**Mockito Framework**

**Exercise 1 – Mocking and Stubbing**

**Code:**

package mocking;

public interface ExternalApi{

String getData();

}

package mocking;

public class Service {

private ExternalApi api;

public Service(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**Mocking and Stubbing**

package mocking;

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

public class ServiceTest {

*@Test*

public void testExternalApi() {

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

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

Service service = new Service(mockApi);

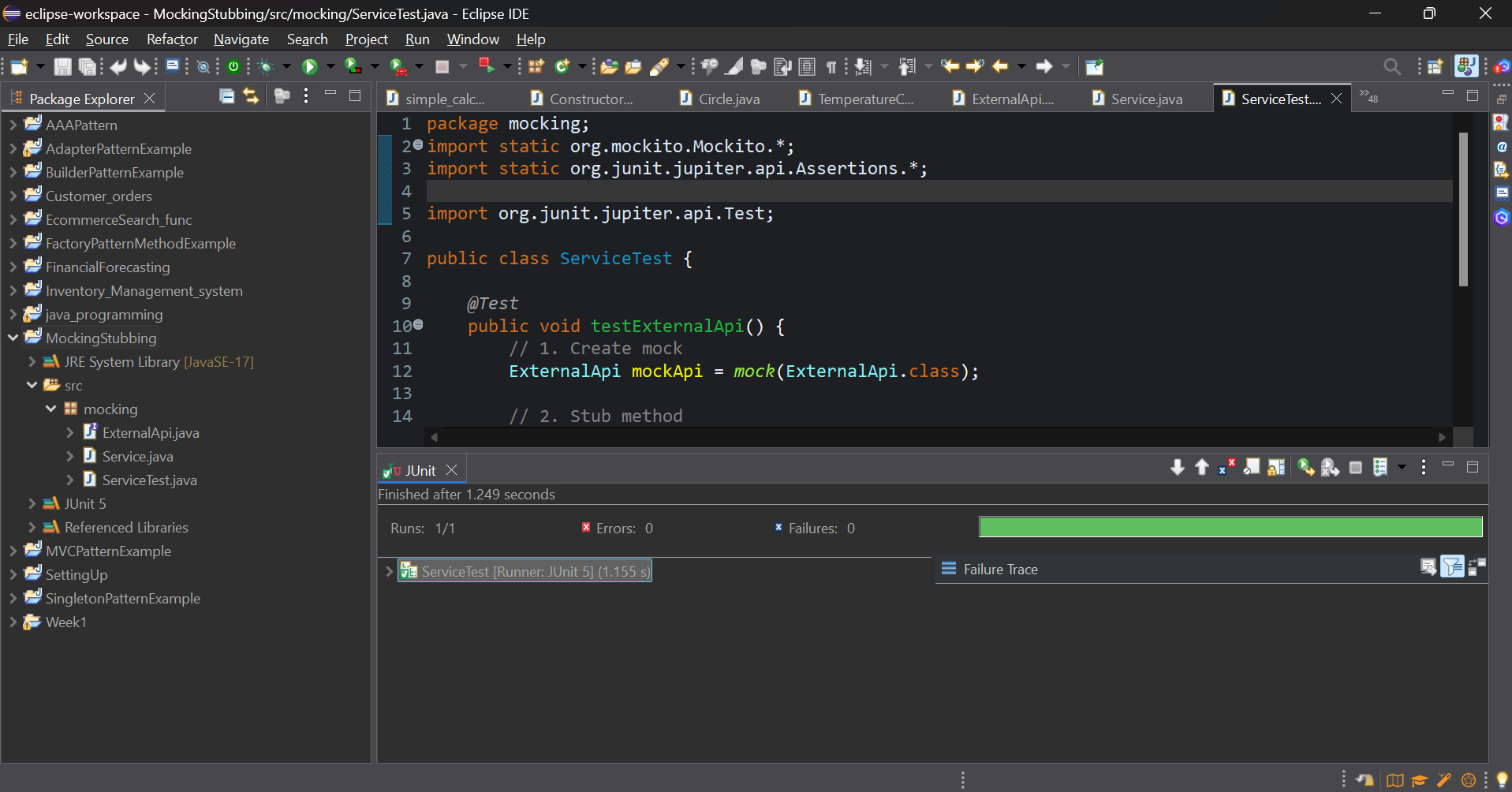
String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

