

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

Лекция 10

**Компонентные и офисные приложения на
платформе Microsoft**

Лекции читает

кандидат технических наук, доцент

Зыков Сергей Викторович

Декларация сборки

Декларация сборки (Manifest) - составная часть сборки.

Это – также набор таблиц метаданных, который:

1. идентифицирует сборку в виде текстового имени, ее версию, культуру и цифровую сигнатуру (если сборка распределяется среди приложений);
2. определяет входящие в состав файлы (по имени и хэшу);
3. указывает типы и ресурсы, существующие в сборке, включая описание тех, которые экспортируются из сборки;
4. перечисляет зависимости от других сборок;
5. указывает набор прав, необходимых сборке для корректной работы.

Схема выполнения .NET-приложения в среде CLR



Домены приложений

Домен приложения (AppDomain) - это логический контейнер сборки, который используется для изоляции приложения в рамках адресного пространства процесса.

Свойства доменов:

1. Домены изолированы друг от друга.
2. CLR способна выгружать домены вместе со всеми сборками, связанными с этими доменами.
3. Возможна дополнительная конфигурация и защита доменов.
4. Для обмена данными между доменами реализован специальный механизм безопасного доступа (маршалинг).
5. В .NET Framework разработана собственная компонентная модель, элементами которой являются .NET-сборки (.NET-assembly), а для прямой и обратной совместимости с моделью COM/COM+ в CLR встроены механизмы (COM Interop), обеспечивающие доступ к COM-объектам по правилам .NET и к .NET-сборкам — по правилам COM. При этом для .NET-приложений не требуется регистрации компонентов в системном реестре Windows.

Виды сборок

В среде .NET выделяют следующие виды сборок:

1. частные (private). Представляют наборы типов, которые могут быть использованы только теми приложениями, где они включены в состав.
2. общего доступа (shared). Предназначены для использования НЕОГРАНИЧЕННЫМ количеством приложений на клиентском компьютере.

Взаимодействие объектов .NET и COM

При вызове объекта COM-клиентом .NET-среда Common Language Runtime:

1. создает **всего одну временную оболочку RCW** (Runtime Callable Wrapper) независимо от количества ссылок на объект. Это гарантирует, что все обращения к объекту происходят единственным путем.
2. на основе метаданных библиотеки создает вызываемый объект и оболочку для возврата данных.
3. выполняет контроль сборки мусора в оболочке, упрощая разработку.

Функции оболочки RCW:

- сокрытие различий между управляемым и неуправляемым кодом;
- управление жизненным циклом COM-объекта;
- передача вызовов методов между управляемым и неуправляемым кодом;
- преобразование параметров методов.

В результате RCW позволяет разработчикам писать код, трактующий COM-объекты как «обычные» .NET-объекты.

