


Web API Design with Spring Boot Week 3 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Here's a friendly tip: as you watch the videos, code along with the videos. This will help you with the homework. When a screenshot is required, look for the icon:  You will keep adding to this project throughout this part of the course. When it comes time for the final project, use this project as a starter.

Project Resources: <https://github.com/promineotech/Spring-Boot-Course-Student-Resources>

Coding Steps:

- 1) In the application you've been building add a DAO layer:
 - a) Add the package, `com.promineotech.jeepp.dao`.
 - b) In the new package, create an interface named `JeepSalesDao`.
 - c) In the same package, create a class named `DefaultJeepSalesDao` that implements `JeepSalesDao`.
 - d) Add a method in the DAO interface and implementation that returns a list of Jeep models (class `Jeep`) and takes the model and trim parameters. Here is the method signature:

```
List<Jeep> fetchJeeps(JeepModel model, String trim);
```

- 2) In the Jeep sales service implementation class, inject the DAO interface as an instance variable. The instance variable should be private and should be named jeepSalesDao. Call the DAO method from the service method and store the returned value in a local variable named jeeps. Return the value in the jeeps variable (we will add to this later).
- 3) In the DAO implementation class (DefaultJeepSalesDao):
 - a) Add the class-level annotation: @Service.
 - b) Add a log statement in DefaultJeepSalesDao.fetchJeeps() that logs the model and trim level. Run the integration test. Produce a screenshot showing the DAO implementation class and the log line in the IDE's console.

The screenshot shows the Spring Tool Suite 4 IDE with the following components:

- Editor:** Displays the `DefaultJeepSalesDao.java` file. The code includes the `@Service` annotation, an `@Autowired` private field `jdbcTemplate`, and the `fetchJeeps` method which logs the model and trim parameters before returning `null`.
- Left Panel:** Shows the project structure with `FetchJeepTest` selected. Below it, the "Failure Trace" section displays an error from `org.springframework.jdbc.datasource.init.ScriptStatementFailedException`.
- Bottom Panel:** The "Console" tab shows the output of the test run, including log messages from the application and the test framework.

```
1 package com.promineotech.jeep.dao;
2
3 import java.util.List;
4
5 @Service
6 @Component
7 @Slf4j
8 public class DefaultJeepSalesDao implements JeepSalesDao {
9
10     @Autowired
11     private NamedParameterJdbcTemplate jdbcTemplate;
12
13     @Override
14     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
15         log.debug("DAO: model={}, trim={}", model, trim);
16         return null;
17     }
18 }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

Failure Trace:

```
org.springframework.jdbc.datasource.init.ScriptStatementFailedException: Failed to execute :
command-palette-input-ready:command-palette#inputReady
command-palette-page-stack-updated:command-palette#updateInputScope
itemsUpdated:command-palette#itemsUpdated
```

Console:

