

JAVA PROGRAMMING

INTRODUCTION TO JAVA PROGRAMMING

Java Basics

- Introduction to Java
 - Java Overview
 - What is Java?
 - Java, A Concurrent OOP
 - Java Virtual Machine
 - JIT Compiler
 - Stable and Powerful Packages
 - Packages Covered in this Course
 - What is the Value Added to the Web?
 - Live Web Pages
 - OO Components of Java Technology
 - Language Elements
 - Portable, Dynamic, and Extensible
- Introduction to Java's Basic OO Syntax
 - Object Outsides vs. Insides
 - Primitive and Object Variables
 - Messages and Object Communication
 - Java Bank Account
 - Java Virtual Machine Storage Areas

Coding in Java

- More Syntax
 - Variables and Storage Classes
 - Data Types, Precision and Default Values
 - Data Types - Literals
 - Conversions
 - Operators and Precedence
 - Arrays
 - String Class
 - Flow of Control
 - Compilation s
 - Packages
 - Symbol Resolution
 - Symbol Resolution Example
 - JDK Tools
- Exceptions
 - Java Exception Keywords
 - Exceptions Syntax
 - Exceptions
 - Exception Inheritance Hierarchy
 - Exceptions, When to Use
 - Exceptions - Complete Example
- I/O in Java
 - Input/Output in Java
 - I/O Objects in Java
 - Console I/O Example
 - File Copy
 - Appending a File
 - System Properties
- Class/Method Syntax
 - Class Syntax

- Inheritance and Accessibility
- Access Specifiers
- Variable (Field) Syntax
- Method Syntax
- Overloading and Overriding Methods
- Constructors
- Calling Constructors Within Constructors
- Object Creation
- Garbage Collection and Finalization
- Java vs. C++

Inheritance and Interfaces

- Inheritance and Interfaces
 - Rules of Java Class Inheritance
 - Using Interfaces in Java

User Interfaces

- Introduction to Applets
 - Java Applet vs. Java Application
 - Displaying Applets
 - Sample Applet HTML
 - Applets
 - Applet Lifecycle
 - Applet Constructor vs. init() Method
 - Layered Security Model
 - Applets and Security
 - Network Loaded Applets
 - Trusted Applets
 - Get Applet by Name
 - Applet Communication
 - Show Document
 - The java.awt and javax.swing Packages
 - Top-Level Containers
 - Top-Level Container Class Hierarchy
 - A Sample of GUI Objects
 - GUI Objects Class Hierarchy
 - Sample of Functions Provided to GUI Objects by Component, JComponent & Container
 - Layout Process and Containment
 - Layout Managers
 - Supplied Layout Managers
 - Border Layout
 - Grid Layout
 - Containment Hierarchy
- Event Handling
 - Event Delegation Model
 - Handler Sample
 - Events
 - Event Listener Interfaces
 - Event Registration
 - Event Delivery and Decoding
- Applets as Applications
 - Running Applets as Applications
 - Tabbed Pane Example

Networking

Networking API

- Some java.net Classes
- Sockets
- Endpoint Naming
- Connecting to a Server (Java)
- Client-Side Sockets
- Server-Side Sockets
- Datagram Sockets
- MulticastSocket
- Uniform Resource Locators (URLs)
- URL and URLConnection Example

Event Adapters and Inner Classes

Event Adapters and Inner Classes

- Presenting a GUI From an Application
- Capturing Window Events
- Using an Adapter Class
- Using an Inner Class
- The System Event Queue
- Posting a Task to the System Event Queue

Graphics and Image Support

Graphics and Image Support

- The Graphics Class
- Setting Font and Color Attributes
- Drawing Geometric Shapes
- Loading and Drawing Images
- Image Example
- Synchronized Loading with MediaTracker

Collections

Collections

- Array and Its Basic Operations
- Array traversal and searching
- Facts About Arrays
- Wrappers for Primitive Types
- Hierarchy of Java Collection Interfaces
- Methods of the Collection Interface
- Iterator
- Set Interface
- SortedSet Interface
- List Interface
- ListIterator Interface
- Map Interface
- SortedMap Interface
- Static Methods of the Collection Class
- Implementing Classes
- Static Methods of the Arrays Class
- The BitSet Class

Threads

Threads

- What is a Thread?
- Creating and Starting a Thread
- Stopping or "Sleeping" a Thread
- Creating a Thread for an Applet
- Creating a Thread That Runs Itself

- Using Synchronized Methods
- Using Synchronized Blocks
- Multithreading with Swing GUI
- The Timer Class
- Coordinating Threads: Using wait() and notify()
- Scope of Variables with Multithreading of Thread Methods

Close-Out s

Close-Out s

- Datagram Design for Chat Room Communications
- RMI Design for Chat Room Communications
- Serialization
- Optional Lab 6 (The Timer Class)
- JNI (Java Native Interface) — The Java Side
- JNI (Java Native Interface) — The C++ Side
- Using Static Final Switches
- Example Using Runtime.exe()
- Sun's "Java Plug-In" for Netscape and Internet Explorer
- Java Development Tools
- Java Development Kit (JDK) Version 1.1
- Java Development Kit (JDK) Version 2 (1.2)
- Java Certification