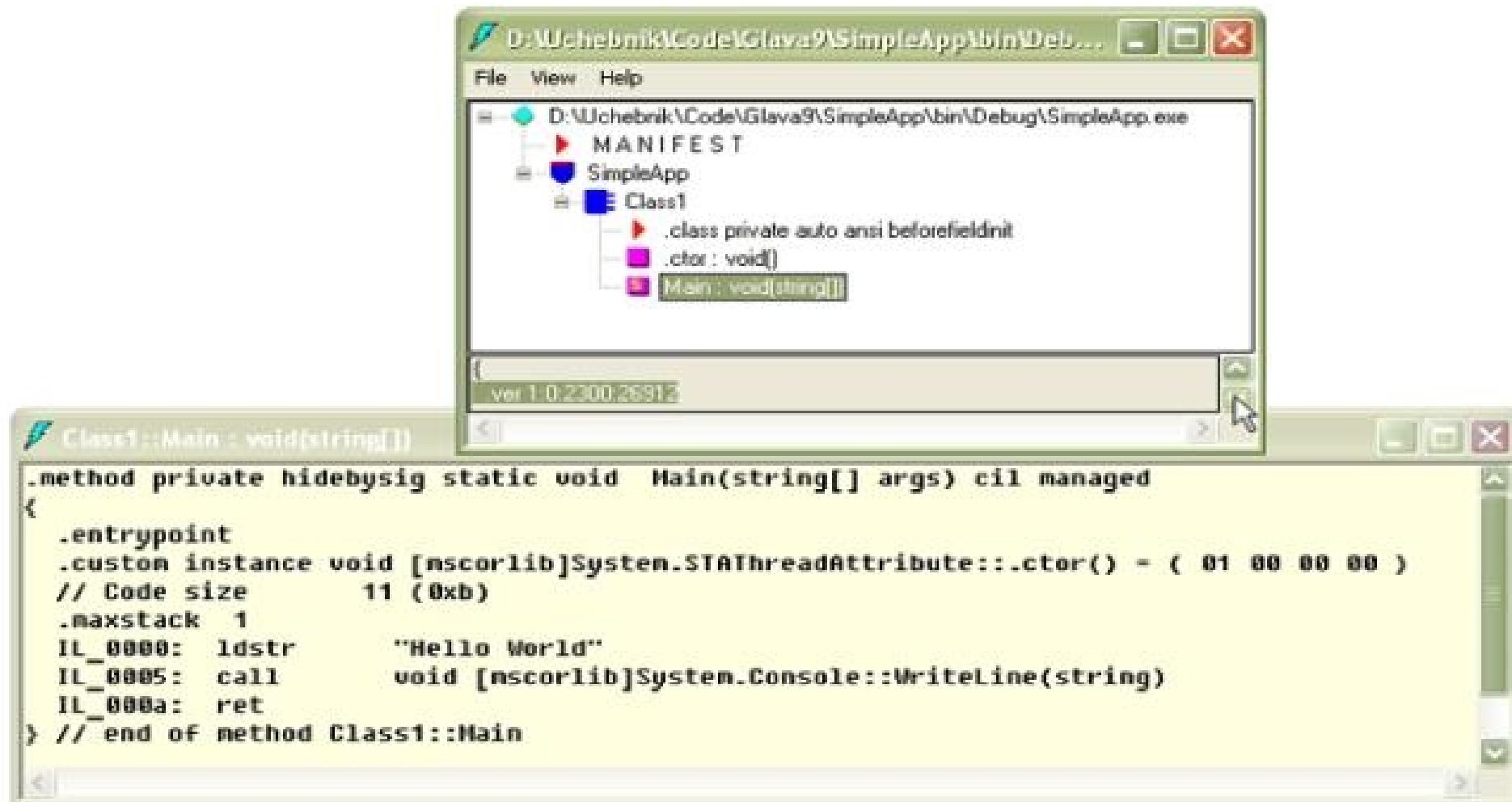
Просмотр содержимого сборки (1)

Содержимое сборки можно просмотреть, запустив дизассемблер "Microsoft Intermediate Language Disassembler" (ildasm.exe).

Пример (простое консольное приложение):

```
using System;
namespace SimpleApp
{
    class Class1
    {
        [STAThread]
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

Просмотр содержимого сборки (2)



Пример работы со сборками (1)

Для работы со сборками используется класс `Assembly` из пространства имен `System.Reflection`.

Пример (сборка `SimpleAssembly.dll`):

```
using System;
namespace SimpleAssembly{
    public class Class{
        public static string HelloWorld(){
            return "Hello world";
        }
    }
}
```

Пример работы со сборками (2)

Пример (приложение для загрузки в форму UsingLoadFrom):

```
using System;
using System.Reflection;

