Mastering JEE Design Patterns

Course Summary

Description

Geared for experienced enterprise Java (JEE) developers, *Mastering JEE Design Patterns* is a lab-intensive Java / JEE design patterns training course which explores the many sophisticated JEE-oriented design patterns and how to use these patterns to develop solid, robust and reusable JEE applications. Technologies such as JPA and EJB3, as well as frameworks such as Spring, web services, and rich interfaces, have significantly impacted previous generations of design patterns. Many of these technologies were heavily influenced by the very problems that previous design patterns addressed. While the basic patterns still ring true, the more advanced patterns have evolved into more robust solutions for secure, stable and scalable enterprise applications.

Topics

- Introduction to Design Patterns
- "Gang of Four" Design Patterns
- Base Patterns
- Business Tier Patterns
- Integration Tier Patterns
- Presentation Tier Patterns
- Crosscutting Patterns
- Working with Patterns

Audience

This is an intermediate level Java EE (JEE) developer course, designed for experienced Java developers, new to JEE, that need to further extend their skills in web development and Struts.

Prerequisites

Students should have practical skills equivalent to or should have received training in the following topic(s) as a pre-requisite:

- Understanding Internet Architectures
- Mastering Java for OO Developers
- Mastering JEE Web Application Development

Attendees should have an extensive working knowledge in developing basic Java applications.

Duration

Five days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Mastering JEE Design Patterns

Course Outline

I. Introduction to Design Patterns
- A. Introduction to Design Patterns
  1. Patterns: Basics
  2. Why Choose a Design Pattern?
  3. Patterns: What is Not a Pattern?
  4. Anti-Patterns
  5. "Gang of Four" Patterns
  6. JEE Base Patterns

II. "Gang of Four" Design Patterns
- A. Patterns and Principles
  1. Patterns and Principles
  2. Gang of Four Patterns
  3. Synopsis of GoF Patterns
- B. Creational Patterns
  1. Factory Method
  2. AbstractFactory
  3. Singleton
  4. Singleton: Multithreading
- C. Structural Patterns
  1. Façade
  2. Composite
  3. Adapter
  4. Proxy
- D. Behavioral Patterns
  1. Iterator
  2. Observer
  3. Command

III. Base Patterns
- A. Base Patterns
  1. Base Patterns and GoF Patterns
  2. Gateway
  3. Separated Interface
  4. Registry
  5. Special Case

IV. Business Tier Patterns
- A. Business Tier Patterns
  1. Domain Model Pattern
  2. Service Locator
  3. Dependency Injection
  4. Service Locator vs. Dependency Injection
  5. Business Delegate
  6. Session Façade
  7. Application Service
  8. Identity Map Pattern

V. Integration Tier Patterns
- A. Integration Tier Patterns
  1. Remote Façade Pattern
  2. Data Transfer Object
  3. Data Access Object
  4. DAO and DTO in JEE : Java Persistence API
  5. Service Activator
  6. Web Service Broker

VI. Presentation Tier Patterns
- A. The MVC "Pattern"
  1. Model View Controller Pattern
- B. Controller Patterns
  1. Front Controller
  2. Application Controller
  3. Combining Patterns
- C. Supporting Patterns
  1. Context Object
  2. Intercepting Filter
- D. View-related Patterns
  1. View Helper
  2. Composite View
- E. Combined Patterns
  1. Service to Worker
  2. Dispatcher View

VII. Crosscutting Patterns
- A. Crosscutting Patterns
  1. Authentication Enforcer
  2. Authorization Enforcer
  3. Intercepting Validator
  4. Secure Base Action
  5. Secure Logger
  6. Secure Pipe
  7. Secure Service Proxy
  8. Intercepting Web Agent

VIII. Working with Patterns
- A. Applying Patterns
  1. Usefulness of Patterns
  2. Selecting the Right Pattern
  3. Adapting an Existing Pattern
  4. Creating a New Pattern
  5. Language Considerations
- B. Exploring Frameworks (Optional)
  1. Patterns and Productivity
  2. Frameworks
  3. Introduction to Spring
  4. What is Struts?
  5. Overview of the Struts 2.0 Architecture