**Module 5 - Spring Core and Maven**

**EXERCISE 1**

// File: LibraryApp.java

package com.library;

import org.springframework.context.annotation.\*;

import org.springframework.context.ApplicationContext;

// Repository class

class BookRepository {

public void saveBook(String title) {

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

}

}

// Service class

class BookService {

private BookRepository bookRepository;

// Setter injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

System.out.println("Adding book: " + title);

bookRepository.saveBook(title);

}

}

// Configuration class (acts like applicationContext.xml)

@Configuration

class AppConfig {

@Bean

public BookRepository bookRepository() {

return new BookRepository();

}

@Bean

public BookService bookService() {

BookService service = new BookService();

service.setBookRepository(bookRepository());

return service;

}

}

public class LibraryApp {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

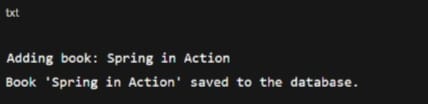
BookService bookService = context.getBean(BookService.class);

bookService.addBook("Spring in Action");

}

}

**OUTPUT**

****

**EXERCISE 2**

**//Embedded XML**

import org.springframework.beans.factory.xml.XmlBeanDefinitionReader;

import org.springframework.context.support.GenericApplicationContext;

import org.springframework.core.io.ByteArrayResource;

// Model

class Book {

private String title;

public Book(String title) {

this.title = title;

}

public String getTitle() {

return title;

}

}

// Repository

class BookRepository {

public void save(Book book) {

System.out.println("Saving book: " + book.getTitle());

}

}

// Service

class BookService {

private BookRepository bookRepository;

// Setter for DI

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

Book book = new Book(title);

bookRepository.save(book);

}

}

// Main Application

public class LibraryManagementApplication {

public static void main(String[] args) {

// Embedded XML configuration

String xmlConfig = """

<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="BookRepository"/>

<bean id="bookService" class="BookService">

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

</bean>

</beans>

""";

// Load Spring context from the XML string

GenericApplicationContext context = new GenericApplicationContext();

XmlBeanDefinitionReader reader = new XmlBeanDefinitionReader(context);

reader.loadBeanDefinitions(new ByteArrayResource(xmlConfig.getBytes()));

context.refresh();

// Get the service bean and use it

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

bookService.addBook("The Power of Habit");

context.close();

}

}

**pom.xml**

<dependency>

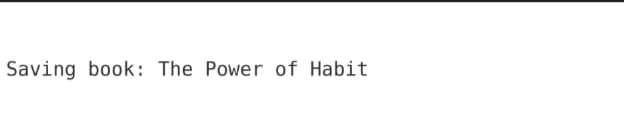
<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.31</version>

</dependency>

**OUTPUT**

****

**Exercise 3**

**LibraryManagementApplication.java**

package com.library;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.\*;

import org.springframework.aop.framework.ProxyFactoryBean;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.\*;

import org.springframework.stereotype.Component;

// =========================

// Logging Aspect

// =========================

@Aspect

@Component

class LoggingAspect {

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

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed();

long end = System.currentTimeMillis();

System.out.println(joinPoint.getSignature() + " executed in " + (end - start) + "ms");

return result;

}

}

// =========================

// Book Service

// =========================

@Component

class BookService {

public void findBooks() {

System.out.println("Finding books...");

try {

Thread.sleep(400); // simulate delay

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

// =========================

// Spring Configuration

// =========================

@Configuration

@ComponentScan(basePackages = "com.library")

@EnableAspectJAutoProxy

class AppConfig { }

// =========================

// Main Class

// =========================

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

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

bookService.findBooks();

}

}

**Pom.xml**

<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>LibraryAOPDemo</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

</project>

**OUTPUT**

****

**Exercise 4**

**//LibraryManagementApp.java**

// File: LibraryManagementApp.java

package com.library;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.\*;

import org.springframework.context.annotation.\*;

import org.springframework.stereotype.Component;

public class LibraryManagementApp {

public static void main(String[] args) {

AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

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

bookService.addBook("Spring in Action");

bookService.removeBook("Effective Java");

context.close();

}

}

// ------------------------------

// Service Class

// ------------------------------

@Component

class BookService {

public void addBook(String title) {

System.out.println("Book added: " + title);

}

public void removeBook(String title) {

System.out.println("Book removed: " + title);

}

}

// ------------------------------

// Logging Aspect

// ------------------------------

@Aspect

@Component

class LoggingAspect {

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

public void logBefore(JoinPoint joinPoint) {

System.out.println("[LOG] Before method: " + joinPoint.getSignature().getName());

