**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

**Step 1:** Created a new Java project named JUnitDemo in Eclipse.

**Step 2:** Adding JUnit 5 in Eclipse:

1. Right-click on the project → Build Path → Add Libraries…
2. Select **JUnit** → Click Next
3. Select **JUnit 5** → Click Finish

**Step 3:** Creating 2 packages com.example.calculator, com.example.calculator.test and 2 classes in them respectively.

**Calculator.java:**

package com.example.calculator;

public class Calculator {

public int add(int a,int b){

return a+b;

}

public int multiply(int a,int b){

return a\*b;

}

}

**CalculatorTest.java:**

package com.example.calculator.test;

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

import org.junit.jupiter.api.Test;

import com.example.calculator.Calculator;

public class CalculatorTest {

Calculator calculator = new Calculator();

@Test

public void testAdd() {

assertEquals(5, calculator.add(2, 3));

}

@Test

public void testMultiply() {

assertEquals(6, calculator.multiply(2, 3));

}

}

**Step 4:** Add requirements in **module-info.java** or simply delete the file:

module JUnitDemo {

requires org.junit.jupiter.api;

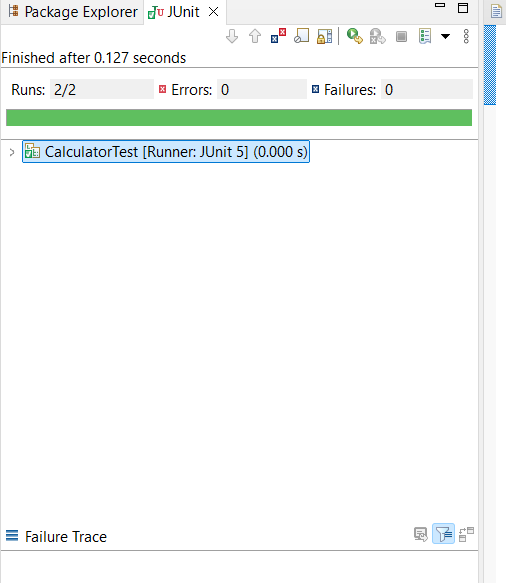
exports com.example.calculator;

}

**Step5:** Running Junit Tests:

1. Open CalculatorTest.java.
2. Right-click inside the file → Choose **Run As → JUnit Test**.
3. Eclipse runs the test and shows the **JUnit panel** with a green bar if all tests pass.

**Output:**



**Exercise 3: Assertions in JUnit**

**Step1:** Creating a class **AssertionsTest** in com.example.calculator.test package.

**AssertionsTest.java:**

package com.example.calculator.test;

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

import org.junit.jupiter.api.Test;

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());

}

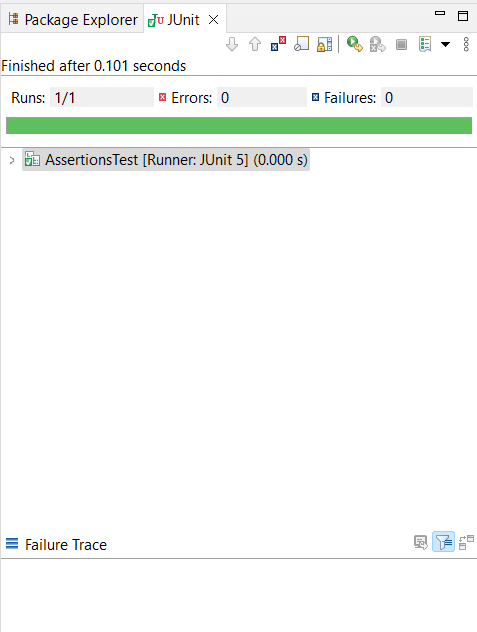
}

**Step2:** Ensure that **JUnit5** library and requirements in **module-info.java** are added.

**Step3:** Running Junit Tests:

1. Open AssertionsTest.java.
2. Right-click inside the file → Choose **Run As → JUnit Test**.
3. Eclipse runs the test and shows the **JUnit panel** with a green bar if all tests pass.

**Output:**



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

**Step1:** Created a new package com.example.bank and added a simple class BankAccount.java.

**BankAccount.java:**

package com.example.bank;

public class BankAccount {

private String accountHolder;

private double balance;

public BankAccount(String accountHolder,double initialBalance){

this.accountHolder = accountHolder;

this.balance = initialBalance;

}

public void deposit(double amount){

if(amount>0)

balance+=amount;

}

public void withdraw(double amount){

if(amount>0 && balance>=amount)

balance-=amount;

}

public double getBalance(){

return balance;

}

}

**Step2:** Created a separate test package com.example.bank.test and added BankAccountTest.java.

**BankAccountTest.java:**

package com.example.bank.test;

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

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.Test;

import com.example.bank.BankAccount;

public class BankAccountTest {

private BankAccount account;

@BeforeEach

public void setUp(){

// Arrange

account = new BankAccount("Alice",1000.0);

System.out.println("Setup complete");

}

@AfterEach

public void tearDown(){

account = null;

System.out.println("Teardown complete");

}

@Test

public void testDeposit(){

// Act

account.deposit(500.0);

// Assert

assertEquals(1500.0,account.getBalance(),0.001);

}

@Test

public void testWithdraw(){

// Act

account.withdraw(200.0);

// Assert

assertEquals(800.0,account.getBalance(),0.001);

