Design Patterns in Java 7.0

Course Summary

Description

This course seeks to develop, for the experienced Java programmer, a strong, shared vocabulary of design patterns and best practices. The course begins with a discussion of how to recognize and apply design patterns - that is, how to incorporate pattern awareness into one's own analysis, design, and implementation practices. The main body of the course focuses on the Gang of Four design patterns, with a chapter each on creational, behavioral, and structural patterns. Classroom time is about evenly split between discussion, group design exercises, and coding labs to reinforce finer points of important patterns

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

- Start to think in terms of design patterns.
- Recognize and apply patterns to specific software development problems.
- Use known patterns as a shared vocabulary in designing and discussing solutions.
- Use Factories and Singletons to control object creation, for a variety of reasons.
- Use Observers, Observables, and Model/View/Controller systems to decouple application behavior and preserve code scalability.
- Understand the full motivation for the Command pattern and take advantage of Command frameworks in JFC.
- Implement Adapters, rather than building redundant classes or creating intermediate data structures for consumption by existing code.
- Understand and apply a range of other J2SE and J2EE patterns to improve code quality and scalability, and to produce high-quality solutions right off the bat.

Topics

- Recognizing and Applying Patterns
- Creational Patterns
- Behavioral Patterns
- Structural Patterns
- J2EE Patterns

Prerequisites

- Solid Java programming experience is essential - especially object-oriented use of the language. Language features and techniques that are integral to some lab exercises include interfaces and abstract classes, threading, generics and collections, and recursive methods.
- Previous experience with UML (Unified Modeling Language) will be helpful, but is not critical. The course uses UML class diagrams extensively but keeps notation fairly simple, and also includes a quick-reference appendix.

Duration

Three 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

Design Patterns in Java 7.0

Course Outline

I. Recognizing and Applying Patterns
- A. Design Patterns
- B. Defining a Pattern
- C. Unified Modeling Language
- D. Seeing Patterns
- E. Warning Signs and Pitfalls

II. Creational Patterns
- A. Factory Patterns
- B. The Singleton Pattern
- C. APIs and Providers
- D. Cascading Factories

III. Behavioral Patterns
- A. The Strategy Pattern
- B. The Template Method Pattern
- C. The Observer Pattern
- D. The Model/View/Controller Pattern
- E. The Command Pattern
- F. The Chain of Responsibility Pattern

IV. Structural Patterns
- A. The Composite Pattern
- B. The Adapter Pattern
- C. The Decorator Pattern
- D. The Façade Pattern
- E. The Flyweight Pattern

V. J2EE Patterns
- A. Model/View/Controller, Redux
- B. The Intercepting Filter Pattern
- C. The Front and Application Controller Patterns
- D. The Business Delegate Pattern
- E. The Service Locator Pattern
- F. The Transfer Object Pattern
- G. The Composite Entity Pattern
- H. The Data Access Object Pattern

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