ADVANCED JAVA PROGRAMMING

Performance and Programming Style

Performance

- Thoughts on Optimization
- Performance Issues
- Overview Of Concerns
- Optimization in Javac
- JIT Compiler
- Operation Benchmarks
- Benchmark Implications
- API Good Guys and Bad Guys
- Open Issues

Java Programming Tips

- OUT Parameters
- Java's Big 4
- Implementation Of Equals
- Implementation of hashCode
- Implementing Clone
- Object Finalization
- Immutable Objects
- Thread Safe Objects
- Common Constructor Code
- Enumerated Types
- Conditional Code Blocks

Event Delegation and Inner Classes

Event Delegation

- Event Delegation
- Event Delegation Framework
- Event Classes
- Event Listeners
- Listener (Handling) Methods
- Registration of Listeners
- Example
- Adapters
- Adapter Classes

Inner Classes

- Motivation for Inner Classes
- Constructing Inner Class Instance
- Multiple Instance References
- List Example
- Anonymous Classes
- Anonymous Adapter
- Inner Class Miscellany

Serialization and Java Databases

Object Serialization

- Representing Objects Outside the VM
- Embedded Object References
- Serialization and Externalization
- Serialization Interface
- Serializing Object Graph
- Transient and ReadObject
- WriteObject
- List Example

- Class Versioning
- Externalizable
- Object Persistence and Relational Model and SQL Primer
 - Naming Persistent Objects
 - Naming and Indexes
 - Structuring the Persistent Store
 - Relational Data Model
 - Bank Accounts Relational Store
 - Standard Query Language (SQL)
 - Selecting Table Rows
 - Table Joins
- JDBC Architecture
 - JDBC Overview
 - Two-tiered Untrusted Applet
 - Application Access
 - Three-tier Access
 - Trusted Applet
 - JDBC Drivers
 - JDBC-ODBC Bridge
- JDBC API
 - DriverManager
 - Connecting to Database
 - Statement and ResultSet
 - ResultSet
 - Mapping SQL Types
 - Result Values
 - Relationships between JDBC Interfaces
 - Example Connect and Select
 - Example Connect and Update
 - PreparedStatement
 - CallableStatement
 - Multiple Result Set, Metadata
 - Transactions

Common Object Request Broker Architecture (CORBA)

- CORBA and IIOP
 - CORBA Introduction
 - Interfaces and Implementations
 - ORB as Transport Layer
 - CORBA Architecture
 - Java and CORBA
 - IIOP
- Hello World Example
 - Architecture
 - IDL
 - Server-Side Java: Server Implementation
 - Server-Side Java: Server
 - Client-Side Java
 - Build Process
 - Deployment Process
- Interface Definition Language (IDL)
 - What is IDL?
 - IDL Components
 - Modules
 - Basic Types

- User-Defined Types: Interfaces
- Interface Operations
- Interface Attributes
- Interface Example
- User-Defined Types: Enumeration and Structures
- User-Defined Types: Unions and typedefs
- User-Defined Types: Exceptions
- Constants and Compound Types

Mapping to Java

- Mapping IDL
- Mapping IDL (Contd.)
- Basic Types
- User-Defined Types: Interfaces
- User-Defined Types: Enumerations
- User-Defined Types: Structures
- User-Defined Types: Unions
- User-Defined Types: Exceptions
- Nesting
- Helpers
- Constants
- Constants Example
- Compound Types and typedefs
- Holders
- Other Generated Java Components

Host Implementation

- First, Some Terminology
- Writing an Impementation
- Preparing a Host
- Preparing a Host Applet
- Preparing a Host Application
- Binding a Servant
- Binding a Servant: Example
- Finding a Servant
- Finding a Servant: Example
- Other Ways of Getting a Servant
- Other Implementation Facilities

Reflection and Java Beans

Reflection

- JDK 1.0 java.lang.Class
- Motivation for Reflection
- Reflection
- Accessor of Class
- Field, Method, and Constructor
- Naming Conventions

Java Beans

- Software Components
- Current Software Component Architectures
- Bean's Objectives
- Operation Modes
- Bean Interface
- Building With Beans
- Application Builders
- Introspection
- Properties

- Bound Properties
- Constrained Properties
- Event Production and Consumption
- BeanInfo
- Property Editors
- Customizers
- Wiring And Adapters
- Bean Instantiation

Java Foundation Classes (JFC)

Swing Components

- Swing Introduction
- Swing Component Heirarchy
- JApplet
- JInternalFrame
- JDialog
- JFileChooser
- JColorChooser
- JMenu, JToolBar
- JTable
- JTree
- JViewport
- JSplitPane
- JEditorPane
- JTextField, JPasswordField
- JComboBox
- More Buttons, JPopupMenu
- JSlider
- JProgressBar
- JToolTip

Swing Features

- Pluggable Look-and-Feel
- PLAF Architecture
- Switching PLAFs
- PLAF Bridge Pattern
- Defining Your Own PLAF
- Threads and Swings
- The Single-Thread Rule
- invokeLater() and invokeAndWait()
- Separate Models
- Model Interfaces
- Example Model Object Model
- Multiple Panes
- JLayeredPane
- JRootPane - The Big Picture
- Document Handling

Accessibility

- Accessibility
- The javax.accessibility Package
- AccessibleContext Basics
- AccessibleContext Extras
- Accessible Value Example: JScrollBar.AccessibleJScrollBar
- Accessible Action Example: JComboBox.AccessibleJComboBox

