МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе № 8

на тему: «Создание собственного com объекта»

по дисциплине: «Проектирование и архитектура программных систем»

Выполнили: Кожухова О.А., Карпикова С.П., Макеева Д.С.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Ужаринский А.Ю., Константинов И.С.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орел, 2019 г.

**Задание на лабораторную работу.**

Создать COM сервер и клиент, использующий его методы, реализующий функции согласно варианту задания.

**Выполнение работы.**

Листинг файла form.cs:

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;

using Lab8;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Lab8cl obj = new Lab8cl();

public Form1()

{

InitializeComponent();

}

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

{

rad.Text = obj.GetRad(Convert.ToDouble(gr.Text)).ToString();

}

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

{

fact.Text = obj.GetFact(Convert.ToInt32(num.Text)).ToString();

}

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

{

roots.Text = "";

List<double> result = obj.GetRoot(Convert.ToDouble(a.Text), Convert.ToDouble(b.Text), Convert.ToDouble(c.Text));

for (int i = 0; i < result.Count; i++)

{

roots.Text += "x" + (i + 1).ToString() + " = ";

roots.Text += result.ElementAt(i).ToString() + " ";

}

if (roots.Text == "")

roots.Text = "Корней нет";

}

}

}

Листинг файла class.cs:

using System;

using System.Text;

using System.Runtime.InteropServices;

using System.Collections.Generic;

namespace Lab8

{

[Guid("02145619-49F9-4B06-9CCA-6A31292541E3")]

internal interface ILab8cl

{

[DispId(1)]

//4. описываем методы которые можно будет вызывать из вне

double GetRad(double gr);

[DispId(2)]

int GetFact(int cnt);

[DispId(3)]

List<double> GetRoot(double a, double b, double c);

}

[Guid("ACFF6853-B772-4B60-AAAE-1742B8794967"), InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]

public interface IMyEvents

{

}

[Guid("5EC817D0-46A1-4F2F-B9C8-729F28B2D947"), ClassInterface(ClassInterfaceType.None), ComSourceInterfaces(typeof(IMyEvents))]

public class Lab8cl : ILab8cl

{

public Lab8cl() { }

public double GetRad(double gr)

{

return Math.Round(gr \* 3.14 / 180, 5);

}

public int GetFact(int cnt)

{

int result = 1;

for (int i = 1; i <= cnt; i++)

{

result = result \* i;

}

return result;

}

public List<double> GetRoot(double a, double b, double c)

{

List<double> result = new List<double>();

double D = b \* b - 4 \* a \* c;

if (D > 0)

{

double x1 = (-b - Math.Sqrt(D)) / 2 \* a;

double x2 = (-b + Math.Sqrt(D)) / 2 \* a;

result.Add(x1);

result.Add(x2);

}

if (D == 0)

{

double x = -b / (2 \* a);

result.Add(x);

}

return result;

}

}

}

**Контрольные вопросы.**

*1. Опишите процедуру создания COM объекта.*

Можно выделить 4 основные фазы создания объекта COM.

1. Клиент вызывает функцию библиотеки COM CoCreatelnstance.

2. Библиотека СОМ находит в системном реестре запись, соответствующую классу данного объекта.

3. Библиотека COM запускает сервер COM и возвращает клиенту указатель на требуемый интерфейс.

4. Клиент вызывает требуемый метод..

*2. Что такое интерфейс СОМ объекта, его назначение?*

Каждый интерфейс объединяет методы объекта, которые обеспечивают доступ к свойствам (данным) и выполнение операций. Обычно в интерфейсе объединяются все методы, выполняющие операции одного типа или работающие с однородными свойствами.

Клиент получает доступ к службам объекта только через интерфейс и его методы. Этот механизм является ключевым. Согласно спецификации COM, уже созданный интерфейс не может быть изменен ни при каких обстоятельствах.

Любой COM-объект в обязательном порядке должен поддерживать интерфейс с именем IUnknown, обеспечивающий два базовых свойства COM-объектов: подсчет обращений и способность запрашивать другие интерфейсы. При помощи интерфейса IUnknown можно определить, какие еще интересующие вас интерфейсы поддерживаются объектом. Интерфейс IUnknown является базовым для всех остальных COM-интерфейсов, так что при наличии указателя на любой интерфейс можно вызвать QueryInterface для любого интерфейса. Все имена COM-интерфейсов начинаются с префикса I, по которому их можно отличить от классов C++ или других объектов.

*3. Назначение библиотеки СОМ. Механизм создания первого экземпляра объекта с помощью библиотеки СОМ и системного реестра.*

Для обеспечения выполнения общих функций и базовых интерфейсов в ОС устанавливается специальная библиотека COM (конкретная реализация может быть различной). Доступ к функциям библиотеки осуществляется стандартным способом, а не через интерфейс. Согласно спецификации, имена всех библиотечных функций начинаются с приставки «Со». При установке использующего COM приложения в системный реестр записывается информация обо всех используемых им объектах COM:

- Идентификатор класса (Class Identifier, CLSID), который однозначно определяет класс объекта;

- Тип сервера объекта – внутренний, локальный или удаленный;

- Для локальных и внутренних серверов сохраняется полное имя динамической библиотеки или исполняемого файла;

- Для удаленных серверов записывается полный сетевой адрес.

Предположим, что клиент пытается использовать некоторый объект COM, который до этого момента не использовался. Следовательно, клиент не может получить указатели на объект и интерфейс. В этом случае он обращается к библиотеке COM и вызывает метод CoCreateInstance, передавая ей в качестве параметра CLSID нужного класса, IID интерфейса и требуемый тип сервера. Библиотека при помощи диспетчера управления службами (Service Control Manager, SCM) обращается к системному реестру, по идентификатору класса – объект, и возвращает библиотеке указатель на запрошенный интерфейс. Библиотека COM передает управление клиенту, который в последствии может обращаться непосредственно к объекту.

*4. Что такое СОМ-сервер?*

СОМ-сервер представляет собой приложение или библиотеку, которая предоставляет услуги приложению-клиенту или библиотеке. СОМ-сервер содержит один или более СОМ-объектов, где СОМ-объекты выступают в качестве наборов свойств, методов и интерфейсов.

Клиенты не знают как СОМ-объект выполняет свои действия. СОМ-объект предоставляет свои услуги при помощи интерфейсов., В дополнение, приложению-клиенту не нужно знать, где находится СОМ-объект. Технология СОМ обеспечивает прозрачный доступ независимо от местонахождения СОМ-объекта.

Когда клиент запрашивает услугу от СОМ-объекта, он передает СОМ-объекту идентификатор класса (CLSID). CLSID - всего лишь GUID, который применяется при обращении к СОМ-объекту. После передачи CLSID, СОМ-сервер должен обеспечить так называемую фабрику класса (см. следующий раздел), которая создает экземпляры СОМ-объектов.

В общих чертах, СОМ-сервер должен выполнять следующее:

- регистрировать данные в системном реестре Windows для связывания модуля сервера с идентификатором класса (CLSID);

- предоставлять фабрику СОМ-класса, создающую экземпляры СОМ-объектов;

- обеспечивать механизм, который выгружает из памяти серверы СОМ, которые в текущий момент времени не предоставляют услуг клиентам.