Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Кафедра «Информационные системы и технологии»

Лабораторная работа №3

по теме: «СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ACTIVEX**»**

Выполнил: студент 3-го курса,

3 группы, факультета ИТ

Буц Е.Ю.

Минск 2017

1. **ХОД РАБОТЫ**

Сначала вносим изменения в MFCActiveXControl1Ctrl.h:

*#define DISPID\_TEST\_METHOD (1025314) //Добавлено*

#pragma once

// MFCActiveXControl1Ctrl.h: объявление класса элемента управления ActiveX для CMFCActiveXControl1Ctrl.

// CMFCActiveXControl1Ctrl: про реализацию см. MFCActiveXControl1Ctrl.cpp.

class CMFCActiveXControl1Ctrl : public COleControl

{

DECLARE\_DYNCREATE(CMFCActiveXControl1Ctrl)

// Конструктор

public:

CMFCActiveXControl1Ctrl();

// Переопределение

public:

virtual void OnDraw(CDC\* pdc, const CRect& rcBounds, const CRect& rcInvalid);

virtual void DoPropExchange(CPropExchange\* pPX);

virtual void OnResetState();

// Реализация

protected:

~CMFCActiveXControl1Ctrl();

DECLARE\_OLECREATE\_EX(CMFCActiveXControl1Ctrl) // Фабрика класса и guid

DECLARE\_OLETYPELIB(CMFCActiveXControl1Ctrl) // GetTypeInfo

DECLARE\_PROPPAGEIDS(CMFCActiveXControl1Ctrl) // ИД страницы свойств

DECLARE\_OLECTLTYPE(CMFCActiveXControl1Ctrl) // Введите имя и промежуточное состояние

// Схемы сообщений

DECLARE\_MESSAGE\_MAP()

// Схемы подготовки к отправке

DECLARE\_DISPATCH\_MAP()

afx\_msg void AboutBox();

*afx\_msg LONG TestMyMethod(LONG param1, LONG param2, BSTR param3);//Добавлено*

// Схемы событий

DECLARE\_EVENT\_MAP()

// Подготовка к отправке и ИД событий

public:

enum {

eventidEventHandlerTest = 1L //Добавлено

};

protected:

*void EventHandlerTest(LPCTSTR name, LPCTSTR surname, LONG age) //Добавлено*

*{*

*FireEvent(eventidEventHandlerTest, EVENT\_PARAM(VTS\_BSTR VTS\_BSTR VTS\_I4), name, surname, age);*

}

};

Затем в MFCActiveXControl1.cpp:

// MFCActiveXControl1Ctrl.cpp: реализация класса CMFCActiveXControl1Ctrl элемента ActiveX.

#include "stdafx.h"

#include "MFCActiveXControl1.h"

#include "MFCActiveXControl1Ctrl.h"

#include "MFCActiveXControl1PropPage.h"

#include "afxdialogex.h"

*#define M\_PI 3.1415926535897932384626433832795 //Добавлено*

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

IMPLEMENT\_DYNCREATE(CMFCActiveXControl1Ctrl, COleControl)

// Схема сообщений

BEGIN\_MESSAGE\_MAP(CMFCActiveXControl1Ctrl, COleControl)

ON\_OLEVERB(AFX\_IDS\_VERB\_PROPERTIES, OnProperties)

END\_MESSAGE\_MAP()

// Схема подготовки к отправке

BEGIN\_DISPATCH\_MAP(CMFCActiveXControl1Ctrl, COleControl)

DISP\_FUNCTION\_ID(CMFCActiveXControl1Ctrl, "AboutBox", DISPID\_ABOUTBOX, AboutBox, VT\_EMPTY, VTS\_NONE)

*DISP\_FUNCTION\_ID(CMFCActiveXControl1Ctrl, "TestMyMethod",DISPID\_TEST\_METHOD, TestMyMethod,VT\_I4, VTS\_I4 VTS\_I4 VTS\_BSTR) //Добавлено*

END\_DISPATCH\_MAP()

// Схема событий

BEGIN\_EVENT\_MAP(CMFCActiveXControl1Ctrl, COleControl)

EVENT\_CUSTOM\_ID("EventHandlerTest", eventidEventHandlerTest, EventHandlerTest, VTS\_BSTR VTS\_BSTR VTS\_I4)

END\_EVENT\_MAP()

// Страницы свойств

// TODO: при необходимости добавьте дополнительные страницы свойств. Не забудьте увеличить значение счетчика.

