**Week 2-TDD using JUnit5 and Mockito**

**Exercise 1: Setting Up JUnit**

**Scenario:**

Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>junitworkingdemo</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

Calculator.java

**package** com.example;

**public** **class** Calculator {

**public** **int** add(**int** a, **int** b) {

**return** a + b;

}

}

CaluculatorTest.java

**package** com.example;

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

**public** **class** CalculatorTest {

@Test

**public** **void** testAdd() {

Calculator calc = **new** Calculator();

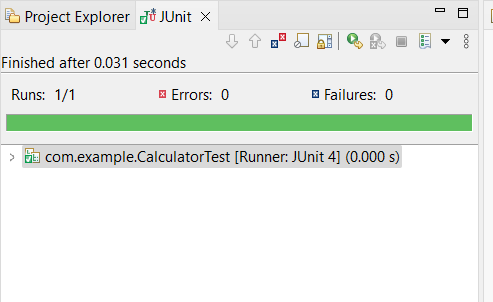
**int** result = calc.add(2, 3);

*assertEquals*(5, result);

}

}

**OUTPUT:**

****

**Exercise 3: Assertions in Junit**

**Scenario:**

AssertionsTest.java

**package** com.example;

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

**public** **class** AssertionsTest {

@Test

**public** **void** testAssertions() {

// Assert equals

*assertEquals*(5, 2 + 3);

// Assert true

*assertTrue*(5 > 3);

// Assert false

*assertFalse*(5 < 3);

// Assert null

*assertNull*(**null**);

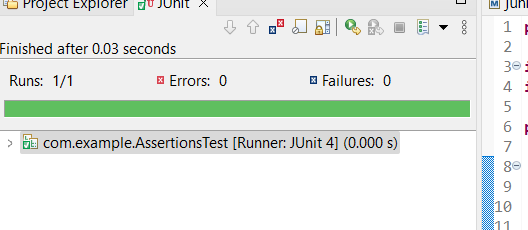
// Assert not null

*assertNotNull*(**new** Object());

}

}

**OUTPUT:**

****

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

**Scenario:**

BankAccount.java

**package** com.example;

**public** **class** BankAccount {

**private** **int** balance = 0;

**public** **void** deposit(**int** amount) {

balance += amount;

}

**public** **void** withdraw(**int** amount) {

balance -= amount;

}

**public** **int** getBalance() {

**return** balance;

}

}

BankAccountTest.java

**package** com.example;

**import** org.junit.Before;

**import** org.junit.After;

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

**public** **class** BankAccountTest {

**private** BankAccount account;

@Before

**public** **void** setUp() {

System.***out***.println("Setting up account...");

account = **new** BankAccount();

account.deposit(100);

}

@After

**public** **void** tearDown() {

System.***out***.println("Cleaning up account...");

account = **null**;

}

@Test

**public** **void** testDeposit() {

account.deposit(50);

// Assert

*assertEquals*(150, account.getBalance());

}

@Test

**public** **void** testWithdraw() {

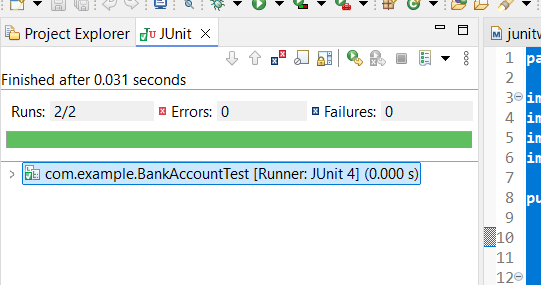
account.withdraw(30);

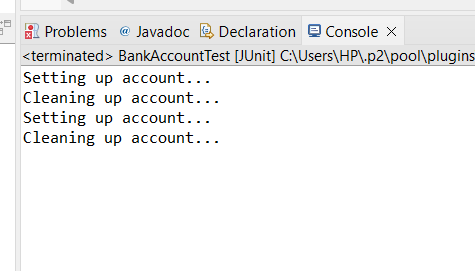
*assertEquals*(70, account.getBalance());

}

}

**OUTPUT:**

****

****

**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**Scenario:**

MyService.java

**package** com.example;

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

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

MyServiceTest.java

**package** com.example;

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

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

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

@Test

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

// Arrange: Create mock and stub

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

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

// Act: Use the mock in the service

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

String result = service.fetchData();

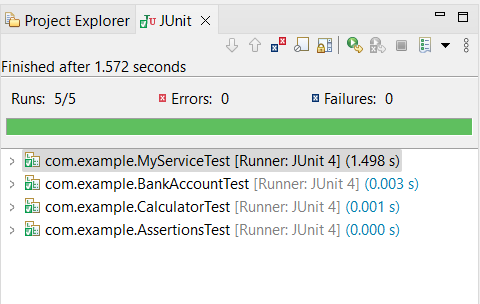
// Assert: Check expected result

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

}

}

**OUTPUT:**

****

**Exercise 2: Verifying Interactions**

**Scenario:**

MyServiceVerifyTest.java

**package** com.example;

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

**import** org.junit.Test;

**public** **class** MyServiceVerifyTest {

@Test

**public** **void** testVerifyInteraction() {

// Step 1: Create a mock

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

// Step 2: Inject mock into the service

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

// Step 3: Call method that should trigger the mock

service.fetchData();

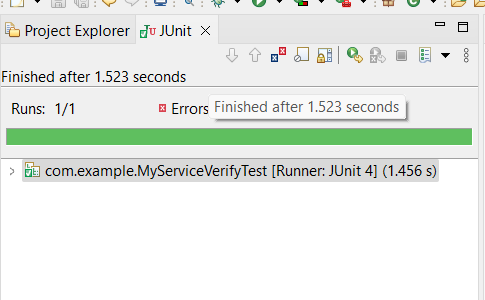
// Step 4: Verify that getData() was called exactly once

*verify*(mockApi).getData();

}

}

**OUTPUT:**

****

**Logging using SLF4J**

**Exercise 1: Logging Error Messages and Warning Levels**

Logging.java

**package** com.example;

**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) {

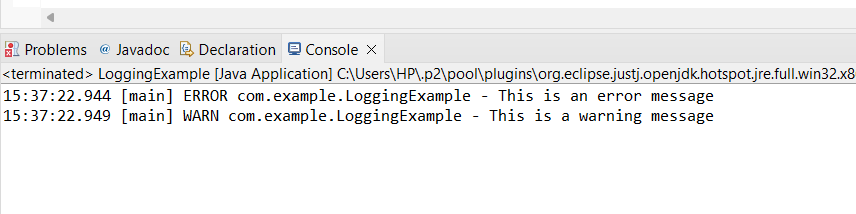
***logger***.error("This is an error message");

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

}

}

**OUTPUT:**

****