

Технологии программирования.

Курсовая работа

Часть №2

Тема: Component Object Model

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

COM (Component 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.

Часть 1.

Для начала работы с COM необходимо проинициализировать соответствующую подсистему вызовом функции CoInitializeEx:

```
CoInitializeEx(0, 2);
```

В нашем случае, инициализацию следует производить до создания компонентов на форме, а поскольку данная функция находится в объекте-обертке OLE, результат будет выглядеть примерно так:

```
public Form1()  
{  
    OLE.CoInitializeEx(IntPtr.Zero, 2);  
    InitializeComponent();  
}
```

Далее необходимо создать экземпляр компонента, реализующего необходимый интерфейс. Для этого воспользуемся функцией CoCreateInstance:

```
CoCreateInstance(CLASS_WebBrowser, nil, CLSCTX_INPROC_SERVER,  
IWebBrowser2, Browser);
```

Соответственно, мы запрашиваем интерфейс IWebBrowser2. Для работы с этим интерфейсом необходимо знать его состав. Необходимое описание интерфейса для используемого вами языка программирования найдите самостоятельно (уникальный номер этого интерфейса {D30C1661-CDAF-11D0-8A3E-00C04FC9E26E}).

После получения ссылки на экземпляр компонента (она будет помещена в переменную Browser) необходимо встроить этот компонент в окно собственного приложения. Для этого воспользуемся технологией OLE. Сначала получим ссылку на OLE-интерфейс браузера:

```
IObject obj = Browser as IObject;
```

Затем выполним команды, встраивающие компонент в окно:

```
obj.SetClientSite(...);  
obj.DoVerb((int)OLEDOVERB.OLEIVERB_INPLACEACTIVATE, IntPtr.Zero, ...,  
0, Handle, ref rect);
```

Для успешного выполнения этих команд необходимо, чтобы вместо многоточия в строках выше стоял объект, корректно реализующий интерфейсы IOLEClientSite и IOLEInPlaceSite, через которые компонент и «договаривается» с встраивающим его приложением о границах областей, параметрах и событиях. Полное описание этих интерфейсов вы можете посмотреть в приложенном файле.

Необходимо объявить и реализовать все методы этих трех интерфейсов. Для примитивной реализации, которая и требуется в данной работе, достаточно, чтобы большая часть этих методов ничего не делала, возвращая успешный результат (HRESULT.S_OK). Методы GetWindowContext, OnPosRectChange и GetWindow должны функционировать так, как это описано в документации. Методы GetMoniker и GetContainer должны возвращать ошибку.

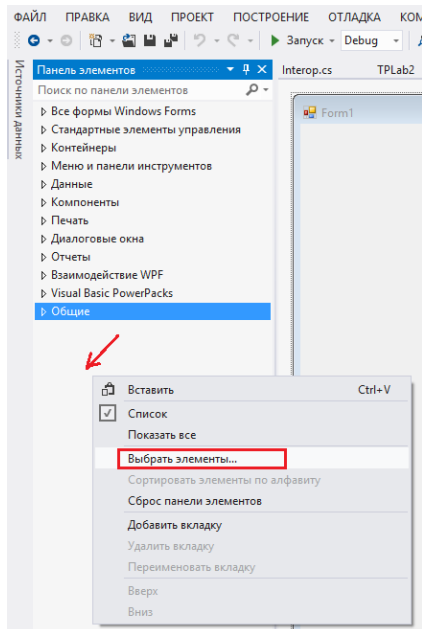
После инициализации, запустим встроенный в наше окно браузер и попросим его открыть какую-нибудь страницу:

```
browser.Navigate("http://ya.ru", null, null, null, null);
```

