**Exercise 1: Mocking and Stubbing**

**CODE:**

**Myservice.java**

package com.example;  
  
public class MyService {  
 private ExternalApi externalApi;  
  
 public MyService(ExternalApi externalApi) {  
 this.externalApi = externalApi;  
 }  
  
 public String fetchData() {  
 return externalApi.getData();  
 }  
}

**MyServiceTest.java**

package com.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
import static org.mockito.Mockito.\*;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = *mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 *assertEquals*("Mock Data", result);  
 }  
}

**ExternalApi.java**

package com.example;  
  
public interface ExternalApi {  
 String getData();  
}

**OUTPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.

Exercise 2: Verifying Interactions

**CODE:**

**MyService.java**

package com.example;  
  
public class MyService {  
 private final ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public void fetchData() {  
 api.getData(); // we are verifying this call  
 }  
}

**MyServiceTest.java**

package com.example;  
  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 // Step 1: Create mock object  
 ExternalApi mockApi = *mock*(ExternalApi.class);  
  
 // Step 2: Use mock in service  
 MyService service = new MyService(mockApi);  
 service.fetchData(); // This should call mockApi.getData()  
  
 // Step 3: Verify the interaction  
 *verify*(mockApi).getData(); // This will pass only if getData() was called  
 }  
}

**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.example</groupId>  
 <artifactId>mockito-interaction-test</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <dependencies>  
 <!-- JUnit 5 -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.9.3</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Mockito -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Required for running JUnit 5 tests -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0-M9</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**ExternalApi.java**

package com.example;  
  
public interface ExternalApi {  
 String getData();  
}

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.