**Spring Core and Maven**

**Exercise-1**: Configuring a Basic Spring 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.

Steps:

1. **Set Up a Spring Project:**

* + Create a Maven project named **LibraryManagement**.

File -> New ->Maven Project ->Name it as LibraryManagement

* + Add Spring Core dependencies in the **pom.xml** file.

To pom.xml add these spring core dependencies

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>6.2.7</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>6.2.7</version>

</dependency>

</dependencies>

2. **Configure the Application Context:**

In **src/main/resources** directory Create a **applicationContext.xml** file and define the beans for **BookService** and **BookRepository** classes as below

<?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"></bean>

</beans>

3. **Define Service and Repository Classes**

In **src/main/java** directorycreate 2 packages as **com.library.service** and **com.library.repository** and add classes as **BookService** and **BookRepository** in their respective classes and add Function as below

**BookRepository class:**

package com.library.repository;

public class BookRepository {

public void getRepository() {

System.***out***.println("Hi..Welcom to BookRepository Class");

}

}

**BookService class:**

package com.library.service

public class BookService {

public void getService() {

System.***out***.println("Hi..Welcom to Book Servie Class");

}

}

4. **Run the Application:**

Create a package com.main and create a Main class in src/main/java directory

**Main class:**

package com.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

import com.library.repository.BookRepository;

public class Main {

public static void main(String[] args) {

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

BookService bs= (BookService) context.getBean("bookservice");

bs.getService();

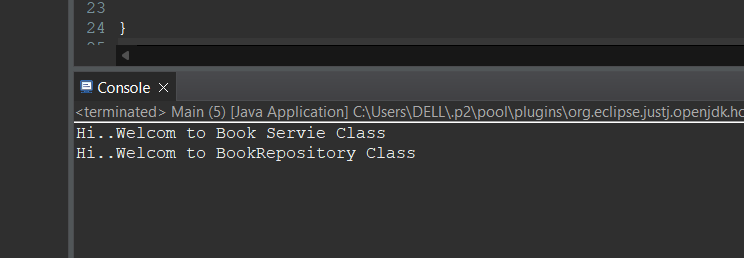
BookRepository br=(BookRepository)context.getBean("bookrepository");

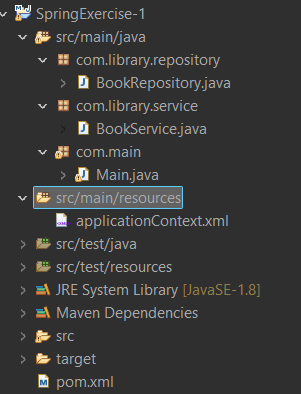
br.getRepository();

}

}

Output:



****

**Exercise 2: Implementing 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.

Steps:

In the above project created change the following

1.ApplicationContext.xml file:

Adding the bookRepository dependency injection through property id

Code:

<?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="bookrepository" class="com.library.repository.BookRepository"></bean>

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

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

</bean>

</beans>

2.BookService class:

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 show() {

System.***out***.println("BookRepository Dependency Injection Implemented");

bookrepository.show();

}

}

3.BookRepository:

package com.library.repository;

public class BookRepository {

public void show() {

System.***out***.println("Hi..Welcom to BookRepository Class");

}

}

4.Main class:

package com.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

import com.library.repository.BookRepository;

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

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

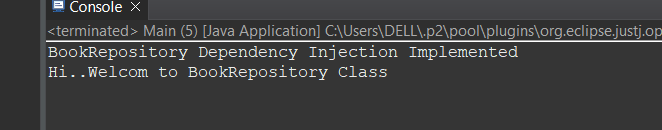
BookService bs= (BookService) context.getBean("bookservice");

bs.show();

}

}

Output:



**Exercise 4: 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.

Steps:

1.Create a maven project

2. GO to maven repository and add these stable dependencies in pom.xml file

i. Spring Context

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>6.2.7</version>

</dependency>

ii. Spring AOP

<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>6.2.6</version>

</dependency>

iii. Spring Web MVC

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>6.2.7</version>

</dependency>

3.Configure the Maven Compiler Plugin for Java version 1.8 in the pom.xml file.

Pom.xml file

<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

https://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>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>6.2.7</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>6.2.6</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>6.2.7</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>