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

<?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

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

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

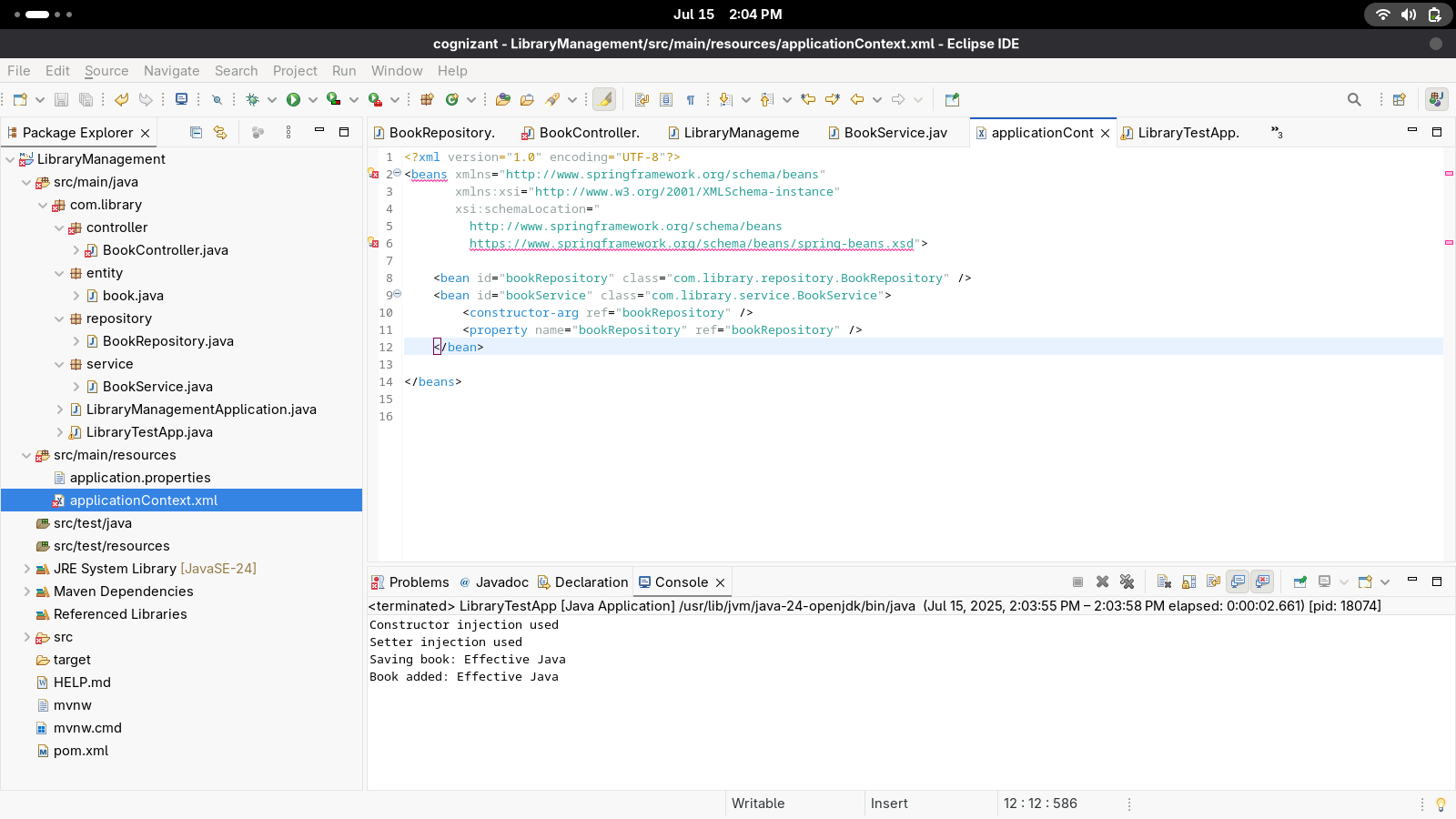
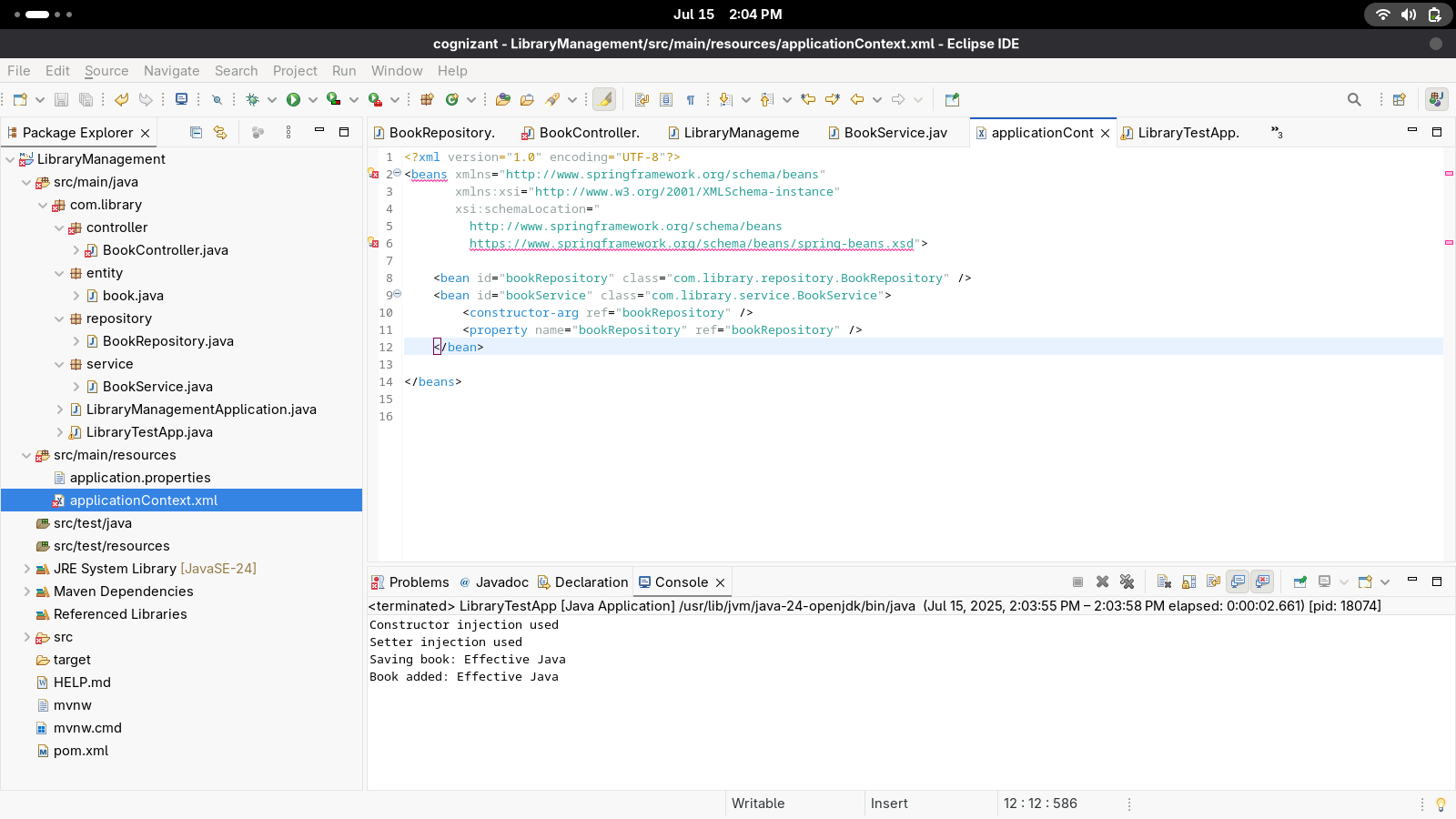
<bean id="bookService" class="com.library.service.BookService">

