# CSE4006: Software Engineering

# Lab 11. Testing (1)

Software Engineering Lab

lab(se);

## What is JUnit ?

**JUnit** is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development.

## Features

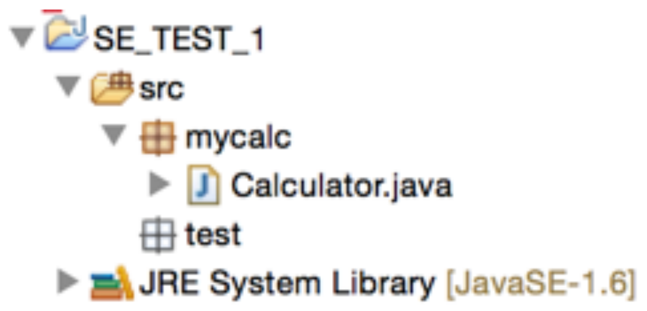
- JUnit is an open source framework which is used for writing & running tests.
- Provides Annotation to identify the test methods.
- Provides Assertions for testing expected results.
- Provides Test runners for running tests.
- JUnit tests allow you to write code faster which increasing quality
- JUnit is elegantly simple. It is less complex & takes less time.
- JUnit tests can be run automatically and they check their own results and provide immediate feedback. There's no need to manually comb through a report of test results.
- JUnit tests can be organized into test suites containing test cases and even other test suites.
- Junit shows test progress in a bar that is green if test is going fine and it turns red when a test fails.

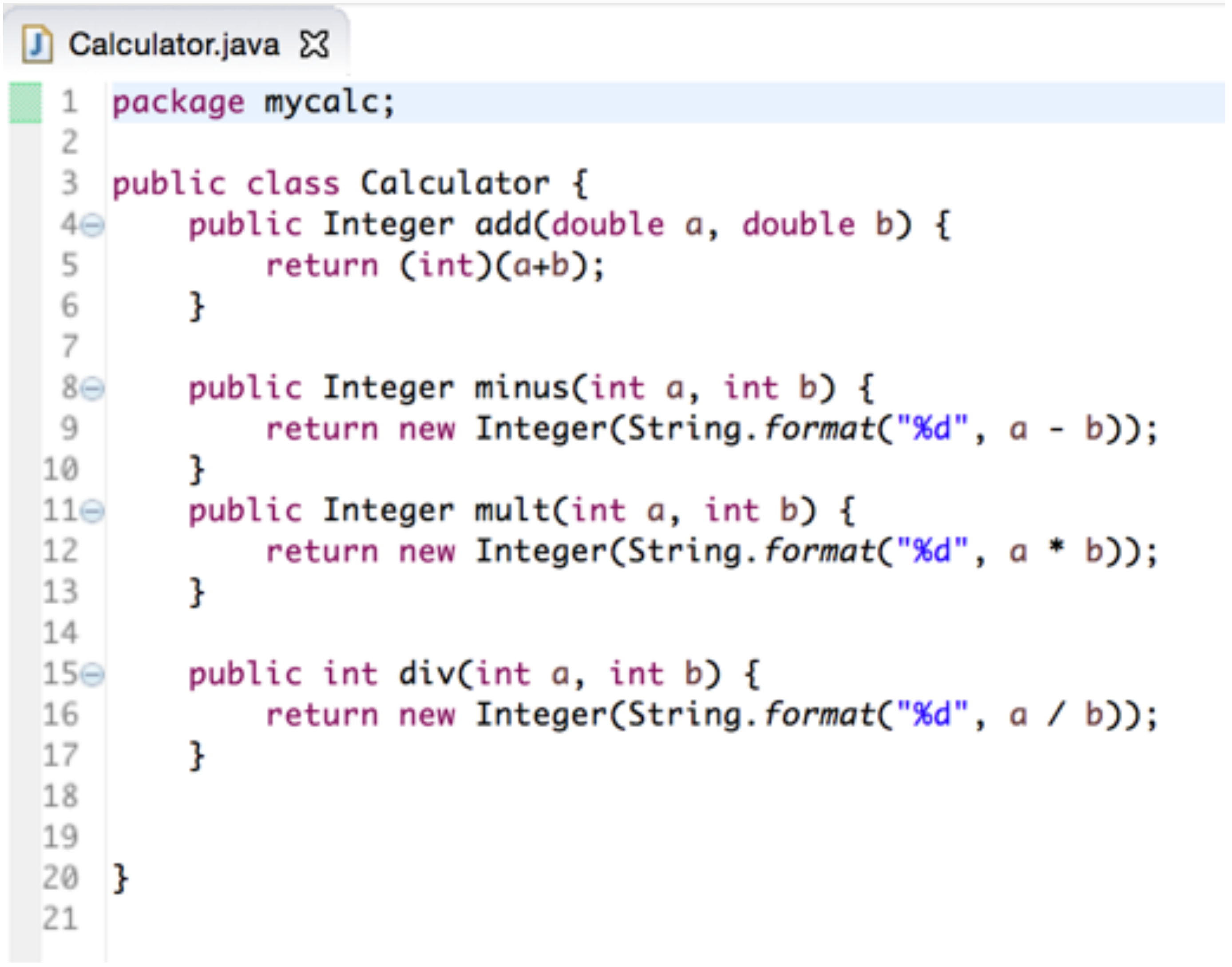
# JUnit test fixture

```
import org.junit.*;

public class TestFoobar {
    @BeforeClass
    public static void setUpClass() throws Exception {
        // Code executed before the first test method
    }

    @Before
    public void setUp() throws Exception {
        // Code executed before each test
    }

    @Test
    public void testOneThing() {
        // Code that tests one thing
    }

    @Test
    public void testAnotherThing() {
        // Code that tests another thing
    }

    @Test
    public void testSomethingElse() {
        // Code that tests something else
    }

    @After
    public void tearDown() throws Exception {
        // Code executed after each test
    }

    @AfterClass
    public static void tearDownClass() throws Exception {
        // Code executed after the last test method
    }
}
```