```
<terminated> FetchJeepTest [JUnit] C:\Users\shgely\Downloads\sts-4.16.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
:: Spring Boot ::
(v2.7.4)
2022-10-15 15:36:03.752 INFO 22384 --- [main] c.p.jeep.controller.FetchJeepTest
2022-10-15 15:36:03.753 DEBUG 22384 --- [main] c.p.jeep.controller.FetchJeepTest
2022-10-15 15:36:03.754 INFO 22384 --- [main] c.p.jeep.controller.FetchJeepTest
2022-10-15 15:36:07.627 INFO 22384 --- [main] c.p.jeep.controller.FetchJeepTest
2022-10-15 15:36:08.116 WARN 22384 --- [main] o.s.test.context.TestContextManag
```

- c) In DefaultJeepSalesDao, inject an instance variable of type NamedParameterJdbcTemplate.
 - d) Write SQL to return a list of Jeep models based on the parameters: model and trim. Be sure to utilize the SQL Injection prevention mechanism of the NamedParameterJdbcTemplate using :model_id and :trim_level in the query.
 - e) Add the parameters to a parameter map as shown in the video. Don't forget to convert the JeepModel enum value to a String (i.e., params.put("model_id", model.toString());)
 - f) Call the query method on the NamedParameterJdbcTemplate instance variable to return a list of Jeep model objects. Use a RowMapper to map each row of the result set. Remember to convert modelId to a JeepModel. See the video for details. Produce a screenshot to show the complete method in the implementation class. 🖥️
- 4) Add a getter in the Jeep class for modelPK. Add the @JsonIgnore annotation to the getter to exclude the modelPK value from the returned object.
 - 5) Run the test to produce a green status bar. Produce a screenshot showing the test and the green status bar. 🖥️

The screenshot shows the Spring Tool Suite 4 IDE. The main editor displays the `FetchJeepTest.java` file. The code includes imports for `org.assertj.core.api.Assertions` and `org.springframework.test.web.servlet.MockMvc`. It defines a `FetchJeepTest` class with a `testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied()` method. The test method uses `MockMvc` to send a GET request to `http://localhost:8080/jeeps?model=WRANGLER&trim=Sport` and asserts that the response status is `OK` and the body contains a list of Jeep models.

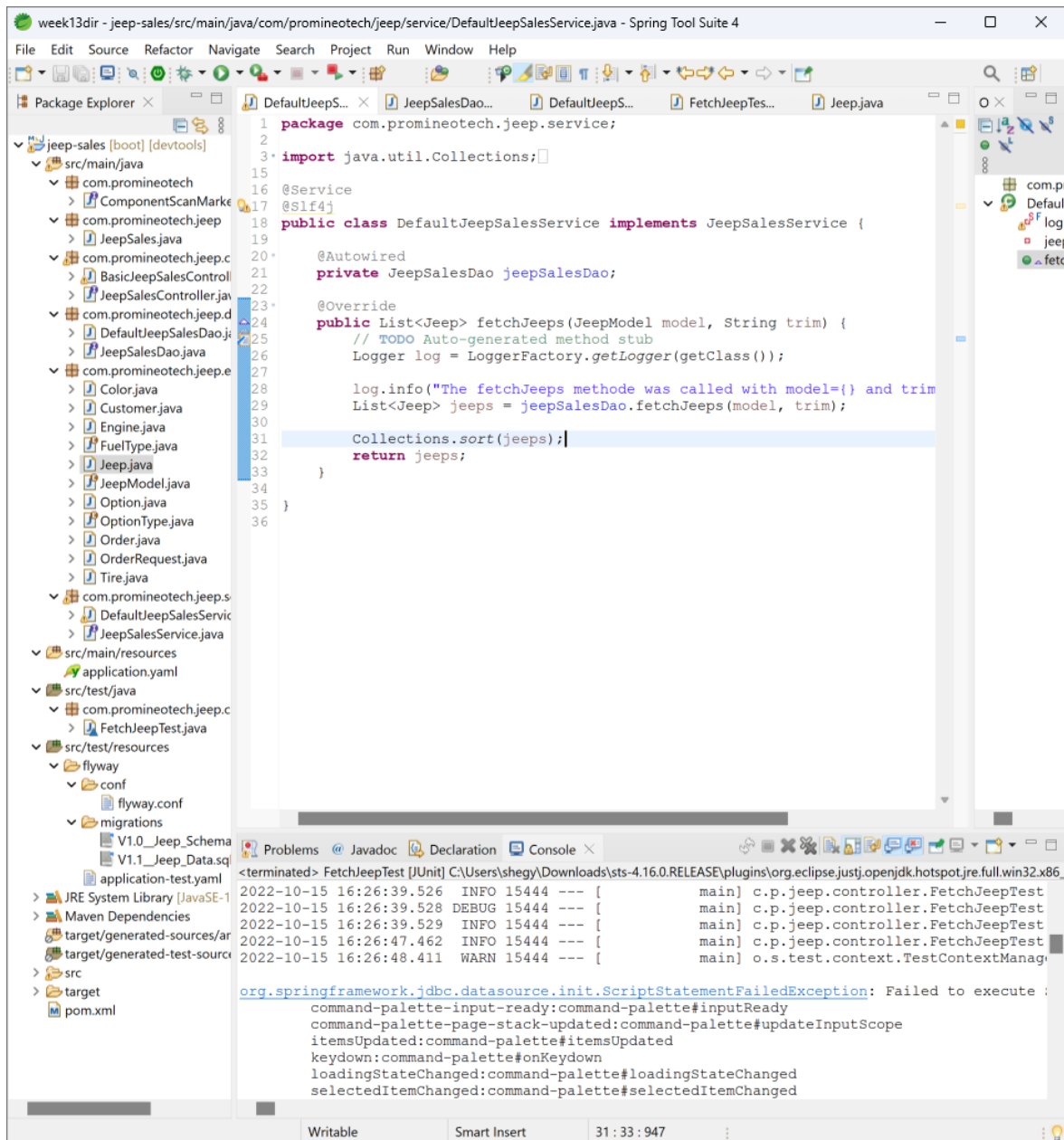
The left sidebar shows the 'Runs' tab with a green status bar indicating the test passed. The bottom pane shows the 'Console' tab with the following output:

