Java Design Patterns

Java Design Patterns

J7 - Version: 3

5 days Course

Description:

In this course, programmers will be introduced to ideas and techniques commonly referred to as Java language patterns. Patterns are reusable solutions to recurring problems arising during software development. This course will introduce the programmer to common patterns and their implementation in the Java language. Patterns will be associated with their uses within the Java API, followed by design and implementation exercises to correlate several design patterns.

To make the best from the course, the first day will be devoted to some java language core issues such as reflection and memory management. The course starts with a "Pitfalls" section in which we will discuss some common misunderstandings in the Java language, clarify them and learn how to avoid them.

Intended audience:

This course is intended for Java programmers, Java designers and Object Oriented analysts that need to design system-level, software-level or module-level components to be implemented in the Java programming language.

Prerequisites:

At least four months of Experience with Java Programming
Fundamental knowledge of software design principles

Objectives:

Learn what design patterns are useful for

Java programmers will associate the ideas with a real-life usage of patterns

Java programmers will be able to apply design patterns in their design work and implement them in the Java programming language

Get to know the advanced topics pertinent to Java programming

Learn to avoid common design and implementation mistakes and pitfalls

Topics:

Java Pitfalls

- Common Java pitfalls and how to avoid them
- Equals(...) vs. = =
- Cloning objects

Memory Management

- GC Overview.
- The finalize method
- Using reference objects

The reflection mechanism

- The java.lang.Class
- Dynamic instantiation
- Method invocation.
- Design Issues and limitations.

º Introduction to Design Patterns

Design Principles

- The Open-Closed Principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Single Responsibility Principle
- Liskov Substitution Principle & Design by Contract

UML Overview

- Introduction to models
- Class diagrams
- Association
- Aggregation
- Generalization

Fundamental design patterns

- Delegation
- Strategy
- Interface
- Immutable
- Marker interface
- Proxy
- The Dynamic Proxy class

Creational patterns

- Singleton
- Factory method
- Abstract factory
- Object pool
- The Prototype Pattern

Structural Patterns

- Composite
- Adapter
- Iterator
- Fa?ade
- Decorator
- Bridge
- Double Dispatch idiom

Behavioral Patterns

- Command
- Template method
- State
- The Flyweight Pattern
- Observer
- Visitor

º Appendix 1. Design Patterns Catalog

º Appendix 2. References.