**Deep Skilling - Week 2 Mandatory Hands-on Exercises**

**Skill : TDD Using JUnit5 and Mockito**

**JUnit5 HandOn**

**Exercise - 1: Setting Up JUnit**

**Code:**

**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.testingexample</groupId>**

**<artifactId>JUnitDemoProgram</artifactId>**

**<version>0.0.1-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.testingexample;**

**public class Calculator {**

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

**return a + b;**

**}**

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

**return a \* b;**

**}**

**}**

**CalculatorTest.java**

**package com.testingexample;**

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

**import org.junit.Test;**

**public class CalculatorTest {**

***@Test***

**public void testAdd() {**

**Calculator calci = new Calculator();**

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

***assertEquals*(5, result);**

**}**

***@Test***

**public void testMul() {**

**Calculator calci = new Calculator();**

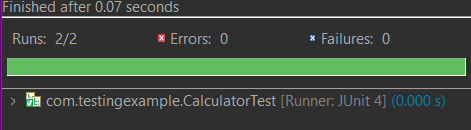
**int result = calci.mul(2, 3);**

***assertEquals*(6, result);**

**}**

**}**

**Output:**

****

**Exercise - 2: Writing Basic JUnit Tests**

**Code:**

**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.exercise2example</groupId>**

**<artifactId>JunitTestsExample</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**CalculatorApp.java**

**package com.exercise2example;**

**public class CalculatorApp {**

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

**return a + b;**

**}**

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

**return a \* b;**

**}**

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

**return a - b;**

**}**

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

**if(b == 0)**

**throw new IllegalArgumentException("Divider cannot be zero");**

**return a / b;**

**}**

**}**

**CalculatorAppTest.java**

**package com.exercise2example;**

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

**import org.junit.Test;**

**public class CalculatorAppTests {**

**CalculatorApp calci = new CalculatorApp();**

***@Test***

**public void testAdd() {**

***assertEquals*(5, calci.add(2, 3));**

**}**

***@Test***

**public void testSubtract() {**

***assertEquals*(2, calci.subtract(5, 3));**

**}**

***@Test***

**public void testMultipy() {**

***assertEquals*(6, calci.multiply(2, 3));**

**}**

***@Test***

**public void testDivide() {**

***assertEquals*(5, calci.divide(15, 3));**

**}**

***@Test*(expected = IllegalArgumentException.class)**

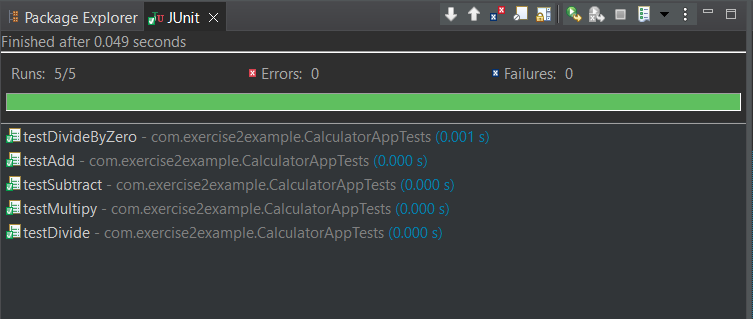
**public void testDivideByZero() {**

**calci.divide(9, 0);**

**}**

**}**

**Output:**

****

**Exercise - 3: Assertions in JUnit**

**Code:**

**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.assertions</groupId>**

**<artifactId>JUnitAssertionsExample</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**AdditionOperation.java**

**package com.example.assertions;**

**public class AdditionOperation {**

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

