**Exercise 2: Verifying Interactions Scenario: You need to ensure that a method is called with specific arguments.**

**Step 1: Create ExternalApi interface**

package com.rindhiyaa.junit;

public interface ExternalApi {

String getData();

}

**Step 2: Create MyService class with dependency injection**

package com.rindhiyaa.junit;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**// Step 3: Create JUnit test using Mockito**

package com.rindhiyaa.junit;

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

verify(mockApi).getData();

assertEquals("Mock Data", result);

System.out.println("Test passed: getData() was called and returned correct data.");

}

}

**OUTPUT**