**Output:**

****A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 2 – Verifying Interactions**

**Code:**

package interactions;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // This is what we want to verify

}

}

**Verifying Interactions**

package interactions;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

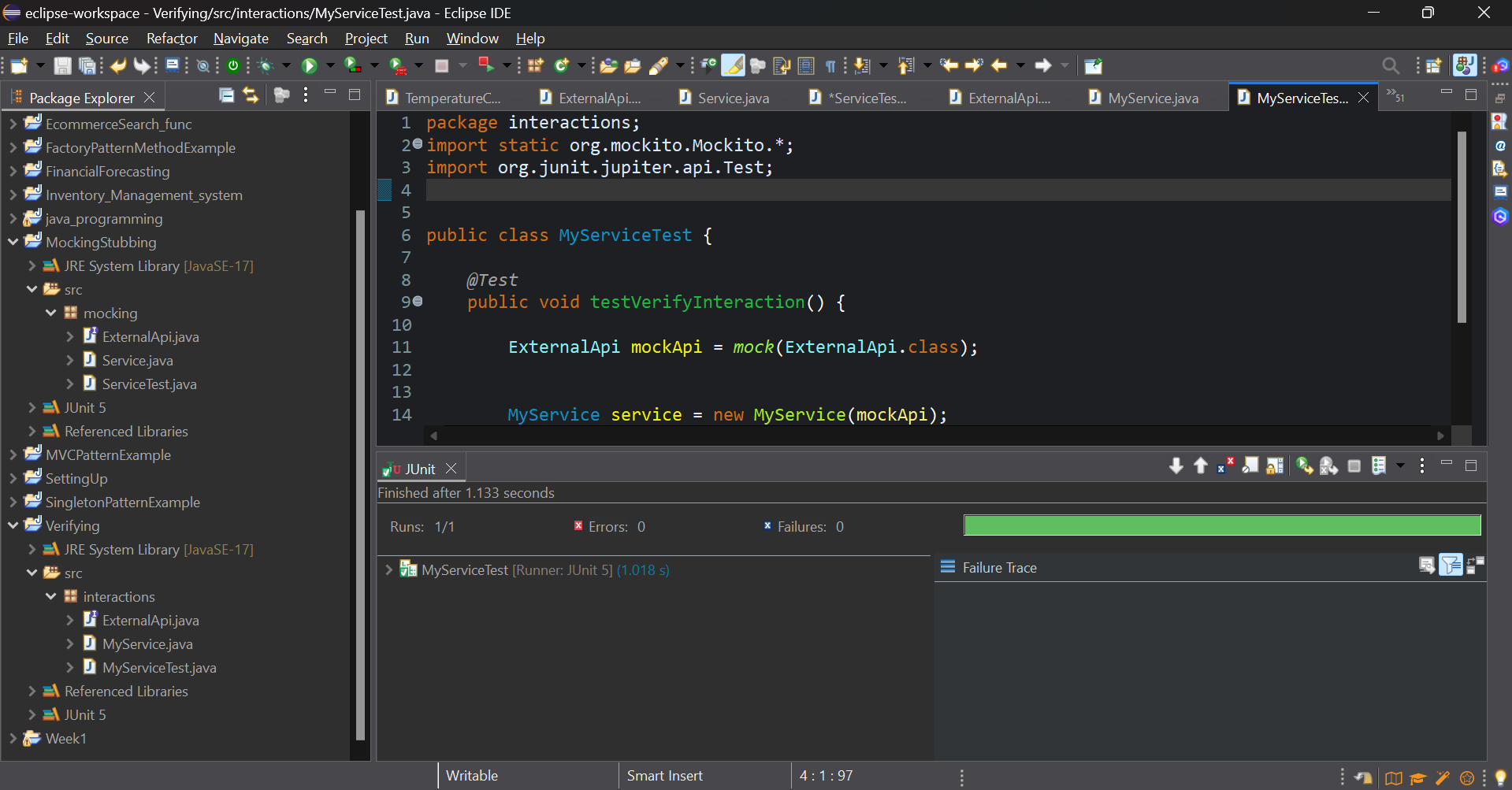
service.fetchData();

*verify*(mockApi).getData();

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.