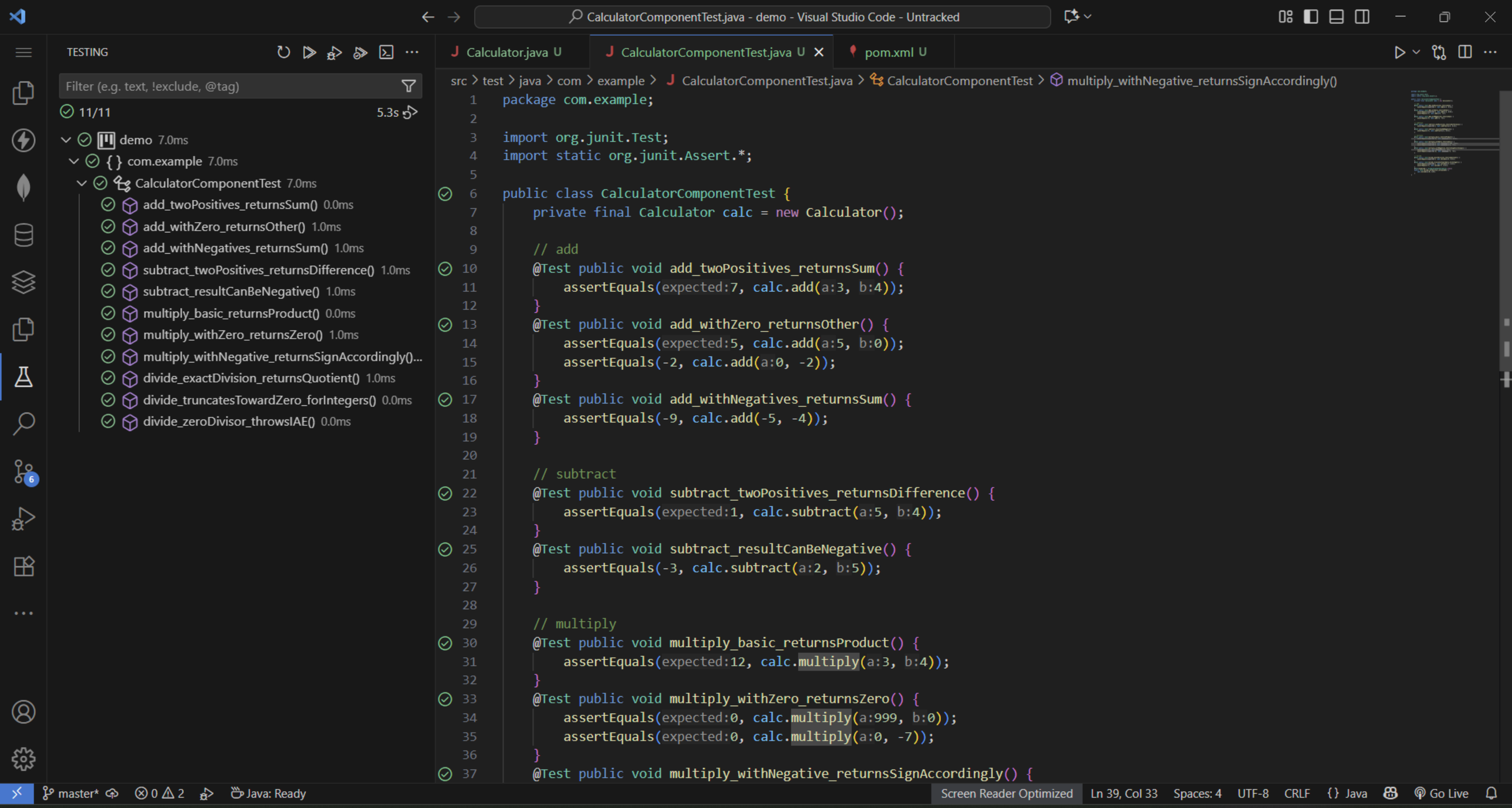
CalculatorComponentTest.java - demo - Visual Studio Code - Untracked
TESTING
Filter (e.g. text, !exclude, @tag)
11/11
5.3s
demo 7.0ms
com.example 7.0ms
CalculatorComponentTest 7.0ms
add_twoPositives_returnsSum() 0.0ms
add_withZero_returnsOther() 1.0ms
add_withNegatives_returnsSum() 1.0ms
subtract_twoPositives_returnsDifference() 1.0ms
subtract_resultCanBeNegative() 1.0ms
multiply_basic_returnsProduct() 0.0ms
multiply_withZero_returnsZero() 1.0ms
multiply_withNegative_returnsSignAccordingly()...
divide_exactDivision_returnsQuotient() 1.0ms
divide_truncatesTowardZero_forIntegers() 0.0ms
divide_zeroDivisor_throwsIAE() 0.0ms
Calculator.java U
CalculatorComponentTest.java U
pom.xml U
src > test > java > com > example > CalculatorComponentTest.java > CalculatorComponentTest > multiply_withNegative_returnsSignAccordingly()
1 package com.example;
2
3 import org.junit.Test;
4 import static org.junit.Assert.*;
5
6 public class CalculatorComponentTest {
7 private final Calculator calc = new Calculator();
8
9 // add
10 @Test public void add_twoPositives_returnsSum() {
11 assertEquals(expected:7, calc.add(a:3, b:4));
12 }
13 @Test public void add_withZero_returnsOther() {
14 assertEquals(expected:5, calc.add(a:5, b:0));
15 assertEquals(-2, calc.add(a:0, -2));
16 }
17 @Test public void add_withNegatives_returnsSum() {
18 assertEquals(-9, calc.add(-5, -4));
19 }
20
21 // subtract
22 @Test public void subtract_twoPositives_returnsDifference() {
23 assertEquals(expected:1, calc.subtract(a:5, b:4));
24 }
25 @Test public void subtract_resultCanBeNegative() {
26 assertEquals(-3, calc.subtract(a:2, b:5));
27 }
28
29 // multiply
30 @Test public void multiply_basic_returnsProduct() {
31 assertEquals(expected:12, calc.multiply(a:3, b:4));
32 }
33 @Test public void multiply_withZero_returnsZero() {
34 assertEquals(expected:0, calc.multiply(a:999, b:0));
35 assertEquals(expected:0, calc.multiply(a:0, -7));
36 }
37 @Test public void multiply_withNegative_returnsSignAccordingly() {
master*
0 2
Java: Ready
Screen Reader Optimized
Ln 39, Col 33
Spaces: 4
UTF-8
CRLF
Java
Go Live