**Exercise 1: Setting Up JUnit**

**Class Calculator.java:**

package org.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

public int division(int a, int b) {

return a / b;

}

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**Class CalculatorTest.java:**

public class CalculatorTest {

@Test

public void testAdd() {

Calculator calc = new Calculator();

int result = calc.add(10, 5);

assertEquals(15, result);

}

@Test

public void testSubtract() {

Calculator calc = new Calculator();

int result = calc.subtract(10, 5);

assertEquals(5, result);

}

@Test

public void testMultiply() {

Calculator calc = new Calculator();

int result = calc.multiply(4, 3);

assertEquals(12, result);

}

@Test

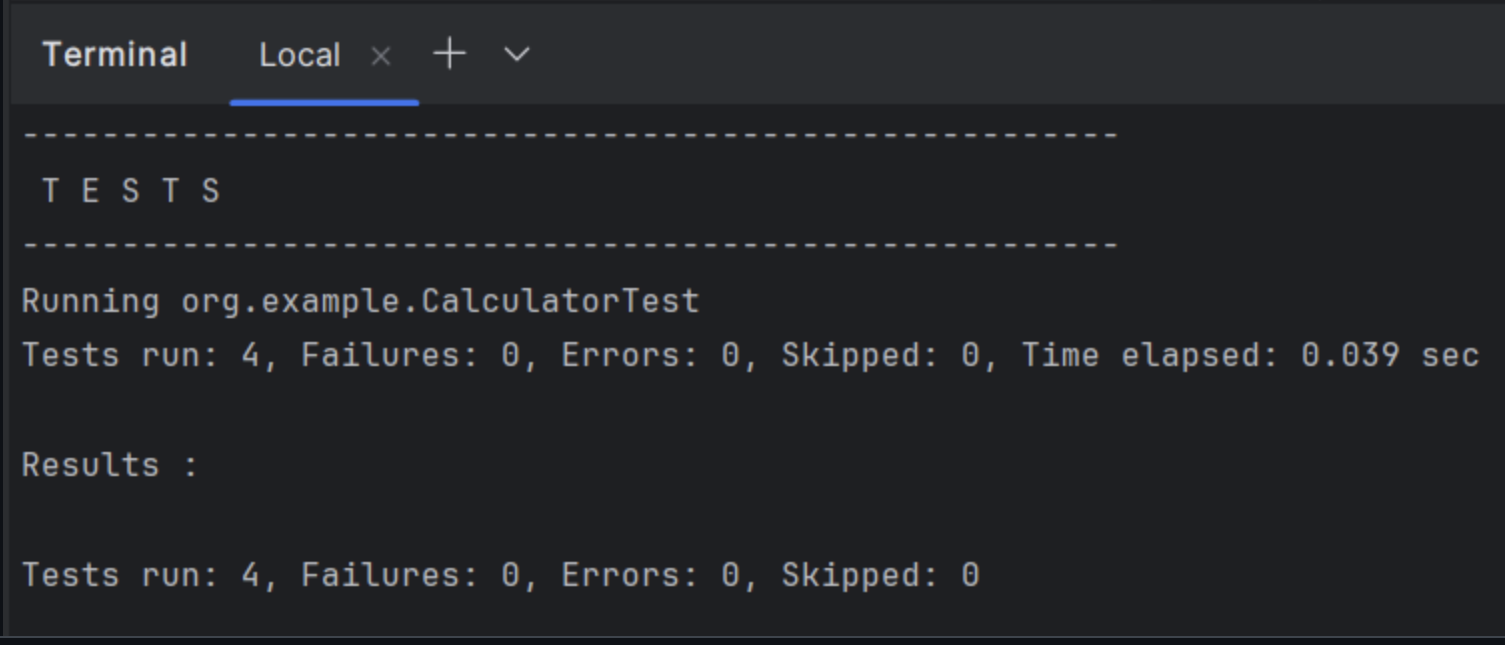
public void testDivision(){

Calculator calc = new Calculator();

int result = calc.division(4,2);

assertEquals(2, result);

}}

**Output**

**Exercise 1: Mocking and Stubbing**

**ExternalApi.java:**

package org.example;

public interface ExternalApi {

String getData();