namespace UsingLoadFrom{
    class Class1{
        [STAThread]
        static void Main(string[] args){
            Assembly privateAss =
                Assembly.LoadFrom("SimpleAssembly.dll");
            MethodInfo info =
                privateAss.GetTypes()[0].GetMethod("HelloWorld");
            Object obj = info.Invoke(null, null);
            Console.WriteLine("Результат выполнения метода:{0}", obj);
        }
    }
}
```

Преимущества компонентной разработки приложений:

1. Снижение стоимости прикладного программного обеспечения
2. Повторное использование кода приложений для новых проектов
3. Унификация обработки объектов различной природы
4. Снижения влияния человеческого фактора на разработку приложений
5. Строгое математическое основание (лямбда-исчисление)
6. Концепция универсальна – в одинаковой мере применима для различных подходов к разработке приложений (напр., функционального программирования и ООП)

Корпоративные системы
Компонентные и офисные приложения

*Разработка офисных
приложений
при помощи расширений
Microsoft Visual Studio .NET*

Корпоративные системы
Компонентные и офисные приложения
Содержание

- Microsoft Office System как платформа
- Преимущества использования Office как платформы
- Visual Studio Tools для Microsoft Office System
- Общая языковая среда выполнения
- Преимущества VSTO
- Что нового в VSTO 2005?
- VSTO Сравнение возможностей
- Расширенные компоненты Office
- Поддержка Actions Pane
- Разработка смарт-тэгов
- Поддержка программирования на уровне схем
- Кэширование данных
- Создание приложений на основе Outlook
- Модель безопасности в VSTO 2005
- Модель развертывания
- Использование панелей команд (пример)