```

2022-10-15 16:26:39.526 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:39.528 DEBUG 15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:39.529 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:47.462 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:48.411 WARN 15444 --- [main] o.s.test.context.TestContextManag

```

Screenshots of Code:



week13dir - jeep-sales/src/main/java/com/promineotech/jeep/dao/JeepSalesDao.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- jeep-sales [boot] [devtools]
 - src/main/java
 - com.promineotech
 - ComponentScanMarker
 - com.promineotech.jee
 - JeepSales.java
 - com.promineotech.jee.c
 - BasicJeepSalesControl
 - JeepSalesController.java
 - com.promineotech.jee.d
 - DefaultJeepSalesDao.java
 - JeepSalesDao.java
 - com.promineotech.jee.e
 - Color.java
 - Customer.java
 - Engine.java
 - FuelType.java
 - Jeep.java
 - JeepModel.java
 - Option.java
 - OptionType.java
 - Order.java
 - OrderRequest.java
 - Tire.java
 - com.promineotech.jee.s
 - DefaultJeepSalesService
 - JeepSalesService.java
 - src/main/resources
 - application.yaml
 - src/test/java
 - com.promineotech.jee.c
 - FetchJeepTest.java
 - src/test/resources
 - flyway
 - conf
 - flyway.conf
 - migrations
 - V1.0_Jeep_Schema
 - V1.1_Jeep_Data.sql
 - application-test.yaml
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - target/generated-sources/annotations
 - target/generated-test-sources/test-annotations
 - src
 - target
 - pom.xml

DefaultJeepS... JeepSalesDao... DefaultJeepS... FetchJeepTes... Jeep.java

