|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  **«Московский технологический университет»**  **МИРЭА** | | |

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

**ОТЧЕТ**

по Лабораторной Работе №2

на тему

«COM»

по дисциплине

«Технологии программирования»

Выполнил студент группы ИСБО-11-16 Шайхуллин С.В.

Принял Cтарший преподаватель Мирзоян Д.И.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_2017 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2017г.

Москва, 2017

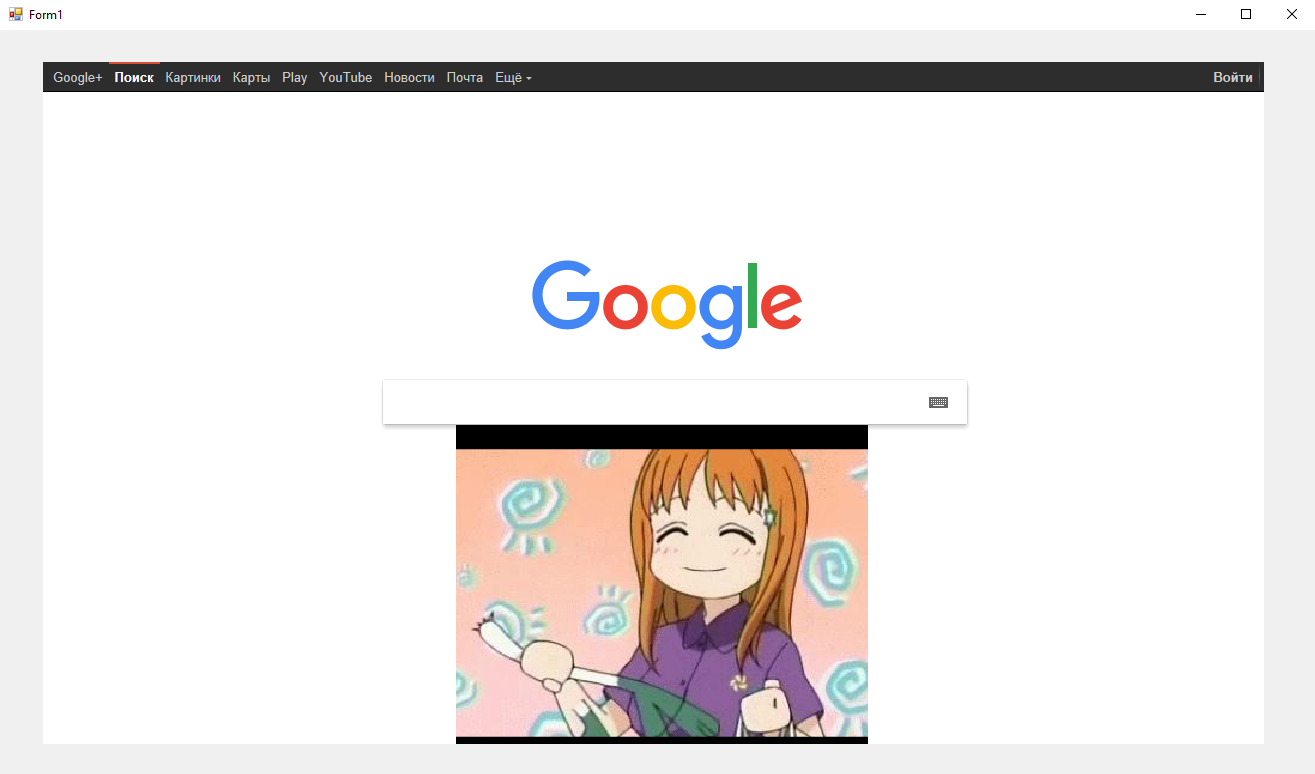
**Теоретическая часть**

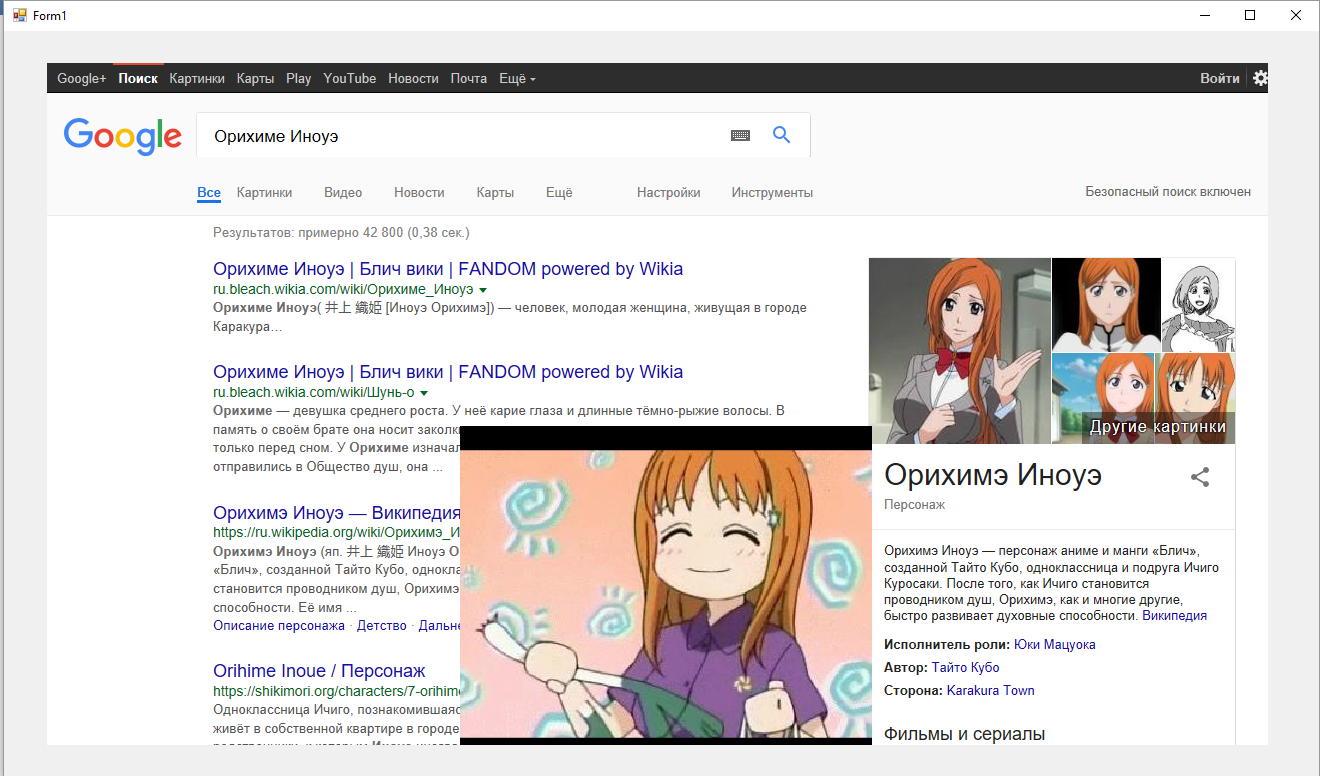
COM (Componen Object Model) — модель объектных компонентов, идеология программирования сложных приложений, предоставленная Microsoft и работающая на операционных системах этого производителя. Основой концепции COM является идея, что каждое приложение состоит из набора взаимодействующих компонентов, для каждого из которых может существовать несколько совместимых вариантов реализации. Само приложение также является COM компонентом. В некоторых источниках эта технология отмечается как устаревшая, хотя именно на ее основе построены такие широко используемые технологии как OLE, DirectX и .NET.

**Задание**

Создать приложение, интегрирующее в свое окно движок браузера Internet Explorer используя COM.

