

Introduction to Design Principles, Patterns and Concurrency Management with Java

Prerequisites: The participants must have strong programming skills in Object Oriented Programming with java.

Objectives

This course introduces following topics in practical oriented approach with Java

- SOLID Design Principles
- Common Design patterns
- Error handling and logging
- Multithreading components
- Java Concurrency components and Usage
- Stream programming
- Database management and practices

Training Strategies: The theoretical topics are discussed interactively and technical details are demonstrated with practical examples. The participants work on the hands on exercises which strengthen the concepts learned.

Course Material: Presentations, documents and sample programs with exercise case studies get shared with the participants.

Machine Setup: The Intel core compatible CPU with minimum 8GB RAM and 500GB HDD with Windows 10 64 bit System and JDK1.8 64 bit, MySQL5.1 Database server and client, Adobe Acrobat Reader, WinZip to be installed. The zip files for Eclipse-Jee-Photon, Apache Tomcat 8.0 web server to be made available.

The participants must have admin rights on their systems to run and configure the applications.

Live internet connection with reasonable speed and download permissions should be provided in the training room to update the required patches.

Training Duration: 7 days online in WebEx mode.

Instructor: Prakash Badhe.

Course Plan

Day1

The participants learn about the design principles and techniques for better design by understanding the problems and implement the solutions by working on example programs and implement the case study implementations.

Design Principles and Practices

- Overview design practices and techniques.

- Design guidelines
- **SOLID Principles**
 - Single responsibility principle : Single behaviour for a class or component
 - Open-closed principle.(Open for extension and closed for modification)
 - Liskov substitution principle : children subclasses in parent world
 - Interface segregation principle.(Separate the required interfaces)
 - Dependency inversion principle: Dependencies managed by someone else.
 - Example scenarios
 - Case studies

Domain Driven Design

- Conventions database operations and queries management.
- Need of Domain Objects
- Factory and Service objects
- Domain Driven Design overview
- Components of DDD
- Advantage of DDD

Error handling and logging

- The error handling process
- Exception management and tracing
- Error logging
- Logging tools and frameworks
- Validation and error logging

Day2

Immutability with Java

- What is Immutability?
- Immutability vs. final
- Advantages of immutability
- Design your own immutable objects

Stream Programming

- Java collections overview
- Streams for collections data management
- Streams API overview
- Map,forEach,Collect,filtering operations
- The flatMap and reduce operations
- Terminate operations
- Functional programming and Lambda expressions
- Functions as objects/parameters

Java Reflection API

- Usage of Reflection API

- Analyse class, methods and variables
- The reflection inside frameworks
- Dependency injection by frameworks

Day3

The Garbage Collector

- The GC types and usage
- The GC control parameters
- Role of GC in performance
- Soft and Weak References
- Tune the GC

Thread Programming

- Java threading overview
- Daemon threads
- Thread and Runnable
- Thread properties and operations
- Thread debugging
- Thread dumps
- Control shared usage of resources in threads
- Synchronized methods vs. Synchronized Blocks.
- The resource locks
- The volatile keyword
- Wait and notify operations
- Deadlocks
- Thread Executors

Day4 and Day5

Java Concurrency Enhancements

- ExecutorService implementations
- The Callable interface to return response
- Synchronous vs. Asynchronous executions
- The Future with Asynchronous processing
- The CompletableFutures usage
- Usage of Latch
- Controlled synchronization with Semaphore
- Usage of Locks
- ACID properties for operations on data
- Atomic operations with Atomic Variables
- Conventional Queues
- The BlockingQueue usage

Day6

Database Design and Query management

- Normalization to avoid redundant data tables

- Queries with Joins, Group by, Having and
- Pivot and Unpivot Queries with Case expression
- Simple Queries with timeseries tables
- The name value pair tables pattern
- The audit tables vs. the audit via bitemporal tables
- The Query plans and Indexing
- The Database partitions
- Optimizing the database design for performance

Day7

The participants learn about the design patterns by way of theoretical discussion about the problems and solutions offered by patterns with simulating the sample example programs and implement the sample hands on exercise scenarios.

Design Patterns Introduction

- Why Design patterns?
- Problems solved by Design Patterns
- Pattern motivation and requirements
- Techniques and implementations
- Design pattern scenarios
- Pattern advantage
- Identify the particular pattern
- Apply design patterns in implementations
- Design pattern categories
- **GOF** design patterns and categories
- Pros and Cons
- Anti-patterns

GOF Creational Patterns

- The Factory Pattern
- The Abstract Factory Pattern
- The Singleton Pattern
- The Builder and Prototype Pattern

GOF Structural Patterns

- The Adapter Pattern
- ~~The Bridge Pattern~~
- ~~The Composite Pattern~~
- The Decorator Pattern
- ~~The Façade Pattern~~
- ~~The Flyweight Pattern~~
- ~~The Proxy Pattern~~

GOF Behavioural Patterns

- ~~Chain of Responsibility~~
- ~~The Command Pattern~~
- ~~The Interpreter Pattern~~
- ~~The Iterator Pattern~~
- ~~The Mediator Pattern~~
- ~~The Memento Pattern~~
- The Observer Pattern
- ~~The State Pattern~~
- The Strategy Pattern
- The Template Method Pattern
- ~~The Visitor Pattern~~