JFC and the AWT

- JFC Features in AWT

- Java2D Graphics
- Java2D Demo
- Java2D Printing
- Printing Example
- Printing Example Running
- Drag and Drop
- Drag and Drop Objects
- Drag and Drop Example Running

Remote Method Invocations (RMI)

- RMI Architecture

- Client-Server Review
- Java C-S with Sockets
- Remote Procedure Calls
- RPC Structure
- RMI Architecture
- RMI Architecture
- Remote References
- RMI Registry
- Remote Object Garbage Collection

- RMI API

- Remote Object
- Remote Interfaces
- setSecurityManager, rebind
- Implementing Interfaces
- Client Implementation
- Build Process
- Thread and Synchronization
- RMI Class Loading
- Server Configurations
- Client Configurations
- Performance

Native Methods

- JNI

- Native Methods
- Declaring Native Method
- Type Encoding
- Native Method Name Resolution
- Name Munging
- Native Method Arguments
- JNI Interface Pointer
- Java Object Field Access
- Example - Field Access
- Java Method Invocation
- Example - Method Calls
- Object Creation, Exceptions
- Building the DLL
- Invocation API

Security

- Security Overview

- Security Goals
- Keys
- Digital Keys
- Session Keys
- Verification

- Message Digests
- Verification Using Digests
- Certificates
- Certificate Authorities
- Security Policies
- Security Managers
- Applet Sandbox
- Security Manager Operation
- Security API

Managing Java Archives

- Loading Applets into a Web Page
- JAR Files
- APPLET Tag Archive Property
- Resources in JAR Files
- Loading Resources Example
- Libraries and Applications

keytool - Managing Keys and Certificates

- Managing Keys and Certificates
- Default Keystores
- Generating Key Pairs
- Other Key-Generation Options
- Self-Signed Certificates
- Reading Certificates
- Importing Certificates
- Chain of Trust
- Root CAs
- Certificate Requests

jarsigner - Signing and Verifying JAR Files

- Signing JARs
- META-INF and the Manifest File
- Initial JAR File
- The jarsigner Tool
- Augmented Manifest File
- Signature File
- Signature File Contents
- Signature Block File
- Signature Block File Contents
- Complete Signed JAR File
- Verification

policytool - Managing Security Policies

- Policy Structure
- Policy Files
- The policytool Application
- Main Window Close-Up
- Policy Entry Window
- Policy Entry Window Close-Up
- Permissions Window
- Permissions Window Close-Up

Overview of Standard Extensions

Standard Extensions

- Extensions Framework
- JavaMail
- Java Advanced Imaging API (JAI)
- JavaBeans Activation Framework

- Java Naming and Directory Interface (JNDI)
- InfoBus
- InfoBus Example
- Java Cryptography Extension (JCE)
- Java Servlets
- Java Servlets Example
- Java Serial Port (COMM)
- JavaHelp
- Java Management API (JMAPI)
- Java 3D
- Java Media Framework
- Other Players
- Other Players

VISUAL PROGRAMMING WITH VISUALAGE FOR JAVA

Overview of Toolset

Overview of Toolset

Philosophy of Java

IBM VisualAge for Java, Version 2.0

IBM VisualAge for Java, Professional and Enterprise Editions

Integrated Development Environment (IDE)

Integrated Development Environment (IDE)

Welcome to VisualAge for Java

Workbench - Different Views of Your Work

Workbench Symbols

Workbench Symbols - Access Modifiers

Workbench Symbols - Based on Keywords

Workbench

Workspace, Repository and Files

Quick Start Smart Guide

Working with a Class

BeanInfo Symbols

Search - Easy Way to Find Classes

Multiple Views

Rapid Application Development

Editor

Error Checking

Error Checking Dialog

Run

Run - Check Class Path

Console

Scrapbook and Log

Repository Explorer

Debugger Features

Debugger Window

Debugger - Set Breakpoints

Debugger - Inspect Variables

Import and Export

Introduction to JavaBeans

Introduction to JavaBeans

Outline

Software Components

Overview

JavaBeans -

Operational Modes

Bean Properties, Events, and Methods

Bean Interface

JavaBean Naming Conventions, Design Patterns for Properties

Properties

Methods

Events

Event Model

Events: Event State Objects

Building with Beans

Application Builders

Manual Integration

Introspection

Boolean Properties

- Indexed Properties
- Bound Properties
- PropertyChange Code Example
- Event Model
- Constrained Properties
- Customization
- Serialization
- Construction
- JavaBeans
- Packaging
- Packaging: Jar File
- JavaBeans and VisualAge for Java
- VisualAge Java and JavaBeans

Introduction to Swing

- Introduction to Swing
 - Java Foundation Classes (JFC)
 - AWT Peer Architecture
 - Swing Architecture - Model/View
 - Swing Architecture - The View
 - Swing Architecture - The Model
 - Some Swing Components
 - JList Model
 - DefaultListModel

Visual Composition Editor: JavaBeans

- Visual Programming
 - Visual Programming
 - VisualAge - Visual Construction
 - Visual Composition Editor
 - Mouse Controls
 - Palette
 - Kinds of Components
 - Types of Beans
 - Example: Employee Bean
 - Employee Example