```
1 package com.promineotech.jee.dao;
2
3 import java.util.List;
4
5
6
7
8 public interface JeepSalesDao {
9
10     /**
11      *
12      * @param model
13      * @param trim
14      * @return
15      */
16
17     List<Jeep> fetchJeeps(JeepModel model, String trim);
18
19
20 }
21
```

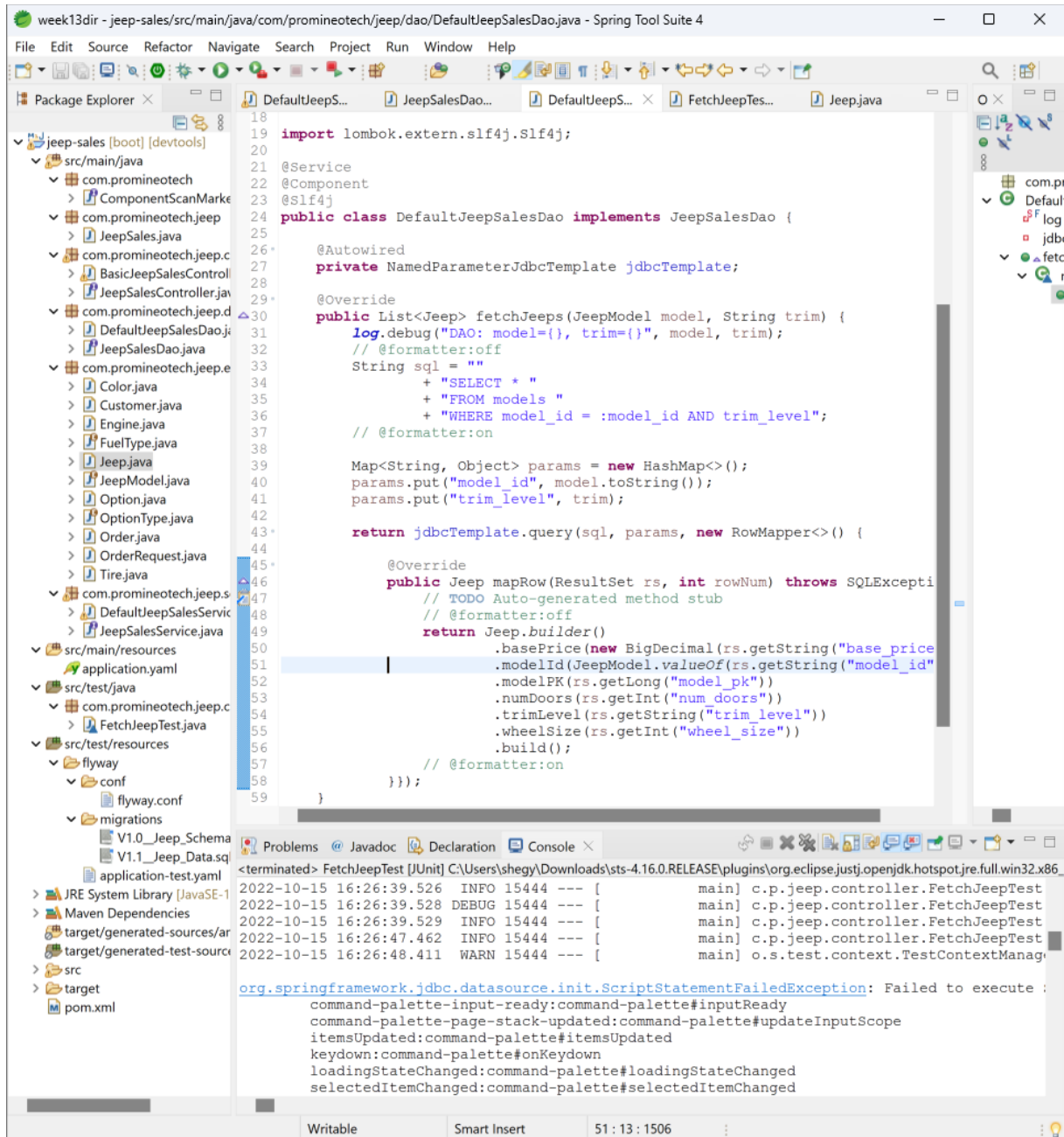
Problems @ Javadoc Declaration Console

<terminated> FetchJeepTest [JUnit] C:\Users\shegy\Downloads\sts-4.16.0.RELEASE\plugins\org.eclipse.jdt.launcher\org.eclipse.jdt.launcher.win32.x86_64.exe

Time	Level	Message
2022-10-15 16:26:39.526	INFO	15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:39.528	DEBUG	15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:39.529	INFO	15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:47.462	INFO	15444 --- [main] c.p.jee.controller.FetchJeepTest
2022-10-15 16:26:48.411	WARN	15444 --- [main] o.s.test.context.TestContextManag

org.springframework.jdbc.datasource.init.ScriptStatementFailedException: Failed to execute :
command-palette-input-ready:command-palette#inputReady
command-palette-page-stack-updated:command-palette#updateInputScope
itemsUpdated:command-palette#itemsUpdated
keydown:command-palette#onKeydown
loadingStateChanged:command-palette#loadingStateChanged
selectedItemChanged:command-palette#selectedItemChanged

Writable Smart Insert 1:1:0



Screenshots of Running Application:


```
week13dir - jeep-sales/src/main/java/com/promineetech/jeep/dao/DefaultJeepSalesDao.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
[terminated] FetchJeepTest [AJR] [C:\Users\shguy\Downloads\sts-4.16.0.RELEASE\plugins\org.eclipse.jdt.launcher\hotspot\jre.full.win32.x86_64_17.0.4\20220903-1038\prebin\java.exe (Oct 15, 2022, 4:26:33 PM - 4:26:49 PM) [pid: 15444]

:: Spring Boot ::
:: (v2.7.4)

