TP08

Salma Bellaou

November 2025

1 exercise 01

```java
public class Calculator {  no usages

    /**
     * Returns the sum of two integers.
     */
    public int add(int a, int b) {
        return a + b;
    }
}
```

Figure 1: Calculator class

```java
import static org.junit.jupiter.api.Assertions.*;
import org . junit . jupiter . api . Test ;


public class CalculatorTest {

    @Test
    void addTwoPositiveNumbersShouldReturnSum() {
        // Arrange
        Calculator calc = new Calculator();

        // Act
        int result = calc.add( a: 2,  b: 3);

        // Assert
        assertEquals( expected: 5, result,  message: "2 + 3 should equal 5");
    }
}
```

Figure 2: test class

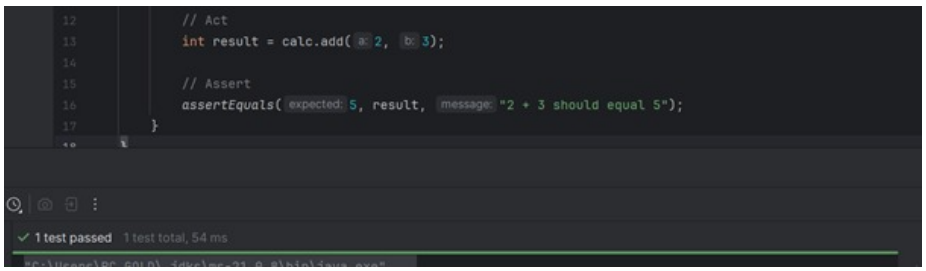

Figure 3: Test passed

```java
public class Calculator {  2 usages

    /**
     * Returns the sum of two integers.
     */
    public int add(int a, int b) {
        return a - b;
    }
}
```

Figure 4: replacing the addition with the subtraction

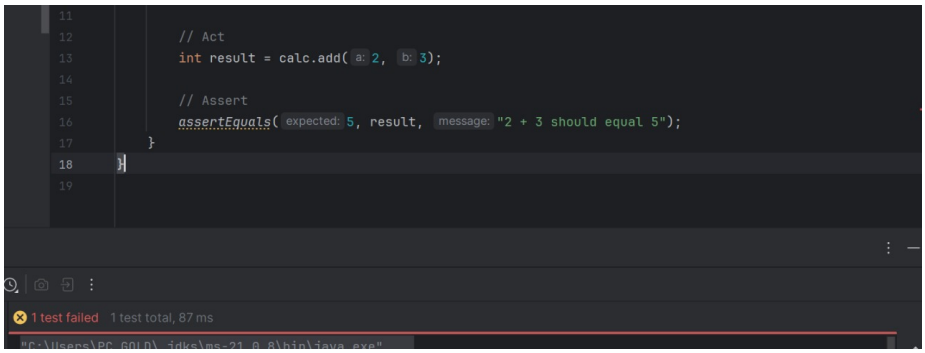

Figure 5: Test failed

```
public class Calculator {  2 usages

    /**
     * Returns the sum of two integers.
     */
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a,int b){  no usages
        return a - b;
    }
    public int multiply(int a, int b){  no usages
        return a * b;
    }
    public float divide(int a, int b){  no usages
        return a / b;
    }
}
```

Figure 6: adding new methods to Calculator

```
    void substracttwoPositiveNumbersShouldReturnSubstraction() {  no usages
        // Arrange
        Calculator calc = new Calculator();

        // Act
        int result = calc.subtract( a: 2, b: 3);

        // Assert
        assertEquals( expected: -1, result, message: "2 - 3 should equal -1");
    }
```