Connections

- Topic
 - The Anatomy of a Connection
 - How to Start a Connection
 - How to Complete a Connection
 - Property to Property Connection
 - Event to Method Connection
 - Event to Code Connection
 - Incomplete Connections
 - Parameter-from-Property Connection
 - Filling in Parameter(s)
 - Connection Information Status Area
 - Connection Names

Properties and Palettes

- Topic
 - Open Properties
 - Some Examples of Properties
 - Beans List
 - Palette Categories
 - Tour of Palette Categories

- Adding Other Beans
- Change the Palette
- Generate and Run

Version Control

- Version Control
 - Support Levels
 - Repository Components
 - Component Relationships
 - Single-Developer Repository
 - Multi-Developer Repository
 - Moving Code in the Workspace
 - Open and Version Editions
 - Component Status And Changeability
 - Loading Another Level of Code
 - Application Development Cycle
 - Tool Dialogs for Version Control

Layout Managers

- Layout Managers
 - Swing Containers
 - Null Layout
 - FlowLayout
 - Setting the Layout Property
 - Flow Layout
 - Border Layout
 - Border Layout in VisualAge for Java
 - GridLayout
 - Grid Bag Layout
 - GridBagLayout Example
 - Constraints
 - Box Layout
 - Card Layout
 - Beans List

Managing Focus in Swing Classes

- Managing Focus in Swing Classes
 - Setting Initial Focus
 - Initial Focus Example
 - Managing Focus
 - Default Tabbing Order
 - Default Tabbing Order Example
 - Controlling the Tabbing Order
 - Next Focusable Component
 - Setting Focus Explicitly
 - Focus Accelerators
 - VCE Assists for Focus
 - Focus Events
 - Focus Manager
 - Key Event Handling

Sheet Music Store Case Study

- Sheet Music Store Case Study
 - Case Study Introduction
 - Lab Exercise
 - Case Study Visual Objects
 - Case Study - Swing Renderers
 - Case Study - Swing Model Invisible JavaBeans

Case Study Invisible Business Objects

Invisible JavaBeans Model/View Architecture

Invisible JavaBeans Model/View Architecture

Types of Beans

Model - View Design Pattern

The General Paradigm

Benefits of Model - View Design Pattern

Benefits of Model - View

Using VA Java and Model/View

Building the View

Characteristics of a Good GUI

Building the Model

Specify the Property

Generate the Property

Specify the Method

Specify the Events

Hidden, Expert, and Preferred Features

ComboBox Model

ComboBox Model

Review of ListModel Interface

ComboBoxModel Interface

ComboBoxModel Implementation

JTable and TableModel

JTable and TableModel

TableModel

JTable

Abstract TableModel

Default TableModel

TableModel Example

JTable and TableModel

Running the Test Program

Additional Capabilities

Visual Composition Editor: Variables and Factory

Visual Composition Editor: Variables and Factory

Class versus Variable

Defining a Variable

When to Use Variables

Using Variables, Example 1

A Reusable View, Example 2

Using Variables, Example 2

Add Variable to the Interface

Promote Feature

Promote Variable as a Whole?

Variables - Tear-Off Property, Example 3

Using Variables, Example 4

Factory

Factory - Creating a Secondary Window

Factory and Invisible Beans

Table Renderers

Table Renderers

Agenda

What is a Renderer?

How Does a Renderer Work?

The TableCellRenderer

- DefaultTableCellRenderer
- Using the TableCellRenderer
- Sample TableCellRenderer
- Sample TableCellRenderer - PriceRenderer
- Sample TableCellRenderer - PriceRenderer
- Sample TableCellRenderer - QuantityRenderer
- Sample TableCellRenderer - QuantityRenderer
- Sample TableCellRenderer - View

Swing and Threads

- Swing and Threads
- Swing and Threads
- Swing Threads Example

JavaBeans - Events

- JavaBeans - Events
- Creating the Counter Bean
- Creating the Counter Bean: Overview
- Creating the Counter Bean: Properties
- Constructors
- Utility Methods
- Events
- Counter Bean Usage Scenarios
- Bound Properties
- Property Change Event Listener
- Action Events
- Action Event Listener
- User-defined Event

Menu Bar and Pop Up Menus

- Menu Bar and Pop Up Menus
- Building a Window Menu Bar
- Pop Up Menus

Application and Applet Deployment

- Application and Applet Deployment
- Application Deployment
- JRE Requirements
- Additional Runtime Libraries
- Enterprise Edition Runtime Files
- Swing Requirements
- Applet Deployment

Customization and JavaBeans

- Customization and JavaBeans
- Outline
- Overview
- Customizer
- PropertyEditor Interface
- PropertyEditor Interface - Example
- BeanInfo Interface

Customizer Interface

- Customizer Interface
- Customizer Interface
- Custom Property Editor for DateTimeInfo
- Customizer Example
- Custom Property Editor for DateTimeInfo

Serialized JavaBeans

Serialized JavaBeans

- Outline
- Serialization
- Serialization - goals
- Serialization - overview
- Serialization - what to save
- Serialization - where to save
- Serializable Interface
- Externalizable Interface
- Creating the SER File
- Using a Serialized JavaBean
- Serialized JavaBean Life Cycle

