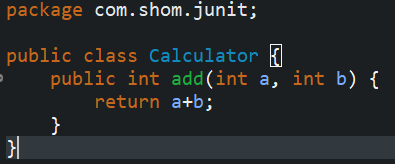
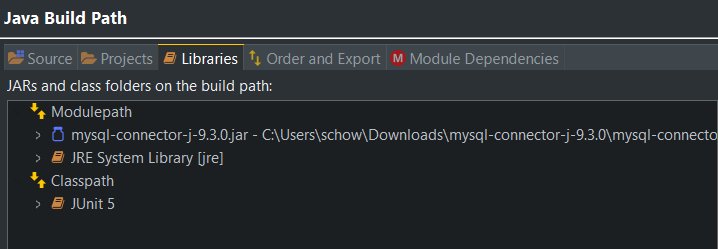
**Junit, Mockito, SLF4J**

**>JUNIT:**

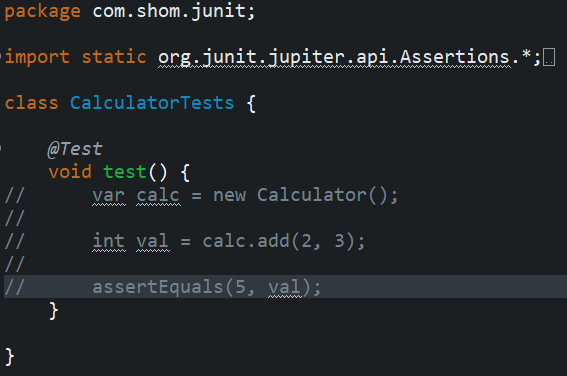
* **Exercise 1: Setting Up Junit**

(creating a java project)

****

**** (Adding Junit dependency)

(creating a new test class)



* **Exercise 3: Assertions in Junit**

package com.shom.junit;

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

import org.junit.jupiter.api.Test;

class CalculatorTests {

@Test

public void assertionTest() {

assertEquals(6, 2+4);

assertTrue(8>3);

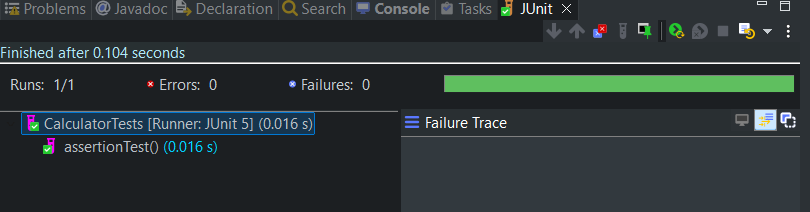
assertFalse(3>8);

assertNull(null);

assertNotNull(new Calculator());

}

}



* **Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

package com.shom.junit;

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

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

class CalculatorTests {

private Calculator calc;

*@BeforeEach*

public void setUp() {

calc = new Calculator();

}

*@AfterEach*

public void tearDown() {

calc = null;

}

*@Test*

public void testAdd() {

//arrange

int a = 2;

int b = 3;

//act

int val = calc.add(a,b);

//assert

*assertEquals*(5, val);

}

*@Test*

public void subTest() {

int a = 5;

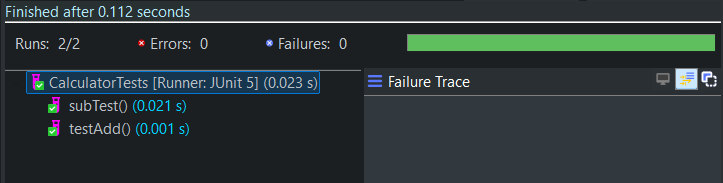
int b = 2;

int val = calc.sub(a,b);

*assertEquals*(3, val);

}

}



>**Mockito**

* **Exercise 1: Mocking and Stubbing**

ExternalApi.java

package com.shom.mockito;

public interface ExternalApi {

String getData();

}

MyService.java

package com.shom.mockito;

public class MyService {

private final ExternalApi extApi;

public MyService(ExternalApi api) {

this.extApi = api;

}

public String fetchData() {

return extApi.getData();

}

}

MyServiceTest.java

package com.shom.mockito;

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

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

class MyServiceTest {

*@Test*

public void testExternalApi() {

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

*when*(mockApi.getData())

.thenReturn("Mock Data");

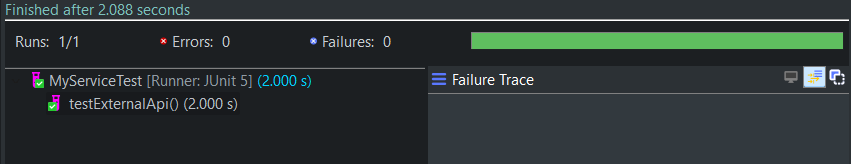
MyService service = new MyService(mockApi);

String result = service.fetchData();

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

}

}

****

* **Exercise 2: Verifying Interactions**

MyServiceTest.java

package com.shom.mockito;

import static org.mockito.Mockito.verify;

import static org.mockito.Mockito.when;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

class MyServiceTest {

@Test

public void testExternalApi() {

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

when(mockApi.getData())

.thenReturn("Mock Data");

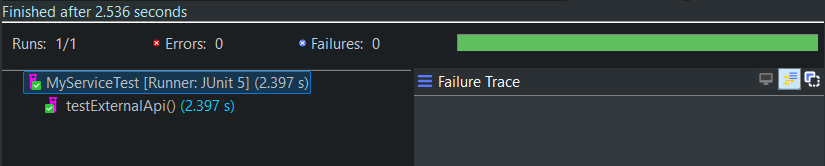
MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

}



**>SLF4J:**

* Exercise 1: Logging Error Messages and Warning Levels

LoggingExample.java

package com.shom.slf4j;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

private static final Logger logger = LoggerFactory.*getLogger*(LoggingExample.class);

public static void main(String[] args) {

logger.error("This is an error message");

logger.warn("This is a warning message");

}

}

