**SPRING CORE\_MAVEN**  
  
  
**Exercise 1: Configuring a Basic Spring Application**

**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 id="bookService" class="com.library.service.BookService"/>  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
  
</beans>

**Pom.xml**

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

<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.calculator</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>23</maven.compiler.source>

<maven.compiler.target>23</maven.compiler.target>

<project.build.sourceEncoding>UTF8

</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

</project>

**BookRepository.java**

public class BookRepository {

public void displayRepository() {

System.out.println("BookRepository is working...");

}

}

**BookService.java**

public class BookService {

public void displayService() {

System.out.println("BookService is working...");

}

}

**MainApp.java**

public class MainApp {

public static void main(String[] args) {

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

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

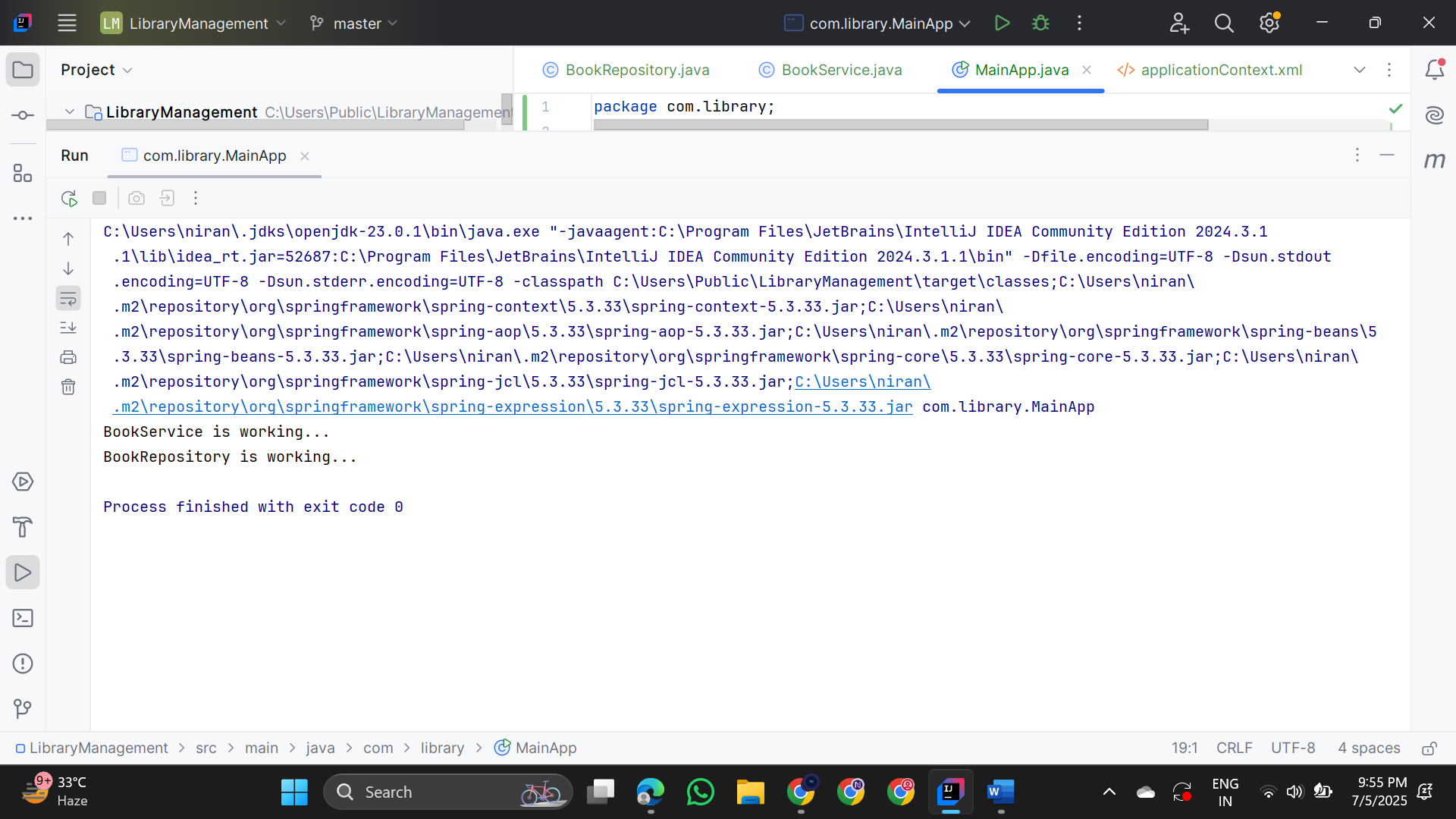
bookService.displayService();