Introduction to Persistence

Introduction to Persistence

- Support for Relational Database Persistence
- Data Access Beans - Select Bean
- Enterprise Data Access Builder
- Persistence Builder
- Enterprise JavaBeans

Data Access Beans

Data Access Beans

- Data Access Beans: Palette
- Select Bean: Query Property
- Select Bean: Connection
- Select Bean: SQL Assist SmartGuide
- SQL Assist SmartGuide (Join)
- SQL Assist SmartGuide (Columns)
- SQL Assist SmartGuide (Sort)
- SQL Assist SmartGuide (Mapping)_
- SQL Assist SmartGuide (SQL)
- Select Bean: Visual Composition
- DBNavigator Bean
- Update, Insert, and Delete
- Host Variables

Internationalization/NLS Support

Internationalization/NLS Support

- What is NLS?
- Language-Specific Applications
- NLS Applications
- NLS - How Does It Work?
- NLS - How Difficult?
- NLS - What Does Java Offer?
- Internationalization Class Library
- VisualAge Java Support for "i18n"
- Externalizing Strings for a Class
- More Resource Bundling

DEVELOPING SERVER SIDE APPLICATIONS USING VISUALAGE FOR JAVA

Overview of Toolset

Overview of Toolset

Philosophy of Java

IBM VisualAge for Java, Version 2.0

IBM VisualAge for Java, Professional Edition Professional Update

Integrated Development Environment (IDE)

Integrated Development Environment (IDE)

Welcome to VisualAge for Java

Workbench - Different Views of Your Work

Workbench Symbols

Workbench Symbols - Access Modifiers

Workbench Symbols - Based on Keywords

Workbench

Workspace, Repository and Files

Quick Start Smart Guide

Working with a Class

BeanInfo Symbols

Search - Easy Way to Find Classes

Multiple Views

Rapid Application Development

Editor

Error Checking

Error Checking Dialog

Run

Run - Check Class Path

Console

Scrapbook and Log

Repository Explorer

Debugger Features

Debugger Window

Debugger - Set Breakpoints

Debugger - Inspect Variables

Import and Export

Introduction to JavaBeans

Introduction to JavaBeans

Outline

Software Components

Overview

JavaBeans -

Operational Modes

Bean Properties, Events, and Methods

Bean Interface

JavaBean Naming Conventions, Design Patterns for Properties

Properties

Methods

Events

Event Model

Events: Event State Objects

Building with Beans

Application Builders

Manual Integration

Introspection

- Boolean Properties
- Indexed Properties
- Bound Properties
- PropertyChange Code Example
- Event Model
- Constrained Properties
- Customization
- Serialization
- Construction
- JavaBeans
- Packaging
- Packaging: Jar File
- JavaBeans and VisualAge for Java
- VisualAge Java and JavaBeans

Visual Composition Editor: JavaBeans

- Visual Programming
 - Topic
 - Visual Programming
 - VisualAge - Visual Construction
 - Visual Composition Editor
 - Mouse Controls
 - Palette
 - Kinds of Components
 - Types of Beans
 - Example: Employee Bean
 - Employee Example
- Connections
 - Topic
 - The Anatomy of a Connection
 - How to Start a Connection
 - How to Complete a Connection
 - Property to Property Connection
 - Event to Method Connection
 - Event to Code Connection
 - Incomplete Connections
 - Parameter-from-Property Connection
 - Filling in Parameter(s)
 - Connection Information Status Area
 - Connection Names
- Properties and Palettes
 - Topic
 - Open Properties
 - Some Examples of Properties
 - Beans List
 - Palette Categories
 - Tour of Palette Categories
 - Adding Other Beans
 - Change the Palette
 - Generate and Run

Version Control

- Version Control
 - Support Levels
 - Repository Components
- Component Relationships

- Single-Developer Repository
- Multi-Developer Repository
- Moving Code in the Workspace
- Open and Version Editions
- Component Status And Changeability
- Loading Another Level of Code
- Application Development Cycle
- Tool Dialogs for Version Control

Invisible JavaBeans Model/View Architecture

- Invisible JavaBeans Model/View Architecture
 - Types of Beans
 - Model - View Design Pattern
 - The General Paradigm
 - Benefits of Model - View Design Pattern
 - Benefits of Model - View
 - Using VA Java and Model/View
 - Building the View
 - Characteristics of a Good GUI
 - Building the Model
 - Specify the Property
 - Generate the Property
 - Specify the Method
 - Specify the Events
 - Hidden, Expert, and Preferred Features

Music Store Case Study

- Music Store Case Study
 - Case Study Introduction
 - Lab Exercise
 - Case Study Invisible Business Objects

JavaBeans - Events

- JavaBeans - Events
 - Creating the Counter Bean
 - Creating the Counter Bean: Overview
 - Creating the Counter Bean: Properties
 - Constructors
 - Utility Methods
 - Events
 - Counter Bean Usage Scenarios
 - Bound Properties
 - Property Change Event Listener
 - Action Events
 - Action Event Listener
 - User-defined Event

Serialized JavaBeans

- Serialized JavaBeans
 - Outline
 - Serialization
 - Serialization - goals
 - Serialization - overview
 - Serialization - what to save
 - Serialization - where to save
 - Serialization
 - Serializable Interface
 - Externalizable Interface

