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 № 7

на тему: «Использование com объектов»

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.**

Разработать приложение для автоматического формирования отчёта по лабораторной работе на основе заданного шаблона и введённых пользователем данных. Результатом работы программы должен быть документ word в который в заданные места на основе шаблона отчёта подставлены данные пользователя. Взаимодействие с word организовать через com объекты.

**Выполнение работы.**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Microsoft.Office.Interop.Word;

using Microsoft.Office.Core;

using System.Reflection;

using Word = Microsoft.Office.Interop.Word;

using System.IO;

using System.Diagnostics;

using System.Drawing.Drawing2D;

namespace Word

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void FindAndReplace(Microsoft.Office.Interop.Word.Application wordApp, object findText, object replaceWithText)

{

object matchCase = true;

object matchWholeWord = true;

object matchWildCards = false;

object matchSoundLike = false;

object nmatchAllForms = false;

object forward = true;

object format = false;

object matchKashida = false;

object matchDiactitics = false;

object matchAlefHamza = false;

object matchControl = false;

object read\_only = false;

object visible = true;

object replace = 2;

object wrap = 1;

wordApp.Selection.Find.Execute(ref findText,

ref matchCase, ref matchWholeWord,

ref matchWildCards, ref matchSoundLike,

ref nmatchAllForms, ref forward,

ref wrap, ref format, ref replaceWithText,

ref replace, ref matchKashida,

ref matchDiactitics, ref matchAlefHamza,

ref matchControl);

}

private void CreateWordDocument(object filename, object savaAs)

{

List<int> processesbeforegen = getRunningProcesses();

object missing = Missing.Value;

Microsoft.Office.Interop.Word.Application wordApp = new Microsoft.Office.Interop.Word.Application();

Microsoft.Office.Interop.Word.Document aDoc = null;

if (File.Exists((string)filename))

{

object readOnly = false; //default

object isVisible = false;

wordApp.Visible = false;

aDoc = wordApp.Documents.Open(ref filename, ref missing, ref readOnly,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing, ref missing);

aDoc.Activate();

//Find and replace:

this.FindAndReplace(wordApp, "<num>", num.Text);

this.FindAndReplace(wordApp, "<students>", students.Text);

this.FindAndReplace(wordApp, "<title>", title.Text);

this.FindAndReplace(wordApp, "<sub>", sub.Text);

this.FindAndReplace(wordApp, "<teachers>", teachers.Text);

this.FindAndReplace(wordApp, "<task>", task.Text);

this.FindAndReplace(wordApp, "<work>", work.Text);

this.FindAndReplace(wordApp, "<questions>", questions.Text);

}

else

{

MessageBox.Show("file dose not exist.");

return;

}

//Save as: filename

aDoc.SaveAs2(ref savaAs, ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing,

ref missing, ref missing, ref missing);

List<int> processesaftergen = getRunningProcesses();

killProcesses(processesbeforegen, processesaftergen);

DialogResult result = MessageBox.Show("Отчёт создан", "Сформировать новый отчёт?", MessageBoxButtons.YesNo,MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

if (result == DialogResult.Yes)

{

num.Clear();

students.Clear();

title.Clear();

sub.Clear();

teachers.Clear();

task.Clear();

work.Clear();

questions.Clear();

}

}

public List<int> getRunningProcesses()

{

List<int> ProcessIDs = new List<int>();

//here we're going to get a list of all running processes on

//the computer

foreach (Process clsProcess in Process.GetProcesses())

{

if (Process.GetCurrentProcess().Id == clsProcess.Id)

continue;

if (clsProcess.ProcessName.Contains("WINWORD"))

{

ProcessIDs.Add(clsProcess.Id);

}

}

return ProcessIDs;

}

private void killProcesses(List<int> processesbeforegen, List<int> processesaftergen)

{

foreach (int pidafter in processesaftergen)

{

bool processfound = false;

foreach (int pidbefore in processesbeforegen)

{

if (pidafter == pidbefore)

{

processfound = true;

}

}

if (processfound == false)

{

Process clsProcess = Process.GetProcessById(pidafter);

clsProcess.Kill();

}

}

}

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

{

if (SaveDoc.ShowDialog() == DialogResult.OK)

{

CreateWordDocument(@"C:\Users\user\Desktop\Универ\Лабораторные\Лабы С#\Word\temp.docx", SaveDoc.FileName);

}

}

}

}

**Контрольные вопросы.**

*1. Что такое объект COM?*

COM (Component Object Model) - это метод разработки программных компонентов, небольших двоичных исполняемых файлов, которые предоставляют необходимые сервисы приложениям, операционным системам и другим компонентам. Другими словами, COM определяет стандартный механизм, с помощью которого одна часть программного обеспечения предоставляет свои сервисы другой независимо от способа их реализации.

