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

Code:

|  |
| --- |
| 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.addBook("Spring in Action");  }  }  package com.library.service;  import com.library.repository.BookRepository;  public class BookService {  private BookRepository bookRepository;  public void setBookRepository(BookRepository bookRepository) {  this.bookRepository = bookRepository;  }  public void addBook(String bookName) {  System.***out***.println("Adding book: " + bookName);  bookRepository.saveBook(bookName);  }  }  package com.library.repository;  public class BookRepository {  public void saveBook(String bookName) {  System.***out***.println("Book '" + bookName + "' saved to database.");  }  } |

Output:

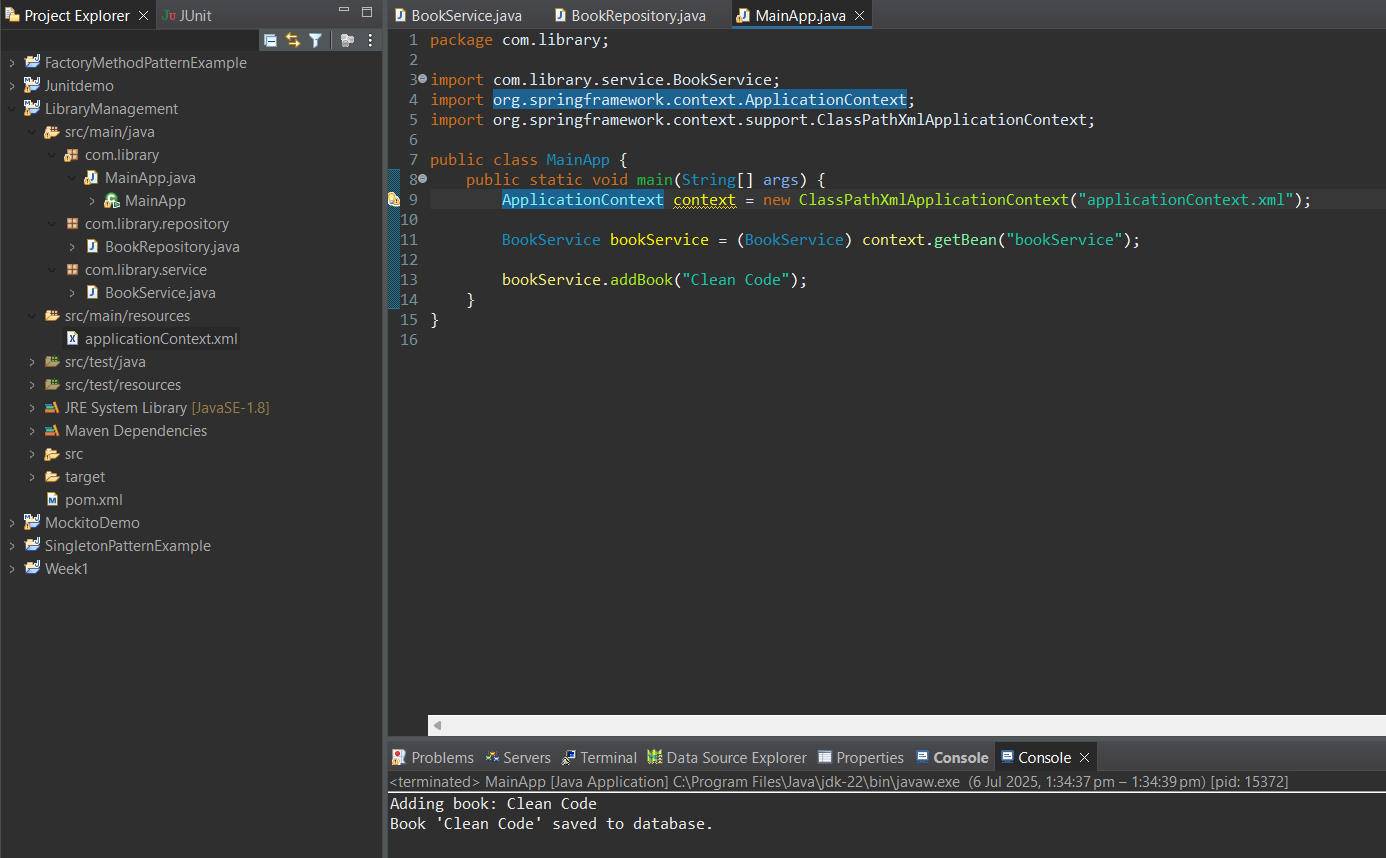
A screenshot of a computer program

AI-generated content may be incorrect.

**Exercise 2: Implementing Dependency Injection**

Code:

|  |
| --- |
| 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.addBook("Clean Code");  }  }  package com.library.service;  import com.library.repository.BookRepository;  public class BookService {  private BookRepository bookRepository;  public void setBookRepository(BookRepository bookRepository) {  this.bookRepository = bookRepository;  }  public void addBook(String bookName) {  System.***out***.println("Adding book: " + bookName);  bookRepository.saveBook(bookName);  }  }  package com.library.repository;  public class BookRepository {  public void saveBook(String bookName) {  System.***out***.println("Book '" + bookName + "' saved to database.");  }  } |

Output:  


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

Code:

|  |
| --- |
| <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>1.8</maven.compiler.source>  <maven.compiler.target>1.8</maven.compiler.target>  </properties>  <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:  
