**Московский Государственный Технический Университет имени Н. Э. Баумана**



**Базовые компоненты интернет-технологий**

**Лабораторная работа №2**

ИСПОЛНИТЕЛЬ:

ФИО Арбузов А.П.

Группа РТ5-31

ПРЕПОДАВАТЕЛЬ:

ФИО Гапанюк Ю. Е.

**Москва 2017**

**Программа 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().

**Текст программы на языке С#.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

/// <summary>

/// Интерфейс IPrint предназначен для вывода информации о геометрической фигуре

/// </summary>

interface IPrint

{

void IPrint();

}

/// <summary>

/// Класс Фигура

/// </summary>

abstract class Figure ///Объявление метода как абстрактного автоматически делает его виртуальным, поэтому в наследуемом классе абстрактный метод должен переопределяться с ключевым словом override. стр.62

{

/// <summary>

/// Тип фигуры

/// </summary>

protected string FigureType { get; set; }

/// <summary>

/// Вычисление площади

/// </summary>

/// <returns> площадь фигуры</returns>

public abstract double Area();

/// <summary>

/// Переопределение метода

/// </summary>

/// <returns></returns>

public override string ToString()

{

return (this.FigureType + " has area " + this.Area().ToString());

}

}

/// Конструктор принимает один параметр и присваивает его переменной класса, доступ к которой выполняется с помощью ключевого слова this

#region Figures

/// <summary>

/// Класс Прямоугольник

/// </summary>

class Rectangle : Figure, IPrint

{

/// <summary>

/// Ширина

/// </summary>

protected double Width { get; set; }

/// <summary>

/// Высота

/// </summary>

protected double Length { get; set; }

/// <summary>

/// Конструктор

/// </summary>

/// <param name="width">ширина</param>

/// <param name="length">высота</param>

public Rectangle(double width, double length)

{

this.FigureType = "Rectangle";

this.Width = width;

this.Length = length;

}

/// <summary>

/// Вычисление площади

/// </summary>

/// <returns> площадь прямоугольника</returns>

public override double Area()

{

return (this.Width \* this.Length);

}

/// <summary>

/// Информация о фигуре

/// </summary>

/// <returns>информация</returns>

public override string ToString()

{

return (this.FigureType + " (width = " + Width.ToString() + ", length = " + Length.ToString() + ") has area " + this.Area().ToString());

}

/// <summary>

/// Вывод ToString()

/// </summary>

public void IPrint()

{

Console.WriteLine(this.ToString());

}

}

/// <summary>

/// Класс Квадрат

/// </summary>

class Square : Rectangle, IPrint

{

/// <summary>

/// Конструктор

/// </summary>

/// <param name="length">длина</param>

public Square(double length)

: base(length, length)

{

this.FigureType = "Square";

this.Length = length;

}

/// <summary>

/// Вычисление площади

/// </summary>

/// <returns> площадь квадрата </returns>

public override double Area()

{

return (this.Length \* this.Length);

}

/// <summary>

/// Информация о фигуре

/// </summary>

/// <returns></returns>

public override string ToString()

{

return (this.FigureType + " (lenght = " + Length.ToString() + ") has area " + this.Area().ToString());

}

}

/// <summary>

/// Класс Круг

/// </summary>

class Cirle : Figure, IPrint

{

/// <summary>

/// Радиус

/// </summary>

private double Radius { get; set; }

/// <summary>

/// Конструктор принимает один параметр и присваивает его переменной класса, доступ к которой выполняется с помощью ключевого слова this

/// </summary>

/// <param name="radius">радиус</param>

public Cirle(double radius)

{

this.FigureType = "Circle";

this.Radius = radius;

}

/// <summary>

/// Площадь круга

/// </summary>

/// <returns></returns>

public override double Area()

{

return Math.PI \* this.Radius \* this.Radius;

}

/// <summary>

/// Переопределение метода

/// </summary>

/// <returns></returns>

public override string ToString()

{

return (this.FigureType + " (radius = " + Radius.ToString() + ") has area " + this.Area().ToString());

}

/// <summary>

/// Вывод информации

/// </summary>

public void IPrint()

{

Console.WriteLine(this.ToString());

}

}

#endregion

class Program

{

static void Main(string[] args)

{

int id; // тип фигуры

Console.WriteLine("1 - Rectangle");

Console.WriteLine("2 - Square");

Console.WriteLine("3 - Circle");

Console.Write("Your choise is : ");

id = Convert.ToInt32(Console.ReadLine());

while ((id < 1) || (id > 3)) // проверка на правильность выбора фигуры

{

Console.WriteLine("\n!You entered wrong data! Please, try again.");

Console.Write("Your choise is : ");

id = Convert.ToInt32(Console.ReadLine()); // применении класса Convert, который содержит большое количество статических методов и может преобразовать значения большинства базовых между собой. стр.20

}

Console.Clear();

switch (id)

{

#region Rectangle

case 1:

double enterWidthRec, enterLengthRec; //входные параметры, которые мы будем считывать с консоли

Console.Write("Enter rectangle's width: ");

enterWidthRec = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter rectangle's length: ");

enterLengthRec = Convert.ToDouble(Console.ReadLine());

Rectangle obj\_rectangle = new Rectangle(enterWidthRec, enterLengthRec);

obj\_rectangle.IPrint(); // вывод получившихся параметров

break; // Аналогично case 2 и case 3

#endregion

#region Square

case 2:

double enterLengthSquare;

Console.Write("Enter square's length: ");

enterLengthSquare = Convert.ToDouble(Console.ReadLine());

Square obj\_square = new Square(enterLengthSquare);

obj\_square.IPrint();

break;

#endregion

#region Circle

case 3:

double enterRadius;

Console.WriteLine("Enter circle's radius: ");

enterRadius = Convert.ToDouble(Console.ReadLine());

Cirle obj\_circle = new Cirle(enterRadius);

obj\_circle.IPrint();

break;

#endregion

default:

Console.WriteLine("\n!You entered wrong data!\nPlease, try again.");

break;

}

Console.Write("\n\nEnter any key for continue...");

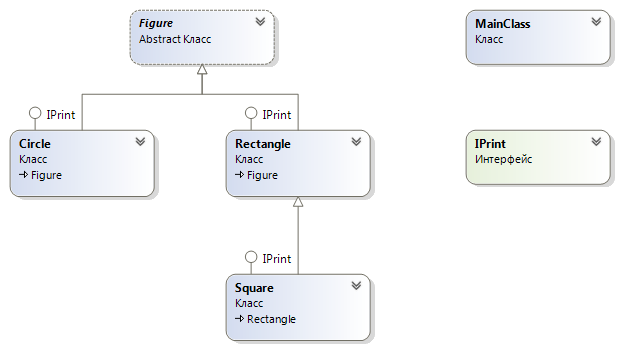
Console.ReadKey();

}

}

}

**Диаграмма классов:**



**Результаты выполнения программы:**

