**Exercise 8: Implementing Basic AOP with Spring**

**Scenario:**

The library management application requires basic AOP functionality to separate cross-cutting concerns like logging and transaction management.

**Steps:**

1. **Define an Aspect:**
   * Create a package **com.library.aspect** and add a class **LoggingAspect**.
2. **Create Advice Methods:**
   * Define advice methods in **LoggingAspect** for logging before and after method execution.
3. **Configure the Aspect:**
   * Update **applicationContext.xml** to register the aspect and enable **AspectJ** auto-proxying.
4. **Test the Aspect:**
   * Run the **LibraryManagementApplication** main class to verify the AOP functionality.

**package** com.library.aspect;

**import** org.aspectj.lang.JoinPoint;

**import** org.aspectj.lang.annotation.Aspect;

**import** org.aspectj.lang.annotation.Before;

**import** org.aspectj.lang.annotation.After;

**import** org.springframework.stereotype.Component;

@Aspect

@Component

**public** **class** LoggingAspect {

@Before("execution(\* com.library.service.\*.\*(..))")

**public** **void** logBefore(JoinPoint joinPoint) {

System.***out***.println("Before" + joinPoint.getSignature().getName());

}

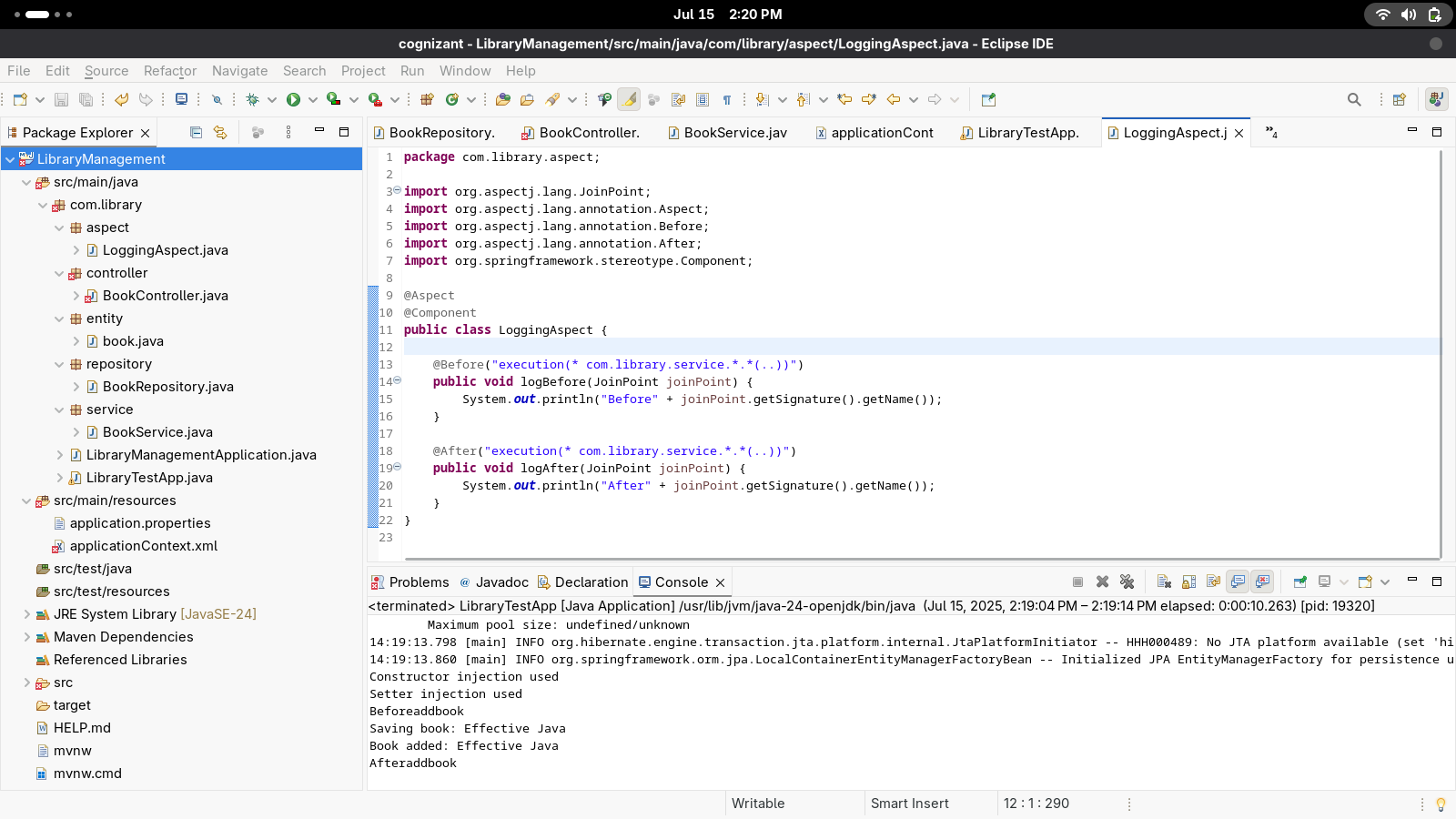
@After("execution(\* com.library.service.\*.\*(..))")