- Библиография

Microsoft Office System как платформа

Модели интеграции различных бизнес-приложений
с продуктами семейства Office:

- Ручная интеграция
- Модель внешней автоматизации
- Интеграция на основе документов
- Интеграция на уровне приложений

Visual Studio Tools для Microsoft Office System

- Использование VSTO позволяет объединить разработку для .NET и для Office
- Предоставляет полный доступ к библиотеке классов .NET Framework
- Предоставляет возможность применения объектно-ориентированных языковых конструкций
- Использование VSTO существенно расширяет набор инструментов и средств
- Применение Office и VSTO дает возможность связать документы с «живыми» бизнес-данными

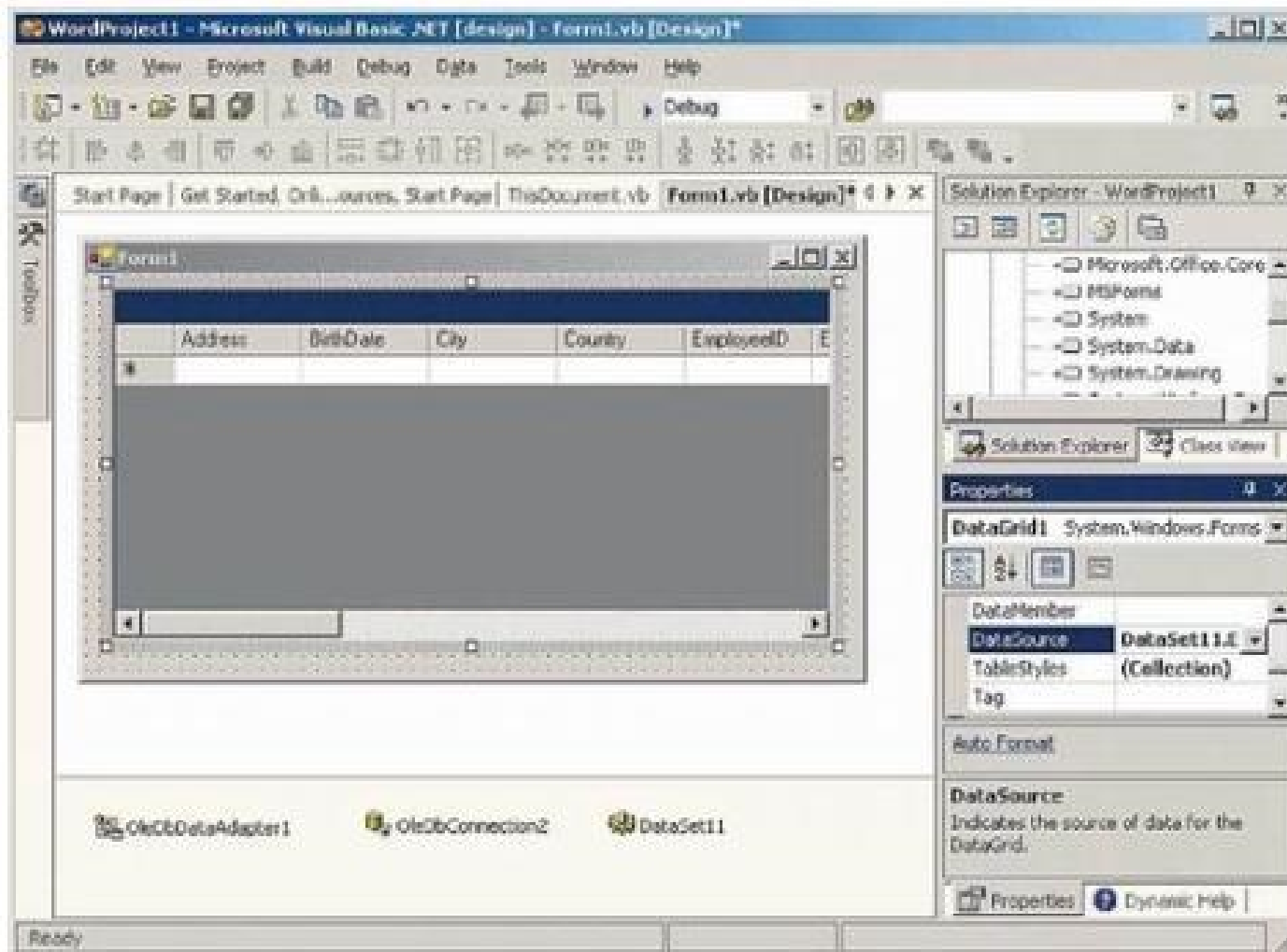
Общая языковая среда выполнения

- Управляет распределением памяти и следит, чтобы ваш код не пытался выполнять запрещенные операции
- Обеспечивает доступ к Microsoft .NET Framework и библиотекам базовых классов
- Новая архитектура VSTO позволяет разрабатывать решения, которые тесно интегрированы и специально ориентированы на использование управляемого кода.

Корпоративные системы

Компонентные и офисные приложения

Преимущества VSTO



Создание кода сборки, используемой в документе Microsoft Office Word

Что нового в VSTO 2005?

Среди новшеств *Visual Studio 2005 Tools for the Microsoft Office System* можно отметить:

- поддержку интерфейсных компонентов, созданных средствами .NET;
- расширенные компоненты Office;
- поддержку Actions Pane;
- разработку Smart Tags;
- поддержку программирования на уровне схем;
- кэширование данных;
- развертывание приложений на основе манифеста.

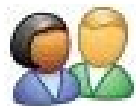
