Agenda

* What is Spring Framework
* Why Spring Framework
* How to Create a Spring based Java Application
* Important Design Patterns used in Spring Framework
* IoC
* DI
* Ways of Configuring Beans
* Bean Scopes (Scope of Beans)
* Application Context
* Bean LifeCycle

For Hands-on

Use cloud Lab

* Open RDP and type ip address and Username
* Click “Connect” button and enter the password

Spring Framework

* It’s very popular Java based Framework
* It’s mainly used to create Loosely coupled Enterprise Java Application.

Loosely coupled

Enterprise Java Application

Tightly Coupled – Example

POJO – Plain Old Java Objects

POJO Class – A Java Class which is not extending other classes nor implementing any interface.

POJO Class – Simple Class

Types of Classes in JAVA

1. Simple Class / Concrete Class/ Complete Class/POJO Class
2. Base Class/ Parent Class/ Super Class
3. Sub Class/ Child Class/Derived Class / Non-POJO class
4. Abstract Class / In-Complete/ Non-Concrete class
5. Wrapper Class
6. Static Class
7. Final Class
8. System Defined / Pre-defined Class
9. Custom/ User-defined Class
10. Bean Class (A Class with properties, Constructors & Getter/Setters)
11. Entity Bean Class (A Bean Class which represents a database table)

Spring Framework – <https://spring.io>

<https://docs.spring.io/spring-framework/docs/current/reference/html/> - Spring official Documentation URL

Modules in Spring Framework

* Spring Core
* Spring AOP (Aspect Oriented Programming)
* Spring Data (JPA/JDBC/R2DBC)
* Spring Web/MVC ( MVC/Reactive)
* Spring Test
* Spring Security

Spring Framework uses two important Design Patterns.

1. IoC (Inversion of Control)
2. DI (Dependency Injection)

IoC – Framework manages the life cycle of Beans (Object creation)

Bean – It is a Class/Object created by the Framework

Example

Constructor – Is a Special Method in Java with the same name as the class name, which is used to initialize all the members with the default values. If it’s primitive numbers, then initialize to 0, if it is boolean then false, if it is object then initialize with null value.

Types of Constructors

* Default Constructor (Will be created automatically when no other parameterized constructor are defined)
* Parameterized Constructor (Fully/Partially parameterized)

Ways of Initializing the object’s properties

* By using Constructor
* By using Setter methods

Design Pattern – Is a proven way of solving problems.

Design Pattern is a general concept. This can be implemented in any programming Lang.

Design Pattern – selecting optimal solution for solving a problem/challenge.

Travel from City A to City B (Both are Metro cities)

* By Flight
* By Train
* By Road ( Car/Bus/Bike)
* By Walk

Ways of Creating Objects

Ways of Programming – OOP, POP, FP, AOP

Ways of Testing – Manual, Automated

Creational,

Behavioral

Structure

MVC (Model, View, Controller)

Singleton (Used to create a Single object from a Class)

Factory

IoC = Inversion of Control – IoC container

DI = Dependency Injection (Injecting the Objects at the Run time)

Spring Framework Architecture

Spring Framework supports JVM based Languages JAVA, Kotlin, Groovy

Spring Core – Core, Beans, Context, SpEL

Type of Java Applications

1. Core Java Applications / Stand-alone /Desktop applications [Download & Install]
2. Web Applications (Accessed using browser and runs on server)
3. Enterprise Application [Core + Web +Additional Features]

Types of Java Applications Packaging

1. JAR – Java Archive
2. WAR – Web Archive
3. EAR – Enterprise Archive

Types of Injection

1. Constructor Injection (This will call, parameterized constructor)
2. Setter Injection (this will use default constructor to create object and setter methods to initialize it)

Singleton is a design pattern.

Singleton class is used to create a Single object only. (Only one instance of the class is allowed at any point of time)

Singleton = Only one object for a Class

By default, all the beans created by spring framework is singleton.