**Exercise 8: Implementing Basic AOP with Spring**

**Scenario:**

The library management application requires basic AOP functionality to separate cross-cutting concerns like logging and transaction management.

**Steps:**

1. **Define an Aspect:**
   * Create a package **com.library.aspect** and add a class **LoggingAspect**.
2. **Create Advice Methods:**
   * Define advice methods in **LoggingAspect** for logging before and after method execution.

**LoggingAspect.java 🡪**

package com.library.aspect;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

private static final Logger logger = LoggerFactory.getLogger(LoggingAspect.class);

//adding the @before advice fired on all the methods of the Services

@Before("execution(\* com.library.services.\*.\*(..))")

public void logBefore() {

System.out.println("before");

logger.info("Method execution started");

}

@After("execution(\* com.library.services.\*.\*(..))")

public void logAfter() {

System.out.println("After");

logger.info("Method execution finished");

}

}

1. **Configure the Aspect:**
   * Update **applicationContext.xml** to register the aspect and enable **AspectJ** auto-proxying.

**applicationContext.xml 🡪**

<beans xmlns="http://www.springframework.org/schema/beans"

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

       xmlns:context="http://www.springframework.org/schema/context"

       xmlns:aop="http://www.springframework.org/schema/aop"

       xsi:schemaLocation="

           http://www.springframework.org/schema/beans

           http://www.springframework.org/schema/beans/spring-beans.xsd

           http://www.springframework.org/schema/context

           http://www.springframework.org/schema/context/spring-context.xsd

           http://www.springframework.org/schema/aop

           http://www.springframework.org/schema/aop/spring-aop.xsd">

    <!-- Enable component scanning -->

    <context:component-scan base-package="com.library" />

    <!-- Enable AspectJ auto-proxying -->

    <aop:aspectj-autoproxy/>

    <!-- Configure constructor injection for BookService -->

    <bean id="bookService" class="com.library.services.BookService">

        <constructor-arg ref="bookRepository"/>

        <property name="bookRepository1" ref="bookRepository"/>

    </bean>

    <!-- Define the BookRepository bean -->

    <bean id="bookRepository" class="com.library.repository.BookRepository"/>

</beans>

1. **Test the Aspect:**
   * Run the **LibraryManagementApplication** main class to verify the AOP functionality.

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main( String[] args )

{

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

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

//call the method

bookService.getBookRepository();

}

}