**Exercise 1: Mocking and Stubbing**

Scenario: You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

**Codes:**

**ExternalApi.java:**

**package** junit;

**public** **interface** ExternalApi {

String getData();

}

**MyService.java:**

**package** junit;

**public** **class** MyService {

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api=api;

}

**public** String fetchData() {

**return** api.getData();

}

}

**MyServiceTest.java:**

**package** junit;

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

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

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

**import** org.mockito.Mockito;

**public** **class** MyServiceTest {

@Test

**public** **void** testExternalApi() {

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

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

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

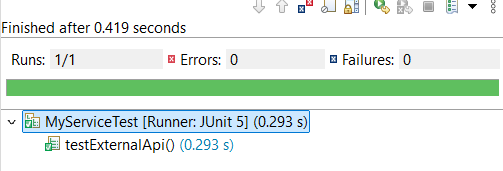
String result=service.fetchData();

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

}

}

**Output:**

****

**Exercise 2: Verifying Interactions**

Scenario: You need to ensure that a method is called with specific arguments.

**Codes:**

**ExternalApi1.java:**

**package** junit;

**public** **interface** ExternalApi1 {

String getData(String str);

}

**MyService1.java:**

**package** junit;

**public** **class** MyService1 {

**private** ExternalApi1 api;

**public** MyService1(ExternalApi1 api) {

**this**.api=api;

}

**public** String fetchData() {

**return** api.getData("Mockito");

}

}

**MyService1Test.java:**

**package** junit;

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

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

**import** org.mockito.Mockito;

**public** **class** MyService1Test {

@Test

**public** **void** testVerifyInteration() {

ExternalApi1 mockApi=Mockito.*mock*(ExternalApi1.**class**);

MyService1 service=**new** MyService1(mockApi);

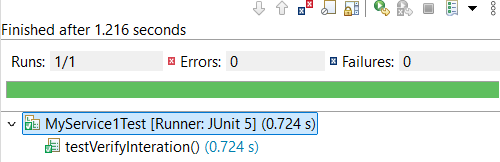
service.fetchData();

*verify*(mockApi).getData("Mockito");

}

}

**Output:**

****