BEGIN\_PROPPAGEIDS(CMFCActiveXControl1Ctrl, 1)

PROPPAGEID(CMFCActiveXControl1PropPage::guid)

END\_PROPPAGEIDS(CMFCActiveXControl1Ctrl)

// Инициализировать фабрику класса и guid

IMPLEMENT\_OLECREATE\_EX(CMFCActiveXControl1Ctrl, "MFCACTIVEXCONTRO.MFCActiveXControCtrl.1",

0xd41da606, 0x3f8d, 0x4d4a, 0x85, 0xd5, 0xc7, 0xf, 0xd5, 0xde, 0xc0, 0xff)

// Введите ИД и версию библиотеки

IMPLEMENT\_OLETYPELIB(CMFCActiveXControl1Ctrl, \_tlid, \_wVerMajor, \_wVerMinor)

// ИД интерфейса

const IID IID\_DMFCActiveXControl1 = { 0x6046BDAF, 0x7F4E, 0x44FF, { 0x95, 0xCB, 0xA, 0xB3, 0x32, 0xFB, 0xD2, 0x6 } };

const IID IID\_DMFCActiveXControl1Events = { 0x9EDEA2EE, 0xC23E, 0x4BB4, { 0x9C, 0xDA, 0x59, 0x4, 0x3C, 0x9, 0xC9, 0x45 } };

// Сведения о типах элементов управления

static const DWORD \_dwMFCActiveXControl1OleMisc =

OLEMISC\_ACTIVATEWHENVISIBLE |

OLEMISC\_SETCLIENTSITEFIRST |

OLEMISC\_INSIDEOUT |

OLEMISC\_CANTLINKINSIDE |

OLEMISC\_RECOMPOSEONRESIZE;

IMPLEMENT\_OLECTLTYPE(CMFCActiveXControl1Ctrl, IDS\_MFCACTIVEXCONTROL1, \_dwMFCActiveXControl1OleMisc)

// CMFCActiveXControl1Ctrl::CMFCActiveXControl1CtrlFactory::UpdateRegistry -

// Добавление или удаление записей системного реестра для CMFCActiveXControl1Ctrl

BOOL CMFCActiveXControl1Ctrl::CMFCActiveXControl1CtrlFactory::UpdateRegistry(BOOL bRegister)

{

// TODO: убедитесь, что элементы управления следуют правилам модели изолированных потоков.

// Дополнительные сведения см. в MFC TechNote 64.

// Если элемент управления не соответствует правилам модели изоляции, то

// необходимо модифицировать приведенный ниже код, изменив значение 6-го параметра с

// afxRegApartmentThreading на 0.

if (bRegister)

return AfxOleRegisterControlClass(

AfxGetInstanceHandle(),

m\_clsid,

m\_lpszProgID,

IDS\_MFCACTIVEXCONTROL1,

IDB\_MFCACTIVEXCONTROL1,

afxRegApartmentThreading,

\_dwMFCActiveXControl1OleMisc,

\_tlid,

\_wVerMajor,

\_wVerMinor);

else

return AfxOleUnregisterClass(m\_clsid, m\_lpszProgID);

}

// CMFCActiveXControl1Ctrl::CMFCActiveXControl1Ctrl - Конструктор

CMFCActiveXControl1Ctrl::CMFCActiveXControl1Ctrl()

{

InitializeIIDs(&IID\_DMFCActiveXControl1, &IID\_DMFCActiveXControl1Events);

// TODO: Инициализируйте здесь данные экземпляра элемента управления.

}

// CMFCActiveXControl1Ctrl::~CMFCActiveXControl1Ctrl - Деструктор

CMFCActiveXControl1Ctrl::~CMFCActiveXControl1Ctrl()

{

// TODO: Выполните здесь очистку данных экземпляра элемента управления.

}

// CMFCActiveXControl1Ctrl::OnDraw - функция рисования

void CMFCActiveXControl1Ctrl::OnDraw(

CDC\* pdc, const CRect& rcBounds, const CRect& /\* rcInvalid \*/)

{

if (!pdc)

return;

// TODO: Замените следующий код собственным кодом рисования.

pdc->FillRect(rcBounds, CBrush::FromHandle((HBRUSH)GetStockObject(WHITE\_BRUSH)));

pdc->Ellipse(rcBounds);

}

// CMFCActiveXControl1Ctrl::DoPropExchange - поддержка сохранения

void CMFCActiveXControl1Ctrl::DoPropExchange(CPropExchange\* pPX)

{

ExchangeVersion(pPX, MAKELONG(\_wVerMinor, \_wVerMajor));

COleControl::DoPropExchange(pPX);

// TODO: Вызывать функции PX\_ для каждого постоянного настраиваемого свойства.

