

Разработка распределенных приложений ориентируемых на Интернет-Инtranет и WEB сервисы с помощью продуктов Borland, Oracle, Sun, Microsoft

На современном этапе развития программного обеспечения уделяется особое внимание созданию надежных, масштабируемых и переносимых приложений, которые не зависят от поставщиков и платформ. Компании, поставляющие готовые решения – SAP, Oracle, HP и многие другие, в настоящий момент уделяют большое внимание взаимодействию между своими приложениями посредством WEB сервисов. На смену технологиям Borland MIDAS и другим, приходят новые, обеспечивающие надежность взаимодействия готовых решений внутри корпорации. Это в первую очередь архитектура, ориентируемая на сервисы со своей технологией SOAP. В этом курсе будут рассмотрены основы создания распределенных приложений с помощью продуктов Borland Delphi, Oracle Jdeveloper, технологии Microsoft .Net и VBA, для серверов приложений стандарта J2EE. С помощью “оркестровки” WEB сервисов, с использованием языка BPEL, в одну корпоративную задачу будут собраны различные службы, поставляющие корпоративные данные из различных бизнес процессов. Использование языка UML позволит наглядно отобразить взаимодействие между процессами масштаба предприятия.

Аудитория: Начинающие разработчики WEB серверных приложений и приложений в архитектуре SOA. Бизнес аналитики, интеграторы, разработчики Xbase, переводящие свои приложения на платформу распределенных вычислений.

Для решения поставленных задач будут использоваться средства разработки Borland Delphi, Oracle Jdeveloper, Microsoft VBA и .NET

Необходимая подготовка: реляционные базы данных, Xbase системы для хранения данных (Clipper, Fox и другие). Знания ООП. Общие понятия распределенных вычислений (DCOM, WEB).

Цель: показать на примерах возможности построения сервис ориентируемых систем с использованием современных средств разработки. Приложения не должны зависеть от платформ, средств разработки, они должны быть настраиваемы, масштабируемы, надежны. Взаимодействие с готовыми решениями (SAP, HP, Oracle) – основная задача интеграторов предприятий. Слушатели получают методики доступа к различным серверам базы данных с использованием технологий J2EE - EJB. В курсе уделяется много внимания современным технологиям обработки данных – WEB, J2EE. В процессе работы создаются приложения, базирующиеся на технологии Web сервисов, обеспечивается доступ к серверам автоматизации MS Word, MS Excel. Создаются WEB сервера для Intranet приложений с реализацией парадигмы MVC (Model, View, Controller) в технологиях STRUTS и JSF. Строятся Web приложения для сервера Apache с использованием технологии Oracle HTML DB. Разрабатывается SOAP сервер с поставкой Web сервисов, а также клиентские места Delphi для доступа к стандартным серверам приложений технологии J2EE. Оркестровка WEB сервисов осуществляется с помощью Oracle BPEL Designer. Интеграция приложений показана на примерах Oracle Portal с созданием Omni WEB сервис портлетов.

Продолжительность курса: 5 дней / 40 ак. часов

Содержание курса:

1. Обзор распределенных вычислений.
2. WEB приложения: CGI и ISAPI стандарты, разработка приложений средствами Delphi для WEB серверов. Компоненты для генерации HTML страниц средствами Delphi. WebSnap компоненты для разработки Web-приложений на Delphi. Возможность плотной интеграции приложений WebSnap в корпоративные Web-сайты, порталы. Быстрая разработка WEB страниц на примерах Delphi Intranet компонент
3. Oracle PLSQL модуль и компоненты HTML DB для быстрой разработки простых web страниц. Примеры использования и интеграции PLSQL приложений в корпоративный портал с использованием Omni портлетов.
4. Web сервисы. Стандарт WSDL W3C. XML – основа обмена данными между приложениями. Краткое введение в XML. Понятие DOM, XSD, механизмы разбора документов.
5. Web-сервис – это просто еще одна программа, выполняющаяся в недрах вашей вычислительной инфраструктуры. Web-сервисы – это технология работы с сообщениями, в которой и передача сообщений основана на XML. Ключевые функции, которые применяются для управления Web-сервисами. В их числе - развертывание, конфигурирование и обеспечение безопасности. SOA не является чем-то новым: IT-отделы компаний успешно создавали и развертывали приложения, поддерживающие сервис-ориентированную архитектуру, уже много лет - задолго до появления XML и Web-сервисов. SOA - это не технология, а способ проектирования и организации информационной архитектуры и бизнес-функциональности.

6. Примеры создания WEB сервисов с помощью Borland Delphi для сервера Apache. Управление безопасности. Ограничения, накладываемые, на Borland Delphi Web сервисы. Преимущества клиентского приложения, реализованного Delphi для Web сервиса, перед Web страницей.
7. Примеры разработки Web сервисов для J2EE сервера. PLSQL сервисы. Oracle база данных– как пользователь сервисов. Управление безопасностью сервисов в J2EE сервере.
8. Доступ к сервисам из продуктов Microsoft. VBA – пользователь сервисов. Создание отчетов в Excel с использованием сервисов. Доступ из .Net к сервисам.
9. До появления концепции SOA при разработке систем в качестве отправного момента для программирования бизнес-логики использовались диаграммы рабочих потоков и блок-схемы систем. Разработанные вручную программы тщательно тестировались, после чего внедрялись. Сегодня ситуация изменилась коренным образом: современные инструменты управления бизнес-процессами позволяют обойтись без ручной разработки и тестирования. Так, с помощью методов моделирования можно проверять корректность исполнения бизнес-логики, представленной в диаграммах, а затем автоматически получать описания этих диаграмм на XML-языках управления бизнес-процессами. Таким средством является язык BPEL
10. SOA "подталкивает" к использованию альтернативных технологий и подходов (таких как обмен сообщениями) для построения приложений посредством связывания сервисов, а не посредством написания нового программного кода. Использование JMS сообщений. Примеры создания JMS сообщений в Borland Delphi.
11. Обмен синхронными и асинхронными сообщениями. Обработка исключений. Общая архитектура BPEL PM. BPEL Designer на базе среды разработки JDeveloper. Инсталляция, конфигурация, начало работы. Конфигурация Oracle BPEL консоли. Запуск сервера и дизайнера. Порядок создания простого BPEL процесса. Конфигурация WSDL интерфейса для BPEL процесса. Конфигурация базового скелетона для процесса. Просмотр MAP процесса. Компиляция, Deploy, тестирование BPEL процесса. Создание параллельных процессов. Пользовательские запросы. Создание пользовательского интерфейса для инициализации BPEL процесса. Настройка производительности процесса. Конфигурация жизненного цикла процесса.
12. Доступ к процессам BPEL из продуктов MSOffice. Запуск, управление процессом из Delphi. Создание клиентских Delphi приложений для работы с бизнес процессами.