BookRepository bookRepository = (BookRepository) context.getBean("bookRepository");

bookRepository.displayRepository();

}

}



**Exercise 2: Implementing Dependency Injection**

**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 id="bookService" class="com.library.service.BookService"/>  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
  
</beans>

**Pom.xml**

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

<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.calculator</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>23</maven.compiler.source>

<maven.compiler.target>23</maven.compiler.target>

<project.build.sourceEncoding>UTF8

</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

</project>

**BookRepository.java**

public class BookRepository {  
 public void displayRepository() {  
 System.*out*.println("Spring Dependency Injection...");  
 }  
}

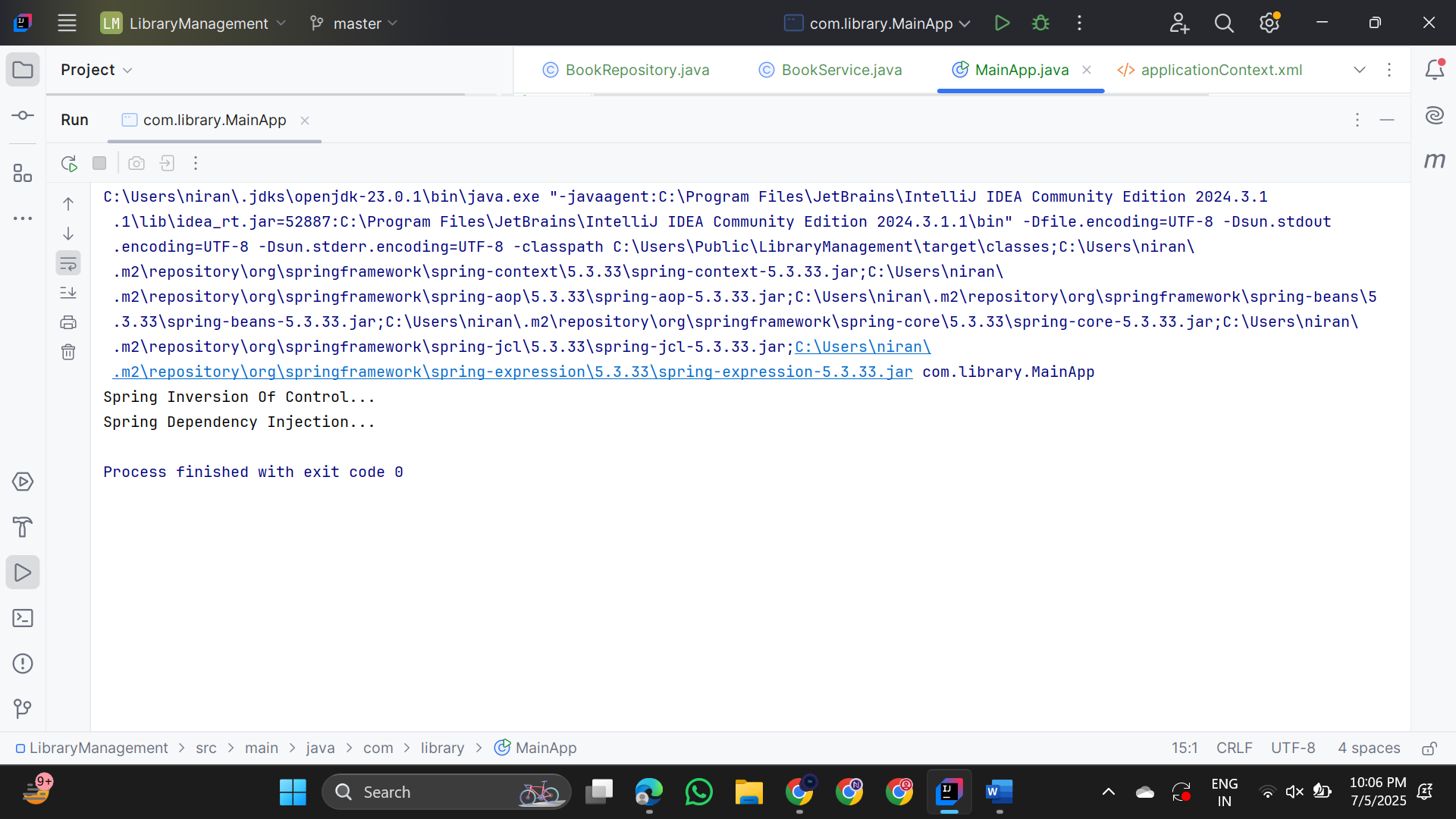
**BookService.java**

public class BookService {  
 private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayService() {  
 System.*out*.println("Spring Inversion Of Control...");  
 if (bookRepository != null) {  
 bookRepository.displayRepository();  
 } else {  
 System.*out*.println("Spring Dependency Injection");  
 }  
 }  
}

**MainApp.java**

public class MainApp {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = (BookService) context.getBean("bookService");  
 bookService.displayService();  
 }



**Exercise 3: Creating and configuring a maven project**

**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 id="bookService" class="com.library.service.BookService"/>  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
  
</beans>

**Pom.xml**

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

<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.calculator</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>23</maven.compiler.source>

<maven.compiler.target>23</maven.compiler.target>

<project.build.sourceEncoding>UTF8

</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

</project>

**BookController.java**

public class BookController {

public void handleRequest() {

System.out.println("Handling a request...");

}

}

**BookRepository.java**

public class BookRepository {  
 public void displayRepository() {  
 System.out.println("Everything was done Successfully...");  
 }  
}

