



Java Design Patterns

Abstract

Design patterns are a must for every developer. Design patterns help the developers write extensible and maintainable code.

Design patterns also provide the developers with common vocabulary for design and allow easy classification of conceptual problems.

The course will focus on the well known GoF patterns and their appliance in the Java language.

Target Audience

Java developers.

Prerequisites

Familiarity with the Java language.

Content:

UML Recap (1 hour):

- Class Diagrams.
- Sequence Diagrams.
- Interaction Diagrams.

Design Principles (2 hours):

- Open/Closed Principle.
- The Liskov Substitution Principle.
- Design by Contract.
- Inversion of Control & Dependency Injection.
- Composition over Inheritance.

Creational Patterns (6 hours)

- Factory.



Abstract Factory.
Builder.
Prototype.
Singleton.

Structural Patterns (6 hours)

Adapter.
Bridge.
Composite.
Decorator.
Façade.
Flyweight.
Proxy.

Behavioral Patterns (8 hours).

Command.
Mediator.
Chain of Responsibility.
Iterator.
Memento.
Observer.
State.
Strategy.
Template Method.
Interpreter.
Visitor.

Overview of non GoF Patterns (1 hour)

Duration: 3 days.