<constructor-arg ref="bookRepository" />

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

</bean>

</beans>

**public** **class** BookService {

**private** BookRepository bookRepository;

**public** BookService(BookRepository bookRepository) {

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

**this**.bookRepository = bookRepository;

}

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

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

**this**.bookRepository = bookRepository;

}

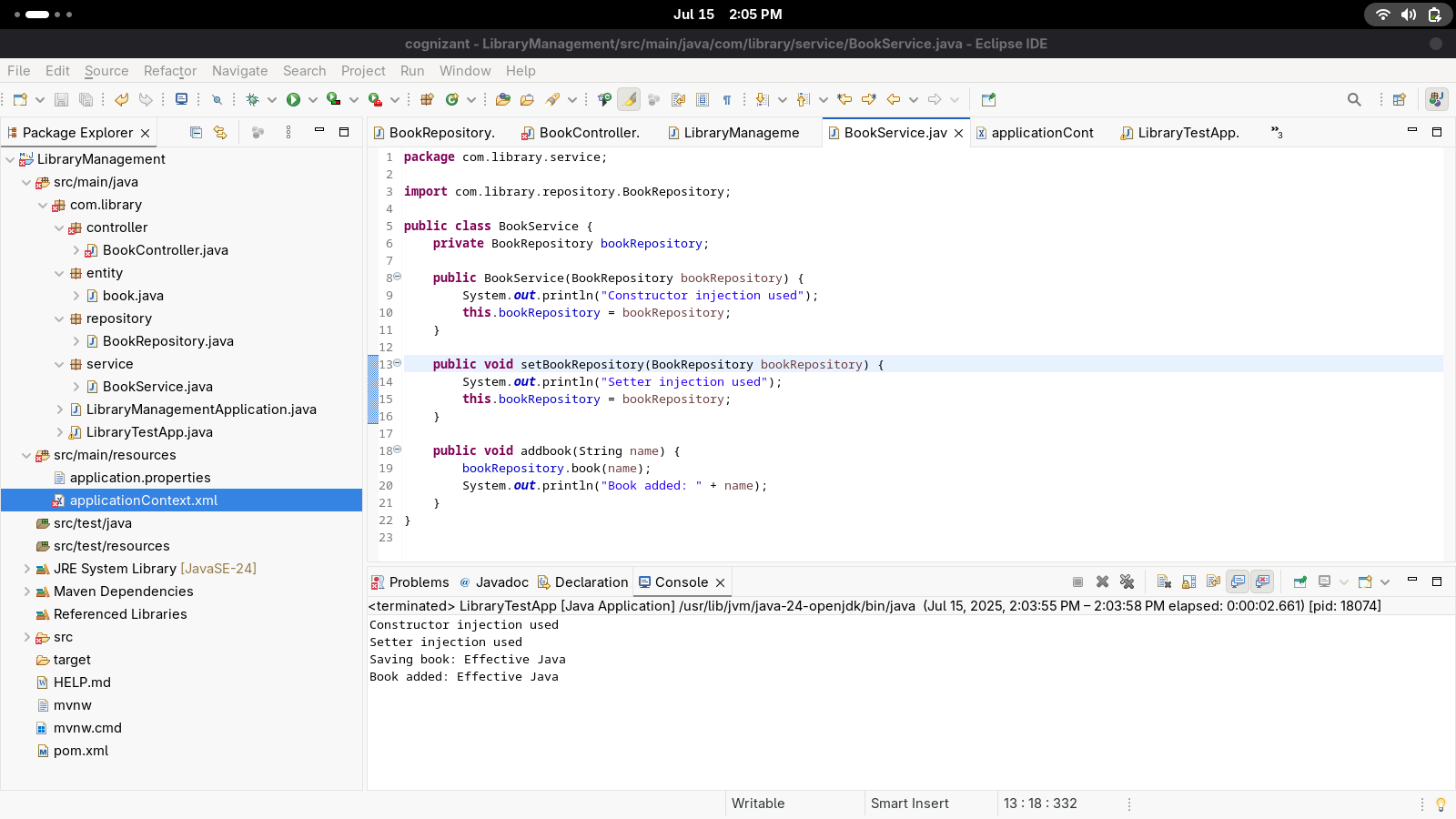
**public** **void** addbook(String name) {

bookRepository.book(name);

System.***out***.println("Book added: " + name);

}

}

Output: 

Constructor injection used

Setter injection used

Saving book: Effective Java

Book added: Effective Java