**JUnit Testing Exercises**

**Exercise 1: Setting Up JUnit**

**junit1.java :**

package junit;

public class junit1 {

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

return a+b;

}

}

**test1.java :**

package junit;

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

import org.junit.jupiter.api.Test;

class test1 {

@Test

void test() {

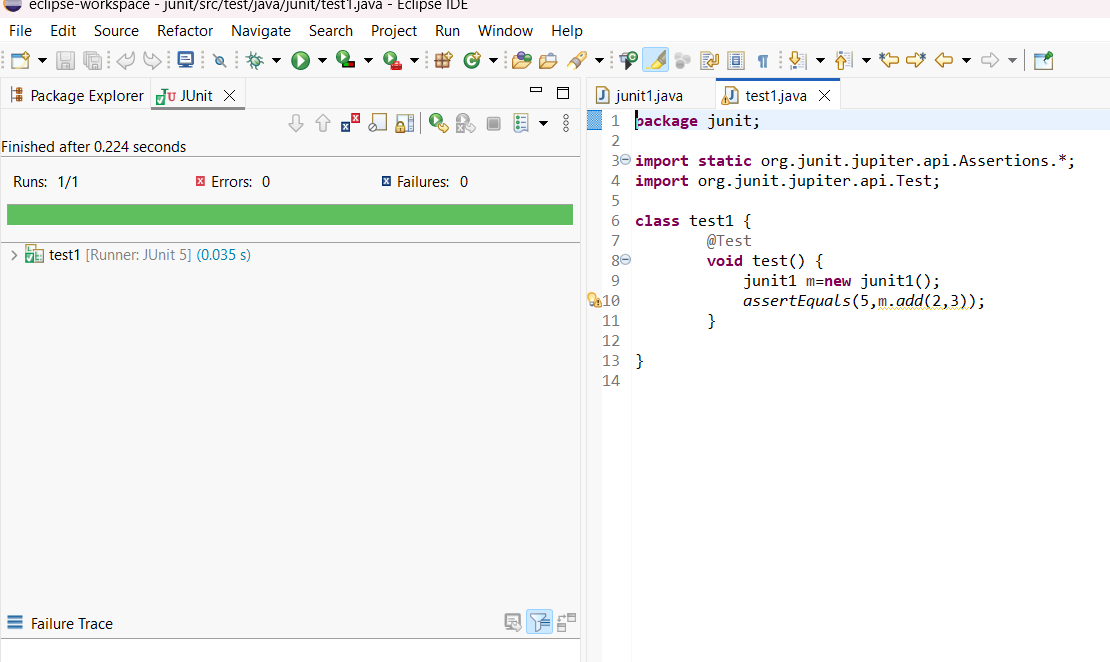
junit1 a=new junit1();

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

}

}

**Output :**

****

**Exercise 3: Assertions in Junit**

**Main.java:**

package assert1;

public class Main {

public int sum(int a, int b) {

return a + b;

}

public boolean isGreater(int a, int b) {

return a > b;

}

public boolean isLess(int a, int b) {

return a < b;

}

public Object returnNull() {

return null;

}

public Object returnNonNull() {

return new Object();

}

}

**junit1.java :**

package assert1;

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

import org.junit.jupiter.api.Test;

class junit1 {

@Test

public void testAssertions() {

// Assert equals

*assertEquals*(5, 2 + 3);

// Assert true

*assertTrue*(5 > 3);

// Assert false

*assertFalse*(5 < 3);

// Assert null

*assertNull*(null);

