\_str, out but\_int);

if (!flag)

{

Console.WriteLine("Вы ввели последовательность символов");

}

return but\_int;

}

static void Main(string[] args)

{

bool f = true;

while (f)

{

Console.WriteLine("Площадь какой фигуры хотите посчитать?");

switch (menu())

{

case 1:

{

double h = 0, w = 0; //ph - высота, pw - ширина

double S;

Console.WriteLine("Введите высоту h и ширину w");

Console.Write("h = ");

try

{

h = Double.Parse(Console.ReadLine());

if (!check(h))

{

break;

}

Console.Write("l = ");

w = Double.Parse(Console.ReadLine());

if (!check(w))

{

break;

}

}

catch (Exception e) {

Console.WriteLine("Вы вводите не число!", e);

break;

}

FigureCollections.Rectangle rect = new FigureCollections.Rectangle(h, w);

S = rect.Area();

rect.Print();

break;

}

case 2:

{

double a; //size - сторона квадрата

double S;

Console.WriteLine("Введите длину стороны квадрата");

Console.Write("a = ");

try

{

a = Double.Parse(Console.ReadLine());

if (!check(a))

{

break;

}

}

catch (Exception e)

{

Console.WriteLine("Вы вводите не число!", e);

break;

}

FigureCollections.Square sq = new FigureCollections.Square(a);

S = sq.Area();

sq.Print();

break;

}

case 3:

{

double r = 0; //pr - радиус

double S;

Console.WriteLine("Введите радиус r");

Console.Write("r = ");

try

{

r = Double.Parse(Console.ReadLine());

if (!check(r))

{

break;

}

}

catch (Exception e)

{

Console.WriteLine("Вы вводите не число!", e);

break;

}

FigureCollections.Circle cir = new FigureCollections.Circle(r);

S = cir.Area();

cir.Print();

break;

}

case 4:

{

f = false;

break;

}

default:

{

Console.WriteLine("Введите целое число от 1 до 4");

break;

}

}

Console.ReadLine();

Console.Clear();

}

}

}

}

namespace FigureCollections

{

abstract class Figure:IComparable //класс фигура

{

public string Type //тип фигуры

{

get

{

return this.\_Type;

}

protected set

{

this.\_Type = value;

}

}

string \_Type;

public abstract double Area(); //вычисление площади

public override string ToString() // Приведение к строке, переопределение метода Object

{

return this.Type + " площадью " + this.Area().ToString();

}

public int CompareTo(object obj) // Сравнение элементов (для сортировки списка)

{

Figure p = (Figure)obj;

if (this.Area() < p.Area()) return -1;

else if (this.Area() == p.Area()) return 0;

else return 1; //(this.Area() > p.Area())

}

}

}

namespace FigureCollections

{

interface IPrint

{

void Print();

}

}

namespace FigureCollections

{

class Rectangle : Figure, IPrint

{

double height; // Высота

double width; // Ширина

// Основной конструктор

public Rectangle(double ph, double pw) //ph - высота, pw - ширина

{

this.height = ph;

this.width = pw;

this.Type = "Прямоугольник";

}

public override double Area() // Вычисление площади

{

double Result = this.width \* this.height;

return Result;

}

public void Print() //вывод

{

Console.WriteLine(this.ToString());

}

}

}

namespace FigureCollections

{

class Square : Rectangle, IPrint

{

public Square(double size)

: base(size, size)

{

this.Type = "Квадрат";

}

}

}

namespace FigureCollections

{

class Circle : Figure, IPrint

{

double radius; // Ширина

public Circle(double pr) //pr - радиус

{

this.radius = pr;

this.Type = "Круг";

}

public override double Area()

{

double Result = Math.PI \* this.radius \* this.radius;

return Result;

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

}

*Вывод*

