

Учебный курс
**Технологии и средства разработки
корпоративных систем**

Лекция 11

**Платформа Microsoft Office - расширенные
возможности**

Лекции читает

кандидат технических наук, доцент

Зыков Сергей Викторович

Корпоративные системы
Компонентные и офисные приложения
Поддержка Actions Panes

Возможности технологий Smart Documents и Action Panes

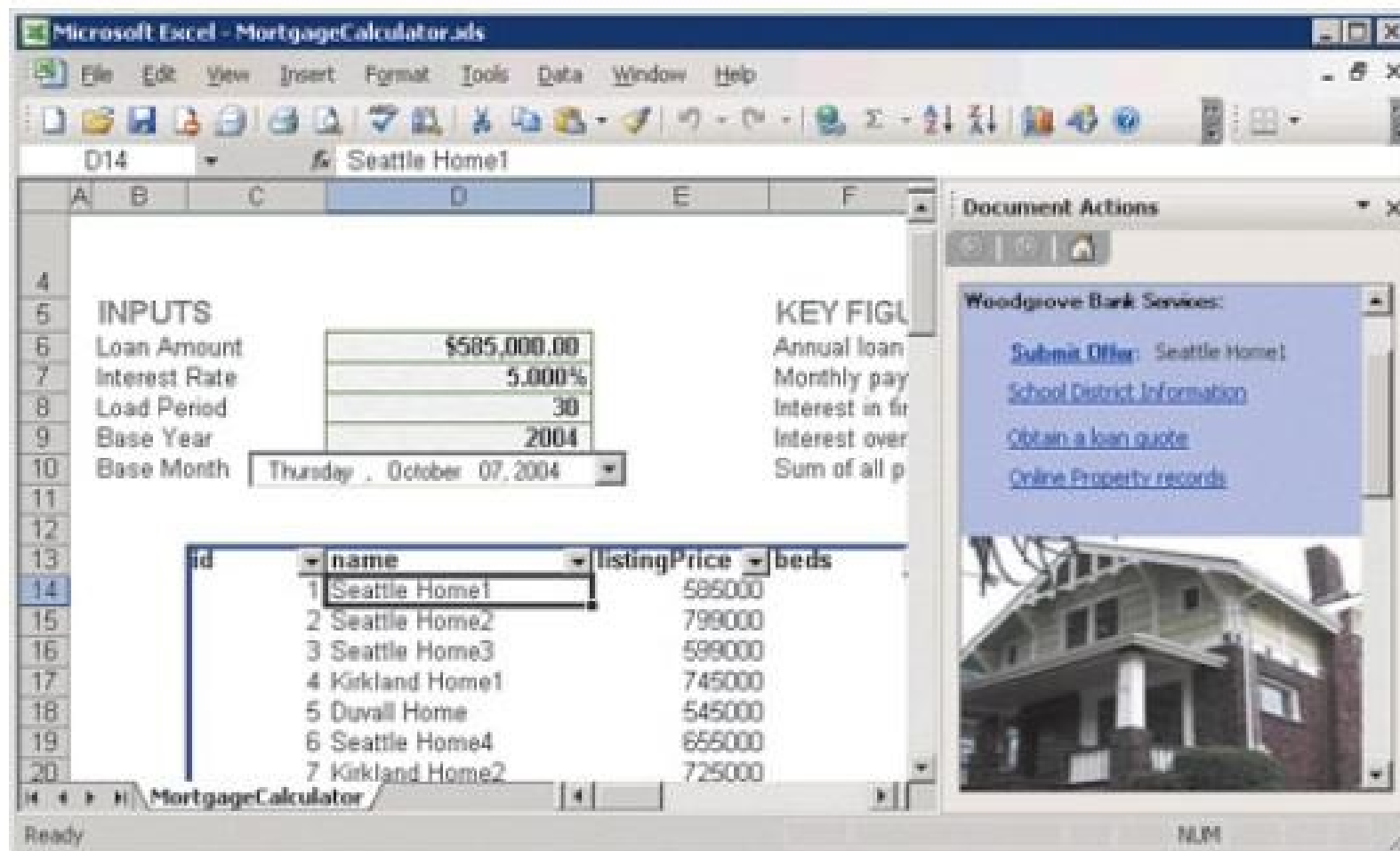
Smart Documents	Action Panes
Требуется реализация интерфейса ISmartDocument	Реализация интерфейса ISmartDocument не требуется — VSTO делает это автоматически
Требуется установка XML Expansion Pack	Установка XML Expansion Pack не требуется — VSTO делает это автоматически
Требуется Word-документ, основанный на XML	Опционально, можно создавать приложения, не базирующиеся на XML
Поддержка компонентов ActiveX	Поддержка компонентов Windows Forms

Разработка смарт-тэгов

- Смарт-тэги (Smart Tags) — это технология, позволяющая ассоциировать распознанный фрагмент текста в документ с каким-либо набором действий.
- Смарт-тэги могут динамически распознавать и обрабатывать данные на основании типа их содержимого
- Смарт-теги могут связываться с определенными ячейками в таблицах Excel 2003 или с определенными полями баз данных Access 2003

Корпоративные системы
Компонентные и офисные приложения

Поддержка программирования на уровне схем



Шаблон для дополнений к MS Office Outlook 2003 в VSTO 2005

Кэширование данных

- VSTO 2005 позволяет реализовать кэширование данных для документов Word и Excel
- Данные из кэша могут быть связаны с документами и отображены в режиме выполнения приложения
- Могут храниться данные, не связанные с интерфейсными элементами
- Асимметричная программная модель

Создание приложений на основе Outlook

Поддержка Microsoft Office Outlook 2003 VSTO 2005:

- позволяет обращаться к объектной модели продукта и кода на C# или на VB .NET
- выполнять такие операции, как создание расширенных меню, экспорт заданий
- совместное использование Outlook и XML Expansion Pack делает возможной интеграцию с Word или с Excel.