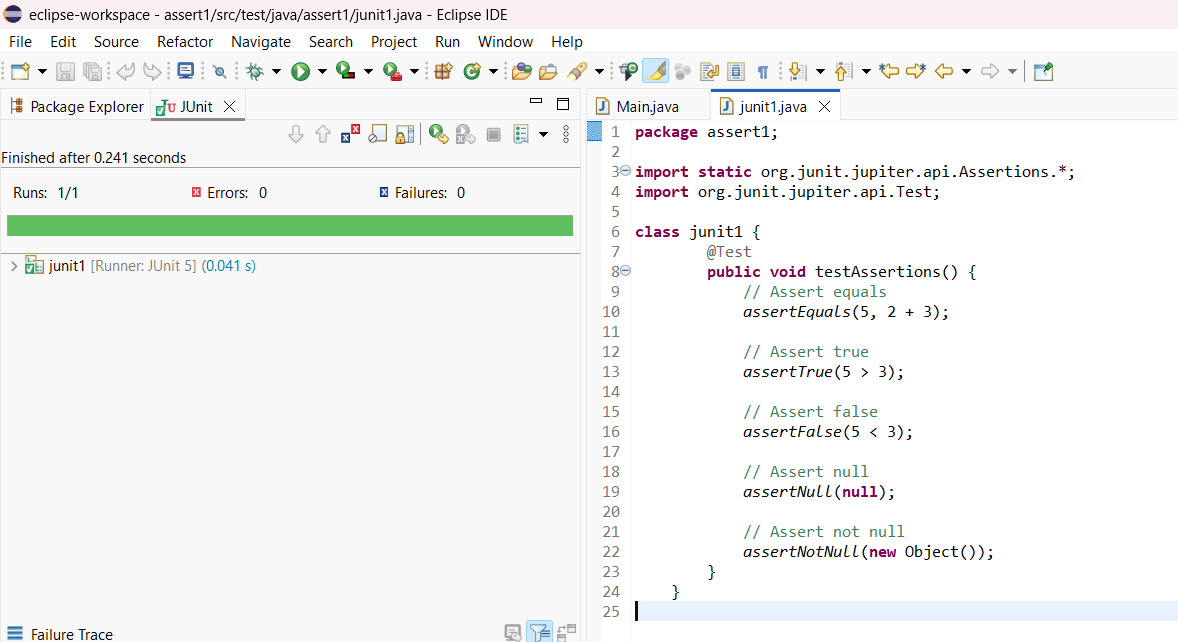
// Assert not null

*assertNotNull*(new Object());

}

}

**Output:**

****

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

**Cal.java:**

package unit1;

public class Cal {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**CalTest.java :**

package unit1;

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

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

public class CalTest {

private Cal cal;

@BeforeEach

void setUp() {

cal = new Cal();

System.*out*.println("Setup before test");

}

@AfterEach

void tearDown() {

cal = null;

System.*out*.println("Teardown after test");

}

@Test

void testAdd() {

int result = cal.add(5, 7);

*assertEquals*(12, result);

}

@Test

void testSub() {

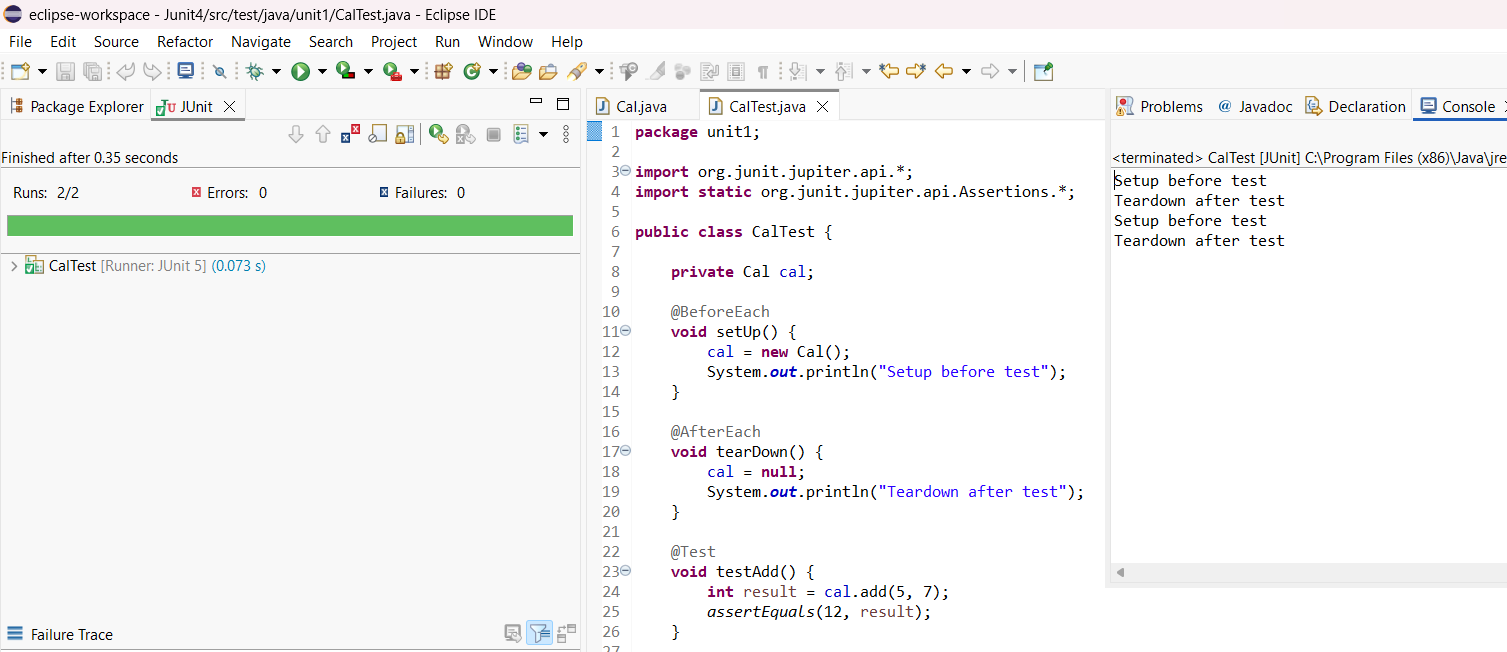
int result = cal.subtract(10, 3);

*assertEquals*(7, result);

}

}

**Output :**

****