**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.

**BookService.java**

package com.library;

import org.springframework.stereotype.Component;

@Component

public class BookService {

    public void addBook(String title) {

        System.out.println("ADDING BOOK: " + title);

        try {

            Thread.sleep(400); // Simulate processing time

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

    }

    public String findBook(String title) {

        System.out.println("FINDING BOOK: " + title);

        try {

            Thread.sleep(300); // Simulate processing time

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

        return title;

    }

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class LibraryManagementApplication {

    public static void main(String[] args) {

        AnnotationConfigApplicationContext context =

                new AnnotationConfigApplicationContext(AppConfig.class);

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

        bookService.addBook("Clean Code");

        bookService.findBook("Effective Java");

        context.close();

    }

}

**LoggingAspect.java**

package com.library;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

    @Around("execution(\* com.library..\*(..))")

    public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

        String methodName = joinPoint.getSignature().toShortString();

        System.out.println("STARTED EXECUTION OF: " + methodName);

        long start = System.currentTimeMillis();

        Object result = joinPoint.proceed();

        long end = System.currentTimeMillis();

        System.out.println("COMPLETED EXECUTION OF: " + methodName);

        System.out.println("EXECUTION TIME: " + (end - start) + " ms\n");

        return result;

    }

}

**AppConfig.java**

package com.library;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.EnableAspectJAutoProxy;

@Configuration

@ComponentScan(basePackages = "com.library")

@EnableAspectJAutoProxy

public class AppConfig {

}

**applictionContext.xml**

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

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

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

       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

         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">

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

    <aop:aspectj-autoproxy />

</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>ex2</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>ex2</name>

  <url>http://www.example.com</url>

  <properties>

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

    <maven.compiler.source>1.8</maven.compiler.source>

    <maven.compiler.target>1.8</maven.compiler.target>

    <java.version>1.8</java.version>

  </properties>

  <dependencies>

    <!-- Spring Context -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-context</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- Spring AOP -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-aspects</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- Spring Web MVC -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-webmvc</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- AspectJ Weaver for AOP -->

    <dependency>

      <groupId>org.aspectj</groupId>

      <artifactId>aspectjweaver</artifactId>

      <version>1.9.20</version>

    </dependency>

    <!-- JUnit for testing -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <!-- Maven Compiler Plugin -->

      <plugin>

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

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

        <version>3.10.1</version>

        <configuration>

          <source>1.8</source>

          <target>1.8</target>

        </configuration>

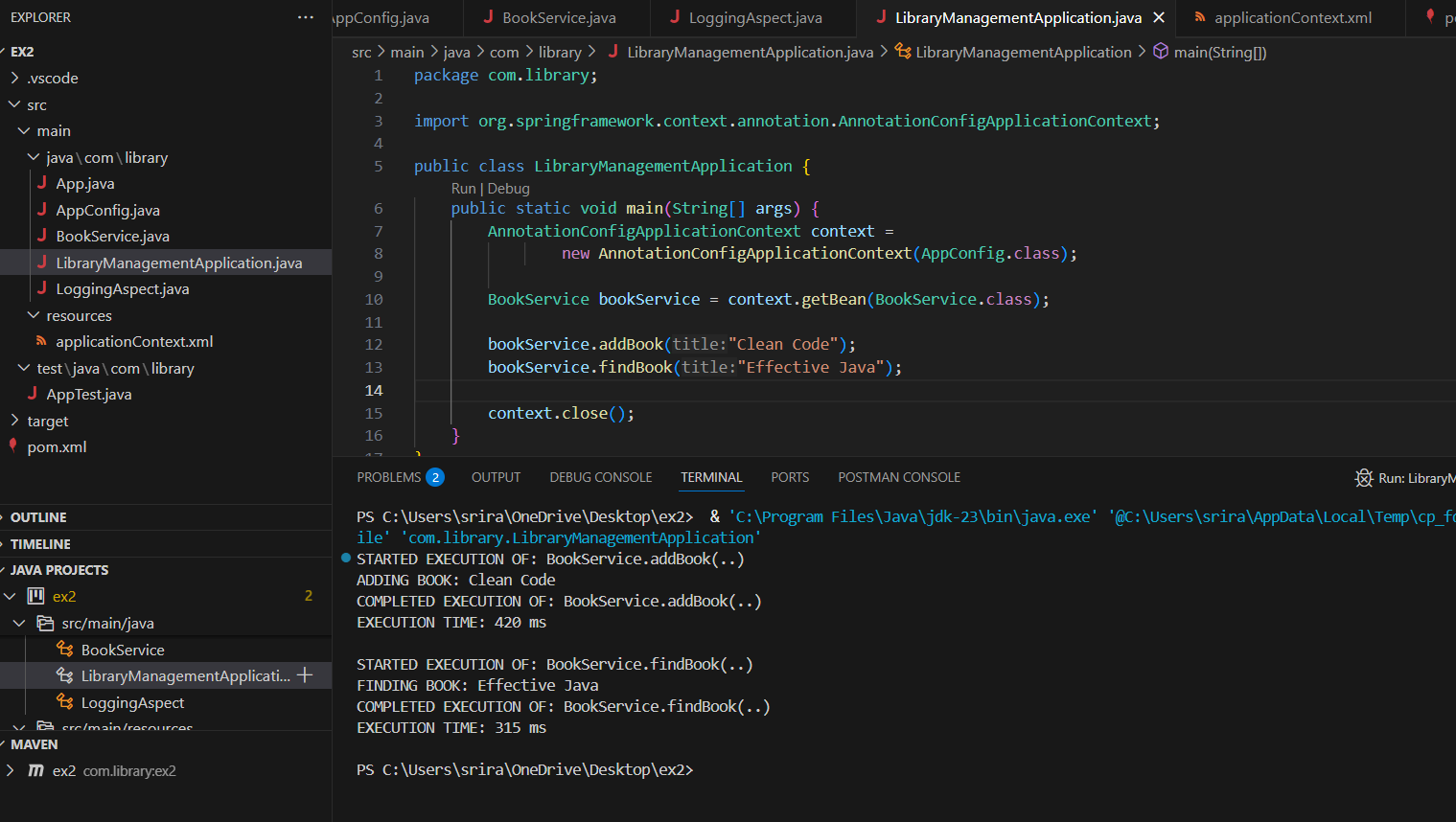
      </plugin>

    </plugins>

  </build>

</project>

**Output:**

****