**Exercise 1: Mocking and Stubbing**

**1. External API Interface (Dependency to Mock) :**

public interface ExternalApi {

String getData();

}

**2. Service Class That Depends on External API**

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

// Calls external API (to be mocked during testing)

return externalApi.getData();

}

}

**3. JUnit Test with Mockito Mocking and Stubbing:**

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Step 1: Create Mock Object

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

// Step 2: Stub Method to Return Predefined Value

when(mockApi.getData()).thenReturn("Mock Data");

// Step 3: Inject Mock into Service

MyService service = new MyService(mockApi);

// Step 4: Call Service Method and Assert Result

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**Dependencies for Mockito and JUnit:**

<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>4.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

**Exercise 2: Verifying Interactions**

1. External API Interface (Dependency to Mock)

public interface ExternalApi {

String getData();

}

2. Service Class That Calls External API

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData(); // Method to be verified

}

}

### 3. JUnit Test with Mockito for Interaction Verification:

### import static org.mockito.Mockito.\*;

### import org.junit.jupiter.api.Test;

### import org.mockito.Mockito;

### public class MyServiceTest {

### @Test

### public void testVerifyInteraction() {

### // Step 1: Create Mock Object

### ExternalApi mockApi = Mockito.mock(ExternalApi.class);

### // Step 2: Inject Mock into Service

### MyService service = new MyService(mockApi);

### // Step 3: Call Service Method

### service.fetchData();

### // Step 4: Verify ExternalApi.getData() was called

### verify(mockApi).getData();

### }

### }

**Example with Arguments Verification:**

public interface ExternalApi {

String getDataById(int id);

}

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData(int id) {

return externalApi.getDataById(id);

}

}

Test with Argument Check:

@Test

public void testVerifyInteractionWithArguments() {

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData(42);

verify(mockApi).getDataById(42); // Verifying correct argument passed

}