Корпоративные системы
 Компонентные и офисные приложения
Visual Studio Tools for Office
Сравнение возможностей

2003 2005



Взаимодействие

Решения VSTO работают на локальной машине, внутри процесса Excel или Word	✓	✓
---	---	---



Мощный UI

Все возможности Word и Excel	✓	✓
Панель задач на .NET		✓



Веб-службы &
 Оффлайн/онлайн

Поддержка веб-служб через .NET Framework	✓	✓
Решения могут работать в оффлайн		✓
Возможности кэширования		✓
Серверный кэш		✓

Корпоративные системы
Компонентные и офисные приложения
Visual Studio Tools for Office
Сравнение возможностей (2)

2003 2005



Для разработчика

Все возможности Visual Studio	✓	✓
Все возможности .NET Framework	✓	✓
Визуальные дизайнеры для проектирования форм Word и Excel		✓
Расширенные элементы управления Office •Word: XmlNode, XMLNodes, Bookmark •Excel: ListObject, NamedRange, XMLMappedRange		✓
Элементы управления Windows Forms		✓
Панель задач на .NET		✓
Декларативная программная модель		✓

Visual Studio Tools for Office

Сравнение возможностей (3)

2003 2005



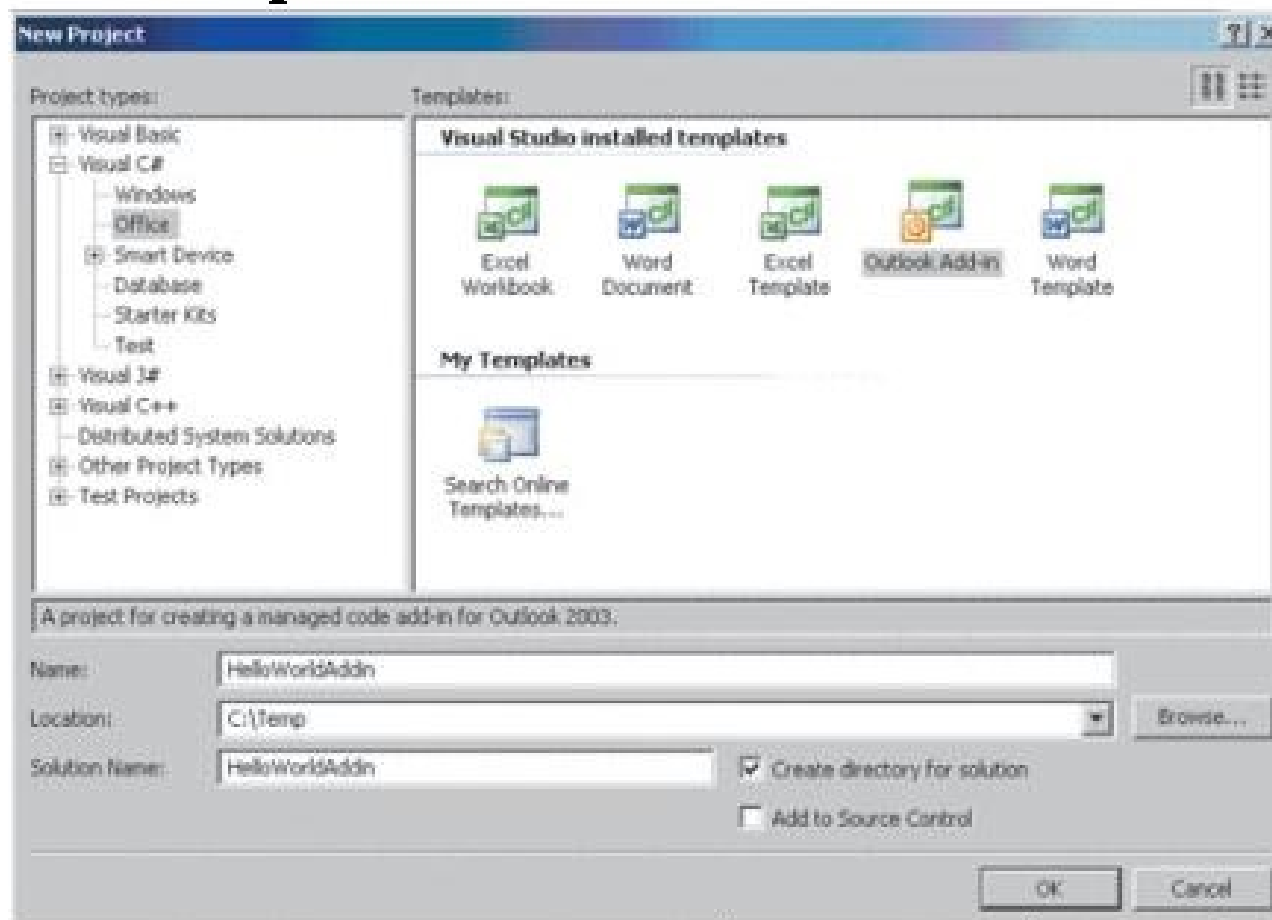
Развертывание



Управление
обновлениями

Документы привязаны к сборке; сборки размещаются на сервере	✓	✓
Развертывание не требует участия пользователя	✓	✓
Модель развертывания ClickOnce		✓

Корпоративные системы
Компонентные и офисные приложения
Расширенные компоненты Office



VSTO 2005 расширяет ряд встроенных объектов Word и Excel. Расширенные компоненты (так называемые Host Controls) доступны через стандартную панель компонентов, их свойства отображаются в Properties Explorer. Они программно доступны через именованные поля, поддерживают связь с данными и расширяют событийную модель.