ProTech

protechtraining.com

"Charting the Course ...

... to Your Success!"

Advanced Java Level III

Course Summary

Description

This course teaches developers and architects how to design and build easily managed and maintainable J2EE applications by applying J2EE and Gang of Four design patterns.

Objectives

At the end of this course, students will be able to:

- Use commonly used J2EE patterns to develop faster and better J2EE applications
- Recognize potential architecture and design problems early on and design and implement an optimal solution
- Select the best combination and implementation of patterns for problem at hand
- Implement design pattern solutions with Servlets, JSPs, with and without EJBs

Topics

- Overview of J2EE Architecture
- UML - Unified Modeling Language
- Object-Oriented Design Principle
- Basic Design Patterns
- JSP Standard Tag Library 1.1
- Review of MVC Paradigm
- Struts
- J2EE Design Patterns
- Servlet and JSP Filter
- Annotations and Servlet
- Overview of Enterprise JavaBeans
- Demo in Creating Enterprise Session Bean 2.1

Audience

This course is designed for enterprise architects, Java technology and J2EE software developers with a good understanding of Object Oriented design and programming.

Prerequisites

This course assumes that you have general competence in Object Oriented design and programming. Proficiency in Java programming is preferred.

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Advanced Java Level III

Course Outline

I. J2EE Architecture
- A. J2EE Architecture
- B. Features
- C. Benefits
- D. Communication Technologies
- E. Messaging Technologies
- F. J2EE Architecture
- G. J2EE Modules
- H. HTTP Servlet
- I. JavaServer Pages
- J. Model-View-Controller
- K. Struts
- L. JSP Standard Library - JSTL
- M. Enterprise Java Beans

II. Overview of UML
- A. Types of UML
- B. Use Case Diagram
- C. Class Diagram
- D. Association
- E. Aggregate Association
- F. Generalization
- G. Association Specifiers
- H. Sequence Diagram

III. Object Oriented Design Concepts
- A. Cohesion
- B. Encapsulation
- C. Coupling
- D. Composition
- E. Interface Inheritancce
- F. Polymorphism
- G. OO Design Principles

IV. Basic Design Patterns
- A. Interface
- B. Abstract Parent Class
- C. Private Methods
- D. Accessor Methods
- E. Constant Data Manager
- F. Immutable Object
- G. Monitor

V. Gang of Four Design Patterns
- A. Template Method
- B. Strategy
- C. Factory Method
- D. Singleton
- E. Adapter
  - 1. Class Adapter
  - 2. Object Adapter
- F. Façade

VI. Review of MVC Paradigm

VII. JSP Standard Tag Libraries 1.1
- A. JSTL 1.1 Libraries
- B. Expression Language
- C. EL Implicit Variables
- D. JSTL Operator
- E. Using JSTL Library
- F. Common Core Library Actions
  - 1. <c:out>
  - 2. <c:set>
  - 3. <c:forEach>
  - 4. <c:choose>
  - 5. <c:if>
  - 6. <c:otherwise>
  - 7. <c:when>

VIII.Struts Framework
- A. Struts Overview
- B. Struts Components
- C. Dynamic Web Project to Support Struts
- D. Struts HTML input form
  - 1. HTML “form” tags
  - 2. The <html:form /> tag
  - 3. The <html:text /> tag
  - 4. The <html:submit /> tag
  - 5. The <html:reset /> tag
- E. ActionServlet – The Controller
  - 1. ActionServlet
  - 2. ActionServlet Class
  - 3. How it Works
  - 4. Configuring the ActionServlet
- F. The struts-config.xml
  - 1. The <form-beans> element
  - 2. The <action-mappings> element
  - 3. Other major elements
- G. ActionForm – The View
  - 1. ActionForm Class
  - 2. Writing AddEmployeeForm class

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

PT7352_ADVANCEDJAVALEVELIII.DOC

Advanced Java Level III

Course Outline (cont'd)

3. Entry in struts-config.xml
4. ActionForm Methods

H. Action – The Controller
1. Action class
2. The execute () method
3. Writing AddEmployeeAction class
4. Action class – an example
5. Entry in struts-config.xml

I. Validating Data in the ActionForm
1. The validate() Method in ActionForm
2. Entry in struts-comfig.xml

IX. J2EE Design Patterns
A. Tier Approach
B. Presentation Tier Patterns
C. Business Tier Patterns
D. Integration Tier Patterns

X. Applying J2EE Design Patterns
A. Service Locator
B. Data Access Object
C. Façade
D. Transfer Object

XI. Servlet and JSP Filters
A. Servlet and JSP Filters
B. What is a Filter?
C. Typical Uses of Filter
D. How Filters Work
E. Filter's Lifecycle
F. Writing the SimpleFilter
G. Deployment Descriptor
H. Multiple Patterns in Mapping

XII. Annotations
A. Annotations
B. Allowable Annotations in a Servlet
C. PoostConstruct Annotation
D. PreDestroy Annotation
E. Resource Annotation
F. Environment Entry Variable
G. DataSource

XIII.Overview of EJB
A. J2EE Containers
B. What is EJB ?
C. EJB Framework Components
D. EJB Server
E. EJB Container
F. EJB Components
G. EJB Deployment Descriptors
H. Roles
I. Benefits
J. When to Use EJB
K. EJB Architecture
L. Component Interface Architecture
M. Home Interface Architecture
N. Types of EnterpriseBeans
O. EJB Session Beans
P. EJB Entity Beans
Q. Message Driven Beans

XIV. Appendix A: Creating Enterprise Session Beans 2.1 Using JBoss 5.0
A. Business Delegate
B. Session Facade

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically