

Object-Oriented Design Patterns and Best Practices in Java (TT1250)

Length: 5 days

Description

The course also contains several “thinking and drawing” lab exercises as a component of the object oriented overview portion of the training course.

This comprehensive training course begins with a review of core concepts of Object Oriented Analysis & Design using UML (approximately one day). Students will learn how to effectively assign responsibilities using the patterns and principles of GRASP (General Responsibility Assignment Software Patterns). Throughout the remainder of the course we explore the following patterns, varying the levels of coverage to drill down on the most commonly used Patterns, and to simply survey others. Students will compare and contrast the patterns and explore the advantages and disadvantages of using certain patterns for explicit development functions in the Java environment.

Audience

Programmers with prior Java and Object-Oriented exposure and background

Prerequisite

Attendees should be familiar with UML and have basic programming experience in Java. This course is not recommended for developers new to Java programming.

Topics

SESSION: REVIEW OF OOAD USING UML

- Concepts of Analysis and Design
- Processes
- Activities
- Classes and Objects
- Providing Services
- Messaging
- Interfaces
- Relationships
- Types
- Dependencies
- Associations
- Generalizations
- Collaborations
- States and Activities
- Events
- State transition

SESSION: OBJECT ORIENTED ANALYSIS

- Use Cases
- Use Case Diagrams
- Defining Use Cases
- Extending Use Cases
- Variations On Use Cases

SESSION: OBJECT ORIENTED DESIGN

- Static Design Concepts
- Multiplicity
- Interface And Implementation
- Good Abstractions
- Constraints And Qualifiers
- Dynamic Design Concepts
- Sequence Diagrams
- Collaboration Diagrams
- Identifying Messages
- Characterizing Messages

Object-Oriented Design Patterns and Best Practices in Java (TT1250)

- State Diagram
- GRASP patterns/principles
- Controller
- Creator
- Information Expert
- Law of Demeter
- Low Coupling/High Cohesion
- Polymorphism
- Pure Fabrication
- Good/Bad packaging principles
- Coupling (allowed and/or required communication paths), layering, and dependencies
- Patterns In Design
- Mapping to Databases
- Mapping to User Interfaces
- About Frameworks
- Designing Components and Interfaces
- Impact of Threading on Singletons
- Options for Addressing Threading Issues
- Challenge of Double-Checked Locking in Java
- Challenge of Clustered Servers
- Builder Design Pattern
- Standard Pattern Description
- Factory Method Design Pattern
- Standard Pattern Description
- Applying the Different Factory Patterns
- Prototype Design Pattern
- Standard Pattern Description
- Survey of Creational Patterns

SESSION: INTRODUCTION TO DESIGN PATTERNS

- Overview of Patterns
- What is a Pattern?
- Why Patterns?
- Crucial Qualities of Good Patterns
- What are NOT Patterns?
- Classifying and Describing Patterns
- Overview of Patterns by Category

SESSION: CREATIONAL PATTERNS

- Class Creation vs. Object Creation
- Abstract Factory Pattern
- Description
- When to Apply
- Structure
- Example
- Benefits and Liabilities
- Issues
- Singleton Design Factory Pattern
 - Standard Pattern Description

SESSION: STRUCTURAL PATTERNS

- Overview of Structural Patterns
- Composite Pattern
- Standard Pattern Description
- Child Management
- Composite Pattern in Java Platform
- Adapter Pattern
- Standard Pattern Description
- Two-Way Adaptors
- Adaptor Pattern in Java Platform
- Proxy Pattern
- Standard Pattern Description
- Types of Proxies
- Proxy Pattern and Web Services
- Bridge Pattern
- Standard Pattern Description
- Façade Pattern
- Standard Pattern Description
- Decorator
- Standard Pattern Description
- Survey of Structural Patterns

SESSION: BEHAVIORAL PATTERNS

- Observer Pattern
- Standard Pattern Description

Object-Oriented Design Patterns and Best Practices in Java (TT1250)

- Java Built-in Support of Observer Pattern
- Observer Pattern in Java Platform
- Strategy Pattern
- Standard Pattern Description
- Iterator Pattern
- Standard Pattern Description
- Controlling Iteration
- Robust Iterators
- Aggregate Classes
- Java Iterator Implementation
- Visitor Pattern
- Standard Pattern Description
- Interpreter Pattern
- Standard Pattern Description
- Chain of Responsibility Pattern
- Standard Pattern Description
- Command Pattern
- Standard Pattern Description
- Command Pattern in Java Platform
- Mediator Pattern
- Standard Pattern Description
- State Pattern
- Standard Pattern Description
- Comparison and Summary

SESSION: APPLYING PATTERNS

- Selecting the Right Pattern
- Adapting an Existing Pattern
- Creating a New Pattern
- Implementation Considerations
- Challenges in Working With Patterns
- Anti-Patterns

APPENDIX: INTRODUCTION TO UML (OPTIONAL)