}

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

public void logAfter(JoinPoint joinPoint) {

System.out.println("[LOG] After method: " + joinPoint.getSignature().getName());

}

}

// ------------------------------

// Configuration Class

// ------------------------------

@Configuration

@ComponentScan("com.library")

@EnableAspectJAutoProxy

class AppConfig {

}

**Pom.xml**

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

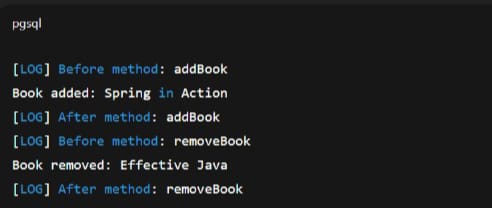
<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

**OUTPUT**

****

**EXERCISE 5**

**//MainApp.java**

package com.library;

import org.springframework.context.annotation.\*;

// BookRepository class

class BookRepository {

public void displayBookInfo() {

System.out.println("Displaying book information from repository.");

}

}

// BookService class with setter-based injection

class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void showBookDetails() {

bookRepository.displayBookInfo();

}

}

// Configuration class simulating applicationContext.xml

@Configuration

class AppConfig {

@Bean

public BookRepository bookRepository() {

return new BookRepository();

}

@Bean

public BookService bookService() {

BookService bookService = new BookService();

bookService.setBookRepository(bookRepository()); // Setter-based DI

return bookService;

}

}

// Main class

public class MainApp {

public static void main(String[] args) {

AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

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

bookService.showBookDetails();

context.close();

}

}

**Pom.xml**

<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>

<dependencies>

<!-- Spring Core/Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

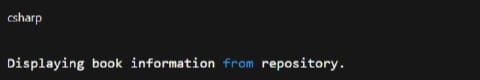
<version>5.3.31</version>

</dependency>

</dependencies>

</project>

**OUTPUT**

****

**Exercise 6**

**//LibraryManagementApplication.java**

package com.library;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

import org.springframework.stereotype.Service;

// Repository

@Repository

class BookRepository {

public String getBookName() {

return "Spring in Action";

}

}

// Service

@Service

class BookService {

private BookRepository bookRepository;

@Autowired

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void displayBook() {

System.out.println("Book Name: " + bookRepository.getBookName());

}

}

// Main Class

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.displayBook();

}

}

**applicationContext.xml**

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

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

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

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">

<!-- Enable annotation-based bean configuration -->

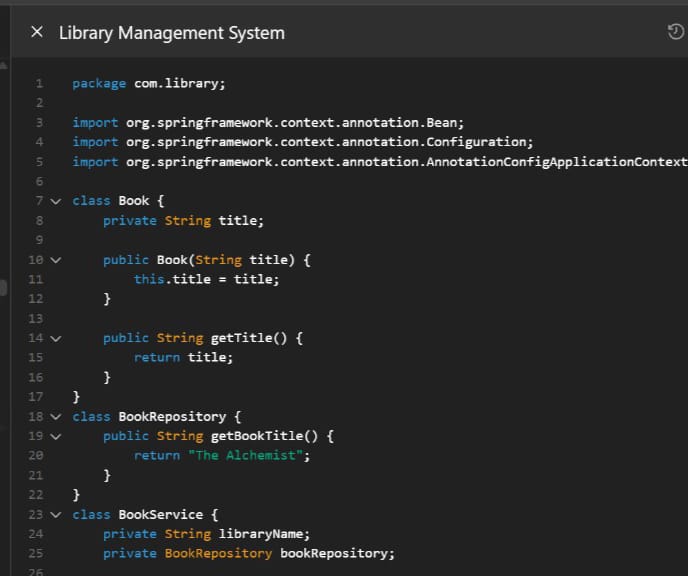
<context:component-scan base-package="com.library"/>