**Пример работы**

****



**Заключение**

В данной лабораторной работе мы научились работать с COM объектами, а также интегрировать готовые модули, используя IDE

**Исходный код**

using System;

using System.Runtime.InteropServices;

using System.Runtime.InteropServices.ComTypes;

namespace Lab2

{

public class ClientSite : IOleClientSite, IOleInPlaceSite

{

private readonly tagRECT rectForm;

private readonly IntPtr groupBoxHandle;

public ClientSite(IntPtr handle, int l, int t, int r, int b)

{

groupBoxHandle = handle;

rectForm = new tagRECT(l, t, r, b);

}

[return: MarshalAs(UnmanagedType.I4)]

public int SaveObject()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int GetMoniker([In, MarshalAs(UnmanagedType.U4)] uint dwAssign, [In, MarshalAs(UnmanagedType.U4)] uint dwWhichMoniker, [MarshalAs(UnmanagedType.Interface), Out] out IMoniker ppmk)

{

ppmk = null;

return HRESULT.E\_FAIL;

}

[return: MarshalAs(UnmanagedType.I4)]

public int GetContainer([MarshalAs(UnmanagedType.Interface), Out] out IOleContainer ppContainer)

{

ppContainer = null;

return HRESULT.E\_NOINTERFACE;

}

[return: MarshalAs(UnmanagedType.I4)]

public int ShowObject()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnShowWindow([In, MarshalAs(UnmanagedType.Bool)] bool fShow)

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int RequestNewObjectLayout()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int GetWindow([In, Out] ref IntPtr phwnd)

{

phwnd = groupBoxHandle;

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int ContextSensitiveHelp([In, MarshalAs(UnmanagedType.Bool)] bool fEnterMode)

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int CanInPlaceActivate()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnInPlaceActivate()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnUIActivate()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int GetWindowContext([MarshalAs(UnmanagedType.Interface), Out] out IOleInPlaceFrame ppFrame, [MarshalAs(UnmanagedType.Interface), Out] out IOleInPlaceUIWindow ppDoc, [In, MarshalAs(UnmanagedType.Struct), Out] ref tagRECT lprcPosRect, [In, MarshalAs(UnmanagedType.Struct), Out] ref tagRECT lprcClipRect, [In, MarshalAs(UnmanagedType.Struct), Out] ref tagOIFI lpFrameInfo)

{

ppFrame = null;

ppDoc = null;

lprcPosRect = rectForm;

lprcClipRect = rectForm;

return HRESULT.S\_OK;

}

int IOleInPlaceSite.Scroll(ref tagSIZE scrollExtent)

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnUIDeactivate([In, MarshalAs(UnmanagedType.Bool)] bool fUndoable)

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnInPlaceDeactivate()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int DiscardUndoState()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int DeactivateAndUndo()

{

return HRESULT.S\_OK;

}

[return: MarshalAs(UnmanagedType.I4)]

public int OnPosRectChange([In, MarshalAs(UnmanagedType.Struct)] ref tagRECT lprcPosRect)

{

lprcPosRect = rectForm;

return HRESULT.S\_OK;

}

}

}

using System;

using System.Windows.Forms;

using static Lab2.OLE;

using static Lab2.OLEDOVERB;

using static Lab2.OLE.CLSCTX;

namespace Lab2

{

public partial class MainWindow : Form

{

private readonly tagRECT tagRect = new tagRECT();

public MainWindow()

{

CoInitializeEx(IntPtr.Zero, 2);

InitializeComponent();

var obj = CoCreateInstance(

rclsid: CLASS\_WEBBROWSER,

pUnkOuter: null,

dwClsContext: CLSCTX\_INPROC\_SERVER,

riid: IID\_IWebBrowser2

) as IOleObject;

var site = new ClientSite(groupBox1.Handle, 0, 0, 1250, 920);

obj.SetClientSite(site);

obj.DoVerb((int)OLEDOVERB\_INPLACEACTIVATE, IntPtr.Zero, site, 0, Handle, ref tagRect);

(obj as IWebBrowser2).Navigate("http://google.com", null, null, null, null);

axShockwaveFlash1.Movie = Application.StartupPath + @"\tuma.swf";

}

}

}