**WEEK 3: ADDITIONAL HANDS ON**

**SPRING CORE AND MAVEN:**

**EXERCISE 5: CONFIGURING THE SPRING IOC CONTAINER**

**STEP 1:** Download the project from Spring Initializer.

**STEP 2:** Update Maven Project.

**STEP 3: Create 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">

<!-- Bean for BookRepository -->

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

<!-- Bean for BookService with DI -->

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

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

</bean>

</beans>

**LibraryManagementApplication.java:**

package com.library.LibraryManagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.LibraryManagement.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.listBooks();

}

}

**BookRepository.java:**

package com.library.LibraryManagement.repository;

public class BookRepository {

public void printBooks() {

System.***out***.println("BookRepository: Displaying all books.");

}

}

**BookService.java:**

package com.library.LibraryManagement.service;

import com.library.LibraryManagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter method for DI

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void listBooks() {

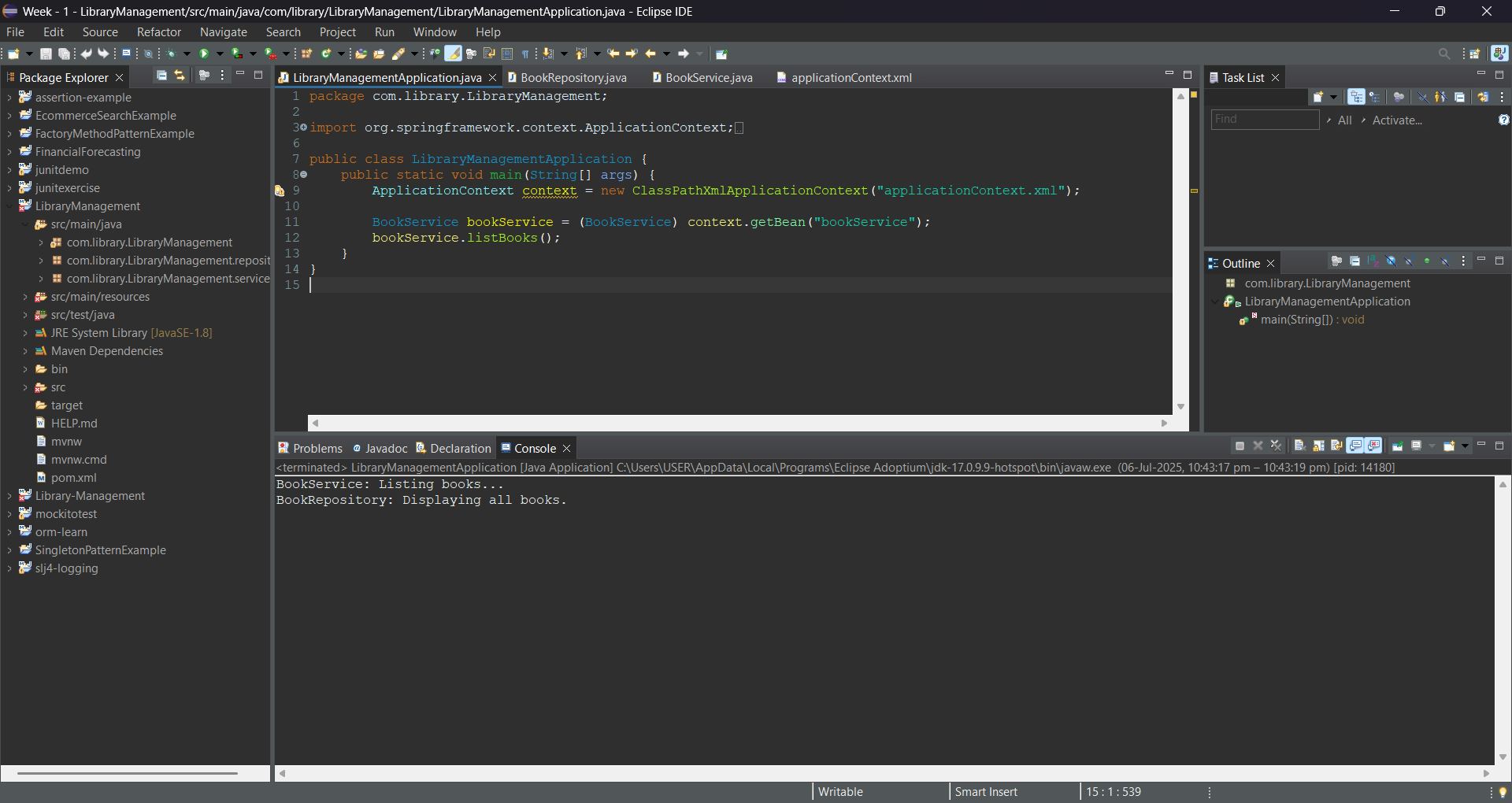
System.***out***.println("BookService: Listing books...");

bookRepository.printBooks();