}

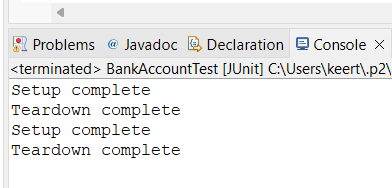
}

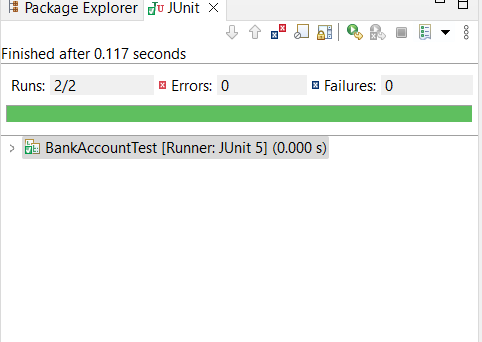
**Step3:** Ensure that **JUnit5** library and requirements in **module-info.java** are added.

**Step4:** Running JUnit Tests

1. Right-click on BankAccountTest.java.
2. Select Run As > JUnit Test.
3. Check the JUnit result tab and console for output.

**Output:**

**Console:**

**JUnit result tab:**

**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**Step1:** Create a New Maven Project in Eclipse:

Go to File → New → Project…

Select Maven Project → Click Next

Choose Create a simple project (skip archetype selection) → Next

Fill:

* Group Id: com.example
* Artifact Id: MockitoTestExample

Click Finish

**Step2:** Add Dependencies to **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

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

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>MockitoTestExample</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.2.5</version>

</plugin>

</plugins>

</build>

</project>

**Step3:** Create a package **com.example** in src/main/java and create following classes:

**ExternalApi.java:**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java:**

package com.example;

public class MyService{

private ExternalApi api;

public MyService(ExternalApi api){

this.api=api;

}

public String fetchData(){

return api.getData();

}

}

**Step4:** Create a package **com.example** in src/test/java and create following class:

**MyServiceTest.java:**

package com.example;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

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);

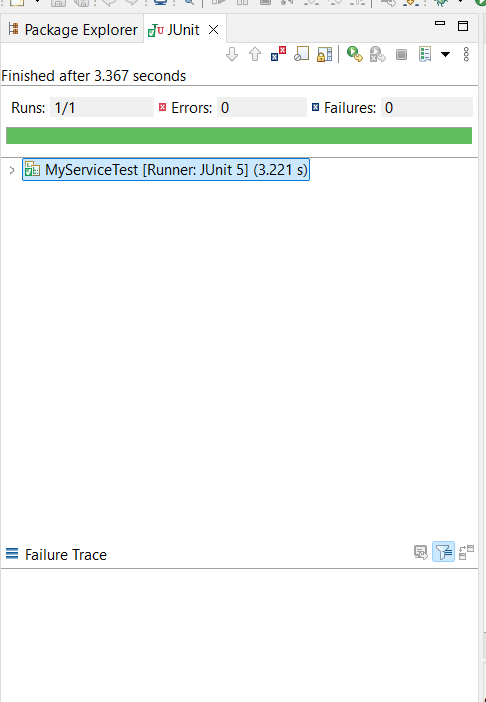
}

}

**Step5:** Run the Test

1. Open MyServiceTest.java.
2. Right-click inside the file → Choose **Run As → JUnit Test**.
3. Eclipse runs the test and shows the **JUnit panel** with a green bar if all tests pass.

**Output:**



**Exercise 2: Verifying Interactions**

**Step1:** Create a New Maven Project in Eclipse:

Go to File → New → Project…

Select Maven Project → Click Next

Choose Create a simple project (skip archetype selection) → Next

Fill:

* Group Id: com.example
* Artifact Id: MockitoVerifyExample

Click Finish

**Step2:** Add JUnit and Mockito Dependencies to **pom.xml.**

**Step3:** Create a package **com.example** in src/main/java and create following classes:

**ExternalApi.java:**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java:**

package com.example;

public class MyService{

private ExternalApi api;

public MyService(ExternalApi api){

this.api=api;

}

public String fetchData(){

return api.getData();

}

}

**Step4:** Create a package **com.example** in src/test/java and create following class:

**MyServiceTest.java:**

package com.example;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction(){

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

MyService service=new MyService(mockApi);

service.fetchData();

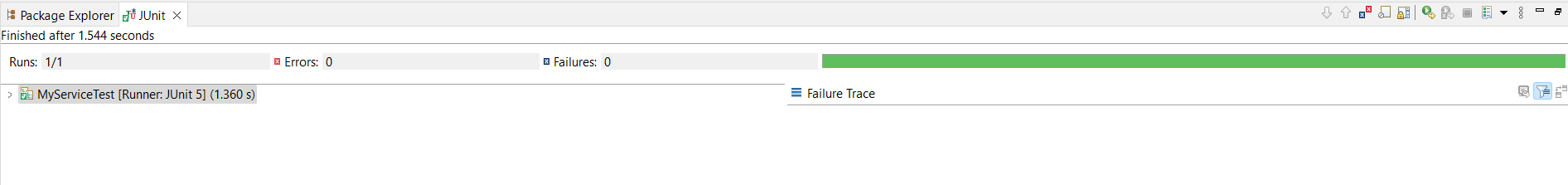
verify(mockApi).getData();

}

}

**Step5:** Run the Test

1. Open MyServiceTest.java.
2. Right-click inside the file → Choose **Run As → JUnit Test**.

**Output:**