**Exercise 2: Implementing Dependency Injection**

**Scenario:** In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

1. **Modify the XML Configuration:**

Update applicationContext.xml to wire BookRepository into BookService.

**applicationContext.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<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>">

**<!—Creating/Initializing bookRepository Bean -->**

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

**<!— bookService Bean is created -->**

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

**<!— bookRepository is wired to bookService using Dependency Injection -->**

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

**</bean>**

</beans>

**BookService.java:**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

**public void setBookRepository(BookRepository bookRepository) {**

**this.bookRepository = bookRepository;**

**}**

public void showBooks() {

System.***out***.println("BookService: Calling BookRepository...");

bookRepository.displayBooks();

}

}

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

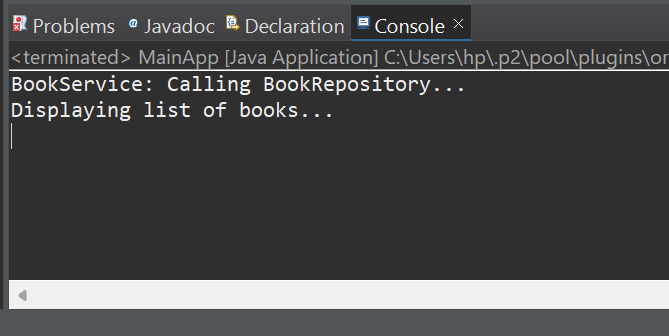
public void displayBooks() {

System.***out***.println("Displaying list of books...");

}

}

**OUTPUT:**



**Output confirms that Spring successfully injected BookRepository into BookService.**