Course SOLID, GOF Design pattern with Java

About this course

It is master course on GOF design patterns & SOLID design principles using Java. This course include indepth coverage of some selected key Java topics to provide solid foundation required for successful software development. Training consist of live coding and real life examples.

Course prerequisites

Good hands on working knowledge on Java

Skill level

Intermediate to advance Level

Duration

Six Half day session (4Hr session)

Some selected core Java topics

Session 1:

Fundamentals

- Quick Revision of OOPs Concept
- Undersanding Abstraction, encapsulation
- Interface vs Abstract class, , loose coupling and high cohesion

Enum in java

- Customise constant value with Enum Constructor,Enum Collections EnumMap, EnumSet
- Enum Abstract function, Singleton Design Pattern

Serilization and De-Serilization

- Serilization and static data
- Serilization and inheritance, composition, Instance control flow, externalization

Collection

- Collection interfaces and classes introduction
- iterators: ListIterator, Iterator, Enumeration
- Comparable, Comparator
- List implemntations: ArrayList, LinkedList, Vector, perforance, use cases
- Set implemntations: HashSet, LinkedHashSet, TreeSet, User define key
- Map implemntations: HashMap, LinkedHashMap, TreeMap, IdentityHashMap, WeakHashMap, ConcurrentHashMap
- Queue, PriorityQueue examples and applications

Session 2:

Generics

- Generics in lists,Autoboxing with generics
- Parameterized Types,Generic lists vs typed arrays
- Create your own generic type, Generics with constructors
- How inheritance works with generic types ,Generics wildcards
- Generics upper bound wildcards ,Generics with multiple type parameters
- Using generics with static methods

Threads

- Introduction to Threads, creation for child threads
- Thread Life cycle
- Need of synchronization, synchronization vs volatile
- Inter-thread communication, Producer and consumer problem
- JUC package, Callable vs Runable, ExecutorService , creating thread pool, submitting jobs
- BlockingArrayList
- Semaphore, CountDownLatch, CyclicBarrier,Exchanger
- Concurrent Collections: ConcurrentHashMap, CopyOnWriteArrayList, CopyOnWriteArraySet
- Apply Atomic Variables
- Applying Locks: ReentrantLock, ReentrantReadWriteLock ,Conditions

Session 3:

Reflection API

- Java reflection introduction
- Using reflection :Classes,Constructors,Fields,Methods,Private Fields and Methods, Array
- Dynamic class loading

Java Annotations

- Built-in Java Annotations: @Deprecated, @Override, @SuppressWarnings
- Custom Annotations
- Annotation processing using java reflection
- @Retention, @Target annotation

Session 4:

SOLID principles

- Single Responsibility Principle
- Open-Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle
- Hands On, Lab assignments

Design Patterns Introduction

- Categories of design patterns

Creational Patterns

Singleton Design Patterns

- Singleton Design Patterns, Example UML
- Eager Singleton, Lazy Singleton, Concurrency Issues
- Pitfall : relflectoion, serilization, multiple class loader
- Initialization Holder
- Singleton Implementation using Enum

Builder Design Patterns

- Builder Design Patterns, Example UML
- Implementation & Design Considerations, Example in Jdk

Factory Method Design Patterns

- Factory Method Design Patterns, Example UML
- Factory Method - Implementation & Design Considerations

Prototype Design Patterns

- PrototypeImplementation Steps,UML

Abstract Factory Design Patterns

- Abstract Factory Design Patterns, Example UML
- Implementation & Design Considerations
- Comparison with Factory Method

Object Pool Design Patterns, Example UML

- Implementation Steps

Session 5:

Structural Design Patterns

Adapter Introduction

- Adapter Design Patterns, Example UML
- Implementation & Design Considerations

Bridge Design Patterns

- Implementation Steps, Example UML

Decorator Design Patterns

- Implementation & Design Considerations

Facade Design Patterns

- Implementation & Design Considerations

Flyweight Design Patterns

- Implementation & Design Considerations
- Comparison with Object Pool

Proxy Design Patterns

- Dynamic Proxy Implementation
- Implementation & Design Considerations

Session 6:

Behavioral Patterns

Chain of Responsibility Design Patterns

- Implementation Steps, Example UML
- Comparison with Command

Command Design Patterns

- Implementation & Design Considerations
- Comparison with Strategy

Mediator Design Patterns
- Implementation & Design Considerations
- Comparison with Observer

Iterator Design Patterns
- Implementation & Design Considerations

Memento Design Patterns
- Implementation & Design Considerations

Observer Design Patterns
- Implementation & Design Considerations
- Comparison with Mediator

Strategy Design Patterns
- Implementation & Design Considerations

Template Method Design Patterns
- Implementation & Design Considerations
- Comparison with Strategy

Visitor Design Patterns
- Implementation & Design Considerations

Null Object Patterns
- Implementation & Design Considerations
- Comparison with Proxy

J2EE Patterns Introduction
- MVC design pattern with Spring framework
- DAO, DTO, Service layer, contorller layer demostraction