Figure 7: adding subtract tester

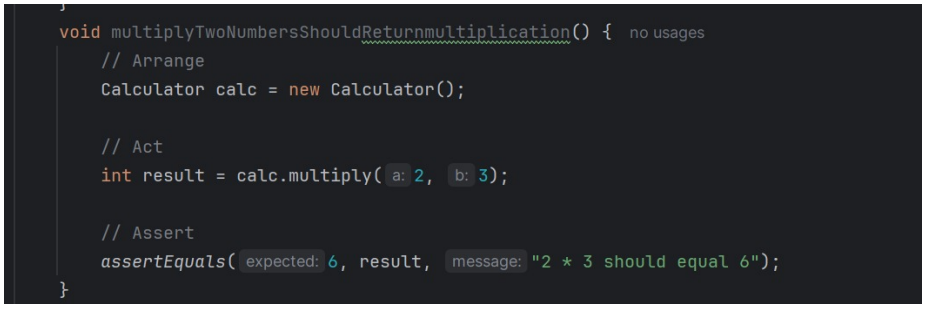

Figure 8: adding multiply tester

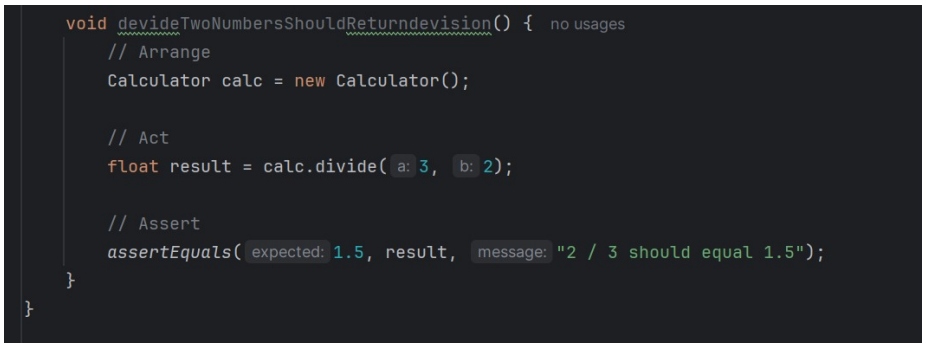

Figure 9: Adding divide tester

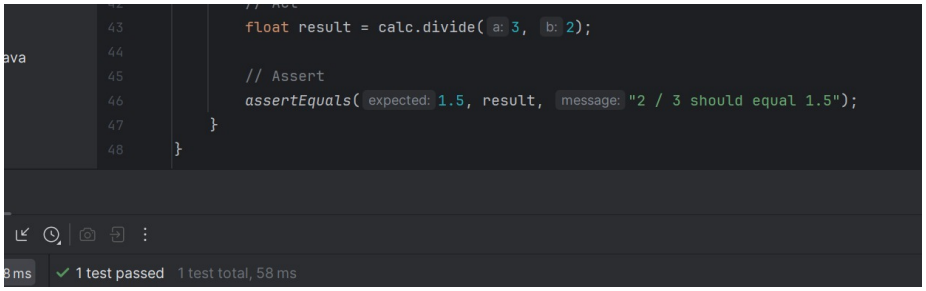

Figure 10: Test passed

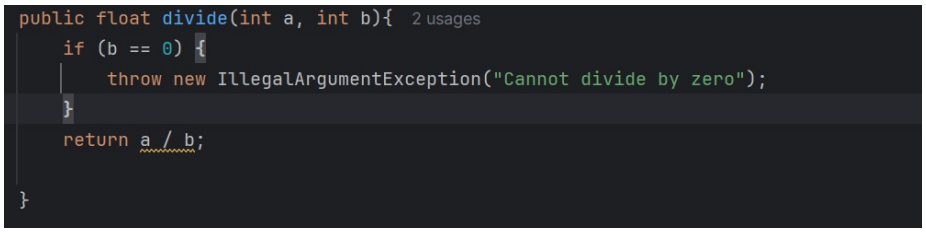

Figure 11: Editing the divide method

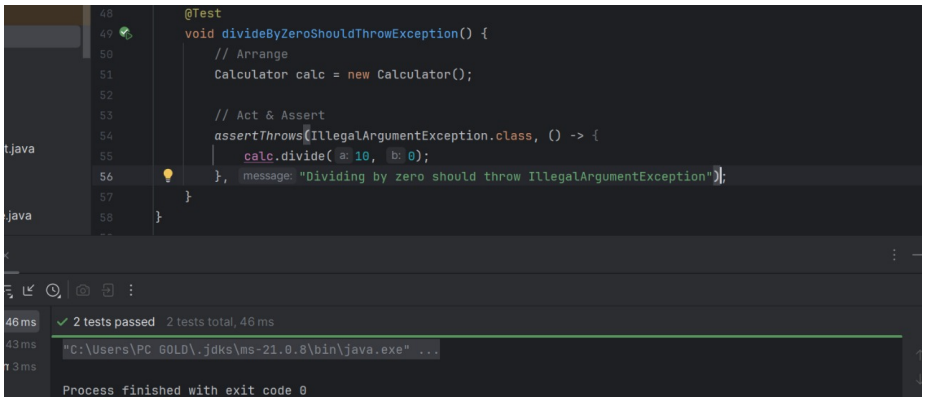

Figure 12: Editing CalculatorTest and we find that the test has passed

2 Exercise 02

Task 1: Boundary Value Analysis Test Cases

The regulator has a tolerance zone of $\pm 0.5°C$ around the target temperature. The following test cases are derived using Boundary Value Analysis (BVA):

1. **Test 1:** Current temperature = target - 0.51°C → Expected action: HEAT
2. **Test 2:** Current temperature = target - 0.50°C → Expected action: STANDBY
3. **Test 3:** Current temperature = target - 0.49°C → Expected action: STANDBY
4. **Test 4:** Current temperature = target → Expected action: STANDBY
5. **Test 5:** Current temperature = target + 0.49°C → Expected action: STANDBY
6. **Test 6:** Current temperature = target + 0.50°C → Expected action: STANDBY
7. **Test 7:** Current temperature = target + 0.51°C → Expected action: COOL