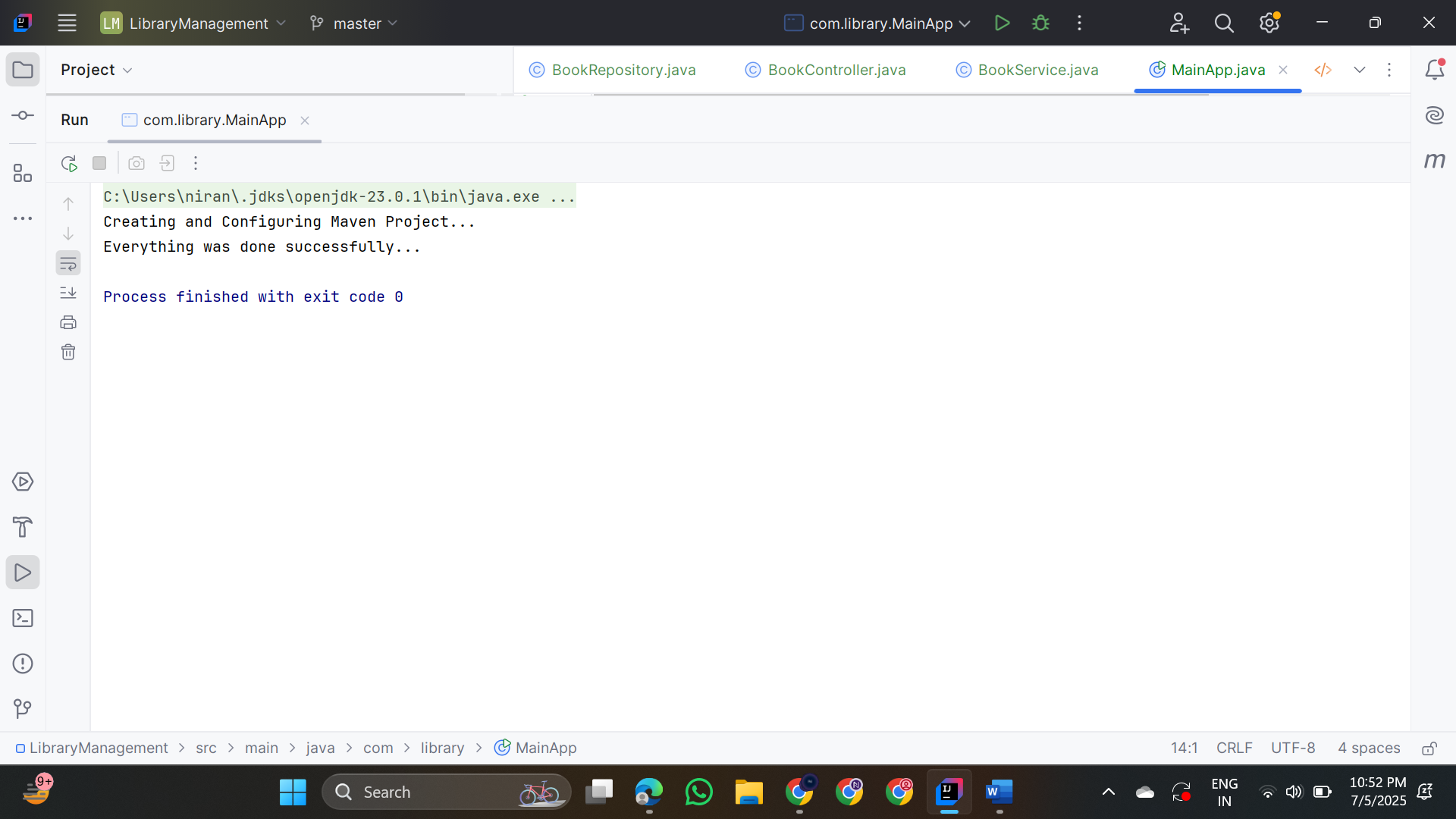
**BookService.java**

public class BookService {  
 private BookRepository bookRepository;  
  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository; }  
 public void displayService() {  
 System.out.println("Creating and Configuring Maven Project...");  
 if (bookRepository != null) {  
 bookRepository.displayRepository();  
 }  
 }  
}

**MainApp.java**

public class MainApp {  
 public static void main(String[] args) {

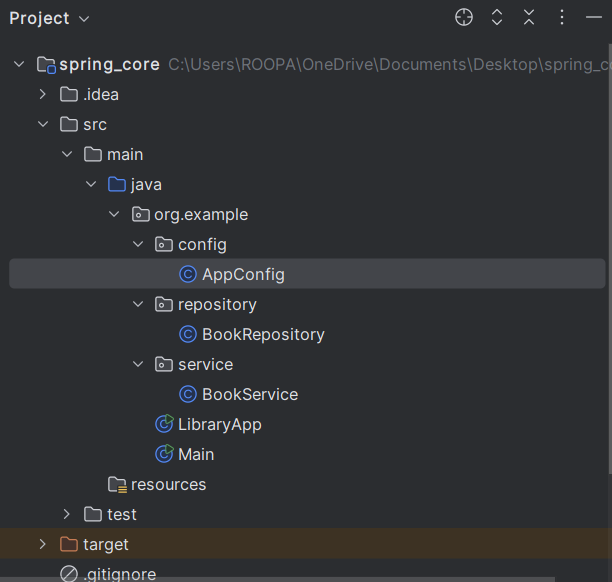
ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService = (BookService) context.getBean("bookService");  
 bookService.displayService();  
 }  
}



**Additional\_Important  
 Handson**

**EXERCISE 5**

**Project structure**

****

**AppConfig.java**

package org.example.config;  
  
import org.example.repository.BookRepository;  
import org.example.service.BookService;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class AppConfig {  
  
 @Bean  
 public BookRepository bookRepository() {  
 return new BookRepository();  
 }  
  
 @Bean  
 public BookService bookService() {  
 BookService service = new BookService();  
 service.setBookRepository(bookRepository());  
 return service;  
 }  
}