- Creating the SER File
- Using a Serialized JavaBean
- Serialized JavaBean Life Cycle

HTTP, URLs, and CGI

- HTTP (Hypertext Transfer Protocol) Overview
- HTTP, URLs and HTML
- HTML Output
- HTML
- HTTP Request Methods from the Web Browser
- HTTP Flows - Simple Request
- HTTP Flows - Script GET Request
- HTTP Flows - Form Requests
- HTTP Flows - Script POST Request
- HTTP Protocol - Request
- HTTP GET Request Format
- HTTP POST Request Format
- HTTP Protocol - Response
- HTTP Response Format
- CGI (Common Gateway Interface) Overview
- CGI Example
- Issues
- Alternatives

An Introduction to Servlets

- What is a Servlet?
- What is a Servlet?
- Multiple Requests
- Generic Servlet Invocation
- Java Servlet API
- Class javax.servlet.GenericServlet
- GET and POST Requests
- Fundamentals of Processing GET Requests
- Simple GET Request Output
- Simple GET Request Example (1 of 2)
- Fundamentals of Processing POST Requests
- Simple POST Request Example (1 of 2)
- Classes and Interfaces that Make Up the Servlet APIs
- Class javax.servlet.http.HttpServlet
- javax.servlet.http.HttpServletRequest (1 of 3)
- javax.servlet.http.HttpServletResponse (1 of 2)
- Interface javax.servlet.ServletConfig
- Developing Servlets with the JSDK
- Running Servlets ..
- Issues

Support for Developing and Debugging Servlets

- Support for Developing and Debugging Servlets
- Why use VisualAge for Java for Servlets?
- Enterprise Update Installation
- Adding WebSphere/JSP/EJB Features
- Debugging Servlets with VisualAge Java

JavaServer Pages

- JavaServer Pages
- Static and Dynamic WEB Content
- Advantages and Disadvantages of Servlets
- Server-Side Scripting

- Simple JSP Example
- JSP Processing
- Invoking a JSP
- What is a JavaServer Page
- A More Complicated JSP Example
- JSP Directives
- JSP Directives - method and import
- JSP Directives - content_type and language
- JSP Directives - implements and extends
- JSP Declarations - Class-wide variable and method declarations
- JSP Scripts - Inline Java code
- JSP Scriptlets - Inline Java code - Example
- JSP expressions
- JSP Beans - <BEAN> tag

Using JavaBeans with JavaServer Pages

- Using JavaBeans with JavaServer Pages
- Disadvantages of JavaServer Pages
- JavaServer Page Model 1
- Separation of Presentation and Content
- JavaBeans and JSP
- <BEAN> Tag Syntax
- <BEAN> Tag Attributes

Introduction to Persistence

- Introduction to Persistence
- Support for Relational Database Persistence
- Data Access Beans - Select Bean
- Enterprise Data Access Builder
- Persistence Builder
- Enterprise JavaBeans

Data Access Beans

- Data Access Beans
- Data Access Beans: Palette
- Select Bean: Query Property
- Select Bean: Connection
- Select Bean: SQL Assist SmartGuide
- SQL Assist SmartGuide (Join)
- SQL Assist SmartGuide (Columns)
- SQL Assist SmartGuide (Sort)
- SQL Assist SmartGuide (Mapping)_
- SQL Assist SmartGuide (SQL)
- Select Bean: Visual Composition
- DBNavigator Bean
- Update, Insert, and Delete
- Host Variables

Persistence Builder Introduction

- Persistence Builder Introduction
- Persistence Challenge
- Modern Persistence Framework
- Enterprise JavaBean Tools
- Application Layers
- Key Elements: Business Object
- Key Elements: Relationship
- Key Elements: Transactions
- Key Elements: Mapping

- Key Elements: Datastore
- Development Environment
- Runtime Environment
- Development Paths
- Forward Engineering
- Backward Engineering
- Outside-In Mapping
- Generation Facilities

Persistence Builder - Forward Engineering

- Persistence Builder - Forward Engineering
 - Forward Engineering: Overview
 - Forward Engineering: Sequence
 - Demonstrations: ITSOBANK Model
 - Demonstrations: ITSOBANK Database
 - Model Browser
 - Model Browser: Editors
 - Defining a Class with Attributes
 - Class Editor
 - Simple Relationship
 - Generate the Schema
 - Schema Browser
 - Schema Browser: Editors
 - Schema Browser: Table Editor
 - Schema Browser: Column Editor
 - Schema Browser: Foreign Key Editor
 - Mapping between Object Model and Schema
 - Map Browser
 - Map Browser: Editors
 - Map Browser: Property Map Editor
 - Map Browser: Broken Map
 - Map Browser: Fixing a Broken Map
 - Generation Facilities
 - Schema Browser: DDL Generation
 - Schema Browser: Tabel Creation
 - Model Browser: Generating Domain Classes
 - Model Browser: Generate Service Classes
 - Model Browser: Generate Service Classes RDB
 - Save the Model - Schema - Map
 - Other Persistence Builder Tools
 - SQL Query Tool
 - Status Tool
 - Lab Exercises: MUSIC Model
 - Lab Exercises: MUSIC Database

Persistence Builder Programming

- Persistence Builder Programming
 - Generation of Domain and Service Classes
 - Model Classes: Generation
 - Service Classes: Generation
 - Service Classes: Relational
 - DataStore
 - Home and Business Object Methods
 - Relationship Methods
 - Transaction Basics
 - Programming: Business Class

