**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

// BookRepository.java

**package** com.library.repository;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** BookRepository {

**private** **final** List<String> books = **new** ArrayList<>();

**public** BookRepository() {

books.add("The Pragmatic Programmer");

books.add("Clean Code");

books.add("Design Patterns: Elements of Reusable OO Software");

}

**public** List<String> findAll() {

**return** books;

}

**public** **void** save(String title) {

books.add(title);

}

}

// BookService.java

**package** com.library.service;

**import** com.library.repository.BookRepository;

**import** java.util.List;

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

**private** **final** BookRepository repository;

**public** BookService(BookRepository repository) {

**this**.repository = repository;

}

**public** List<String> getAllBooks() {

**return** repository.findAll();

}

**public** **void** addBook(String title) {

repository.save(title);

}

}

// Main.java

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ApplicationContext ctx = **new** ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = ctx.getBean(BookService.**class**);

System.***out***.println("Current books:");

bookService.getAllBooks().forEach(System.***out***::println);

bookService.addBook("Introduction to Algorithms");

System.***out***.println("\nAfter adding a book:");

bookService.getAllBooks().forEach(System.***out***::println);

}

}

// 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

       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"/>

    </bean>

</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.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

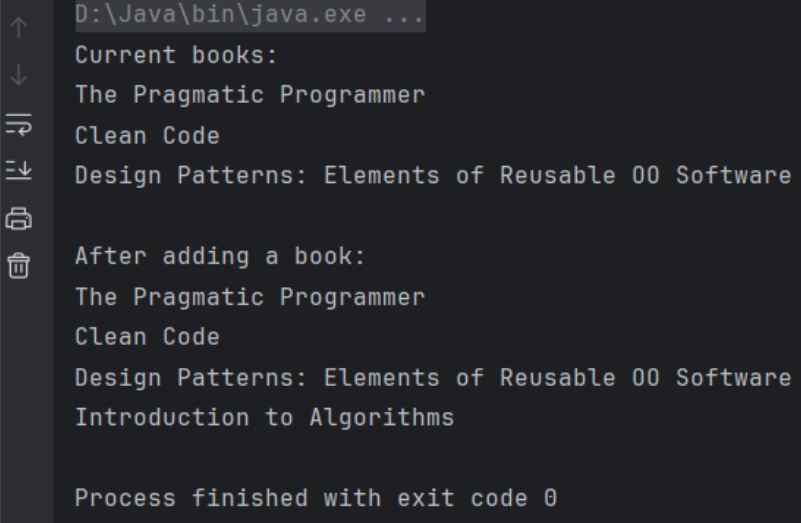
<version>6.2.0</version>

</dependency>

</dependencies>

</project>

**Output:**



**Exercise 2: Implementing Dependency Injection**

// BookRepository.java

**package** com.library.repository;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** BookRepository {

**private** **final** List<String> books = **new** ArrayList<>();

**public** BookRepository() {

books.add("Refactoring: Improving the Design of Existing Code");

books.add("Working Effectively with Legacy Code");

}

**public** List<String> findAll() {

**return** books;

}

**public** **void** save(String title) {

books.add(title);

}

}

// BookService.java

**package** com.library.service;

**import** com.library.repository.BookRepository;

**import** java.util.List;

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

**private** BookRepository bookRepository;

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

**this**.bookRepository = bookRepository;

}

**public** List<String> getAllBooks() {

**return** bookRepository.findAll();

}

**public** **void** addBook(String title) {

bookRepository.save(title);

}

}

// Main.java

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ApplicationContext ctx =

**new** ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = ctx.getBean(BookService.**class**);

System.***out***.println("Current books:");

bookService.getAllBooks().forEach(System.***out***::println);

bookService.addBook("Head First Design Patterns");

System.***out***.println("\nAfter adding a book:");

bookService.getAllBooks().forEach(System.***out***::println);

}

}

// 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

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">

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

</bean>

</beans>

// pom.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

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

<bean id="bookRepository"

class="com.library.repository.BookRepository" />

<bean id="bookService"

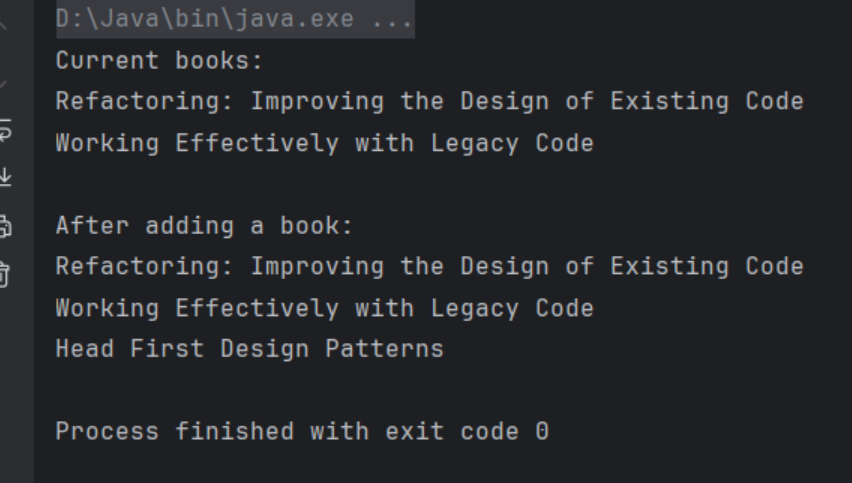
class="com.library.service.BookService">

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

</bean>

</beans>

**Output:**



**Exercise 4: Creating and Configuring a Maven Project**

// pom.xml

<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>Library</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

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

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

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>6.2.7</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>6.1.5</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>6.2.6</version>

</dependency>

<dependency>

<groupId>jakarta.servlet</groupId>

<artifactId>jakarta.servlet-api</artifactId>

<version>6.0.0</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.12.1</version>

<configuration>

<source>23</source>

<target>23</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Output:**

