**Exercise 7: Implementing Constructor and Setter Injection**

**Scenario:**

The library management application requires both constructor and setter injection for better control over bean initialization.

**Steps:**

1. **Configure Constructor Injection:**
   * Update applicationContext.**xml** to configure constructor injection for **BookService**.
2. **Configure Setter Injection:**
   * Ensure that the **BookService** class has a setter method for **BookRepository** and configure it in **applicationContext.xml**.

**applicationContext.xml 🡪**

<beans xmlns="http://www.springframework.org/schema/beans"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xsi:schemaLocation="http://www.springframework.org/schema/beans

           http://www.springframework.org/schema/beans/spring-beans.xsd">

    <!-- Define BookRepository bean -->

    <bean id="bookRepository" class="com.library.repository.BookRepository">

    </bean>

    <!-- Define BookService bean with both constructor and setter injection -->

    <bean id="bookService" class="com.library.service.BookService">

        <constructor-arg ref="bookRepository" />

        <property name="bookRepository" ref="bookRepository" />

    </bean>

</beans>

1. **Test the Injection:**
   * Run the **LibraryManagementApplication** main class to verify both constructor and setter injection.

**LibraryManagementApplication.java🡪**

package com.library;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main( String[] args )

{

ClassPathXmlApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean(BookService.class);

// Test the bookService bean here

System.out.println("BookService Bean: " + bookService);

}

}