- Programming: Relationship
- Programming: Scrapbook
- Visual Programming: Palette
- Visual Programming: Composition
- Visual Programming: Table Model
- Business Method Implementation
- Adding Business Logic

Persistence Builder - Backward Engineering

- Persistence Builder - Backward Engineering
 - Backward Engineering
 - Backward Engineering: Sequence
 - Import Tables into a Schema
 - Schema Browser: Tailor the Schema
 - Model Browser: Tailor the Model
 - Map Browser: Verify the Map
 - Generate and Save

Persistence Builder - Relationships and Inheritance

- Persistence Builder - Relationships and Inheritance
 - One-to-Many Relationship
 - One-to-One Relationship
 - Two Tables for One Class
 - Many-to-Many Relationship
 - Inheritance
 - Inheritance: Object Model
 - Inheritance: Schema with Single Table
 - Inheritance: Single Table Mapping Root
 - Inheritance: Single Table Mapping Subclass
 - Inheritance: Schema with Multiple Tables
 - Inheritance: Multiple Table Schema
 - Inheritance: Multiple Table Mapping Root
 - Inheritance: Multiple Table Mapping Subclass

Persistence Builder - Many-to-Many Relationships

- Persistence Builder - Many-to-Many Relationships
 - Many-to-Many Relationship
 - M:M Relationship: Business Model
 - M:M Relationship: Specification
 - M:M Relationship: Intermediate Class
 - M:M Relationship: Programming
 - M:M Relationship: Tailored Methods

PB - Persistence Builder - Transactions and Locking

- PB - Transactions/Locking
 - Transaction Basics
 - Nested Transactions
 - Starting Transactions: Methods
 - Ending Transactions: Commit or Rollback
 - Business Object Versions
 - Transacted Variable
 - Business Transaction
 - Concurrent Transactions
 - Locking Policies: Design Decision
 - Isolation Policies: Runtime Specification per Tx
 - Non-Locking Conflict Detection
 - Visual Programming Transaction Beans
 - Visual Programming: Windows

Persistence Builder Additional Topics

Persistence Builder Additional Topics

Custom Queries: QueryPool class

Custom Queries: Examples

Custom Queries: Implementation

Custom Queries: Usage

Lite Collections: What are they?

Lite Collections: Specification

Lite Collections: Usage

Preload Path

Complex Mapping: Object to Multiple Tables

Complex Mapping: Composers

Complex Mapping: Converters

Model Extension

Using Existing Model in Persistence Builder

Persistence Builder - Deployment

Deployment Issues

Persistence Builder JAR Files

Servlet Development Environment

DEVELOPING OO APPLICATIONS WITH JAVA: THE FULL LIFE CYCLE

The Application Development Process

- Methodologies Vs. Lifecycle Process
 - Basic issues in software development
 - Methodologies
 - Object-oriented methodologies
 - The Software Lifecycle
 - Process Attributes
- Evolution of the Process
 - Classical phases in software development
 - The Waterfall Lifecycle Model
 - Waterfall model
 - The Prototyping Lifecycle Model
 - Prototyping model
 - OO Lifecycle Models
 - Incremental development
 - Iterative development
 - What about Project Management?
 - Spiral model
- A Process
 - Lifecycle Process Models
 - An Application Development Process
 - System Analysis and Design
 - Increments
 - Requirements and Project Planning
 - Analysis and Design
 - Testing
 - Increment Focus over Time
- Methodologies and Work Products
 - Looking at OO methodologies again..
 - Modeling Techniques for SW Development
 - Modeling Views
 - Static models
 - Dynamic models
 - Work Products
 - VMT Analysis Overview
 - VMT System Design Overview
 - VMT Object Design Overview
- Application Development Roles
 - Sample Application Development Project Roles
 - Sample Roles and Responsibilities

Basic Principles of Object-Oriented

- Real-World Modeling
 - Real-world modeling
 - Real-world modeling using objects
 - What is an object?
 - The Object/Message paradigm
 - Basic principles of object-orientation
- Principle 1: Abstract Data Types
 - Principle 1: abstract data types
 - Classes
 - Classes in typed versus typeless languages
 - Benefits of abstract data typing
 - Messages and methods

- Principle 2: Encapsulation
 - Principle 2: encapsulation
- Principle 3: Polymorphism
 - Principle 3: Polymorphism
 - Ad hoc polymorphism - overloading
 - Dynamic binding (late binding)
 - Advantages of dynamically binding overloaded operations
 - Parametric polymorphism
- Principle 4: Inheritance
 - Principle 4: Inheritance
 - Finding Commonality
 - Object at Runtime
 - More inheritance
 - Inheritance of structure
 - Inheritance of behavior
 - AWT class hierarchy
 - Inheritance versus buy-relationships
 - Inheritance versus buy-relationships
 - Single versus multiple inheritance
 - Single versus multiple inheritance
 - Advantages of inheritance
- Principle 5: Object Identity
 - Principle 5: object identity
 - Object identity
 - Object identity
 - Identity: Implementation
 - The type/state/identity trichotomy
 - Inheritance of structure
 - Inheritance of behavior

