Учебный курс

Технологии и средства разработки корпоративных систем

Лекция 9

Модели, принципы и структура компонентных приложений

Лекции читает

кандидат технических наук, доцент

Зыков Сергей Викторович

Понятие компонента (1)

Компонент – выделенная структурная единица с четко определенным интерфейсом. Абсолютно все его зависимости от окружения должны быть описаны в рамках этого интерфейса. Один компонент может также иметь несколько интерфейсов, играя несколько разных ролей в системе. Эти ограничения являются так называемым **интерфейсным контрактом** или **программным контрактом** компонента.

Интерфейсный контракт для каждой операции самого компонента (или используемой им) определяет **предусловие** и **постусловие** ее вызова. Предусловие операции должно быть выполнено при ее вызове, иначе корректность результатов не гарантируется. С постусловием все наоборот – постусловие вызванной у компонента операции должно быть выполнено после ее вызова, и это – обязанность компонента. Постусловие операции определяет, какие ее результаты считаются корректными.

Понятие компонента (2)

Компонент— единица развертывания. Он может быть присоединен к остальной системе, когда она уже некоторое время работает, и должен после этого выполнять все свои функции, если в исходной системе уже были все компоненты, от которых он зависит.

Набор правил определения интерфейсов компонентов и их реализаций, а также правил, по которым компоненты работают в системе и взаимодействуют друг с другом, принято объединять под именем **компонентной модели (component model)**. В компонентную модель входят правила, регламентирующие жизненный цикл компонента, т.е. то, через какие состояния он проходит при своем существовании в рамках некоторой системы (незагружен, загружен и пассивен, активен, находится в кэше и пр.) и как выполняются переходы между этими состояниями.

Компонент - более узкое понятие, чем программный модуль.

Возможность включения и исключения компонента из системы делает его отдельным элементом коммерческого ПО.

Существующие компонентные модели:

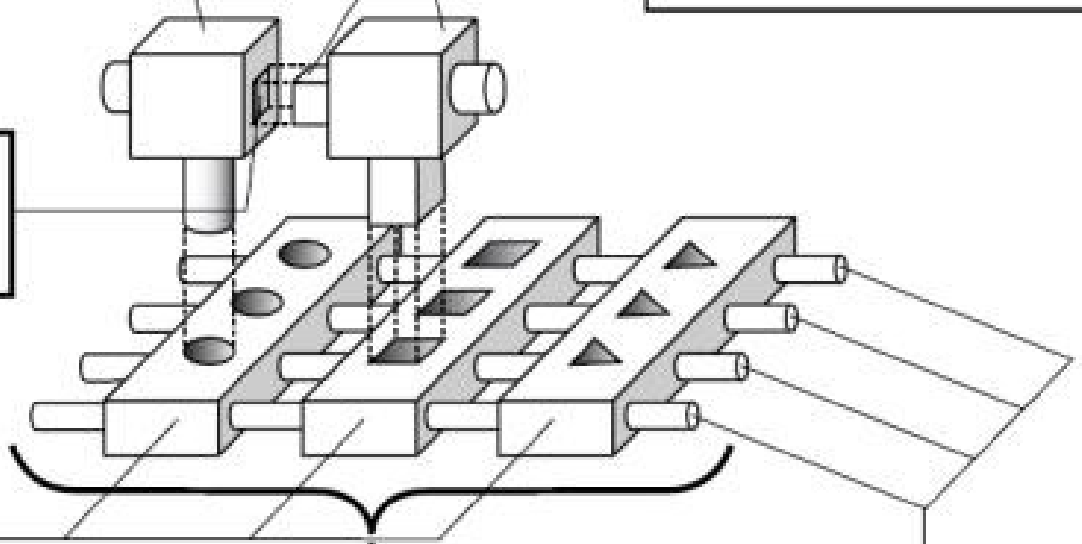
- COM (Component object Model) – технологический стандарт компании Microsoft
- JavaBeans – технологический стандарт компании Sun Microsystems для компонентов (не является независимым от языка программирования)
- CORBA (брокеры объектных запросов) – громоздкий IDL-интерфейс; сложность отображения одного языка реализации в другой

