

Java Design Patterns Ed 1

This Java Design Patterns training reviews common patterns specific to Java SDK & EE development. Lab exercises teach you to identify, apply and re-factor these patterns into code, using a NetBeans or Eclipse IDE and the GlassFish Application Server v3.

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

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
 - $\circ\,$ Apply the Strategy pattern
 - $\circ\,$ Apply the Observer pattern
 - Apply the Composite pattern
 - $\circ\,$ Review the Model-View-Controller (MVC) patterns
- Implementing Patterns in Java
 - $\circ\,$ Use implementation patterns designed for Java
 - List forces affecting class, state, and behavioral patterns
 - $\circ\,$ Describe how patterns, idioms and refactoring differ from each other
- Exploring Changes in Java EE Technology
 - $\circ\,$ Describe the design goals of the Java EE model
 - $\circ\,$ Describe improvements in the Java EE 6 model
- Implementing Integration Patterns



İçerenköy Mah. Eski Üsküdar Yolu Cad. Bodur İş Merkezi No:8 Kat:3 D:13, İstanbul, Ataşehir, 34752, Türkiye www.methodtr.com



- 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
 - $\circ\,$ Describe the Singleton pattern
 - $\circ\,$ Describe the Bean Locator pattern
 - $\circ\,$ Describe the Resource Binder pattern
- Implementing More Infrastructure Patterns
 - $\circ\,$ Describe how Java EE interceptors work
 - Describe the Dependency Injection Extender pattern
 - $\circ\,$ Describe the Payload Extractor pattern
 - Describe the Context Holder pattern
 - $\circ\,$ Describe the Thread Tracker pattern
- Exploring Anti-Patterns
 - $\circ\,$ Describe the Law of Leaky Abstractions
 - Define AntiPatterns
 - Describe Integration Tier AntiPatterns
 - Describe Business Tier AntiPatterns
 - $\circ\,$ Describe Presentation Tier AntiPatterns
- Selecting Patterns for Architecture
 - $\circ\,$ Define the roles of architect, designer, and developer
 - $\circ\,$ Describe the relationship between design patterns and architecture
 - List guidelines for applying patterns to an architectural solution

