

# How to Write Unit Tests

## Parameterized Tests

With the parameterized test feature of JUnit, you can test the same functionalities with different parameters (<https://github.com/junit-team/junit4/wiki/Parameterized-tests>). To do so, you need to add at least two annotations of *@RunWith* and *@Parameters* to your test class. Here is an example from the JUnit wiki page (<https://github.com/junit-team/junit4/wiki/Parameterized-tests#identify-individual-test-cases>).

```
import static org.junit.Assert.assertEquals;

import java.util.Arrays;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;

@RunWith(Parameterized.class)
public class FibonacciTest {

    @Parameters(name = "{index}: fib({0})={1}")
    public static Iterable<Object[]> data() {
        return Arrays.asList(new Object[][] {
            { 0, 0 }, { 1, 1 }, { 2, 1 }, { 3, 2 }, { 4, 3 }, { 5, 5 }, { 6, 8 }
        });
    }

    private int input;
    private int expected;

    public FibonacciTest(int input, int expected) {
        this.input = input;
        this.expected = expected;
    }

    @Test
    public void test() {
        assertEquals(expected, Fibonacci.compute(input));
    }
}

public class Fibonacci {
    ...
}
```

It is highly recommended to provide a test name explicitly to easily figure out failed tests.