Lifecycle Process: Analysis

- Actors, Use Cases, and Important Objects
 - Object-oriented analysis and design - basic rules
 - An Application Development Process
 - OO A&D - data-driven versus behavior-driven school
 - OO A&D - a complementary approach
 - Object-oriented Analysis and Design
 - Analysis activities
 - VMT Analysis Overview
 - The automated warehouse case
 - Requirements modeling
 - Requirements modeling - the automated warehouse case
 - Finding candidate objects
 - Finding candidate objects - object behavior stereotypes
 - Finding candidate objects - the automated warehouse case
 - Case of the flying boat
 - Preparing a data dictionary
 - Preparing a data dictionary - the automated warehouse case
 - VMT Analysis Overview
- Object Modeling
 - Object modeling & CRC analysis
 - Object modeling
 - Messages and methods
 - Attributes, values, and domains
 - Classes

- Classes - the automated warehouse case
- Links
- Associations
- Association attributes
- Association attributes versus classified associations
- Associations - the automated warehouse case
- Aggregations
- Constraints
- Constraints - the automated warehouse case
- Inheritance
- Inheritance - the automated warehouse case

CRC Analysis

- CRC analysis
- Responsibility analysis
- Collaborations
- CRC analysis - the automated warehouse case
- VMT Analysis Overview

Dynamic Modeling

- Dynamic modeling
- Messages and responses
- Interaction diagrams
- Interaction diagrams - the automated warehouse case
- Object states
- State transitions
- State transition diagrams
- State transition diagrams - sending messages or responses
- State transition diagrams - conditions/guards
- Relation interaction diagrams / state transition diagrams
- Relation dynamic model / CRC cards
- State transition diagrams - the automated warehouse case
- VMT Analysis Overview
- An Application Development Process

Lifecycle Process: Design

Design Activities

- An Application Development Process
- Design activities
- Object-Oriented analysis and design -- a complementary approach
- Design activities

Application Architecture

- Application architecture
- Application architecture -- issues
- VMT System Design Overview
- Application architecture -- basic principles
- Weak coupling -- information hiding
- Client-server versus peer-to-peer
- Layered construction
- Layered construction: example
- Tree impurity
- Sage complexity metric
- Architectural components
- The "observer" design pattern
- The Model-View-Controller framework

The Model-View-Controller framework in a client/server environment

The Model-View-Controller framework -- the counter example

Advantages of the Model-View-Controller framework

Object Design

VMT Object Design Overview

Object design

Representing associations

Representing associations -- one-way associations

Representing associations -- two-way associations

Representing associations -- classified associations

Representing associations -- association attributes

Derived Data Policy

Persistence

Persistence

Persistence -- translating the object model in an ER model

Persistence -- replicate object data policies

Persistence -- database integrity strategies

Detailed CRC Cards

Detailed CRC cards

Detailed CRC cards -- example

VMT Object Design Overview

An Application Development Process

GUI Design

What is a User Interface?

An Application Development Process

What is a user interface?

What is an OOUI?

Characteristics of an OOUI

Mapping the User Model

Mapping the user model

User's conceptual model

Programmer's model

Designer's model

The usability iceberg

GUI Components

GUI components

Views

Choices

Action choice

Routing choice

Settings choice

Controls

Menus

Entry field

List box

Combination box

Drop-down list

Drop-down combination box

Spin button

Push button

Push button

Predefined push buttons

Radio button

- Value set
- Check box
- Slider
- Group Box

Lifecycle Process: Coding with VisualAge for Java

An Application Development Process

- Java Programming Language

 - What is Java?

 - Java Development Kit (JDK)

 - Build process

 - Applet vs. application

- Basic OO Principles in Java

 - Basic OO Principles in Java

 - Classes and Objects

 - Examples of Sending Messages

 - Java Example (Encapsulation)

 - Polymorphism

 - Inheritance

 - Java Example of Inheritance

 - Access Modifiers

 - Class Methods and Class Variables

 - Declaring Class Variables in Java

- IBM VisualAge for Java

 - VisualAge for Java

 - VisualAge for Java components

 - Anatomy of a Bean

 - VisualAge for Java object creation

 - Visual Composition Editor

 - VisualAge for Java Editor and Debugger

Java Class Hierarchy

- Java Class Hierarchy

 - Java Class Hierarchy

 - Packages

 - Interfaces

 - Single vs. multiple inheritance

 - Combining Classes and Interfaces

 - Interface example

 - Interface flexibility

 - Packages

 - java.lang package

 - java.util package

 - java.awt package

 - java.applet package

 - java.io package

 - java.lang.reflect package

 - java.net package

 - java.sql package

 - java.beans package

 - Java - Remote Method Invocation (RMI)

 - CORBA

 - Object Request Broker (ORB)

 - Static/Dynamic Invocation

Lifecycle Process: Testing

- Testing the Application

An Application Development Process
Traditional Testing in the Lifecycle Process
testing
Integration testing
Use Case testing
Use Case testing example
Test objects
Acceptance testing
Testing of analysis/design work products

Versioning and Building with VisualAge for Java

Versioning and Building with VisualAge for Java
An Application Development Process
Class/Package/Project Containment
Versioning Control
A Sample Flow
Class/Package/Project Containment
Workspace vs. Repository
Exporting the Code
Java Runtime Execution Environment