**return a + b;**

**}**

**}**

**AdditionTest.java**

**package com.example.assertions;**

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

**import org.junit.Test;**

**public class AdditionTest {**

**AdditionOperation add = new AdditionOperation();**

***@Test***

**public void testOperation() {**

***assertEquals*(5, add.performOp(2, 3));**

***assertTrue*(5 > add.performOp(1, 2));**

***assertFalse*(5 > add.performOp(10, 48));**

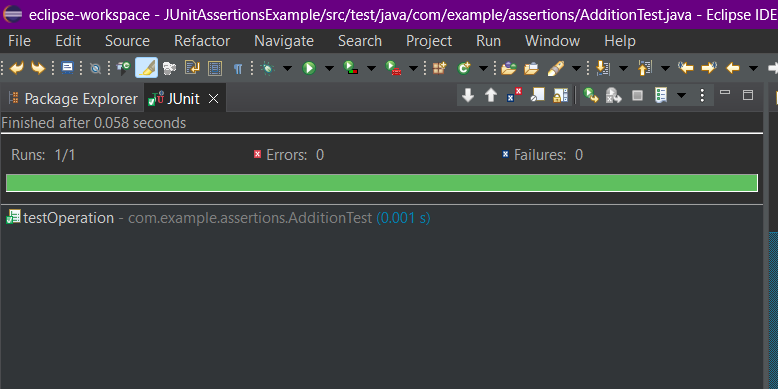
***assertNull*(null);**

***assertNotNull*(new AdditionOperation());**

**}**

**}**

**Output:**

****

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

**Code:**

**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>teardownMethods</groupId>**

**<artifactId>JUnitTeardownExample</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**BankAccountApp.java**

**package teardownMethods;**

**public class BankAccountApp {**

**private int balance;**

**public BankAccountApp(int initialBalance) {**

**this.balance = initialBalance;**

**}**

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

**balance += amount;**

**}**

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

**if (amount > balance) {**

**throw new IllegalArgumentException("Insufficient balance");**

**}**

**balance -= amount;**

**}**

**public int getBalance() {**

**return balance;**

**}**

**}**

**BankAppTest.java**

**package teardownMethods;**

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

**import org.junit.Test;**

**import org.junit.Before;**

**import org.junit.After;**

**public class BankAppTest {**

**private BankAccountApp account;**

***@Before***

**public void setUp() {**

**// Setup**

**account = new BankAccountApp(1000);**

**System.*out*.println("Before Test: New account created with 1000 balance");**

**}**

***@After***

**public void tearDown() {**

**// Teardown**

**System.*out*.println("After Test: Test finished\n");**

**}**

***@Test***

**public void testDeposit() {**

**// Arrange**

**int depositAmount = 500;**

**// Act**

**account.deposit(depositAmount);**

**// Assert**

***assertEquals*(1500, account.getBalance());**

**}**

***@Test***

**public void testWithdraw() {**

**// Arrange**

**int withdrawAmount = 400;**

**// Act**

**account.withdraw(withdrawAmount);**

**// Assert**

***assertEquals*(600, account.getBalance());**

**}**

***@Test*(expected = IllegalArgumentException.class)**

**public void testWithdrawInsufficientBalance() {**

**// Arrange**

**int withdrawAmount = 2000;**

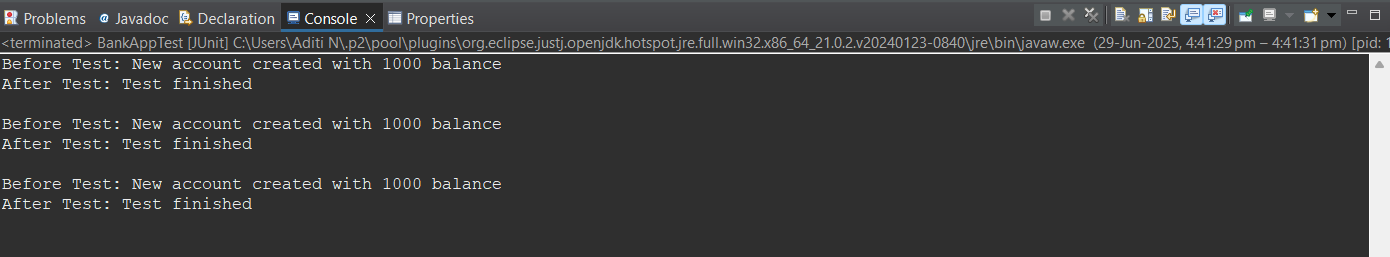
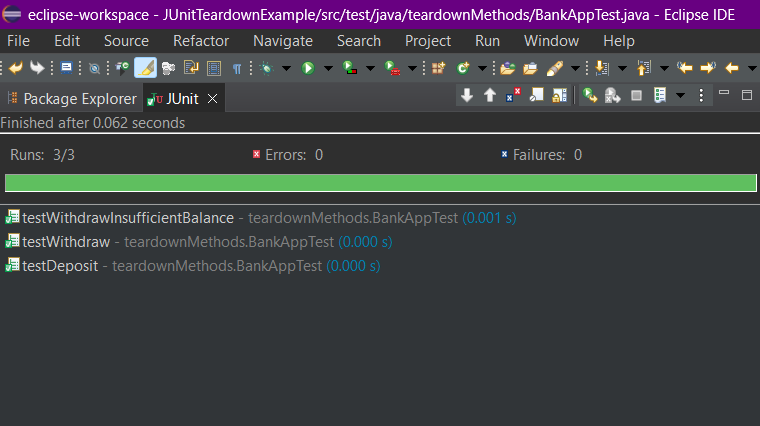
**// Act**