# JUnit Tutorial

File > New Java Project
File > New Package
File > New Class

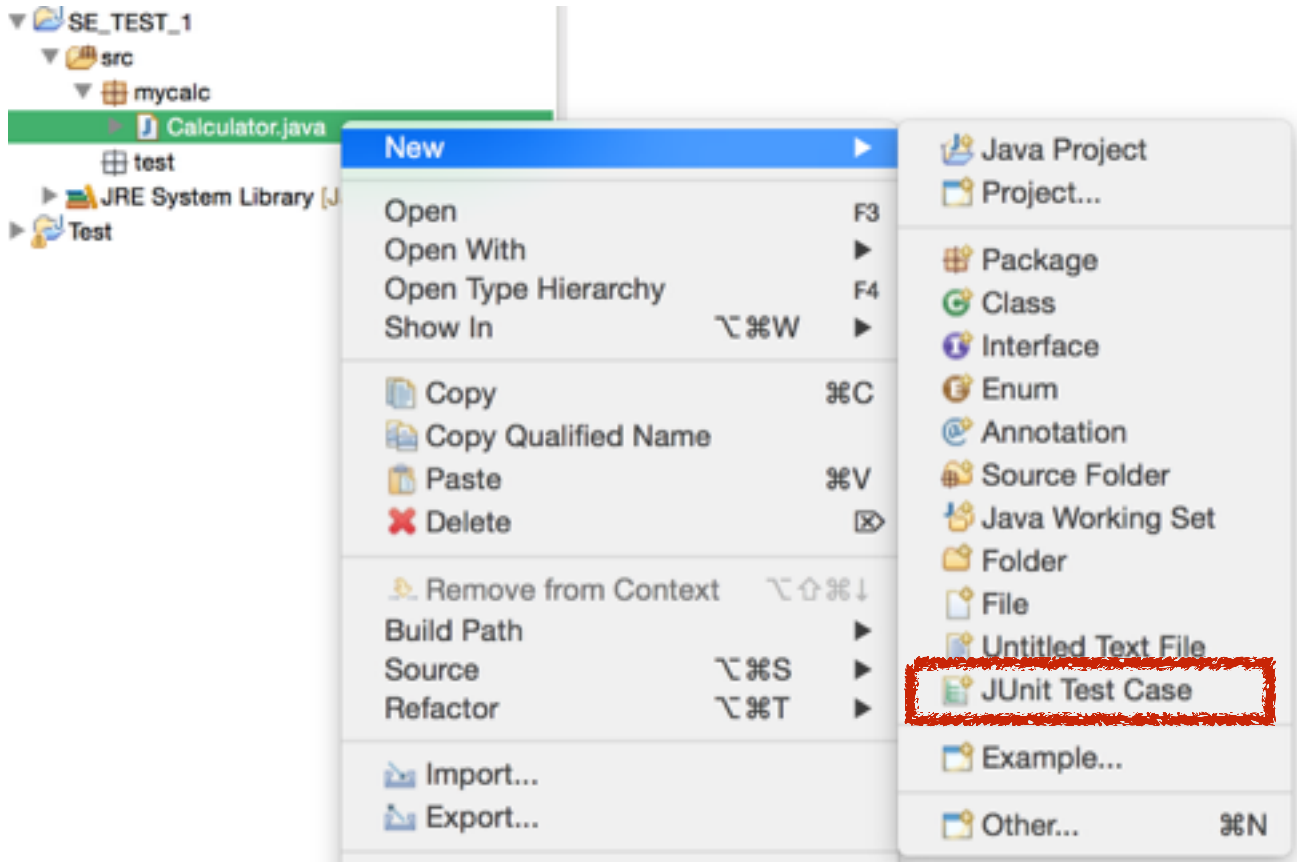




```java
package mycalc;

public class Calculator {
    public Integer add(double a, double b) {
        return (int)(a+b);
    }

    public Integer minus(int a, int b) {
        return new Integer(String.format("%d", a - b));
    }
    public Integer mult(int a, int b) {
        return new Integer(String.format("%d", a * b));
    }

    public int div(int a, int b) {
        return new Integer(String.format("%d", a / b));
    }


}

