**TIGHTLY COUPLED JAVA CODE**

package com.in28minutes.learn\_spring\_framework;

import com.in28minutes.learn\_spring\_framework.game.GameRunner;

import com.in28minutes.learn\_spring\_framework.game.MarioGame;

import com.in28minutes.learn\_spring\_framework.game.SuperContraGame;

public class AppGamingBasicJava {

public static void main(String[] args) {

//var marioGame = new MarioGame();

var superContraGame = new SuperContraGame();

var gameRunner = new GameRunner(superContraGame);

gameRunner.run();

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** GameRunner {

// MarioGame game ; **GAME RUNNER CLASS IS TIGHTLY COUPLED WITH MARIOGAME CLASS**

SuperContraGame game;

**public** GameRunner(SuperContraGame game) {

**this**.game = game ;

}

**public** **void** run()

{

System.***out***.println("RUNNING GAME IS " +game);

game.down();

game.up();

game.left();

game.rigtht();

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** SuperContraGame {

**public** **void** up()

{

System.***out***.println(" SUPER JUMP");

}

**public** **void** down()

{

System.***out***.println(" SUPER Go into hole");

}

**public** **void** left()

{

System.***out***.println(" SUPER Go back");

}

**public** **void** rigtht()

{

System.***out***.println(" SUPER Accerlate");

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** MarioGame {

**public** **void** up()

{

System.***out***.println("JUMP");

}

**public** **void** down()

{

System.***out***.println("Go into hole");

}

**public** **void** left()

{

System.***out***.println("Go back");

}

**public** **void** rigtht()

{

System.***out***.println("Accerlate");

}

}

---------------------------------------------------------------------------------------------------------------------

**LOOSE COUPLED JAVA CODE FOR ABOVE SCENARIO**

package com.in28minutes.learn\_spring\_framework;

import com.in28minutes.learn\_spring\_framework.game.GameRunner;

import com.in28minutes.learn\_spring\_framework.game.MarioGame;

import com.in28minutes.learn\_spring\_framework.game.PacmanGame;

import com.in28minutes.learn\_spring\_framework.game.SuperContraGame;

public class AppGamingBasicJava {

public static void main(String[] args) {

//var game = new MarioGame();

//var game = new SuperContraGame();

var game = new PacmanGame();

var gameRunner = new GameRunner(game);

gameRunner.run();

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** MarioGame **implements** GamingConsole{

**public** **void** up()

{

System.***out***.println("JUMP");

}

**public** **void** down()

{

System.***out***.println("Go into hole");

}

**public** **void** left()

{

System.***out***.println("Go back");

}

**public** **void** right()

{

System.***out***.println("Accerlate");

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** PacmanGame **implements** GamingConsole {

**public** **void** up()

{

System.***out***.println("PacmanGame JUMP");

}

**public** **void** down()

{

System.***out***.println("PacmanGame Go into hole");

}

**public** **void** left()

{

System.***out***.println(" PacmanGame Go back");

}

**public** **void** right()

{

System.***out***.println("PacmanGame Accerlate");

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** SuperContraGame **implements** GamingConsole {

**public** **void** up()

{

System.***out***.println(" SUPER JUMP");

}

**public** **void** down()

{

System.***out***.println(" SUPER Go into hole");

}

**public** **void** left()

{

System.***out***.println(" SUPER Go back");

}

**public** **void** right()

{

System.***out***.println(" SUPER Accerlate");

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **class** GameRunner {

// MarioGame game ;

GamingConsole game;

**public** GameRunner(GamingConsole game) {

**this**.game = game ;

}

**public** **void** run()

{

System.***out***.println("RUNNING GAME IS " +game);

game.down();

game.up();

game.left();

game.right();

}

}

**package** com.in28minutes.learn\_spring\_framework.game;

**public** **interface** GamingConsole {

**void** up();

**void** down();

**void** left();

**void** right();

}

­

A **Spring Bean** is an object that is managed by the Spring IoC (Inversion of Control) container. These beans are the backbone of a Spring application and are created, managed, and destroyed by the container. A bean is simply an instance of a class that is registered with the Spring container.

In Spring Boot, **@Configuration** is an annotation used to indicate that a class declares one or more @Bean methods. These methods are used to define beans that should be managed by the Spring IoC (Inversion of Control) container. The @Configuration annotation marks a class as a source of bean definitions, which are registered with the Spring container.

In Spring Boot, a **Bean** is an object that is instantiated, assembled, and managed by the Spring IoC (Inversion of Control) container. Beans are the key components of a Spring application, and they are defined in the application context, where they can be injected and used throughout the application. Indicates that a method produces a bean to be managed by the Spring container.

**What is a Bean?**

* **Managed Object**: A Spring Bean is an object whose lifecycle is managed by the Spring container. The container creates the bean, manages its dependencies, and ultimately destroys it when no longer needed.
* **Scope**: Beans can have different scopes (singleton, prototype, etc.), determining how they are created and used within the application.
* **Configuration**: Beans can be configured via annotations, XML configuration, or Java configuration.

