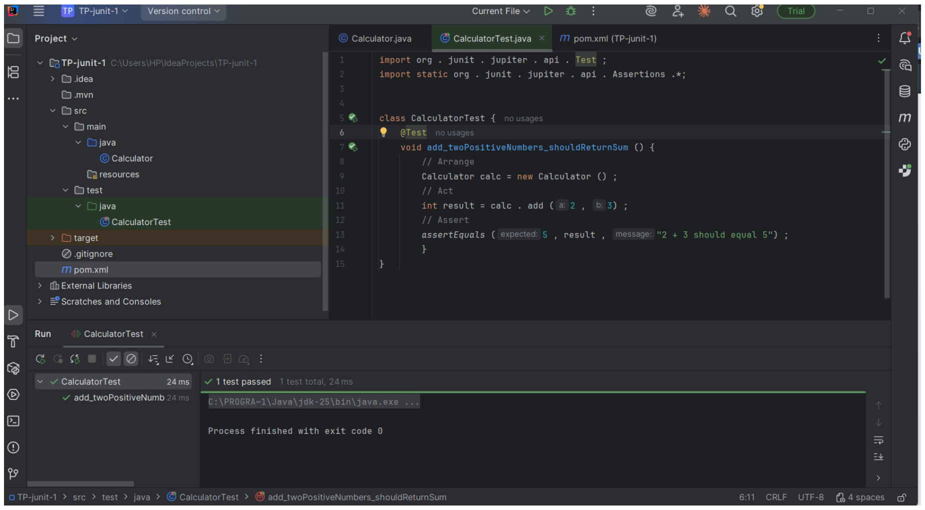
Testing

El Gourji Nasreddine

November 2025

1 EX1

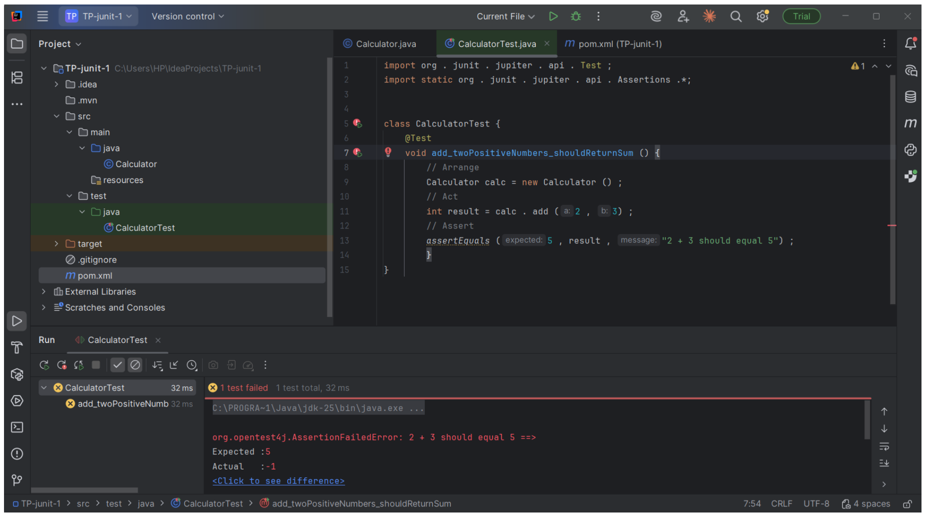
1.1 Success Test :



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


class CalculatorTest {  no usages
    @Test  no usages
    void add_twoPositiveNumbers_shouldReturnSum () {
        // 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") ;
        }
}
```

```
CalculatorTest 24 ms
  add_twoPositiveNumb 24 ms

1 test passed  1 test total, 24 ms
C:\PROGRA~1\Java\jdk-25\bin\java.exe ...

