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: Марочкин М.А. Шифр: 170584   
 Шорин В.Д. Шифр: 171406  
 Щекотихин С.Е. Шифр: 170590  
ИПАИТ  
Направление: 09.03.04 «Программная инженерия»  
Группа: 71-ПГ  
Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
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.

Вариант 21.

**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.

**Листинг программы**

**«COMLib.cs»**

using System;

using System.Runtime.InteropServices;

namespace COMServer {

[Guid("E5B23F96-411D-4A67-AFBA-B3DF6B3B0ACB")]

[ComVisible(true)]

public interface COMInterface {

string Ping();

double Min(double a, double b, double c);

double Distance(double x1, double y1, double z1, double x2, double y2, double z2);

int Randomize(int x);

}

[Guid("8C034F6A-1D3F-4DB8-BC99-B73873D8C297")]

[ClassInterface(ClassInterfaceType.None)]

[ComVisible(true)]

public class COMServer : COMInterface {

public string Ping() {

return "Pong";

}

public double Min(double a, double b, double c) {

return a < b ? a < c ? a : c : b < c ? b : c;

}

public double Distance(double x1, double y1, double z1, double x2, double y2, double z2) {

return Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2) + Math.Pow(z2 - z1, 2));

}

public int Randomize(int x) {

Random r = new Random();

return r.Next(-x, 2 \* x);

}

}

}

**«Program.cs»**

using System;

namespace COMClient {

internal class Program {

public static void Main(string[] args) {

COMServer.COMServer com = new COMServer.COMServer();

Console.WriteLine("Ping: " + com.Ping());

Console.WriteLine("Random: " + com.Randomize(10));

Console.WriteLine("Distance: " + com.Distance(1, 1, 1, 0, 0, 0));

Console.WriteLine("Min: " + com.Min(55, 22, 13));

}

}

}