}

**Main.java:**

package org.example;

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**MyService.java:**

package org.example;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**MyServiceTest.java:**

package org.example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.mock;

import static org.mockito.Mockito.when;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Step 1: Create mock

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Stub the method

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

// Step 3: Inject mock into service

MyService service = new MyService(mockApi);

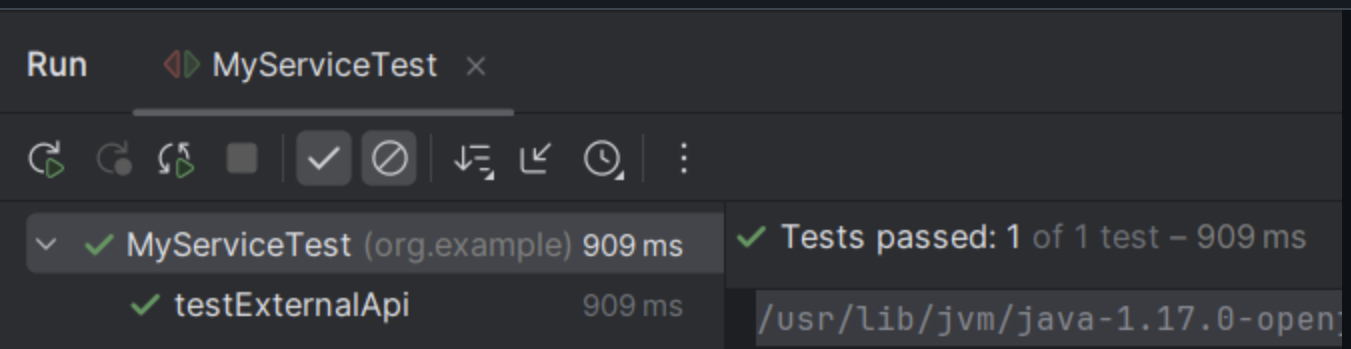
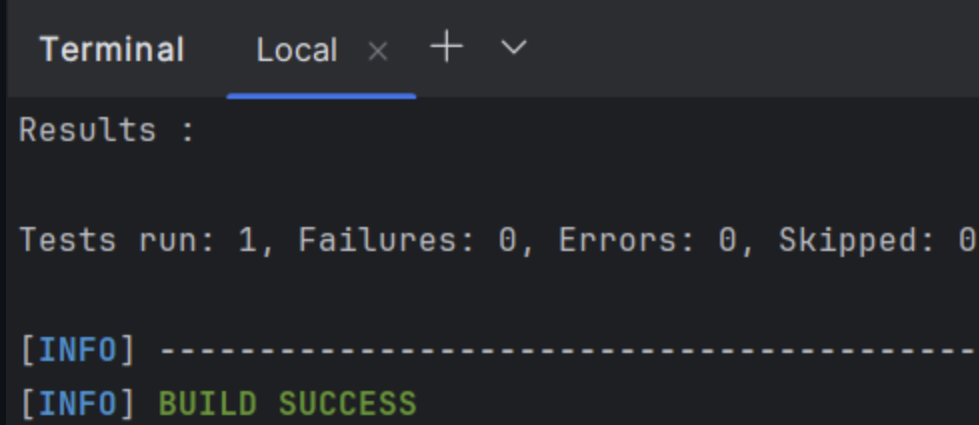
// Call and assert

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**Output:**

**Exercise 2: Verifying Interactions**

**ExternalApi.java:**

package org.example;

public interface ExternalApi {

String getData(String source);

}

**Main.java:**

package org.example;

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**MyService.java:**

package org.example;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData("database");

}

}

**MyServiceTest.java:**

package org.example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: created a mock object

ExternalApi mockApi = mock(ExternalApi.class);

//Step 2: Call method on service

MyService service = new MyService(mockApi);

String result= service.fetchData();

//verify the getData() was called on the mock

verify(mockApi).getData(“database");

}

}

**Exercise 3: Assertions in JUnit**

**AssertionTest.java:**

package org.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());

// Assert same reference

Object ref = new Object();

assertSame(ref, ref);

// Assert not same reference

assertNotSame(new Object(), new Object());