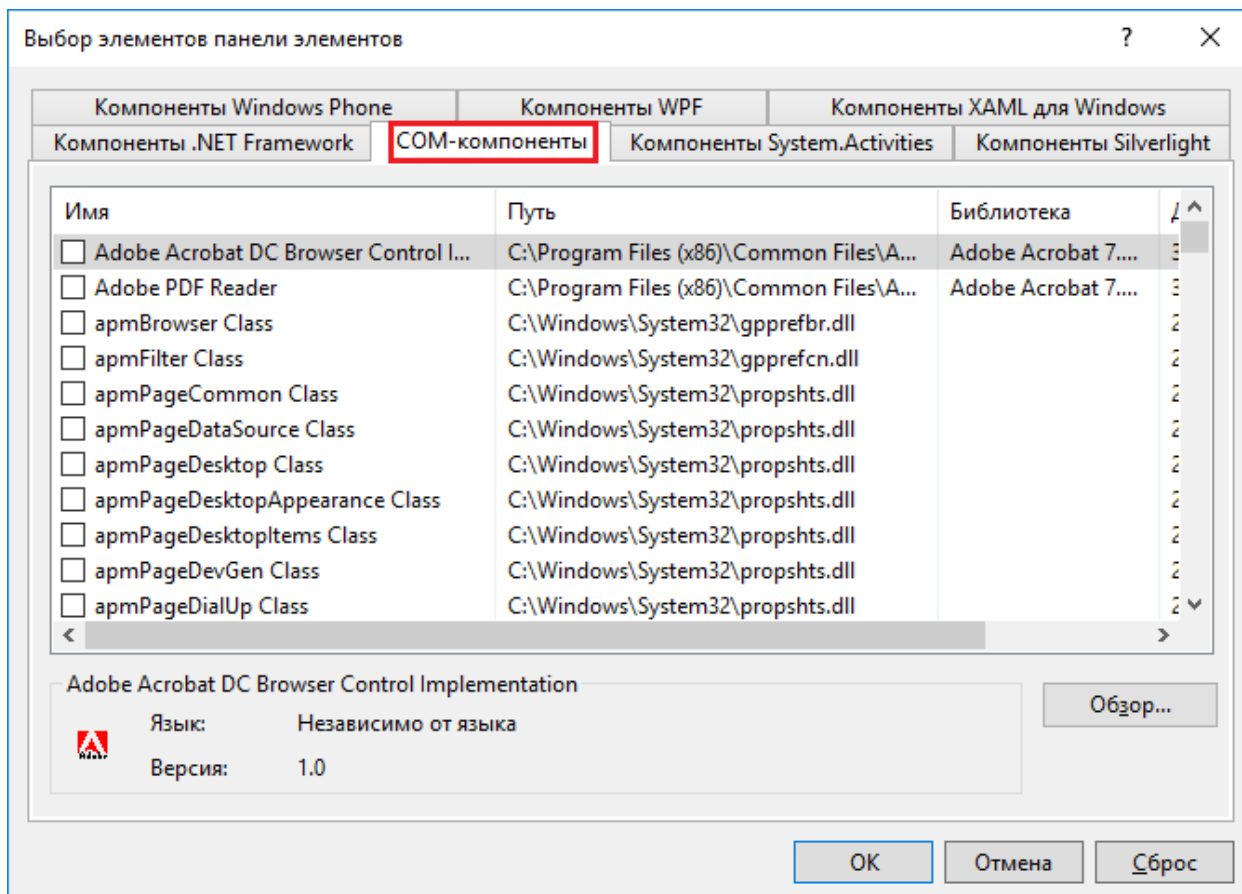
Часть 2.

В первой части мы рассмотрели как происходит работа со встраиваемым компонентом COM на низком уровне. Однако, нам известно, что большинство компонентов снабжено специальным описанием, позволяющим осуществлять их встраивание в полуавтоматическом режиме. Поскольку Веб-браузер обычно уже присутствует в панели компонентов, рассмотрим какой-нибудь другой COM компонент. Например, проигрыватель роликов Adobe Flash.

Для этого откроем диалог выбора визуальных компонентов:



Далее переключимся на вкладку «COM-компоненты»:



Найдем в списке компонент «Shockwave Flash Object» и отметим его галочкой. Если этого компонента в списке нет, его нужно добавить, щелкнув кнопку «Обзор» и выбрав файл C:\Windows\System32\Macromed\Flash\Flash.ocx

Загрузка и интеграция компонента в среду может занять некоторое время (1-2 минуты). После загрузки, мы можем просто разместить компонент на форме, как и любой другой визуальный элемент управления.

Для проверки работы компонента, возьмите файл tuma.swf, приложенный к лабораторной работе и поместите в папку приложения. После этого загрузите его в компонент проигрывателя (через свойство Movie) и запустите воспроизведение.