Java Design Patterns

Duration: 4 Days

What you will learn

The Java language and popular Java-based frameworks incorporate more proven development practices into their programming interfaces with each major release. These practices, referred to as design patterns, document well-known names, code implementation and re-factoring techniques, and the risks and trade-offs associated with using them. In design patterns, the responsibility of each component is identified by role. The conventions of design pattern documentation make it easier for development teams to communicate their programming intentions and provide a reference point for the entire Java development community.

The Java Patterns course reviews common and emerging patterns specific to Java SDK and EE development. Students learn the depth and evolution of pattern-based techniques in Java with particular emphasis on Java EE 6 conventions. The lab exercises show students how to identify, apply and re-factor selected patterns into code, using a NetBeans or Eclipse IDE and the GlassFish Application Server v3. Students also learn a subset of UML notation to expedite communicating through design instead of code.

Learn To:

Dinstinguish between Java EE 5 and Java EE 6 pattern-based features
Implement relevent patterns in each tier of the Java EE environment
Re-factor code to improve inter-tier communications
Relate pattern-based development to an implementation architecture
Apply object-oriented principles and design guidelines
Implement well-known patterns to Java-specific code problems

A Live Virtual Class (LVC) is exclusively for registered students; unregistered individuals may not view an LVC at any time. Registered students must view the class from the country listed in the registration form. Unauthorized recording, copying, or transmission of LVC content may not be made.

Audience

Application Developers
Architect
J2EE Developer
Java Developers
Java EE Developers

Related Training

Required Prerequisites

Experience with Java SE and Java EE development

Developing Applications for the Java EE 6 Platform

Course Objectives

Identify key design principles of object-oriented development

Apply Java-specific implementation techniques to well-known patterns

Use patterns to complete a Java application design

Use patterns to complete a web-tier application design

Use patterns to complete a business-tier application design

Use patterns to improve communication between Java EE tiers

Identify and refactor anti-patterns in working code

Using part of a sample architecture scheme, select design patterns for implementing the scheme

Course Topics

Reviewing Object-Oriented Principles in Java

Describe how OO concepts apply to Java
Describe how OO principles apply to Java
List the goals of an OO language
Interpret Unified Modeling Language (UML) notation and create UML diagrams
Identify selected design patterns

Reviewing Gang of Four Patterns

List key behavioral, creational and structural patterns
Apply the Facade pattern
Apply the Strategy pattern
Apply the Observer pattern
Apply the Composite pattern
Review the Model-View-Controller (MVC) patterns

Implementing Patterns in Java

Use implementation patterns designed for Java
List forces affecting class, state, and behavioral patterns
Describe how patterns, idioms and refactoring differ from each other

Exploring Changes in Java EE Technology

Describe the design goals of the Java EE model
Describe improvements in the Java EE 6 model

Implementing Integration Patterns

Describe design patterns for the integration tier
Review Java EE integration changes that apply design patterns
Identify use cases for applying integration tier patterns

Implementing Patterns in Business Components
Describe the role of an enterprise bean
Describe design patterns for the business tier

Implementing Infrastructural Patterns in Java EE
Describe the role of infrastructural Java EE patterns
Describe the Service Starter pattern
Describe the Singleton pattern
Describe the Bean Locator pattern
Describe the Resource Binder pattern

Implementing More Infrastructure Patterns
Describe how Java EE interceptors work
Describe the Dependency Injection Extender pattern
Describe the Payload Extractor pattern
Describe the Context Holder pattern
Describe the Thread Tracker pattern

Exploring Anti-Patterns
Describe the Law of Leaky Abstractions
Define AntiPatterns
Describe Integration Tier AntiPatterns
Describe Business Tier AntiPatterns
Describe Presentation Tier AntiPatterns

Selecting Patterns for Architecture
Define the roles of architect, designer, and developer
Describe the relationship between design patterns and architecture
List guidelines for applying patterns to an architectural solution