Unit testing with JUnit

NICOLE KERSTEN – TINFB2 – 12.02.2020

Content

Unit testing

- Automated tests
- Prove that behavior of code is as expected
- A test should focus on one single action
- Requires a module based structure
- Tests should be independent

What is JUnit?

- Open Source Framework for automated unit testing in Java
- One of a family of unit testing Frameworks (xUnit)
- PHPUnit, SUnit, Nunit (.Net), …
- Developed by Erich Gamma and Kent Beck
- Latest version 5.6.2
- JUnit 5 or JUnit Jupiter
- Integrated in IntelliJ and other dev environments

Version History

JUnit 3:

- Extend JUnit class
- Tests methods have to start with test...
- override methods of superclass (setUp(), tearDown() ...)

JUnit 4:

- @Test annotation
- No overriding of methods
- AssertThat() method

JUnit 5 (JUnit Jupiter):

- No assertThat() method
- Very common to use Hamcrest or AssertJ

Why JUnit?

- Graphical and textual Interface
- Shows test progress in a bar that is green if testing is going fine and it turns red when a test fails
- Runs a suite of tests and reports results
- Easy to rerun tests
- Separates tests and productive code

Additional libraries

Assertions

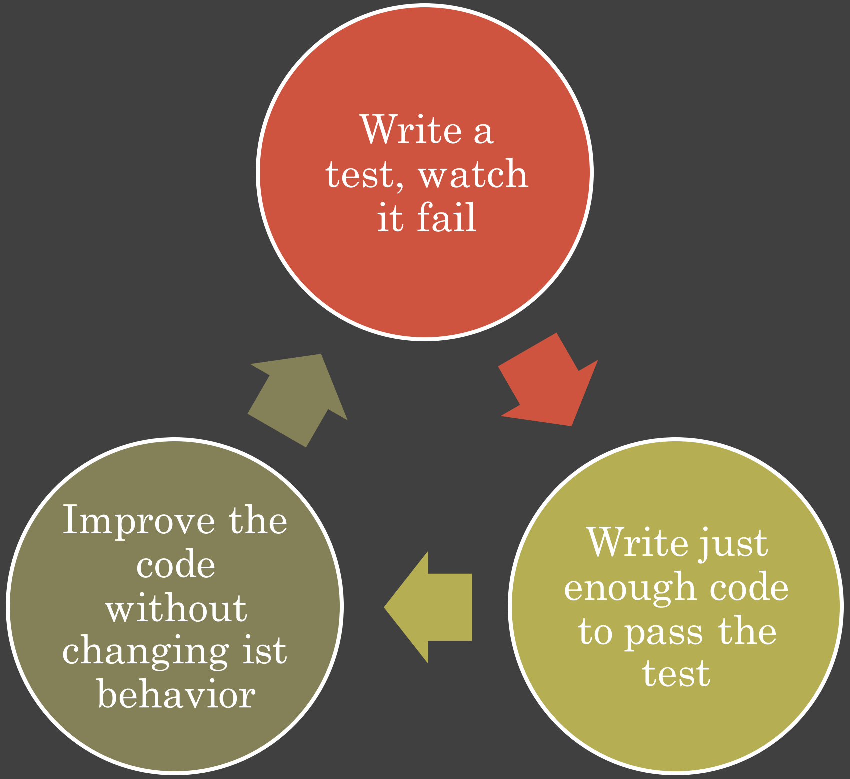
- Hamcrest
- AssertJ

Mocking

- EasyMock
- JMock
- Mockito

Test Driven Development (TDD)

- Better productivity
- Better code quality
- Easier to maintain


Write a test, watch it fail
Write just enough code to pass the test
Improve the code without changing ist behavior

Writing tests

- void method with informative name
- Annotation @Test
- Call method you want to test
- Check result with an assertion
- JUnit executes the test

Annotations

- @Test
- @RepeatedTest
- @BeforeEach
- @AfterEach
- @BeforeAll
- @AfterAll
- @Disabled
- @Timeout