}

}

**OUTPUT:**

****

**EXERCISE 7: IMPLEMENTING CONSTRUCTOR AND SETTER INJECTION**

**STEP 1:** Download the project from Spring Initializer.

**STEP 2:** Update Maven Project.

**STEP 3: Create 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">

<!-- Bean for BookRepository -->

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

<!-- Bean for BookService with constructor and setter injection -->

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

<!-- Constructor injection -->

<constructor-arg ref="bookRepository"/>

<!-- Setter injection -->

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

</bean>

</beans>

**BookService.java**

package com.library.LibraryManagement.service;

import com.library.LibraryManagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Constructor for constructor injection

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

System.***out***.println("Constructor injection called");

}

// Setter for setter injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

System.***out***.println("Setter injection called");

}

public void listBooks() {

System.***out***.println("BookService: Listing books...");

bookRepository.printBooks();

}

}

**BookRepository.java**

package com.library.LibraryManagement.repository;

public class BookRepository {

public void printBooks() {

System.***out***.println("BookRepository: Displaying all books.");

}

}

**LibraryManagementApplication.java:**

package com.library.LibraryManagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.LibraryManagement.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

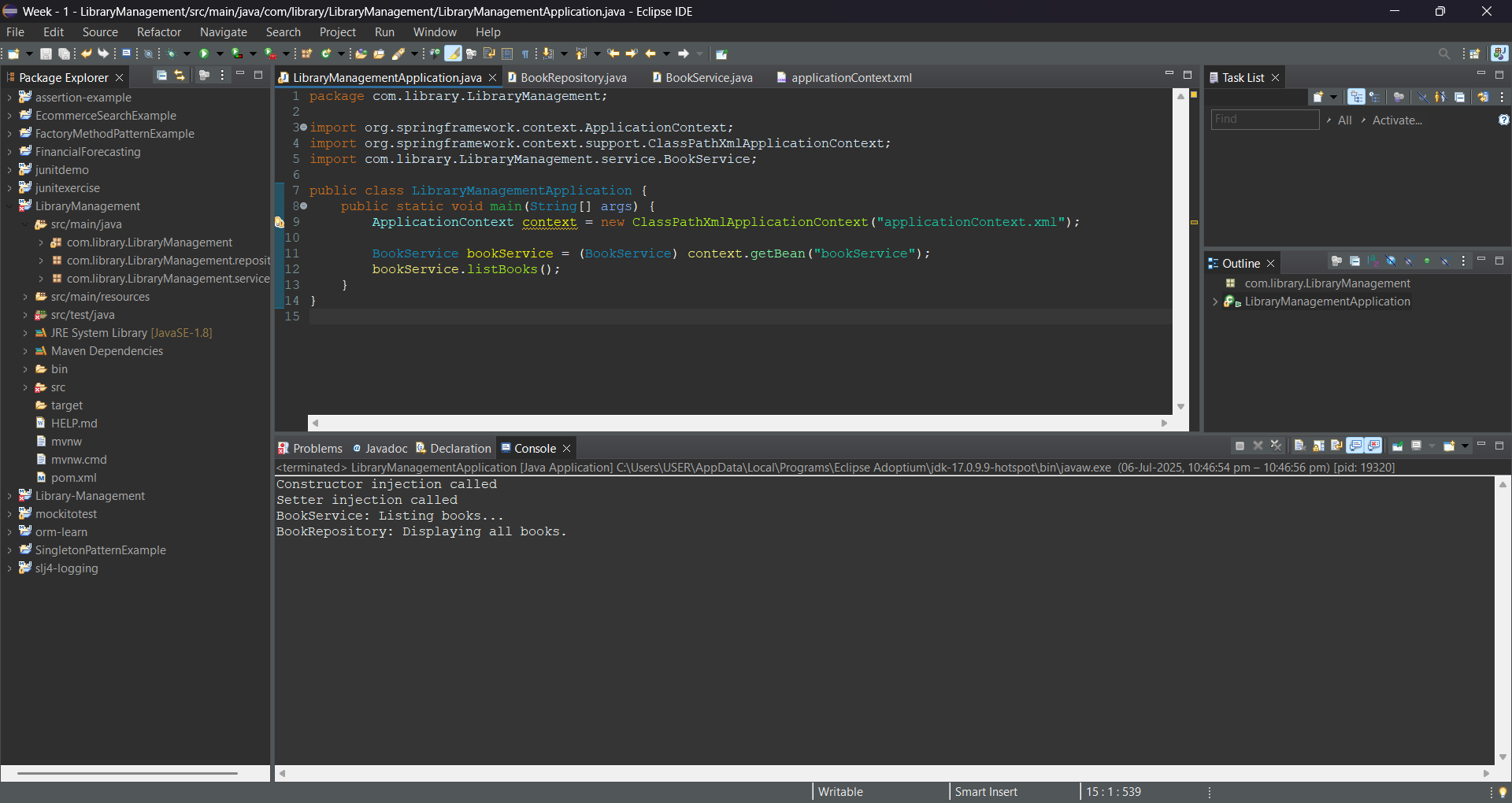
BookService bookService = (BookService) context.getBean("bookService");

bookService.listBooks();

}

}

**OUTPUT:**

****