}

}

**Main.java:**

package org.example;

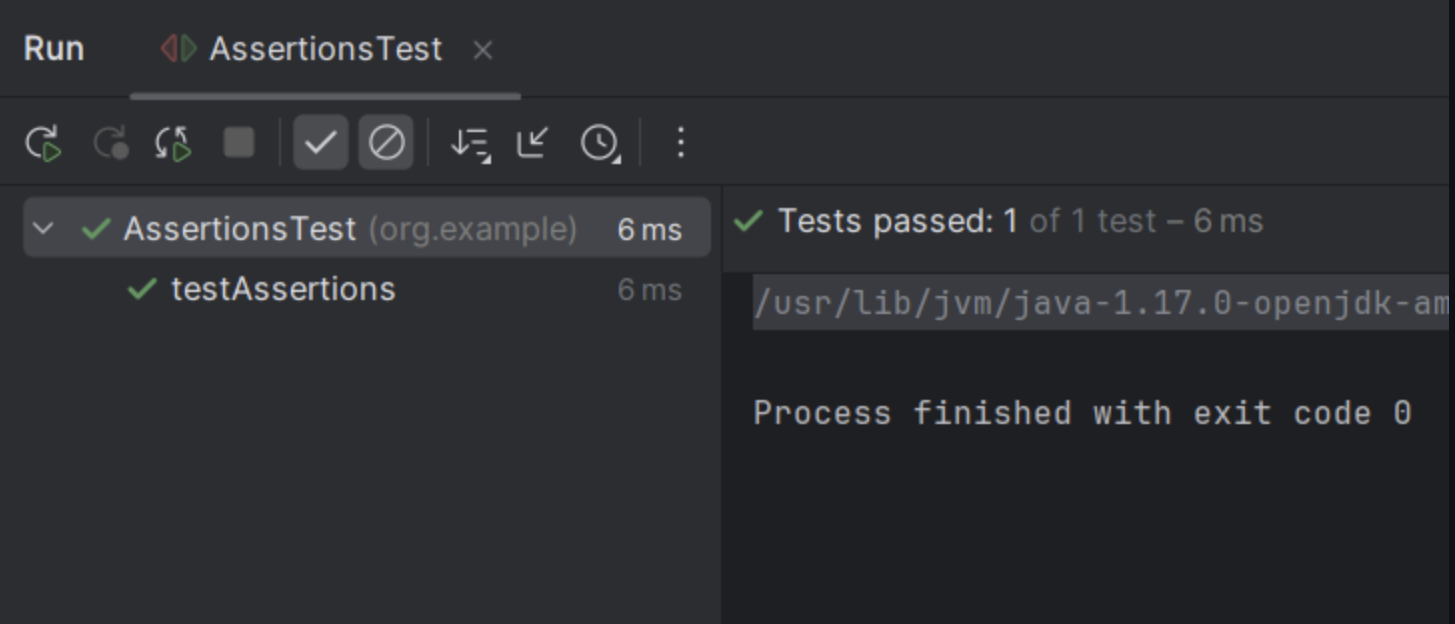
public class Main {

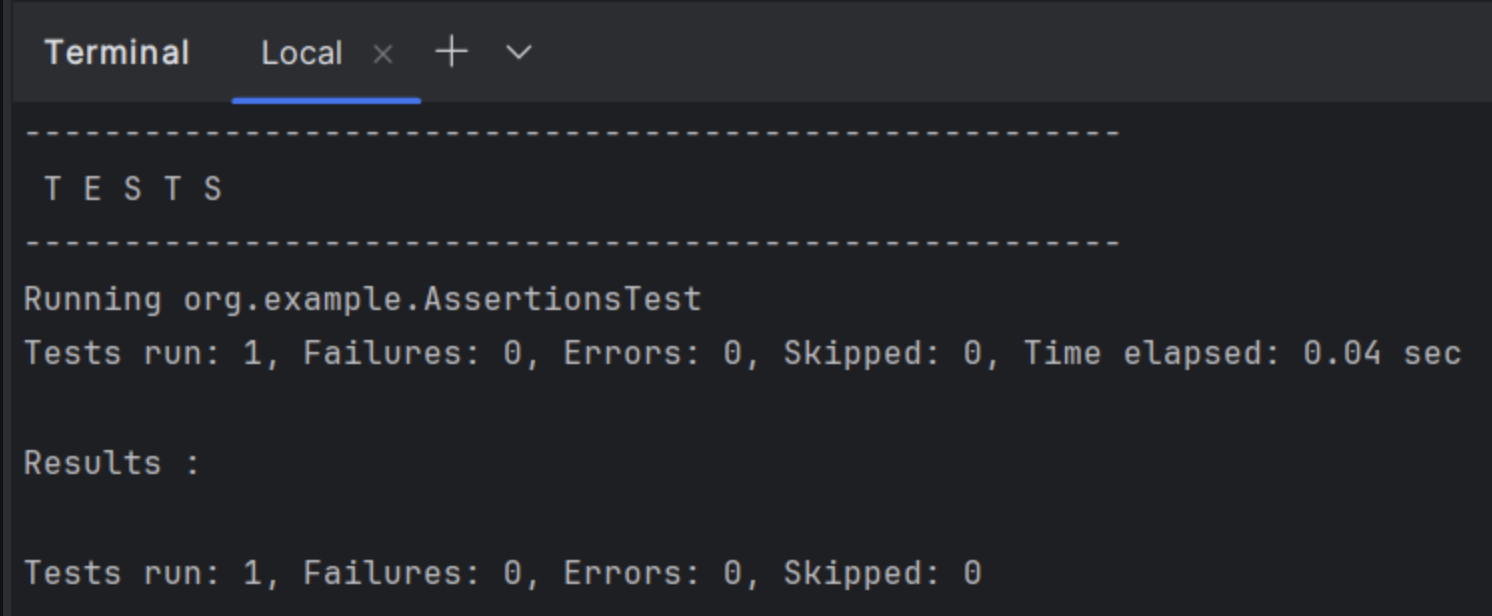
public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**Output:**



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

**Calculator.java:**

package org.example;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int multiply(int a, int b) {

return a \* b;

}

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**CalculatorTest.java:**

package org.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

//setup is done

calculator = new Calculator();

System.out.println("Setup: New Calculator created");

}

@After

public void tearDown() {

//teardown is done

calculator = null;

System.out.println("Teardown: Calculator object is equals null");

}

@Test

public void testAddition() {

// Arrange

int a = 10;

int b = 5;

// Act

int result = calculator.add(a, b);

// Assert

assertEquals(15, result);

}

@Test

public void testMultiplication() {

// Arrange

int a = 4;

int b = 3;