Assertions in JUnit5

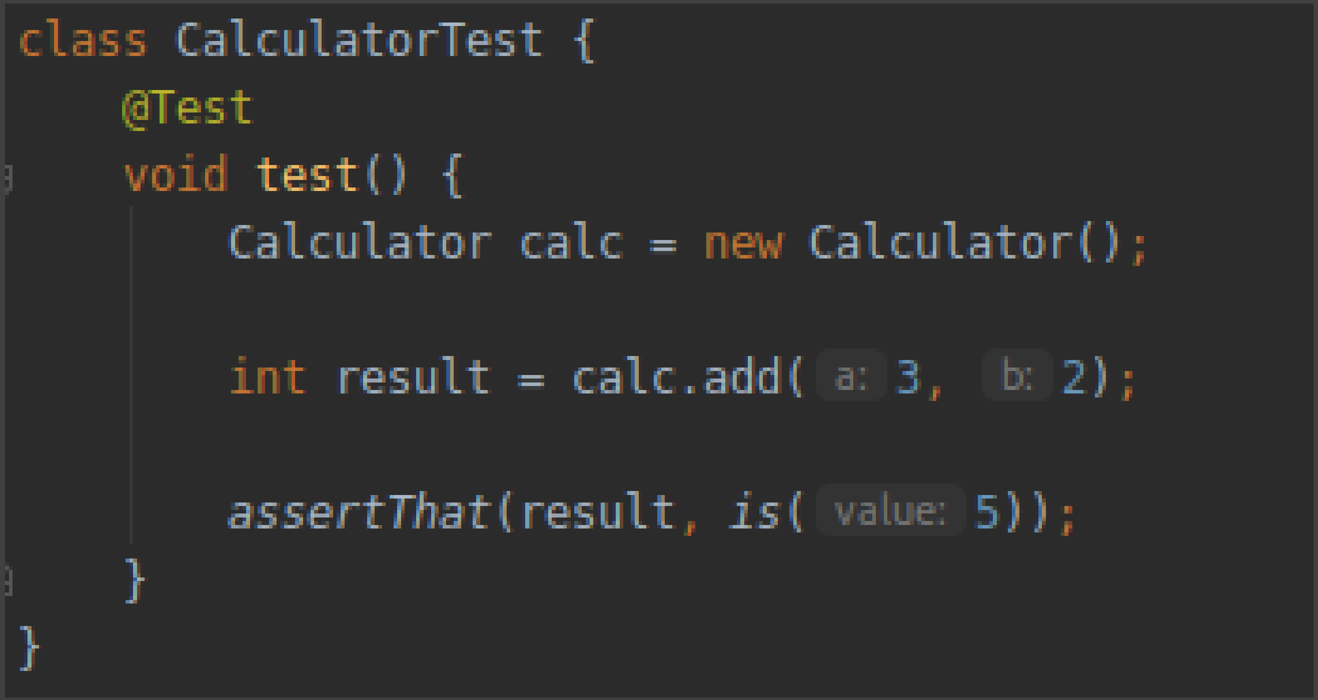
- assertTrue(condition)
- assertFalse(condition)
- assertEquals(expectedObject, actualObject)
- assertSame(expectedObject, actualObject)
- assertNotSame(expectedObject, actualObject)
- assertNull(object)
- assertNotNull(object)
- assertArrayEquals(expectedArray, actualArray)
- fail()

Better option -> Hamcrest

Hamcrest

- assertThat(<actual>, is(<expected>);

- is()
- equalTo()
- instanceOf()
- greaterThan()
- closeTo()
- containsInAnyOrder()



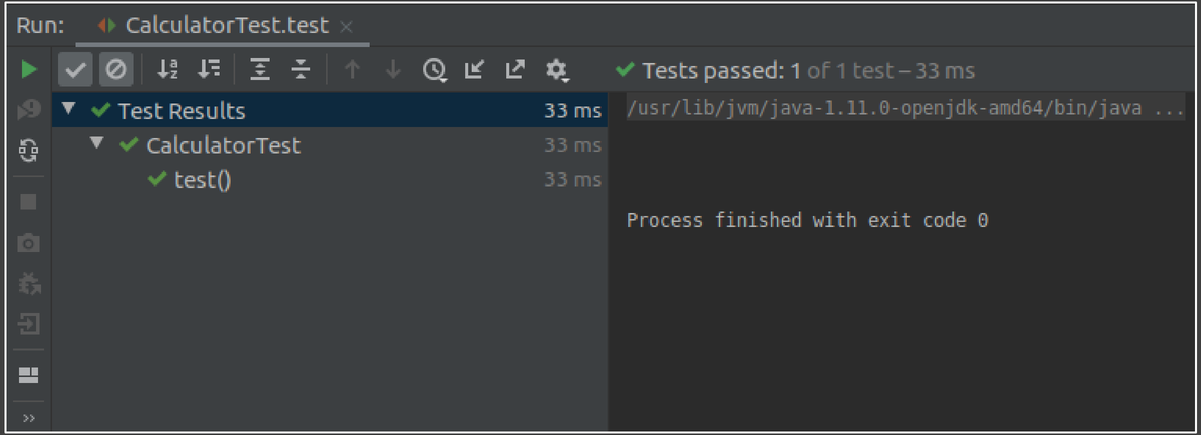
```
class CalculatorTest {
    @Test
    void test() {
        Calculator calc = new Calculator();

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

        assertThat(result, is( value: 5));
    }
}
```

JUnit example

Running tests

IntelliJ: Shift + F10

Eclipse: Ctrl + F11

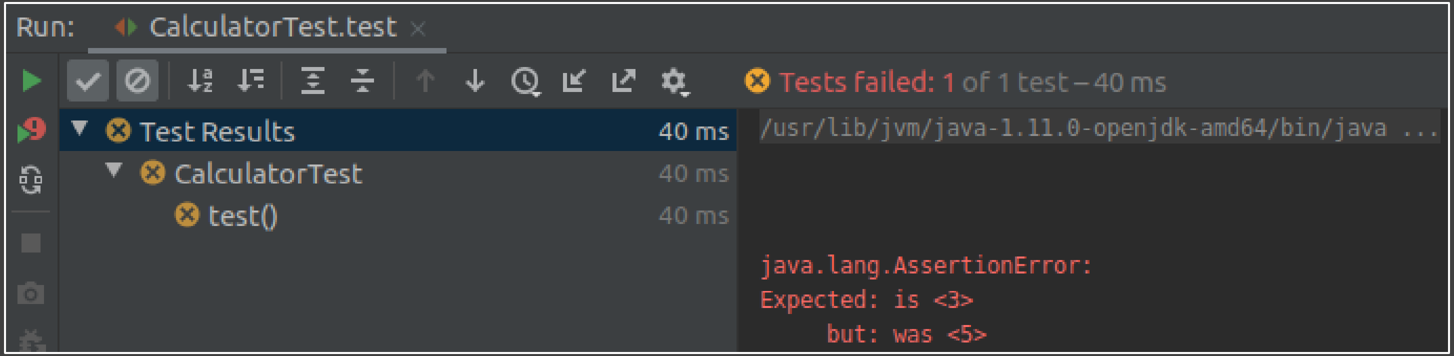


Run: CalculatorTest.test

Tests passed: 1 of 1 test – 33 ms

Test Results	33 ms
CalculatorTest	33 ms
test()	33 ms

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java ...


Process finished with exit code 0
```



Run: CalculatorTest.test

Tests failed: 1 of 1 test – 40 ms

Test Results	40 ms
CalculatorTest	40 ms
test()	40 ms

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java ...


java.lang.AssertionError:
Expected: is <3>
     but: was <5>
```

Live Demo