JUnit Testing Exercises

Exercise 1: Setting Up JUnit

CalculatorTest.java. import org.junit.Test;

import static org.junit.Assert.\*; public class CalculatorTest {

@Test

public void sampleTest() { int expected = 5;

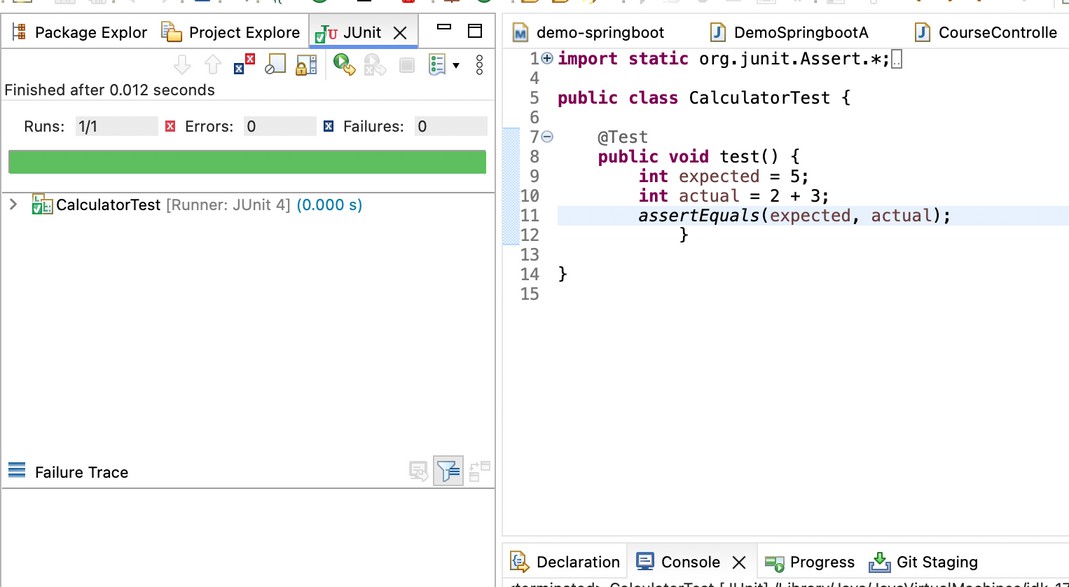
int actual = 2 + 3;

assertEquals(expected, actual);

}

}

Output



Exercise 2: Writing Basic JUnit Tests

Calculator.java

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 divide(int a, int b) { if (b == 0) {

throw new IllegalArgumentException("Cannot divide by zero");

}

return a / b;

}

}

## CalclulatorTest.java

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 divide(int a, int b) { if (b == 0) {

throw new IllegalArgumentException("Cannot divide by zero");

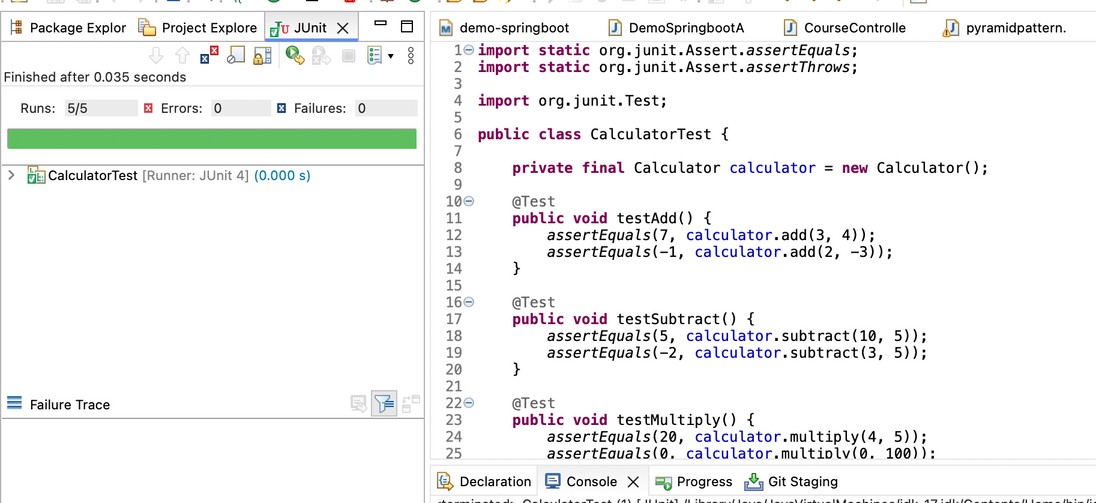
}

return a / b;

}

}

# output



Exercise 3: Assertions in JUnit

## AssertionsTest.java

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

public class AssertionsTest { @Test

public void testAssertions() {

// Assert equals assertEquals(5, 2 + 3);

// Assert true assertTrue(5 > 3);

// Assert false assertFalse(5 < 3);

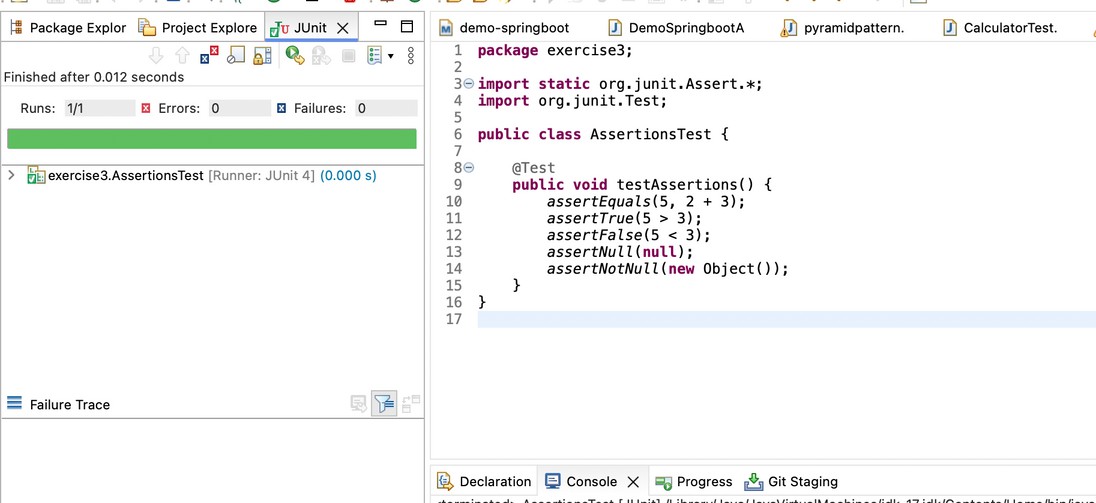
// Assert null assertNull(null);

// Assert not null assertNotNull(new Object());

}

}

# output



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

## CalculatorAAATest.java

import static org.junit.Assert.assertEquals; import org.junit.After;

import org.junit.Before; import org.junit.Test;

public class CalculatorAAATest { private Calculator calculator;

// Setup method runs before each test @Before

public void setUp() {

// Arrange: initialize Calculator object before each test calculator = new Calculator(); System.out.println("Setup complete.");

}

// Teardown method runs after each test @After

public void tearDown() {

// Cleanup if needed (not mandatory here) calculator = null; System.out.println("Teardown complete.");

}

@Test

public void testAdd() {

// Arrange int a = 5; int b = 3;

// Act

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

// Assert assertEquals(8, result);

}

@Test

public void testSubtract() {

// Arrange int a = 10; int b = 7;

// Act

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

// Assert assertEquals(3, result);

}

}

output

