***EXERCISE 1: CONFIGURING A BASIC STIRNG APPLICATION***

***Scenario:***

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

**MainApp.java:**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

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

bookService.showBooks();

}

}

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

public void displayBooks() {

System.out.println("Displaying all books from the repository.");

}

}

**pom.xml(Library Management):**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

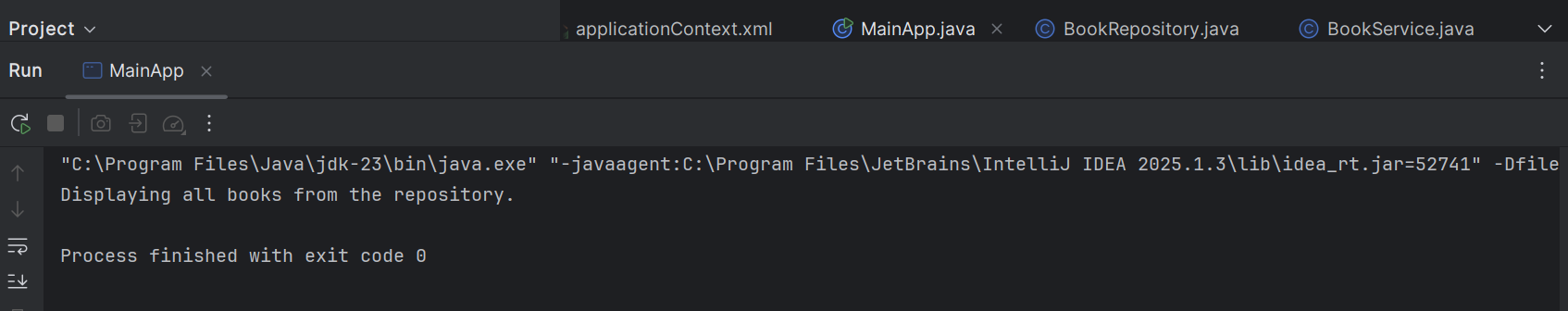
<version>5.3.20</version>

</dependency>

</dependencies>

</project>

**OUTPUT:**



***EXERCISE 2: IMPLEMENTING THE 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.

**LibraryManagementApplication.java:**

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

bookService.performService();

}

}

**BookRepository.java:**

package com.example.library;

public class BookRepository {

public void printRepository() {

System.out.println("BookRepository: Accessing books...");

}

}

**BookService.java:**

package com.example.library;

public class BookService {

private BookRepository bookRepository;

// Setter method for dependency injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void performService() {

System.out.println("BookService: Performing service...");

bookRepository.printRepository();

}

}

**pom.xml(Library Management):**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

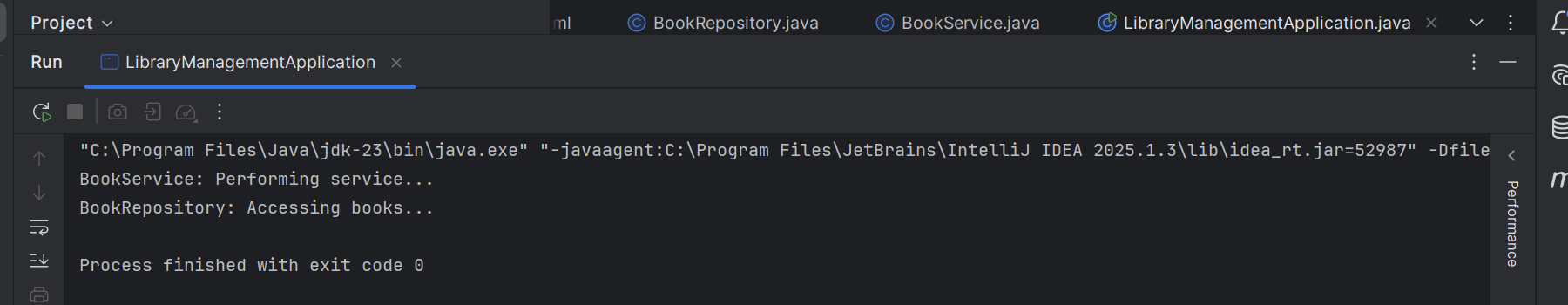
<version>5.3.30</version>

</dependency>

</dependencies>

</project>

**OUTPUT:**

****

***EXERCISE 3: CREATING AND CONFIGURING A MAVEN PROJECT***

**Scenario:**

You need to set up a new Maven project for the library management application and add Spring dependencies.

**App.java:**

package com.example.library;

public class App {

public static void main(String[] args) {

System.out.println("Library Management Project Running...");

}

}

**Pom.xml(Library Management):**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>pom</packaging>

<modules>

<module>src</module>

</modules>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.33</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

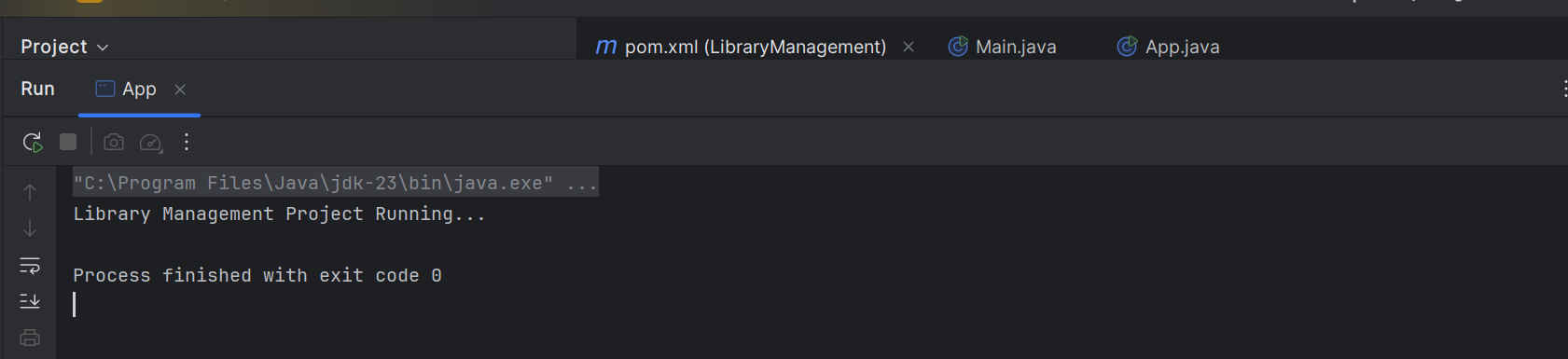
</plugin>

</plugins>

</build>

</project>

**OUTPUT:**

****