**account.withdraw(withdrawAmount); // should throw exception**

**// Assert is handled by the expected exception**

**}**

**}**

**Output:**

**Mockito HandOn**

**Exercise - 1: Mocking and Stubbing**

**Code:**

**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>mockitoexercise1</groupId>**

**<artifactId>MockitoDemo</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>org.junit.jupiter</groupId>**

**<artifactId>junit-jupiter</artifactId>**

**<version>5.9.3</version>**

**<scope>test</scope>**

**</dependency>**

**<dependency>**

**<groupId>org.mockito</groupId>**

**<artifactId>mockito-core</artifactId>**

**<version>5.11.0</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**DemoAPI.java**

**package mockito1;**

**public interface DemoAPI {**

**String getData();**

**}**

**ReqService.java**

**package mockito1;**

**//import mockito1.DemoAPI;**

**public class ReqService {**

**private DemoAPI myapi;**

**public ReqService(DemoAPI api) {**

**this.myapi = api;**

**}**

**public String fetch() {**

**return myapi.getData();**

**}**

**}**

**ServiceTest.java**

**package mockito1;**

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

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

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

**class ServiceTest {**

***@Test***

**public void testExternalApi() {**

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

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

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

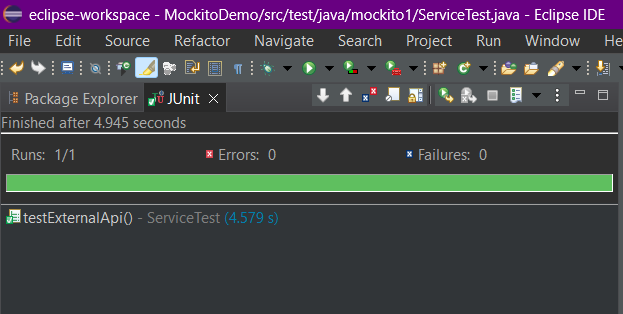
**String result = service.fetch();**

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

**}**

**}**

**Output:**

****

**Exercise - 2: Verifying Interactions**

**Code:**

**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>mockitoexercise2</groupId>**

**<artifactId>MockitoInteractions</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>org.junit.jupiter</groupId>**

**<artifactId>junit-jupiter</artifactId>**

**<version>5.9.3</version>**

**<scope>test</scope>**

**</dependency>**

**<dependency>**

**<groupId>org.mockito</groupId>**

**<artifactId>mockito-core</artifactId>**

**<version>5.11.0</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**SampleAPI.java**

**package mockitoexercise2;**

**public interface SampleAPI {**

**String getData();**

**}**

**Service.java**

**package mockitoexercise2;**

**public class Service {**

**private SampleAPI externalApi;**

**public Service(SampleAPI externalApi) {**

**this.externalApi = externalApi;**

**}**

**public String fetchData() {**

**return externalApi.getData();**

**}**

**}**

**ServiceTest.java**

**package mockitoexercise2;**

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

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

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

**class ServiceTest {**

***@Test***

**public void testVerifyInteraction() {**

**// Arrange**

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

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

**// Act**

**service.fetchData();**

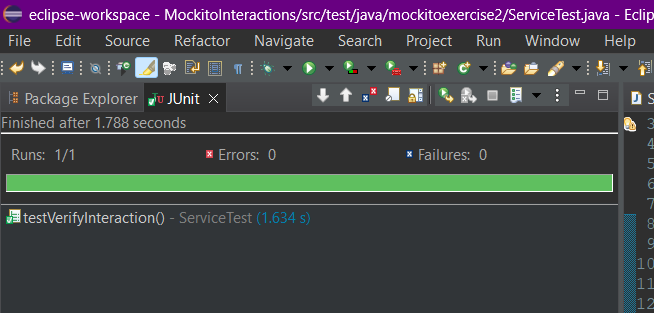
**// Assert (verify)**

***verify*(mockApi).getData(); // Confirms getData() was called once**

**}**

**}**

**Output:**

****