Process finished with exit code 0
```

1.2 Failed Test after :



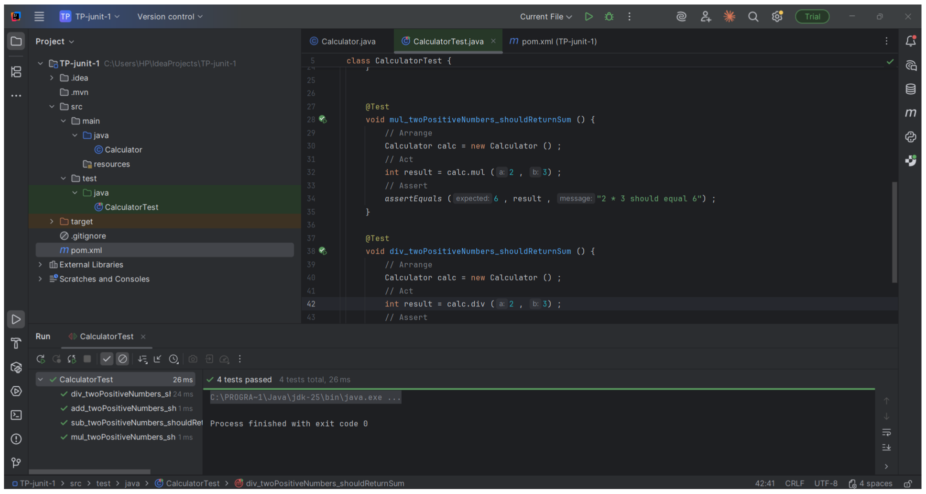
```
import org . junit . jupiter . api . Test ;
import static org . junit . jupiter . api . Assertions .*;


class CalculatorTest {
    @Test
    void add_twoPositiveNumbers_shouldReturnSum () {
        // 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") ;
        }
}
```

```
CalculatorTest 32 ms
    add_twoPositiveNumb 32 ms
```

1 test failed 1 test total, 32 ms

```
C:\PROGRA~1\Java\jdk-25\bin\java.exe ...

org.opentest4j.AssertionFailedError: 2 + 3 should equal 5 ==>
Expected :5
Actual   :-1
<Click to see difference>
```

1.3 Success Test after adding the sub and mul features :



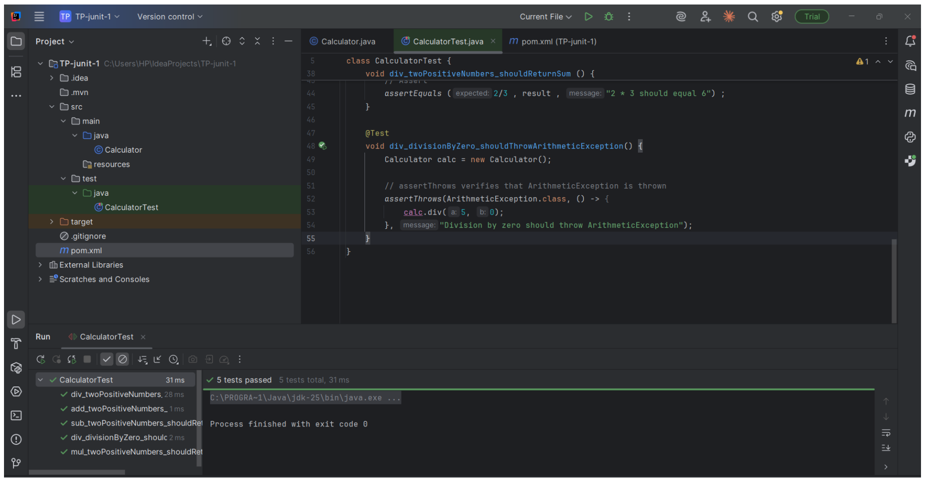
```
class CalculatorTest {


    @Test
    void mul_twoPositiveNumbers_shouldReturnSum () {
        // Arrange
        Calculator calc = new Calculator () ;
        // Act
        int result = calc.mul ( a: 2 ,  b: 3) ;
        // Assert
        assertEquals ( expected: 6 , result , message: "2 * 3 should equal 6") ;
    }

    @Test
    void div_twoPositiveNumbers_shouldReturnSum () {
        // Arrange
        Calculator calc = new Calculator () ;
        // Act
        int result = calc.div ( a: 2 ,  b: 3) ;
        // Assert
```

```
CalculatorTest 26 ms
    div_twoPositiveNumbers_sl 24 ms
    add_twoPositiveNumbers_sh 1 ms
    sub_twoPositiveNumbers_shouldRet
    mul_twoPositiveNumbers_sh 1 ms
```

4 tests passed 4 tests total, 26 ms

```
C:\PROGRA~1\Java\jdk-25\bin\java.exe ...