```

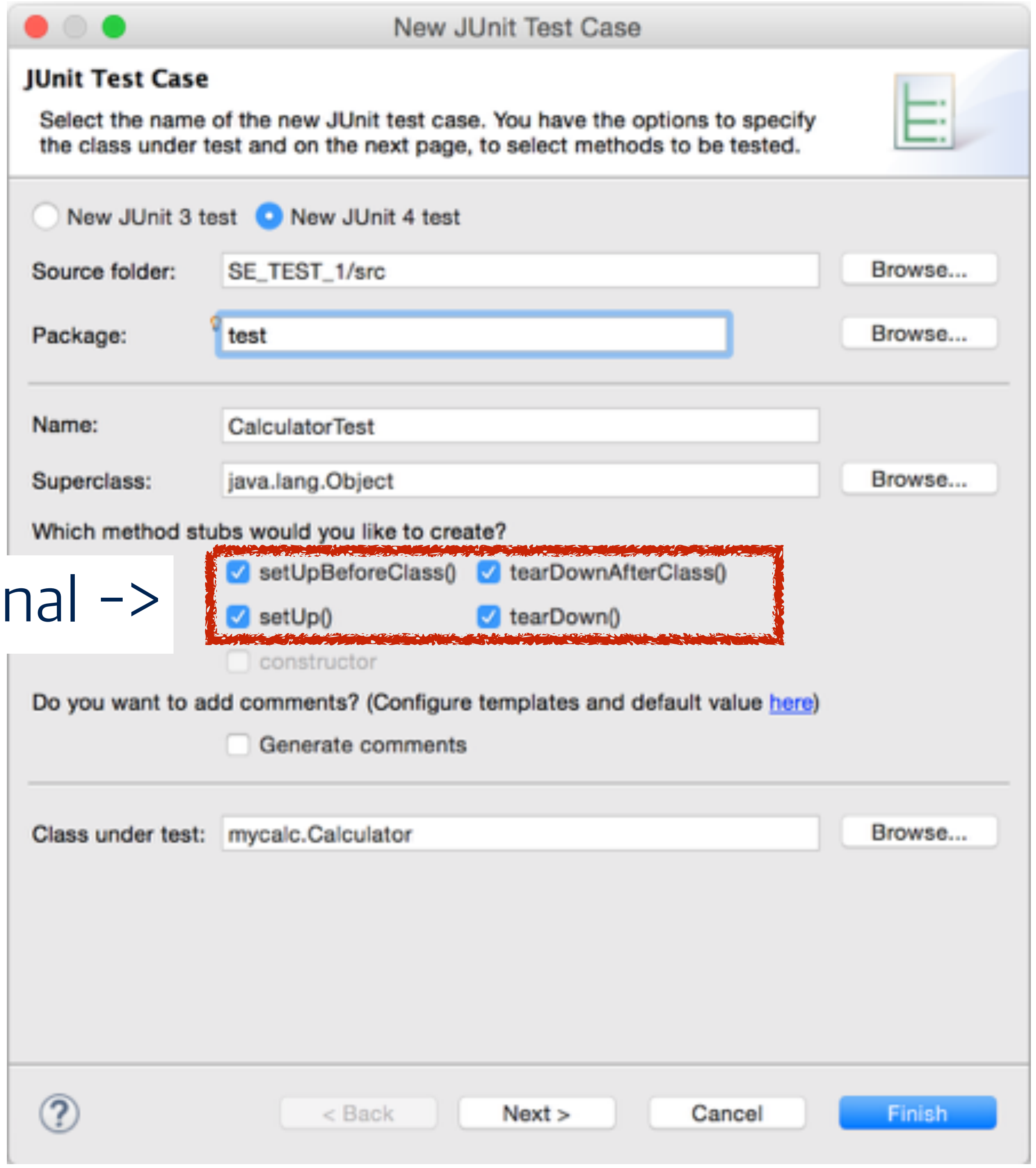
Simple Calculator JavaCode
download code : http://pasted.co/b32693f3