2022-10-15 16:26:39.526 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest : Starting FetchJeepTest using Java 17.0.4.1 on DESKTOP-9HNL7LA with PID 15444 (started by shguy in
2022-10-15 16:26:39.528 DEBUG 15444 --- [main] c.p.jee.controller.FetchJeepTest : Running with Spring Boot v2.7.4, Spring v5.3.23
2022-10-15 16:26:39.529 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest : The following 1 profile is active: "test"
2022-10-15 16:26:47.462 INFO 15444 --- [main] c.p.jee.controller.FetchJeepTest : Started FetchJeepTest in 8.651 seconds (JVM running for 10.933)
2022-10-15 16:26:48.411 WARN 15444 --- [main] o.s.test.context.TestContextManager : Caught exception while invoking "beforeTestMethod" callback on TestExecutionListener [org.springframework
org.springframework.jdbc.datasource.init.ScriptStatementFailedException: Failed to execute SQL script statement #1 of class path resource [flyway/migrations/V1.0_jeep_schema.sql]: <DOCTYPE html> <
command-palette-input-ready:command-palette#inputReady
command-palette-page-stack-updated:command-palette#updateInputScope
itemsUpdated:command-palette#itemsUpdated
keydown:command-palette#onKeyDown
loadingStateChanged:command-palette#loadingStateChanged
selectedItemsChanged:command-palette#selectedItemsChanged
pageFetchError:command-palette#pageFetchError
" > <command-palette-mode data-char="*" data-scope-types="["<quot;<quot;"]" data-placeholder="Search issues and pull requests" ></command-palette-mode> <command-palette-mode data-char="*" data-sco
command-palette-input:command-palette#onInput
command-palette-select:command-palette#onSelect
command-palette-descope:command-palette#onDescope
command-palette-cleared:command-palette#onInputClear
" > <div class="js-search-icon &-flex flex-items-center mr-2" style="height: 26px"> <svg aria-hidden="true" height="16" viewbox="0 0 16 16" version="1.1" width="16" data-view-component="true"
at org.springframework.jdbc.datasource.init.ScriptUtils.executeSqlScript (ScriptUtils.java:282) ~[spring-jdbc-5.3.23.jar:5.3.23]
at org.springframework.jdbc.datasource.init.ResourceDatabasePopulator.populate (ResourceDatabasePopulator.java:273) ~[spring-jdbc-5.3.23.jar:5.3.23]
at org.springframework.jdbc.datasource.init.DatabasePopulatorUtils.execute (DatabasePopulatorUtils.java:54) ~[spring-jdbc-5.3.23.jar:5.3.23]
at org.springframework.jdbc.datasource.init.ResourceDatabasePopulator.execute (ResourceDatabasePopulator.java:249) ~[spring-jdbc-5.3.23.jar:5.3.23]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.lambda$executeSqlScript$4 (SqlScriptsTestExecutionListener.java:273) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.transaction.support.TransactionOperations.lambda$executeWithoutResult$0 (TransactionOperations.java:68) ~[spring-tx-5.3.23.jar:5.3.23]
at org.springframework.transaction.support.TransactionOperations.execute (TransactionOperations.java:140) ~[spring-tx-5.3.23.jar:5.3.23]
at org.springframework.transaction.support.TransactionOperations.executeWithoutResult (TransactionOperations.java:63) ~[spring-tx-5.3.23.jar:5.3.23]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.executeSqlScript (SqlScriptsTestExecutionListener.java:273) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.lambda$executeSqlScript$0 (SqlScriptsTestExecutionListener.java:203) ~[spring-test-5.3.23.jar:5.3.23]
at java.base/java.lang.Iterable.forEach (Iterable.java:75) ~[na:na]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.executeSqlScript (SqlScriptsTestExecutionListener.java:203) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.executeSqlScript (SqlScriptsTestExecutionListener.java:146) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener.beforeTestMethod (SqlScriptsTestExecutionListener.java:118) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.test.context.TestContextManager.beforeTestMethod (TestContextManager.java:120) ~[spring-test-5.3.23.jar:5.3.23]
at org.springframework.test.context.junit.JupiterSpringExtension.beforeEach (SpringExtension.java:174) ~[spring-test-5.3.23.jar:5.3.23]
at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.lambda$invokeBeforeEachCallbacks$2 (TestMethodTestDescriptor.java:143) ~[junit-jupiter-engine-5.8.2.jar:5.8.2]
at org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute (ThrowableCollector.java:73) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.invokeBeforeMethodOnCallbacksUntilExceptionOccurs (TestMethodTestDescriptor.java:139) ~[junit-jupiter-engine-5.8.2.jar:5.8.2]
at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.execute (TestMethodTestDescriptor.java:120) ~[junit-jupiter-engine-5.8.2.jar:5.8.2]
at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.execute (TestMethodTestDescriptor.java:160) ~[junit-jupiter-engine-5.8.2.jar:5.8.2]
at org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecursively$94 (NodeTestTask.java:151) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute (ThrowableCollector.java:73) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecursively$98 (NodeTestTask.java:137) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecursively$99 (NodeTestTask.java:139) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute (ThrowableCollector.java:73) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.NodeTestTask.executeRecursively (NodeTestTask.java:138) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at org.junit.platform.engine.support.hierarchical.NodeTestTask.execute (NodeTestTask.java:95) ~[junit-platform-engine-1.8.2.jar:1.8.2]
at java.base/java.util.ArrayList.forEach (ArrayList.java:151) ~[na:na]
```

URL to GitHub Repository:

<https://github.com/shguyes/Week13Assignment>