Объект СОМ – это некоторая сущность, имеющая состояние и методы доступа, позволяющие изменять это состояние. СОМ-объекты можно создавать прямым вызовом специальных функций, но напрямую уничтожить его невозможно. Вместо прямого уничтожения используется механизм самоуничтожения, основанный на подсчете ссылок. В COM присутствует понятие класса. Класс в COM носит название CoClass. Согласно правилам обозначения объектов COM, базовый интерфейс IUnknown, который имеется у любого объекта, обозначается как кружок, примыкающий к верхней стороне прямоугольника объекта. Остальные интерфейсы обозначаются справа и слева.

*2. Чем отличаются СОМ объекты от обычных объектов?*

Объект COM может иметь любое число интерфейсов (если это число больше нуля), причем, каждый интерфейс обладает собственным указателем. Это первое отличие объектов COM от обычных.

У объектов COM имеется особенность еще в одном объектом механизме – наследовании. Вообще различают два способа наследования. Наследование реализации подразумевает передачу родителем потомку всего программного кода. Наследование интерфейса означает передачу только объявления методов, их программный код потомок должен предоставить самостоятельно. Объекты COM поддерживают только наследование интерфейса, избегая тем самым возможного нарушения инкапсуляции родителя. Тем не менее, просто так выбросить наследование реализации нельзя. Вместо нее объекты COM используют механизм включения, т.е. при необходимости потомок вызывает нужный метод родителя. Также применяется механизм агрегирования, когда один или несколько интерфейсов одного объекта на время включаются в другой объект путем передачи указателей.

*3. Объясните основные принципы построения архитектуры приложения на основе com объектов.*

В технологии COM приложение предоставляет для использования своей службы, применяя для этого объекты COM. Одно приложение содержит как минимум один объект. Каждый объект имеет один или несколько интерфейсов. 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. Это гарантирует постоянную работоспособность приложений на основе COM, невзирая на любые модернизации.

Объект всегда работает в составе сервера COM, который может быть динамической библиотекой или исполняемым файлом. Объект может иметь собственные свойства и методы или использовать данные и службы сервера.

Для доступа к методам объекта клиент должен получить указатель на соответствующий интерфейс после чего клиент может использовать службы объекта, просто вызывая его методы. Доступ к свойствам объектов осуществляется только через его методы.

Взаимодействие между клиентом и объектом обеспечивается базовыми механизмами COM. При этом от клиента скрыто, где именно расположен объект: в адресном пространстве того же процесса, в другом процессе или на другом компьютере. Механизм обеспечения взаимодействия между удаленными элементами COM называется маршалингом (marshalling).

*4. Расскажите о порядке работы с com-объектами.*

Любой объект COM является обычным экземпляром некоторого класса, описывающего его свойства и методы. Информация обо всех зарегистрированных и доступных в данной ОС классах COM собрана в специальной библиотеке COM, которая используется для запуска экземпляра класса – объекта.

Сначала объект обращается к библиотеке COM, передавая ей имя требуемого класса и необходимого в первую очередь интерфейса. Библиотека находит нужный класс и сначала запускает сервер, который затем создает объект – экземпляр класса. После этого библиотека возвращает клиенту указатели на объект и интерфейс. В последующей работе клиент может обращаться непосредственно к объекту и его интерфейсам.

После создания наступает очередь инициализации – объект должен загрузить необходимые данные, считать настройки из системного реестра и т.д. За это отвечают специальные объекты COM, которые называются моникерами (monikers). Они работают скрытно от клиента. Обычно моникер создается вместе с классом.

Чтобы работать с COM-объектом необходимо создать объект и получить информацию о нем. 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.

*5. Опишите преимущества и недостатки архитектуры, основанной на com объектах.*

Преимущества:

1. COM предоставляет стандартный набор функций для доступа к провайдеру сервиса (COM-серверу), получения информации о предоставляемых им сервисах и вызова требуемого сервиса. В качестве COM-сервера может выступать операционная система или приложение.

2. COM использует объектно-ориентированные концепции для обеспечения модульности при построении сложных распределенных систем, а также для повторного использования готовых компонентов и их разработки с сохранением совместимости с предыдущими версиями.

3. COM реализует модель вычислений "клиент-сервер", что обеспечивает преимущества распределенной обработки данных.

4. COM обеспечивает вызов сервисов в сетевом окружении, независимо от расположения COM-сервера.

Недостатки:

Первое: компоненты COM могут быть трудными для кодирования.

Второе: компоненты COM могут оказаться трудными для развертывания. Разработчики COM серверных компонентов предполагали обеспечить совместимость новых версий компонентов с более старыми, но не всегда это удавалось, поэтому установка нового приложения, которое ссылается на новую версию компонента COM, может внезапно дать отказ существующих приложений.