}

// CMFCActiveXControl1Ctrl::OnResetState - сброс элемента управления к состоянию по умолчанию

void CMFCActiveXControl1Ctrl::OnResetState()

{

COleControl::OnResetState(); // Сбрасывает значения по умолчанию, найденные в DoPropExchange

// TODO: Сбросьте здесь состояние любого другого элемента управления.

}

// CMFCActiveXControl1Ctrl::AboutBox - показывает пользователю панель "О программе"

void CMFCActiveXControl1Ctrl::AboutBox()

{

CDialogEx dlgAbout(IDD\_ABOUTBOX\_MFCACTIVEXCONTROL1);

dlgAbout.DoModal();

}

*LONG CMFCActiveXControl1Ctrl::TestMyMethod(LONG param1, LONG param2, BSTR param3) //Добавлено*

*{*

*return M\_PI\*(param1\*param1);*

*}*

// обработчики сообщений CMFCActiveXControl1Ctrl

И наконец в MFCActiveXControl1.idl:

// MFCActiveXControl1.idl: источник библиотеки типов для проекта элементов управления ActiveX.

// Этот файл будет обработан компилятором MIDL для

// создания библиотеки типов (MFCActiveXControl1.tlb), которая станет ресурсом в

// MFCActiveXControl1.ocx.

#include <olectl.h>

#include <idispids.h>

*#define DISPID\_TEST\_METHOD (1025314)* *//Добавлено*

[ uuid(0804CFE8-DB55-42FC-8882-169886F4468A), version(1.0),

control ]

library MFCActiveXControl1Lib

{

importlib(STDOLE\_TLB);

// Первичный интерфейс диспетчеризации для CMFCActiveXControl1Ctrl

[

uuid(6046BDAF-7F4E-44FF-95CB-0AB332FBD206)

]

dispinterface \_DMFCActiveXControl1

{

properties:

methods:

[id(DISPID\_ABOUTBOX)] void AboutBox();

*[id(DISPID\_TEST\_METHOD)] LONG TestMyMethod(LONG param1,LONG param2, BSTR param3);* *//Добавлено*

};

// Интерфейс отправки событий для CMFCActiveXControl1Ctrl

[

uuid(9EDEA2EE-C23E-4BB4-9CDA-59043C09C945)

]

dispinterface \_DMFCActiveXControl1Events

{

properties:

// Интерфейс событий не имеет свойств

methods:

*[id(1)] void EventHandlerTest(BSTR name, BSTR surname, LONG age); //Добавлено*

};

// Информация о классе для CMFCActiveXControl1Ctrl

[

uuid(D41DA606-3F8D-4D4A-85D5-C70FD5DEC0FF)

]

coclass MFCActiveXControl1

{

[default] dispinterface \_DMFCActiveXControl1;