Модель безопасности в VSTO 2005

Модель очень удобна для разработчиков, желающих интегрировать свои решения с Office 2003. Код должен явно быть признан доверенным перед тем, как у него появится возможность выполнения. На компьютере каждого из пользователей содержится набор правил, определяющих, какому коду разрешено выполнение и что этот код может делать

Корпоративные системы

Компонентные и офисные приложения

Модель развертывания

Модель	Характеристики
Локальная/локальная	• Загрузка документа не требует доступа к сети;
	• пользователи должны полностью установить приложение на базе Office System;
	• пользователи работают с локальной копией документа;
	• обновления документа требуют копирования его на все компьютеры;
	• обновления требуют установки на всех компьютерах
Локальная/сетевая	• Обычно загрузка документа требует доступа к сети;
	• пользователи работают с локальной копией документа;
	• обновления происходят из централизованного хранилища;
	• обновление документа требует копирования его на все компьютеры
Сетевая/сетевая	• Загрузка документа требует доступа к сети;
	• пользователи работают с документом в режиме разделения доступа;
	• обновления документа требуют его публикации на сервере

Модель развертывания

Развертывание базируется на манифесте, который содержит всю необходимую для выполнения операции информацию. Для того чтобы приложения, созданные средствами VSTO 2005, могли работать, на клиентском компьютере должны быть установлены:

- Office 2003 Professional Edition;
- .NET Framework 2.0;
- VSTO 2005 Runtime;
- Office PIAs;
- соответствующие политики безопасности.

Корпоративные системы
Компонентные и офисные приложения
Использование панелей команд

Добавление новой строки меню в Excel:

- Присваиваем пространству имен Microsoft.Office.Core более простой псевдоним, например, Office:

```
// C#  
// В начале ThisWorkbook.cs.  
using Office = Microsoft.Office.Core;
```

- Затем определяем три переменных уровня модуля:

```
// В секции объявлений класса OfficeCodeBehind.  
private Office.CommandBar MainMenuBar;  
private Office.CommandBarControl MenuItem;  
private Office.CommandBarButton MenuItem;
```

Использование панелей команд (2)

- Пишем процедуру для создания пункта строки меню:

```
private void InitMenuBarItems(string Caption)
{
    try
    {
        this.MainMenuBar = ThisApplication.CommandBars[
            "Worksheet Menu Bar"];
        this.MenuBarItem = this.MainMenuBar.Controls.Add(
            Office.MsoControlType.msoControlPopup, Type.Missing,
            Type.Missing, Type.Missing, true);
        Office.CommandBarControl cbc =
            (Office.CommandBarControl) this.MenuBarItem;
        cbc.Caption = Caption;
        cbc.Visible = true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}
```

Корпоративные системы
Компонентные и офисные приложения
Использование панелей команд (3)

- Создаем новый пункт меню:

```
private Office.CommandBarButton CreateButton(  
    Office.CommandBarPopup Parent, string Caption)  
{  
    Office.CommandBarControl cbc = null;  
    try  
    {  
        cbc = Parent.Controls.  
            Add(Office.MsoControlType.msoControlButton,  
                Type.Missing, Type.Missing, Type.Missing, true);  
        cbc.Caption = Caption;  
        cbc.Visible = true;  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message, ex.Source, MessageBoxButtons.OK,  
            MessageBoxIcon.Error);  
    }  
    return (Office.CommandBarButton) cbc;  
}
```

Использование панелей команд (4)

- Вызывает вышеупомянутые процедуры из события `ThisWorkbook_Open`:

```
InitMenuBarItems("&Travel Tools");  
this.MenuItem = this.CreateButton(  
    (Office.CommandBarPopup) this.MenuBarItem,  
    "&Create Expense Report");  
this.MenuItem.Click +=  
    new  
    Office._CommandBarButtonEvents_ClickEventHandler(  
        MenuItem_Click);
```

Использование панелей команд (5)

- Добавим код, выполняющийся при выборе пользователем нового пункта меню:

```
private void MenuItem_Click(Office.CommandBarButton Ctrl,
    ref bool CancelDefault)
{
    frmReport frm = new frmReport();
    switch (frm.ShowDialog())
    {
        case DialogResult.OK:
            // Создать отчет.
            break;

        case DialogResult.Cancel:
            // Не делать ничего.
            break;
    }
    frm.Close();
    frm.Dispose();
}
```

Библиография (1)

1. Кулямин В.В. Компонентный подход в программировании
(<http://www.intuit.ru/department/se/compprog/>)
2. Марченко А.Л. Введение в программирование на C# 2.0
(<http://www.intuit.ru/department/pl/csharp20/>)
3. Кариев Ч.А. Создание Windows-приложений на основе Visual C#
(<http://www.intuit.ru/department/pl/visualcsharp/>)
4. Зыков С.В. Введение в теорию программирования. Объектно-ориентированный подход
(<http://www.intuit.ru/department/se/tppobj/>)

Библиография (2)

- <http://msdn2.microsoft.com/library/>
- <http://itc.ua/node/23618/>
- <http://compress.ru/Archive/CP/2005/8/54/>
- Alvin Bruney. Professional VSTO 2005: Visual Studio 2005 Tools for Office. WROX, 2006, 313 pp.

Благодарю за внимание!

Вопросы?

- <http://zykov.altweb.ru>
- szykov@hotmail.com
- sergey.zykov@tekama.com