**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

**Scenario:** You need to set up JUnit in your Java project to start writing unit tests.

**CODE:**

import org.junit.Test;

import static org.junit.Assert.\*;

public class Main {

public static void main(String[] args) {

org.junit.runner.JUnitCore.main("Main$CalculatorTest");

}

public static class Calculator {

public int add(int a, int b) {

return a + b;

}

}

public static class CalculatorTest {

@Test

public void testAdd() {

Calculator c = new Calculator();

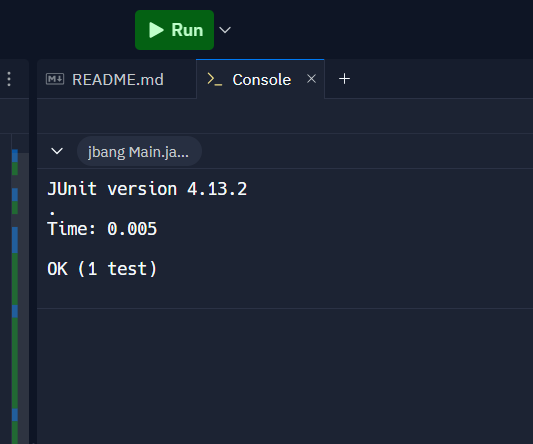
assertEquals(5, c.add(2, 3));

}

}

}

**OUTPUT:**

****

**Exercise 3: Assertions in Junit**

**Scenario:** You need to use different assertions in JUnit to validate your test results.

**CODE:**

///usr/bin/env jbang "$0" "$@" ; exit $?

//DEPS junit:junit:4.13.2

import org.junit.Test;

import static org.junit.Assert.\*;

public class Main {

public static void main(String[] args) {

org.junit.runner.JUnitCore.main("Main$AssertionsTest");

}

public static 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

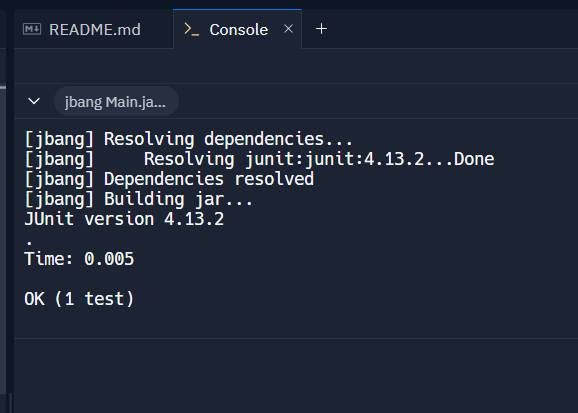
assertNotNull(new Object());

}

}

}

**OUTPUT:**

****

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

**Scenario:** You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

**CODE**: ///usr/bin/env jbang "$0" "$@" ; exit $?

//DEPS junit:junit:4.13.2

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Main {

public static void main(String[] args) {

org.junit.runner.JUnitCore.main("Main$CalculatorTest");

}

// Class under test

public static class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

// JUnit Test Class

public static class CalculatorTest {

private Calculator calc;

@Before

public void setUp() {

System.out.println("Before each test: Calculator initialized");

calc = new Calculator();

}

@After

public void tearDown() {

System.out.println("After each test: Calculator cleaned up");

calc = null;

}

@Test

public void testAdd() {

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

assertEquals(5, result);

}

@Test

public void testSubtract() {

int result = calc.subtract(10, 4);

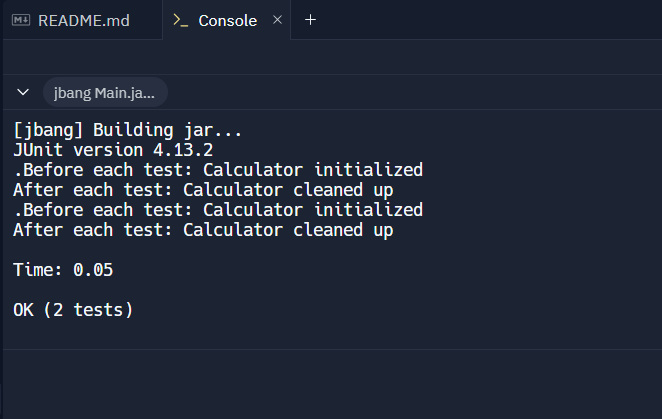
assertEquals(6, result);

}

}

}

**OUTPUT:**

****