Spring Framework Exercises - Library Management

Exercise 1: Configuring a Basic Spring Application

Step 1: Set Up a Spring Project

- Create a Maven project named LibraryManagement.

- Add the following dependencies in pom.xml:

```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.3.29</version>
    </dependency>
</dependencies>
```

Step 2: Configure the Application Context

- Create applicationContext.xml in src/main/resources:

```
<?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 id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository"/>
    </bean>

</beans>
```

Step 3: Define Service and Repository Classes

```
// BookRepository.java
package com.library.repository;

public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Saving book: " + bookName);
    }
}

// BookService.java
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 addBook(String bookName) {
        System.out.println("Adding book: " + bookName);
        bookRepository.saveBook(bookName);
    }
}
```

Step 4: Run the Application

```
// MainApp.java
package com.library.main;

import com.library.service.BookService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
        BookService bookService = (BookService) context.getBean("bookService");
        bookService.addBook("Spring in Action");
    }
}
```

Output:

```
Adding book: Spring in Action
```

Saving book: Spring in Action

Exercise 2: Implementing Dependency Injection

(Steps are already covered in the applicationContext.xml and setter injection in BookService above)