**public** **void** logAfter(JoinPoint joinPoint) {

System.***out***.println("After" + joinPoint.getSignature().getName());

}

}



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

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

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

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

https://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/aop

https://www.springframework.org/schema/aop/spring-aop.xsd">

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

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

<constructor-arg ref="bookRepository" />

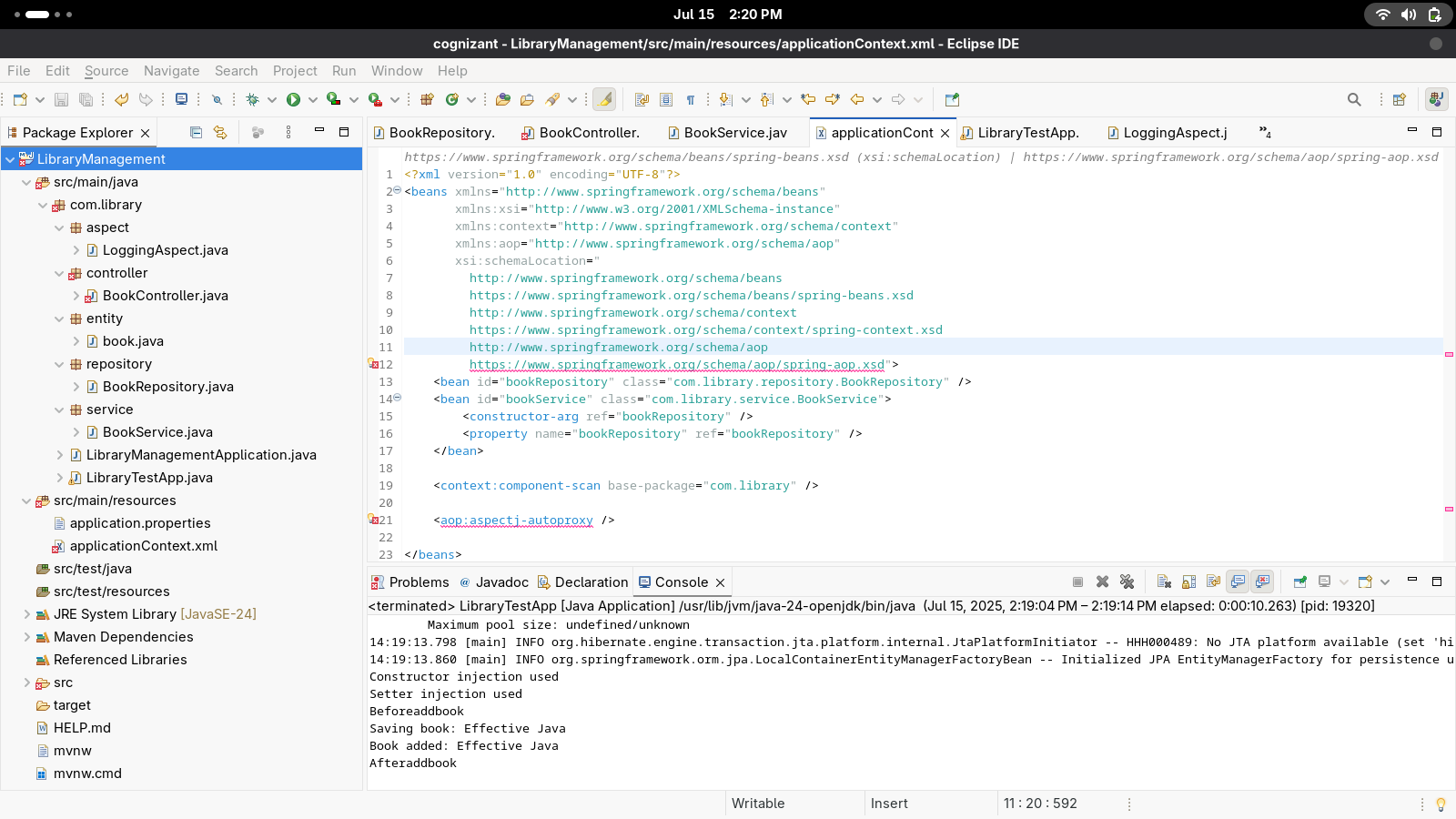
<property name="bookRepository" ref="bookRepository" />

</bean>

<context:component-scan base-package="com.library" />

<aop:aspectj-autoproxy />

**</beans>**



**Output:**

Constructor injection used

Setter injection used

Beforeaddbook

Saving book: Effective Java

Book added: Effective Java

**Afteraddbook**