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

**Scenario:**

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

**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">

<!-- Bean for BookRepository -->

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

<!-- Constructor Injection for BookService -->

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

<constructor-arg ref="bookRepository"/>

</bean>

</beans>

**BookService.java**

package com.example.library;

public class BookService {

private BookRepository bookRepository;

// Constructor for constructor injection

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

// Setter for setter injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void manageBooks() {

// Logic to manage books

bookRepository.doSomething();

}

}

**BookRepository.java**

package com.example.library;

public class BookRepository {

public void doSomething() {

// Logic for book repository

System.out.println("BookRepository doing something...");

}

}

package com.example.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.manageBooks();

}

}