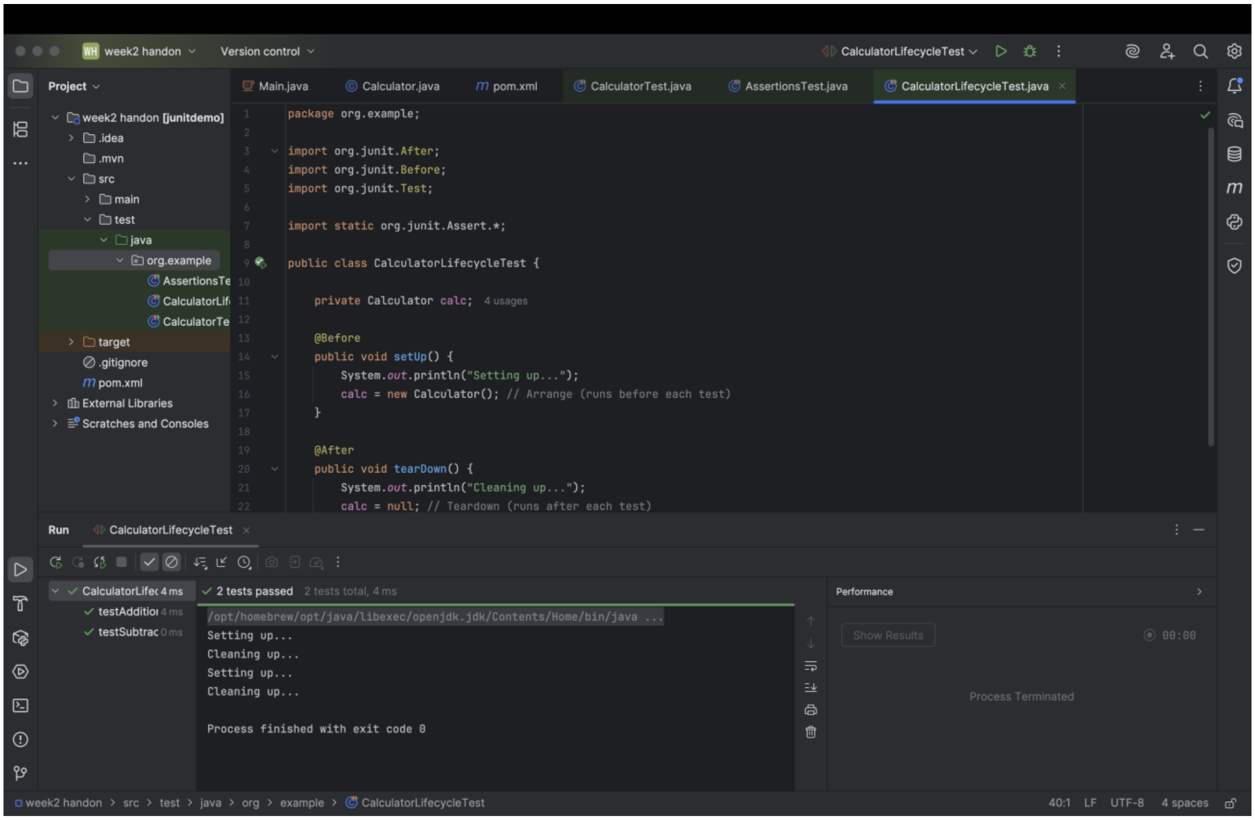
Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit

```
package org.example;

 import org.junit.After;

import org.junit.Before;

import org.junit.Test;

 import static org.junit.Assert.*;

 public class CalculatorLifecycleTest {

     private Calculator calc;

    @Before

  public void setUp() {

        System.out.println("Setting up...");

        calc = new Calculator(); // Arrange (runs before each test)

    }

     @After

    public void tearDown() {

        System.out.println("Cleaning up...");

        calc = null; // Teardown (runs after each test)

   }

     @Test

    public void testAddition() {

        // Act

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

        // Assert

        assertEquals(5, result);
   }
   @Test
public void testSubtraction() {
        int result = calc.subtract(10, 4);
        assertEquals(6, result);
   }
}
```

Output:



```java
package org.example;

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

import static org.junit.Assert.*;

public class CalculatorLifecycleTest {

    private Calculator calc;  4 usages

    @Before
    public void setUp() {
        System.out.println("Setting up...");
        calc = new Calculator(); // Arrange (runs before each test)
    }

    @After
    public void tearDown() {
        System.out.println("Cleaning up...");
        calc = null; // Teardown (runs after each test)
```

✓ 2 tests passed 2 tests total, 4 ms

```
/opt/homebrew/opt/java/libexec/openjdk.jdk/Contents/Home/bin/java ...
Setting up...
Cleaning up...
Setting up...
Cleaning up...

Process finished with exit code 0
```

SUMMARY:

- Used Arrange-Act-Assert (AAA) pattern
- Used @Before for setup
- Used @After for cleanup
- Ensured each test runs in an isolated environment