// Act

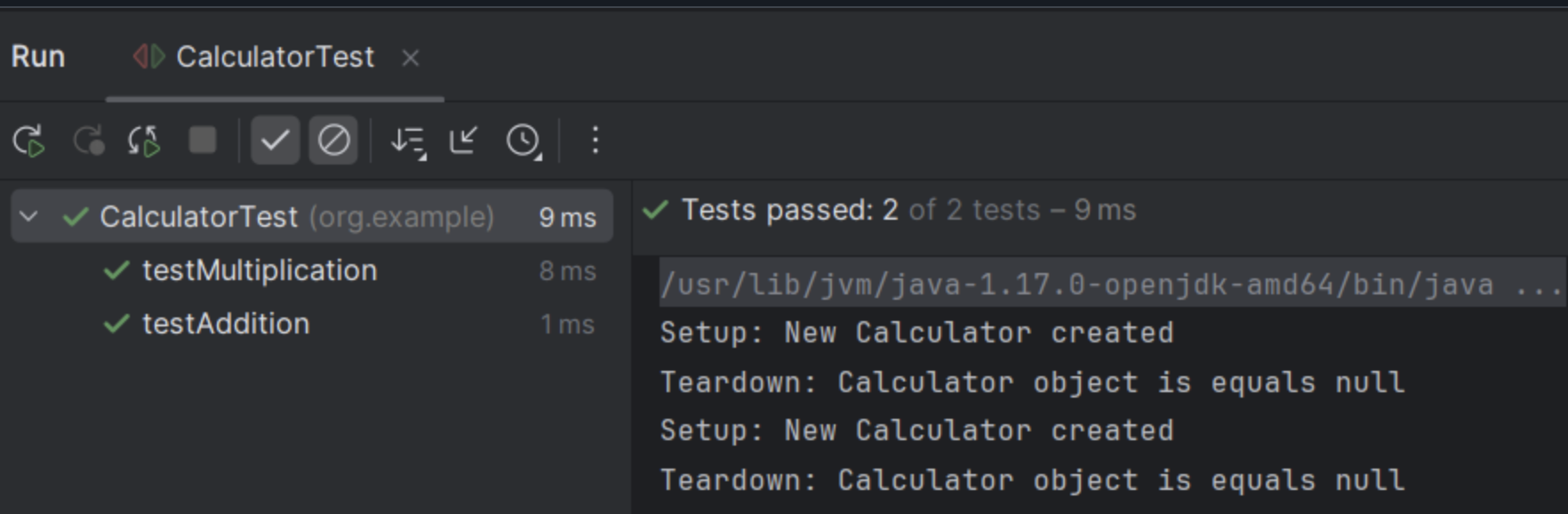
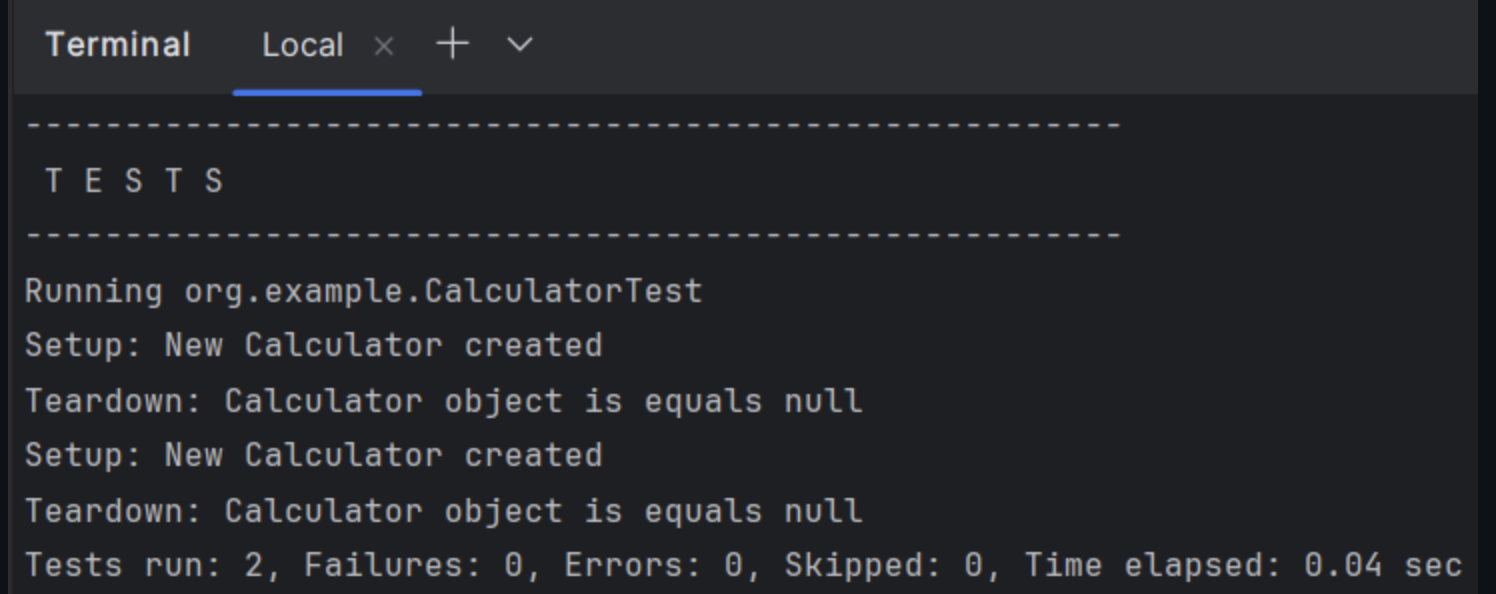
int result = calculator.multiply(a, b);

// Assert

assertEquals(12, result);

}

}



**Output:**