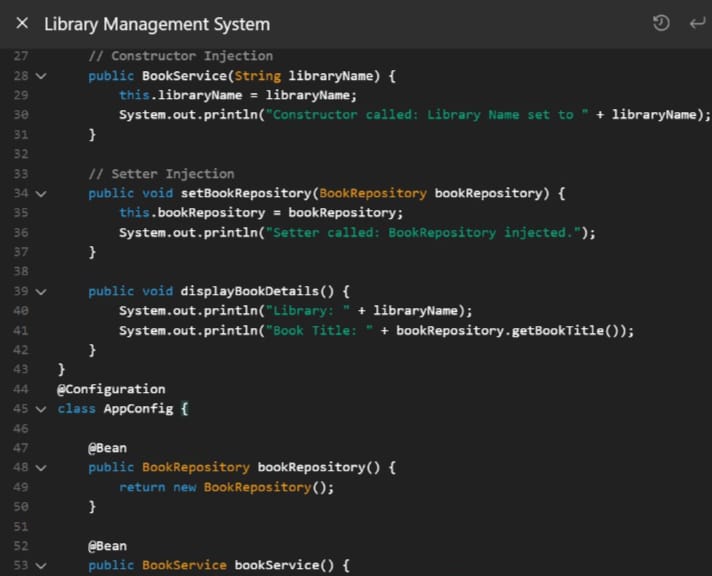
</beans>

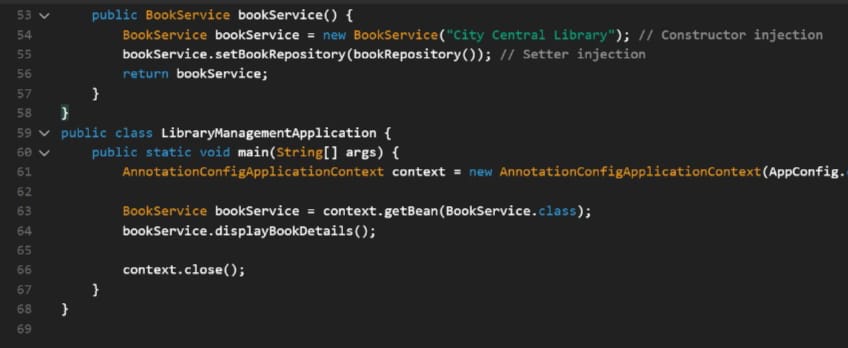
**OUTPUT**

****

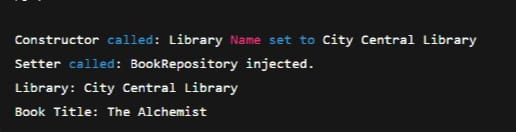
**Exercise 7**

****

****

****

**OUTPUT**

****

**EXERCISE 8**

package com.library;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.springframework.context.annotation.\*;

import org.springframework.stereotype.Component;

// Main application

public class LibraryManagementApplication {

public static void main(String[] args) {

AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

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

bookService.addBook();

bookService.removeBook();

context.close();

}

// ---------- Book Service ----------

@Component

public static class BookService {

public void addBook() {

System.out.println("Book added to the library.");

}

public void removeBook() {

System.out.println("Book removed from the library.");

}

}

// ---------- Aspect for Logging ----------

@Aspect

@Component

public static class LoggingAspect {

@Before("execution(\* com.library.LibraryManagementApplication.BookService.\*(..))")

public void beforeAdvice() {

System.out.println("[LOG] Method is about to be executed.");

}

@After("execution(\* com.library.LibraryManagementApplication.BookService.\*(..))")

public void afterAdvice() {

System.out.println("[LOG] Method has been executed.");

}

}

// ---------- Java Config ----------

@Configuration

@ComponentScan(basePackages = "com.library")

@EnableAspectJAutoProxy

public static class AppConfig {

}

}

**POM.XML**

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

**<**artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

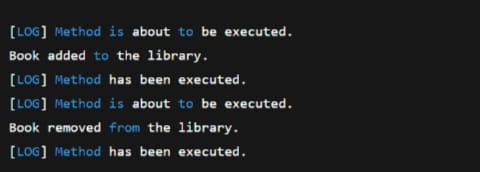
<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

**OUTPUT**

****

**EXERCISE 9**

// File: src/main/java/com/library/LibraryManagementApplication.java

package com.library;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.\*;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import org.springframework.beans.factory.annotation.Autowired;

import jakarta.persistence.\*;

import java.util.List;

@SpringBootApplication

public class LibraryManagementApplication {

public static void main(String[] args) {

SpringApplication.run(LibraryManagementApplication.class, args);

}

// ✅ Book Entity

@Entity

public static class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

private double price;

public Book() {

}

public Book(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

// Getters & Setters

public Long getId() {

return id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

}

// ✅ Repository Interface

@Repository

public interface BookRepository extends JpaRepository<Book, Long> {

}

// ✅ REST Controller

@RestController

@RequestMapping("/api/books")

public static class BookController {

@Autowired

private BookRepository bookRepository;

@GetMapping

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return bookRepository.findById(id).orElse(null);

}

@PostMapping

public Book addBook(@RequestBody Book book) {

return bookRepository.save(book);

