JUnit Testing

JUnit is a popular testing framework for Java used to write and run unit tests.
versions: JUnit 4 and JUnit 5 (JUnit Jupiter)

JUnit Architecture

Test Class: Contains test methods
Test Method: Annotated with @Test
Assertions: Methods to check expected results (assertEquals, assertTrue, etc.)

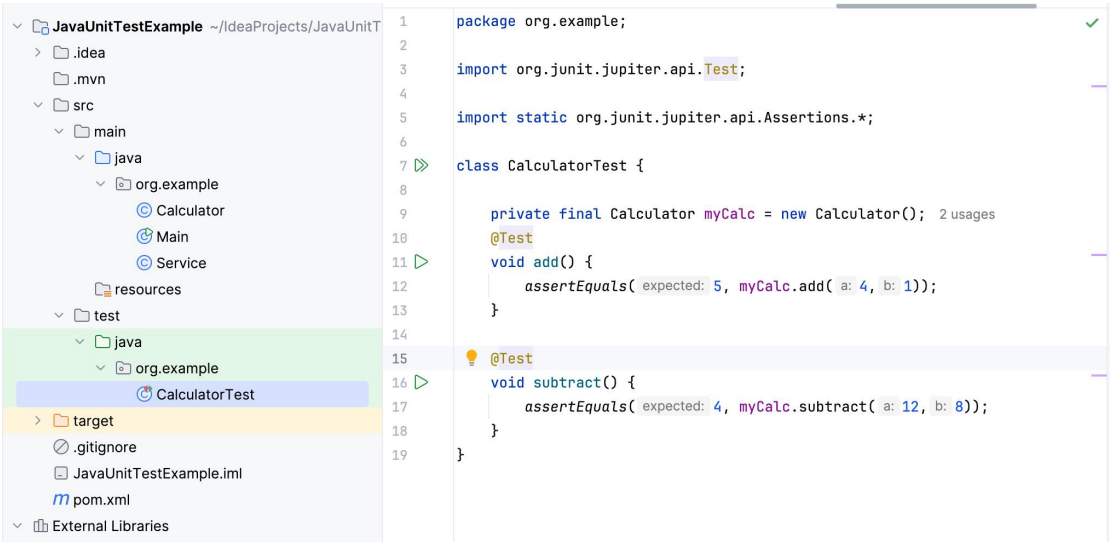
Annotations:

`@Test` – marks a test method

`@BeforeEach` / `@Before` – runs before each test

`@AfterEach` / `@After` – runs after each test

`@BeforeAll` / `@AfterAll` – runs once before/after all tests



Example of a Test Class Structure:

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
class CalculatorTest {
    @Test
    void testAddition() {
        Calculator calc = new Calculator();
        assertEquals(5, calc.add(2, 3));
    }
}
```

Common Assertions in JUnit

Assertion	Description	Example
`assertEquals(expected, actual)`	Checks if two values are equal	`assertEquals(5, calc.add(2,3));`
`assertTrue(condition)`	Checks if condition is true	`assertTrue(calc.isPositive(3));`
`assertFalse(condition)`	Checks if condition is false	`assertFalse(calc.isPositive(-1));`
`assertThrows(Exception.class, executable)`	Checks if a method throws an exception	`assertThrows(ArithmeticException.class, () -> calc.divide(5,0));`
`assertNull(object)`	Checks if object is null	`assertNull(calc.getValue());`
`assertNotNull(object)`	Checks if object is not null	`assertNotNull(calc.getValue());`

Writing Your First JUnit Test

Step-by-step process:

Create a class to test (`Calculator.java`)

Create a JUnit test class (`CalculatorTest.java`)

Write test methods using `@Test`

Use assertions to validate expected behavior

Run tests using IDE or Maven/Gradle

```
class Calculator {
    int add(int a, int b) { return a + b; }
    int subtract(int a, int b) { return a - b; }
}

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

class CalculatorTest {
    Calculator calc = new Calculator();
    @Test
    void testAdd() {
        assertEquals(5, calc.add(2, 3));
    }
    @Test
    void testSubtract() {
        assertEquals(2, calc.subtract(5, 3));
    }
}
```