**Example**:

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public** **class** App02HelloWorldSpring {

**public** **static** **void** main(String[] args) {

// Launch a spring context

**var** context = **new** AnnotationConfigApplicationContext(HelloWorldConfiguration.**class**);

// CONFIGURE THE THINS THAT WE WANT SPRING TO MANAGE@Configuration

// HelloWorldConfiguration - @Configuration

//name - @Bean

//3.Retrieving beans managed by spring

System.***out***.println(context.getBean("name"));

}

}

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

@Configuration

**public** **class** HelloWorldConfiguration {

@Bean

**public** String name() {

**return** "SIVA";

}

}

In Spring Boot, **records** are a modern and concise way to create immutable data-holding classes. They are particularly useful for DTOs, request/response objects, and simple data models. By reducing boilerplate code and promoting immutability, records help make your Spring Boot application cleaner and more maintainable.

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public** **class** App02HelloWorldSpring {

**public** **static** **void** main(String[] args) {

// Launch a spring context

**var** context = **new** AnnotationConfigApplicationContext(HelloWorldConfiguration.**class**);

// CONFIGURE THE THINS THAT WE WANT SPRING TO MANAGE@Configuration

// HelloWorldConfiguration - @Configuration

//name - @Bean

//3.Retrieving beans managed by spring

System.***out***.println(context.getBean("name"));

System.***out***.println(context.getBean("age"));

System.***out***.println(context.getBean("person"));

System.***out***.println(context.getBean("address"));

}

}

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**record** Person (String name , **int** age) {};

**record** Address(String firstLine , String city) {};

@Configuration

**public** **class** HelloWorldConfiguration {

@Bean

**public** String name() {

**return** "SIVA";

}

@Bean

**public** **int** age() {

**return** 21 ;

}

@Bean

**public** Person person()

{

**return** **new** Person("praveen" , 22);

}

@Bean

**public** Address address()

{

**return** **new** Address("ramalayam" , "guntur");

}

}

---------------------------------------------------------------------------------------------------------------------

**DIFFERENT OF ACCESSING BEANS :**

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**record** Person (String name , **int** age , Address address) {};

**record** Address(String firstLine , String city) {};

@Configuration

**public** **class** HelloWorldConfiguration {

@Bean

**public** String name() {

**return** "SIVA";

}

@Bean

**public** **int** age() {

**return** 21 ;

}

@Bean

**public** Person person()

{

**return** **new** Person("praveen" , 22 , **new** Address("shivalayam" , "MEDIKINDURU"));

}

@Bean

**public** Person person3Parameters(String name , **int** age , Address address1)

{

**return** **new** Person(name,age,address1);

}

@Bean(name = "address1")

**public** Address address()

{

**return** **new** Address("ramalayam" , "guntur town");

}

@Bean

**public** Person person2MethodCall()

{

**return** **new** Person(name() , age() , address());

}

}

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public** **class** App02HelloWorldSpring {

**public** **static** **void** main(String[] args) {

// Launch a spring context

**var** context = **new** AnnotationConfigApplicationContext(HelloWorldConfiguration.**class**);

// CONFIGURE THE THINS THAT WE WANT SPRING TO MANAGE@Configuration

// HelloWorldConfiguration - @Configuration

//name - @Bean

//3.Retrieving beans managed by spring

System.***out***.println(context.getBean("name"));

System.***out***.println(context.getBean("age"));

System.***out***.println(context.getBean("person"));// by defining method name

System.***out***.println(context.getBean("person2MethodCall"));

System.***out***.println(context.getBean("address1"));// by defining name

System.***out***.println(context.getBean(Address.**class**));//by using class name

System.***out***.println(context.getBean("person3Parameters"));

Arrays.*stream*(context.getBeanDefinitionNames()).forEach(System.***out***::println);

// **List of beans managed by spring beans**

}

}

In Spring Boot, **primary annotations** and **qualifiers** are used to manage and control the injection of beans when there are multiple beans of the same type available in the Spring context.

**@Primary**

* **Definition**: The @Primary annotation is used to mark a bean as the primary candidate when multiple beans of the same type are available in the Spring context. If you do not explicitly specify which bean to inject using @Qualifier, the @Primary annotated bean will be injected by default.

**@Qualifier**

* **Definition**: The @Qualifier annotation is used to specify exactly which bean should be injected when multiple beans of the same type are available. This helps in situations where @Primary might not be sufficient or where you want to inject a specific bean without marking it as primary.

---------------------------------------------------------------------------------------------------------------------

**USING SPRING FRAMEWORK TO MANAGE BEANS FOR JAVA GAMING APP**

package com.in28minutes.learn\_spring\_framework;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import com.in28minutes.learn\_spring\_framework.game.GameRunner;

import com.in28minutes.learn\_spring\_framework.game.GamingConsole;

import com.in28minutes.learn\_spring\_framework.game.MarioGame;

import com.in28minutes.learn\_spring\_framework.game.PacmanGame;

import com.in28minutes.learn\_spring\_framework.game.SuperContraGame;

public class App03GamingSpringBeans {

public static void main(String[] args) {

var context = new AnnotationConfigApplicationContext(GamingConfiguration.class);

context.getBean(GamingConsole.class).up();

context.getBean(GameRunner.class).run();

}

}

**package** com.in28minutes.learn\_spring\_framework;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** com.in28minutes.learn\_spring\_framework.game.GameRunner;

**import** com.in28minutes.learn\_spring\_framework.game.GamingConsole;

**import** com.in28minutes.learn\_spring\_framework.game.PacmanGame;

@Configuration

**public** **class** GamingConfiguration {

@Bean

**public** GamingConsole game()

{

**var** game = **new** PacmanGame();

**return** game;

}

@Bean

**public** GameRunner gameRunner()

{

**var** gameRunner = **new** GameRunner(game());

**return** gameRunner;

}

}