Process finished with exit code 0
```

1.4 Success Test for handling the zero exception:



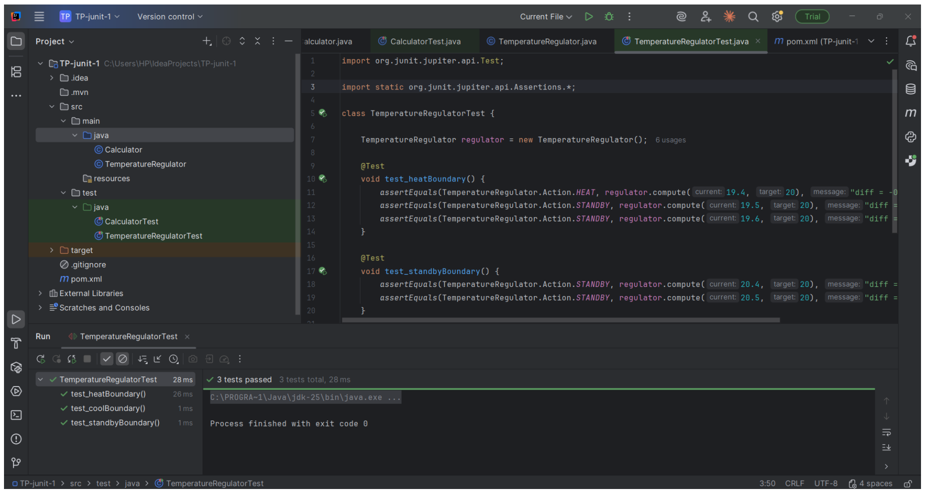
```java
class CalculatorTest {
    void div_twoPositiveNumbers_shouldReturnSum () {
        assertEquals ( expected: 2/3 , result , message: "2 * 3 should equal 6") ;
    }

    @Test
    void div_divisionByZero_shouldThrowArithmeticException() {
        Calculator calc = new Calculator();

        // assertThrows verifies that ArithmeticException is thrown
        assertThrows(ArithmeticException.class, () -> {
            calc.div( a: 5,  b: 0);
        }, message: "Division by zero should throw ArithmeticException");
    }
}
```

```
✓ CalculatorTest 31 ms
  ✓ div_twoPositiveNumbers, 28 ms
  ✓ add_twoPositiveNumbers_ 1 ms
  ✓ sub_twoPositiveNumbers_shouldRet
  ✓ div_divisionByZero_shoulc 2 ms
  ✓ mul_twoPositiveNumbers_shouldRet
```

```
✓ 5 tests passed 5 tests total, 31 ms
C:\PROGRA~1\Java\jdk-25\bin\java.exe ...

Process finished with exit code 0
```

2 EX1

2.1 Success Test :



```java
import org.junit.jupiter.api.Test;

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

class TemperatureRegulatorTest {

    TemperatureRegulator regulator = new TemperatureRegulator();  6 usages

    @Test
    void test_heatBoundary() {
        assertEquals(TemperatureRegulator.Action.HEAT, regulator.compute( current: 19.4,  target: 20),  message: "diff = -0
        assertEquals(TemperatureRegulator.Action.STANDBY, regulator.compute( current: 19.5,  target: 20),  message: "diff =
        assertEquals(TemperatureRegulator.Action.STANDBY, regulator.compute( current: 19.6,  target: 20),  message: "diff =
    }

    @Test
    void test_standbyBoundary() {
        assertEquals(TemperatureRegulator.Action.STANDBY, regulator.compute( current: 20.4,  target: 20),  message: "diff =
        assertEquals(TemperatureRegulator.Action.STANDBY, regulator.compute( current: 20.5,  target: 20),  message: "diff =
    }
```

```
✓ TemperatureRegulatorTest 28 ms
  ✓ test_heatBoundary() 26 ms
  ✓ test_coolBoundary() 1 ms
  ✓ test_standbyBoundary() 1 ms
```

```
✓ 3 tests passed 3 tests total, 28 ms
C:\PROGRA~1\Java\jdk-25\bin\java.exe ...

Process finished with exit code 0
```