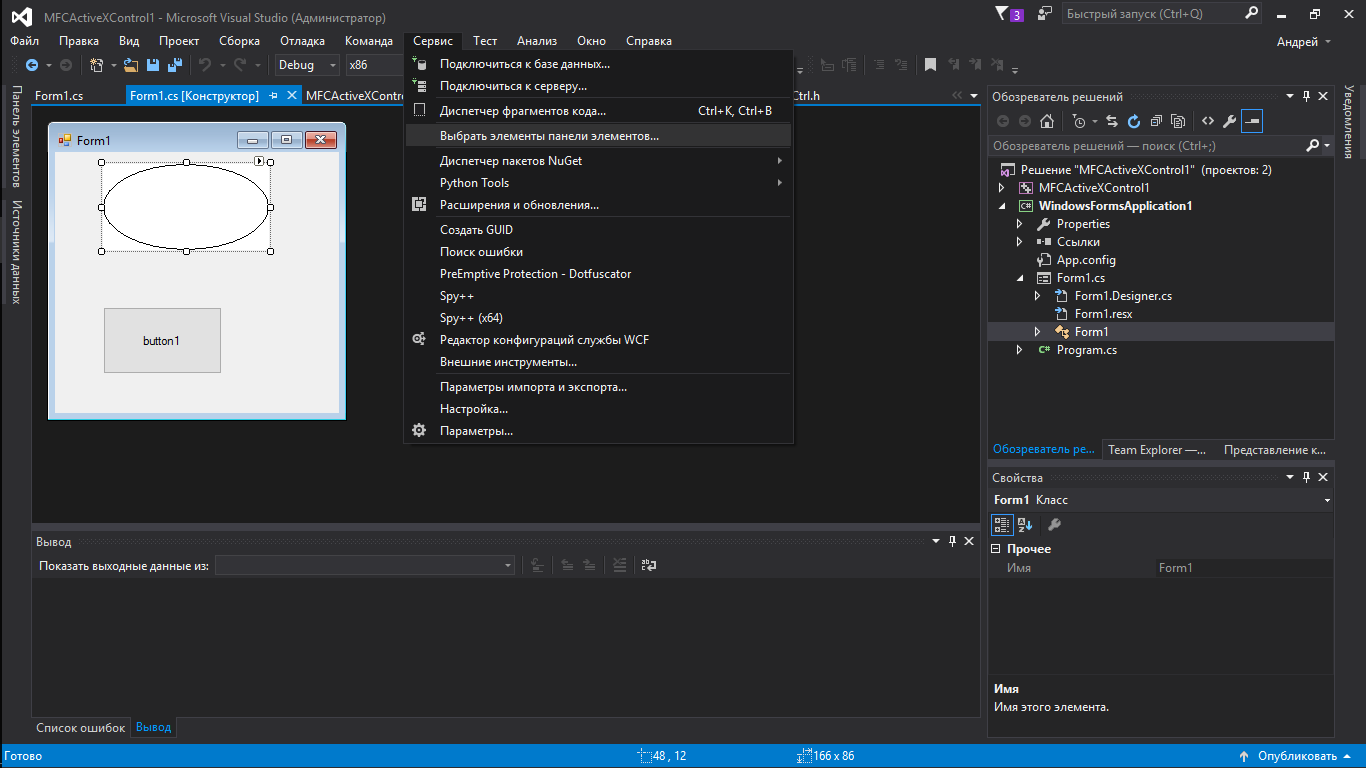
[default, source] dispinterface \_DMFCActiveXControl1Events;

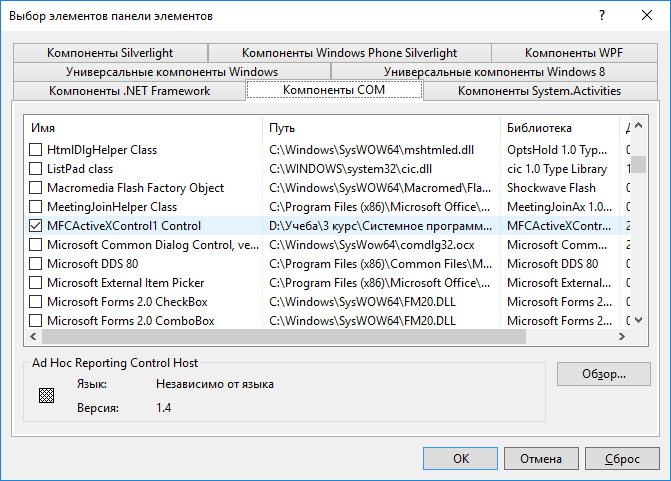
};

};

Собираем проект (Среда разработки должна быть запущена от имени администратора). После этого используем ActiveX в других проектах.

Потом создаем C#-проект (оконное приложение) и используем там наш компонент. На форме создаем кнопку и перетаскиваем компонент. Добавить его можно так:





Код для приложения такой:

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 MFCActiveXControl1Lib;

using AxMFCActiveXControl1Lib;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

static int k = 10;

public Form1()

{

InitializeComponent();

k = axMFCActiveXControl11.TestMyMethod(5, 4, "Hi");

axMFCActiveXControl11.EventHandlerTest += new AxMFCActiveXControl1Lib.\_DMFCActiveXControl1Events\_EventHandlerTestEventHandler(axMFCActiveXControl11\_EventHandlerTest);

}

private void axMFCActiveXControl11\_EventHandlerTest(object sender, \_DMFCActiveXControl1Events\_EventHandlerTestEvent e)

{

int age = e.age;

string name = e.name;

string surname = e.surname;

}

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

{

MessageBox.Show("Active returned: "+k);

}

}

}

Результат выполнения:

