МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский государственный университет им. А.Н. Косыгина   
(Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе № 2

по дисциплине «Технологии программирования»

Тема: «Работа с классами»

Выполнил: Стычинский С.В. ,МВИ-121

Проверили: Кузьмина Т. М. и Адаев Р. Б.

Москва 2023

Цель работы

Познакомится с понятиями класс, объект класса, научится

создавать свои классы и работать с ними

Задание

Класс, описывающий окружность. Класс должен иметь методы, решающие следующие задачи:

a. расчет длины окружности;

b. расчет площади окружности

(Задание 4)

Теоретический материал

Классы Язык C# позволяет разработчику определить новый тип данных – класс. Простейшее описание класса имеет следующий вид. class имя класса {< список членов класса>} имя класса – это любой идентификатор, уникальный в пределах своей сферы действия; список членов класса – список, объявляющий члены класса с указанием уровня доступа. Члены класса – это поля (члены-данные), методы (функции-члены), свойства и события. В данной лабораторной работе будут создаваться классы, имеющие в качестве членов поля и методы.

Например:

class Myclass {

public int i; /\* слово public обозначает, что данный член класса является общедоступным\*/

public float x;

public float f()

{

return i\*y;

}

}

Во всех объектно-ориентированных языках четко различаются понятия «класс» и «объект». Класс – это определяемый пользователем тип данных, его можно представить себе как чертеж, по которому будут строиться сами данные, т.е. объекты (экземпляры) класса. 10 В C# создание объектов класса происходит с помощью операции new.

Например: Myclass a1=new Myclass(); или Myclass a2; a2=new Myclass(); Ключевое слово new означает, что среде выполнения следует вызвать конструктор класса и выделить необходимое количество оперативной памяти под экземпляр класса. Поскольку класс относится к ссылочному типу, то выделении памяти производится из «кучи», находящейся в распоряжении среды выполнения .NET . Команды Myclass a1, Myclass a2, определяют ссылки на объекты класса, а не сами объекты, объекты создаются с помощью операции new . Но после того как объекты будут созданы, мы будем их называть объект (экземпляр) a1, и объект (экземпляр) a2. Рассмотрим еще один пример: Myclass c=new Myclass(); Myclass d; d=c; В этом примере создан один объект и две ссылки на него. Доступ к членам класса осуществляется с помощью операции точка «.». Например: c.x=7; //полю х объекта с присвоено значение 7. d.i=6; //полю i объекта d присвоено значение 6. a1.x=c.f();/\*полю х объекта a1 присвоено значение, которое возвращает метод f, вызванный для объекта с, поскольку ссылка d также указывает на этот же объект, то метод f вернет значение 42.\*/ Приведение типов Автоматическое приведение типов в языке C# происходит только в том случае, когда не происходит потери точности.

Например. int i=9; float x=i; Если в результате приведения одного типа к другому может произойти потеря точности, то его программист должен выполнить сам. Для явного приведения типов перед переменной в круглых скобках пишется имя нового типа, например (int)i. Пример:

float i=9; int l=(int)i; i=l;

//в этом случае потеря точности не грозит, и приведение типов

//выполняется автоматически.

Многие функции преобразования (приведения) типов включены в библиотечный класс Convert. Например. int i= Convert.ToInt32(textBox1.Text); В этом примере, текстовая строка, записанная в поле ввода ( textBox1.Text ) будет преобразована в значение типа int. Если в textBox1.Text записана строка, которую нельзя прочитать как целое число, то будет сгенерировано исключение. В языке C# , стандартные числовые типы, такие как int, float т.д., имеют статические функции-члены (методы), которые также решают задачи преобразования типов, например: int i= int.Parse(textBox1.Text); Для того чтобы преобразовать число в строку, лучше использовать метод ToString, определенный во всех стандартных типах.

Например: textBox1.Text=i.ToString(); Хотя тоже можно использовать методы из класса Convert. Математические функции 12 При решении некоторых задач, могут потребоваться функции извлечения корня или возведения в степень. Математические функции хранятся в библиотечном классе Math. Если нам надо вычислить sin x, то на языке C# это будет выглядеть следующим образом: Math.Sin(x); Отметим, что возвращаемые значения математических функций, как правило имеют тип double. Если мы работаем с переменными типа float, то надо будет выполнить явное приведение типов. Например: float u= (float) Math.Sin(x);

Этапы выполнения лабораторной работы

1. Продумать структуру класса, предназначенного для решения задач, определенных вариантом индивидуального задания. Имена класса и его членов должны быть информативны. В классе не должно быть полей, в которые пользователь мог бы поместить противоречивые данные. Например, в классе «Окружность» не могут быть одновременно поля «Радиус» и «Диаметр», надо выбрать одно из них, а второе заменить методом. Или, если рассматривается класс «Равнобокая\_трапеция», то набор полей «Основание1», «Основание2», «Боковая\_сторона» может получить значения, по которым не построишь трапецию (например, 5, 105,10). Но если поле «Боковая\_сторона» заменить на поле «Высота», то проблем не будет.

2. Описать класс. Описание класса должно располагаться рядом с описанием класса формы. Хотя язык C# позволяет использовать кириллицу для записи идентификаторов, рекомендуется использовать латиницу.

3. Поскольку значения полей класса должен задавать пользователь, то на форму надо поместить столько полей ввода (компонентов TextBox), сколько полей в классе.

4. (Создание локального объекта, т.е. объекта с которым может работать только один метод). Поместить на форму кнопку. Обработчик события Click должен создать объект Вашего класса, считать информацию из полей ввода 13 и поместить ее в поля класса, вызвать методы, выполняющие необходимые вычисления и вывести на экран возвращенные ими значения.

5. (Создание объектов класса, с которыми могут работать разные методы класса главной формы). Создать объект класса, который является полем главной формы. Поместить на форму две кнопки, которые будут работать с вновь созданным объектом, первая кнопка должна вводить информацию (значения полей), вторая выводить на форму (и значения полей, и расчетные данные).

6. Отладить программу.

Распечатка программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab2\_arch

{

public partial class Form1 : Form

{

okr o1 = new okr(10); // поле главной формы

public Form1()

{

InitializeComponent();

}

// кнопка ввод

private void button1\_Click(object sender, EventArgs e)

{

o1.R = (double)int.Parse(textBox1.Text);

button1.Text = o1.R.ToString();

}

// кнопка вывод

private void button2\_Click(object sender, EventArgs e)

{

textBox2.Text = o1.DlinaOkr().ToString();

textBox3.Text = o1.Sokr().ToString();

}

// кнопка расчет

private void button3\_Click(object sender, EventArgs e)

{

double r = (double)int.Parse(textBox1.Text);

okr b = new okr(r);

button3.Text = "Длина окружности:\n" + b.DlinaOkr().ToString() + "\nПлощадь окружности:\n" + b.Sokr().ToString();

}

class okr

{

public double R;

public okr(double r)

{

this.R = r;

}

public double DlinaOkr()

{

return 2 \* Math.PI \* this.R;

}

public double Sokr()

{

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

}

public void setR(double r)

{

this.R = r;

}

public double getR()

{

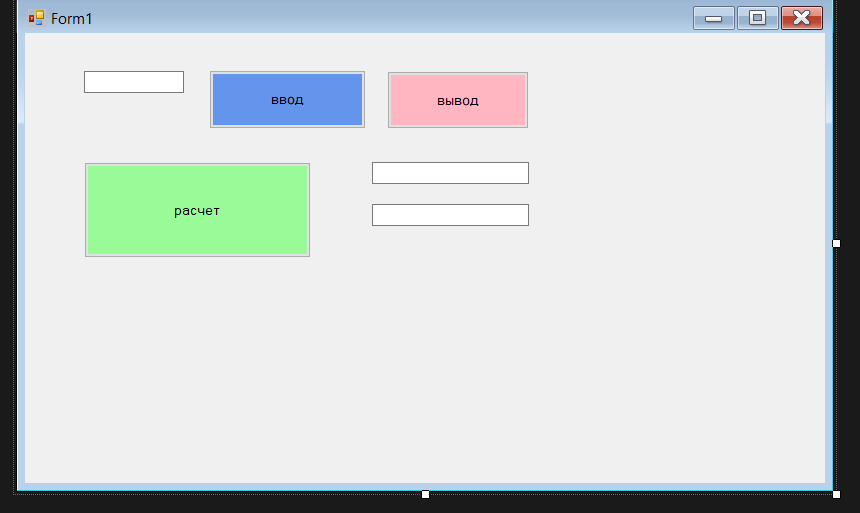
return this.R;

}

}

}

}



Результат работы программы