**BookRepository.java**

package org.example.repository;  
import java.util.Arrays;  
import java.util.List;  
  
public class BookRepository {  
 public List<String> getAllBooks() {  
 return Arrays.*asList*("Effective Java", "Clean Code", "Spring in Action");  
 }  
}

**BookService.java**

package org.example.service;  
  
import org.example.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayBooks() {  
 bookRepository.getAllBooks().forEach(System.*out*::println);  
 }  
}

**LibraryApp.java**

package org.example;

import org.example.config.AppConfig;

import org.example.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class LibraryApp {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

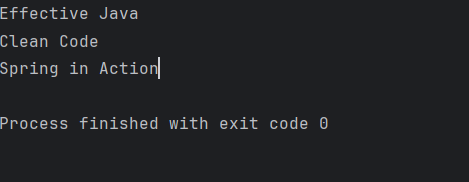
BookService service = context.getBean(BookService.class);

service.displayBooks();

}

}

**OUTPUT**



**EXERCISE 7**

**AppConfig.java**

package org.example.config;  
import org.example.repository.BookRepository;  
import org.example.service.BookService;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class AppConfig {  
 @Bean  
 public BookRepository bookRepository() {  
 return new BookRepository();  
 }  
 @Bean  
 public BookService bookService() {  
 // Constructor Injection:  
 BookService service = new BookService(bookRepository());  
  
 // Setter Injection:  
 service.setBookRepository(bookRepository());  
  
 return service;  
 }  
}

**BookService.java**

package org.example.service;  
  
import org.example.repository.BookRepository;  
  
public class BookService {  
  
 private BookRepository bookRepository;  
  
 // Constructor injection  
 public BookService(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 System.*out*.println("Constructor injection used");  
 }  
  
 // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 System.*out*.println("Setter injection used");  
 }  
  
 public void printBooks() {  
 for (String title : bookRepository.getBookTitles()) {  
 System.*out*.println(title);  
 }  
 }  
}

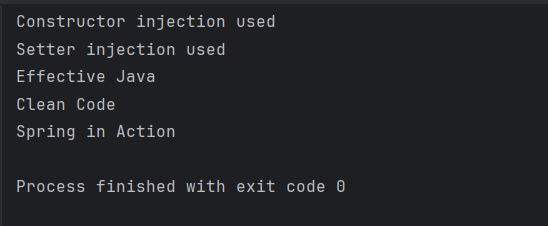
**BookRepository.java**

package org.example.repository;  
  
import java.util.Arrays;  
import java.util.List;  
  
public class BookRepository {  
 public List<String> getBookTitles() {  
 return Arrays.*asList*("Effective Java", "Clean Code", "Spring in Action");  
 }  
}

**LibraryApp.java**

package org.example;  
  
import org.example.config.AppConfig;  
import org.example.service.BookService;  
import org.springframework.context.annotation.AnnotationConfigApplicationContext;  
  
public class LibraryApp {  
 public static void main(String[] args) {  
 AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);  
 BookService bookService = context.getBean(BookService.class);  
 bookService.printBooks();  
 context.close();  
 }  
}

**OUTPUT**

****

**EXERCISE 9**

**BookController.java**

package com.example.librarymanagement.controller;

import com.example.librarymanagement.model.Book;

import com.example.librarymanagement.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookRepository bookRepository;

@GetMapping

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

@PostMapping

public Book addBook(@RequestBody Book book) {

return bookRepository.save(book);

}

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return bookRepository.findById(id).orElse(null);

}

@DeleteMapping("/{id}")

public void deleteBook(@PathVariable Long id) {

bookRepository.deleteById(id);

}

}

**BookRepository.java**

package com.example.librarymanagement.repository;  
  
import com.example.librarymanagement.model.Book;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface BookRepository extends JpaRepository<Book, Long> {  
}

**Book.java**

package com.example.librarymanagement.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
  
@Entity  
public class Book {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String title;  
 private String author;  
  
 public Book(Long id, String title, String author) {  
 this.id = id;  
 this.title = title;  
 this.author = author;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public String getAuthor() {  
 return author;  
 }  
  
 public void setAuthor(String author) {  
 this.author = author;  
 }  
}