}

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book bookDetails) {

Book book = bookRepository.findById(id).orElse(null);

if (book != null) {

book.setTitle(bookDetails.getTitle());

book.setAuthor(bookDetails.getAuthor());

book.setPrice(bookDetails.getPrice());

return bookRepository.save(book);

}

return null;

}

@DeleteMapping("/{id}")

public String deleteBook(@PathVariable Long id) {

bookRepository.deleteById(id);

return "Book deleted successfully!";

}

}

}

**Pom.xml**

<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>1.0.0</version>

<packaging>jar</packaging>

<name>LibraryManagement</name>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version>

</parent>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<!-- Spring Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 In-Memory Database -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

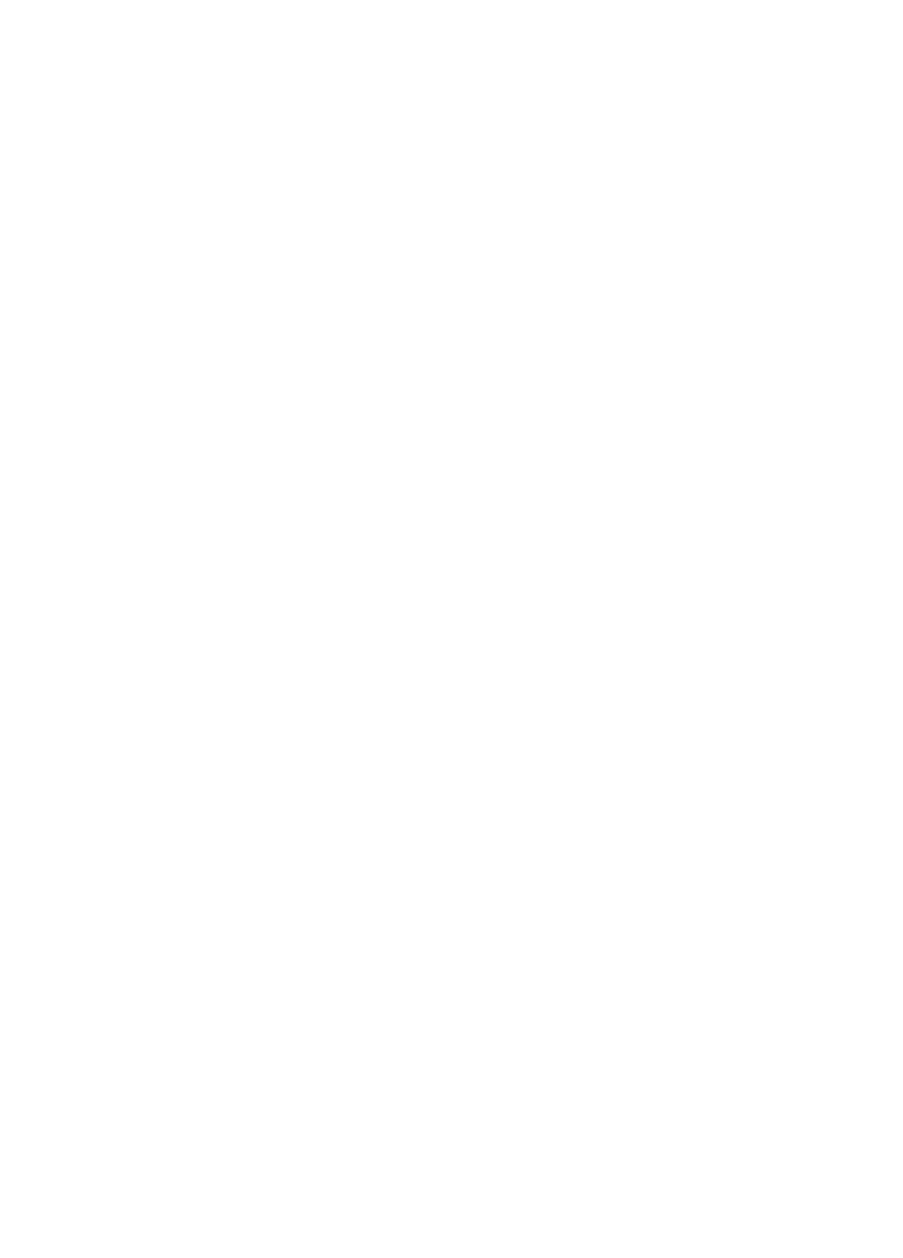
</dependency>

</dependencies>

</project>

**OUTPUT**



****