Основные элементы компонентных приложений

Компоненты
Предоставляют или требуют интерфейсы
Соответствуют компонентной модели
Развертываются в компонентной среде

Интерфейсы
Предоставляемые компонентами
Описываются контрактами

Интерфейс
Требуемый компонентом
Описывается контрактом



Компонентная модель
Определяет требования к работающим в ее рамках компонентам, виды компонентов
Иногда требует реализовать определенные интерфейсы

Компонентная среда
В ней развертываются компоненты
Предоставляет компонентную модель и набор базовых служб

Базовые службы
Обеспечивают работоспособность набора компонентов

Отличия компонентов от классов ЯООП

- Класс определяет не только набор реализуемых интерфейсов, но и саму их реализацию, поскольку он содержит код определяемых операций. Контракт компонента, чаще всего, не фиксирует реализацию его интерфейсов.
- Класс написан на определенном языке программирования. Компонент же не привязан к определенному языку, если, конечно, его компонентная модель этого не требует, — компонентная модель является для компонентов тем же, чем для классов является язык программирования.
- Обычно компонент является более крупной структурной единицей, чем класс. Реализация компонента часто состоит из нескольких тесно связанных друг с другом классов.

Среда Microsoft .NET. Основные понятия

Платформа – в контексте информационных технологий – среда, обеспечивающая выполнение программного кода. Платформа определяется характеристиками процессоров, особенностями операционных систем.

Framework – это инфраструктура среды выполнения программ, нечто, определяющее особенности разработки и выполнения программного кода на данной платформе.

Microsoft .NET – платформа

.NET Framework - инфраструктура платформы Microsoft .NET. Включает следующие основные компоненты: Common Language Runtime (CLR) и .NET Framework Class Library (.NET FCL).

Common Language Specification (CLS) – обобщенная спецификация языков программирования.

.NET-приложение – приложение, разработанное для выполнения на платформе Microsoft .NET. Реализуется на языках программирования, соответствующих CLS.

Функции CLR

- Управление кодом (загрузка и выполнение).
- Управление памятью при размещении объектов.
- Изоляция памяти приложений.
- Проверка безопасности кода.
- Преобразование промежуточного языка в машинный код.
- Доступ к метаданным (расширенная информация о типах).
- Обработка исключений, включая межъязыковые исключения.
- Взаимодействие между управляемым и неуправляемым кодами (в том числе и COM-объектами).
- Поддержка сервисов для разработки (профилирование, отладка и т.д.).

Библиотека классов FCL

FCL (.NET Framework Class Library) - соответствующая CLS-спецификации объектно-ориентированная библиотека классов, интерфейсов и системы типов

FCL могут использовать ВСЕ .NET-приложения, независимо от назначения архитектуры используемого при разработке языка программирования, и в частности:

- встроенные (элементарные) типы, представленные в виде классов (на платформе .NET все построено на структурах или классах);
- классы для разработки графического пользовательского интерфейса (Windows Form);
- классы для разработки web-приложений и web-служб на основе технологии ASP.NET (Web Forms) - http://asp.net/;
- классы для разработки XML и Internet-протоколов (FTP, HTTP, SMTP, SOAP);
- классы для разработки приложений, работающих с базами данных (ADO .NET) и многое другое.

Сборки на платформе .NET

- ***Сборка*** (assembly) - базовый строительный блок приложения в .NET Framework. Управляемые модули обьединяются в сборки. Сборка является логической группировкой одного или нескольких управляемых модулей или файлов ресурсов. Управляемые модули в составе сборок исполняются в Среде Времени Выполнения (CLR). Сборка может быть либо исполняемым приложением (при этом она размещается в файле с расширением .exe), либо библиотечным модулем (в файле с расширением .dll).

- Сборка является аналогом компонента в среде программирования .NET. Для CLR все сборки одинаковы, независимо оттого, на каких языках программирования они были написаны. Главное - это чтобы они соответствовали CLS.