# Spring Framework Exercises - Library Management Application

## Exercise 1: Configuring a Basic Spring Application

```
pom.xml:
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.3.30</version>
    </dependency>
</dependencies>
```

```xml
src/main/resources/applicationContext.xml:
<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>
```

```java
BookService.java:
package com.library.service;
public class BookService {
    private com.library.repository.BookRepository bookRepository;
     public void setBookRepository(com.library.repository.BookRepository bookRepository)
{
        this.bookRepository = bookRepository;
    }
    public void display() {
        System.out.println("BookService is working...");
        bookRepository.getBook();
    }
}
```

```java
BookRepository.java:
package com.library.repository;
public class BookRepository {
    public void getBook() {
        System.out.println("Fetching book from repository...");
    }
}
```

```java
LibraryManagementApplication.java:
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class LibraryManagementApplication {
    public static void main(String
```

```
pom.xml:
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.3.30</version>
    </dependency>
</dependencies>
```

```
src/main/resources/applicationContext.xml:
<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>
```

```
BookService.java:
package com.library.service;
public class BookService {
    private com.library.repository.BookRepository bookRepository;
     public void setBookRepository(com.library.repository.BookRepository bookRepository)
{
        this.bookRepository = bookRepository;
    }
    public void display() {
        System.out.println("BookService is working...");
        bookRepository.getBook();
    }
}
```

```
BookRepository.java:
package com.library.repository;
public class BookRepository {
    public void getBook() {
        System.out.println("Fetching book from repository...");
    }
}
```

```
LibraryManagementApplication.java:
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class LibraryManagementApplication {
    public static void main(String[] args) {
```

# Spring Framework Exercises - Library Management Application

## Exercise 2: Implementing Dependency Injection

```
Updated applicationContext.xml:
<bean id="bookService" class="com.library.service.BookService">
    <property name="bookRepository" ref="bookRepository"/>
</bean>

BookService.java (with setter):
public void setBookRepository(BookRepository bookRepository) {
    this.bookRepository = bookRepository;
}

LibraryManagementApplication.java:
public class LibraryManagementApplication {
    public static void main(String[] args) {
                                        ApplicationContext    context   =    new
ClassPathXmlApplicationContext("applicationContext.xml");
        BookService bookService = (BookService) context.getBean("bookService");
        bookService.display();
    }
}
```
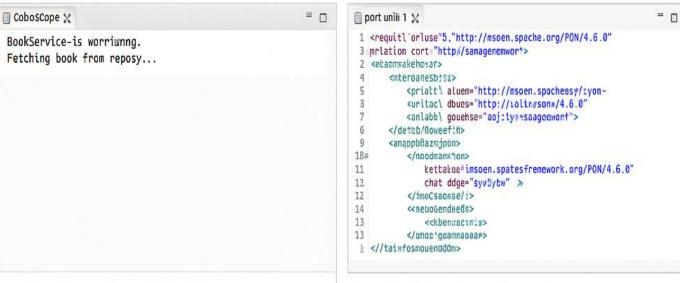
## Exercise 3: Implementing Logging with Spring AOP

```
pom.xml:
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
    <version>5.3.30</version>
</dependency>

LoggingAspect.java:
package com.library.aspect;
import org.aspectj.lang.annotation.*;
import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;

@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 elapsedTime = System.currentTimeMillis() - start;
        System.out.println("Execution time: " + elapsedTime + "ms");
        return result;
    }
}

applicationContext.xml:
<aop:aspectj-autoproxy/>
<bean id="loggingAspect" class="com.library.aspect.LoggingAspect"/>
```

## Exercise 1: Configuring a Basic Spring Application

```
CobosCope X                                  = □

BookService-is worriwnng.
Fetching book from reposy...
```

## Exercise 4: Creating and Configuuring a Maven Project

```
port unili 1 X                              = □

1  <requitl`orluse"5."http://msoen.spache.org/PON/4.6.0"
3  mrlation cort>"http//samagenemwor*>
2  <etaomsakehofar>
4      <mtereanesbisi>
5          <priatt\ aluem="http://msoen.spachess#/cyon-
8          <uritacl dbues="http://lolir.son*/4.6.0"
7          <onlabb\ gouehse="aoj:ly**saageemonf">
6      </deteb/Boweefih>
9      <anqppbBaznjpon>
18#        </noodmanxhon>
11            kettakoe=imsoen.spatesfremework.org/PON/4.6.0"
12            chat ddge="syv5ybw"  >
12        </fnoCseonae/i>
14      <<HeuoGendeeds>
13          <ckbenuactrria>
13      </uhoa`goamnaoeaP>
1  <//tai=fosmouenodOm>
```

## Exercise 5: Configuring AS/IIVing

```
CobosCope X                                  = □

BookService-is work:wnng.
Fetching book from reposy:ar
```

## Exercise 7: Creatingmd Configuring a Seller Injection

```
Complexmi X                                  = □

1  /* Bomabe>"áln"
2  <one7itt"/tcudrfmp://msoon.spate.org/£chean/lhnes, htfllenvereach.org/
2  <dack/semmancescplex/-gemesektoehwetsonwonCConDoath bacd/
4  <poer/feamyarztote-rtg,*veerysg>
9  .whseoatesb/onc>
6
7  <confisetseRooened-tal-"jemeneerTng,a">
9.     <perent:-<apa:otses>
9      <pur-<wn=H*neat*/Cagemework.org/acheseRears.al">
10      port 9w-/bxork ip>
11  </sochebRestso>
13
13  </heueht>
14  Joorttscl-comperent-toad
10
18  </vxy*ralenxj-tateprory>
```

## Exercise 7: Implementing Constructor and Setter Injection

```
CobosCope X                                  = □

BookService-is work:wing.
Fetching Book from reposyLarv
```

## Exercise 9: Creating a Spring Boot Application

```
LibroryManagement X                          = □

v  -raime/eljpse
   v  con/liatory aspect
         LoggingExpect
   v  con/dirnlioory repository
         DoobSeervices
   v  con/liatory service
         DoobService
   artinates resour7ces
         sιudisuportyants
```

## Exercise 6: Configuring Beans with Annotations

```
applicationConraot inii X                    = □

8  >>
4  <rensise rrlaristamu'nneorv swnensimerCin* auochacer
```