**CODE:**

eclipse.preferences.version=1

org.eclipse.jdt.core.compiler.codegen.targetPlatform=1.8

org.eclipse.jdt.core.compiler.compliance=1.8

org.eclipse.jdt.core.compiler.problem.enablePreviewFeatures=disabled

org.eclipse.jdt.core.compiler.problem.forbiddenReference=warning

org.eclipse.jdt.core.compiler.problem.reportPreviewFeatures=ignore

org.eclipse.jdt.core.compiler.release=disabled

org.eclipse.jdt.core.compiler.source=1.8

activeProfiles=

eclipse.preferences.version=1

resolveWorkspaceProjects=true

version=1

package com.library.aspect;  
  
import org.aspectj.lang.ProceedingJoinPoint;  
import org.aspectj.lang.annotation.Around;  
import org.aspectj.lang.annotation.Aspect;  
  
@Aspect  
public class LoggingAspect {  
  
@Around("execution(\* com.library.service.\*.\*(..))")  
public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {  
long start = System.currentTimeMillis();  
  
Object result = joinPoint.proceed();  
  
long end = System.currentTimeMillis();  
System.out.println("[LOG] Method " + joinPoint.getSignature() + " executed in " + (end - start) + "ms");  
  
return result;  
}  
}

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book '" + bookName + "' saved to the database.");

}

}

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter for DI

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

bookRepository.saveBook(bookName);

}

}

package com.library;

import com.library.service.BookService;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

// Properly close the context to avoid resource leak

try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml")) {

BookService service = context.getBean("bookService", BookService.class);

service.addBook("Spring in Action");

}

}

}

<?xml version="1.0" encoding="UTF-8"?>

<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 and AOP -->

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

<aop:aspectj-autoproxy />

<!-- Beans -->

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

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

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

</bean>

<!-- Aspect bean -->

<bean class="com.library.aspect.LoggingAspect" />

</beans>

<?xml version="1.0" encoding="UTF-8"?>

<classpath>

<classpathentry kind="src" output="target/classes" path="src/main/java">

<attributes>

<attribute name="optional" value="true"/>

<attribute name="maven.pomderived" value="true"/>

</attributes>

</classpathentry>

<classpathentry excluding="\*\*" kind="src" output="target/classes" path="src/main/resources">

<attributes>

<attribute name="maven.pomderived" value="true"/>

<attribute name="optional" value="true"/>

</attributes>

</classpathentry>

<classpathentry kind="src" output="target/test-classes" path="src/test/java">

<attributes>

<attribute name="optional" value="true"/>

<attribute name="maven.pomderived" value="true"/>

<attribute name="test" value="true"/>

</attributes>

</classpathentry>

<classpathentry excluding="\*\*" kind="src" output="target/test-classes" path="src/test/resources">

<attributes>

<attribute name="maven.pomderived" value="true"/>

<attribute name="test" value="true"/>

<attribute name="optional" value="true"/>

</attributes>

</classpathentry>

<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.8">

<attributes>

<attribute name="maven.pomderived" value="true"/>

</attributes>

</classpathentry>

<classpathentry kind="con" path="org.eclipse.m2e.MAVEN2\_CLASSPATH\_CONTAINER">

<attributes>

<attribute name="maven.pomderived" value="true"/>

</attributes>

</classpathentry>

<classpathentry kind="output" path="target/classes"/>

</classpath>

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>logging-with-AOP</name>

<comment></comment>

<projects>

</projects>

<buildSpec>

<buildCommand>

<name>org.eclipse.jdt.core.javabuilder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.m2e.core.maven2Builder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>org.eclipse.jdt.core.javanature</nature>

<nature>org.eclipse.m2e.core.maven2Nature</nature>

</natures>

</projectDescription>

<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>LibraryAOPExample</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Spring Core & Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.32</version>

</dependency>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.32</version>

</dependency>

<!-- AspectJ Weaver -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

</project>

**OUTPUT:**

A computer screen shot of a program

Description automatically generated