# JUnit Tutorial (cont'd)



File > New > JUnit Test Case

# JUnit Tutorial (cont'd)



New JUnit Test Case

**JUnit Test Case**

Select the name of the new JUnit test case. You have the options to specify the class under test and on the next page, to select methods to be tested.

New JUnit 3 test [✓ New JUnit 4 test]

Source folder: SE_TEST_1/src

Package: test

Name: CalculatorTest

Superclass: java.lang.Object

Which method stubs would you like to create?

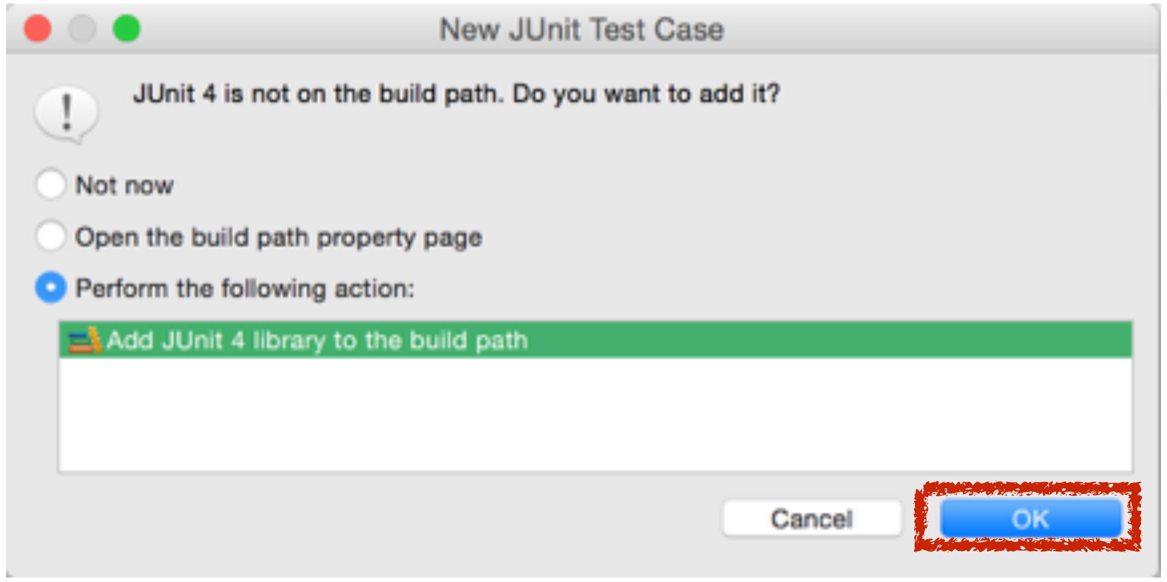
optional ->

- ✓ setUpBeforeClass()
- ✓ tearDownAfterClass()
- ✓ setUp()
- ✓ tearDown()
- constructor

Do you want to add comments? (Configure templates and default value here)

- Generate comments

Class under test: mycalc.Calculator



New JUnit Test Case

JUnit 4 is not on the build path. Do you want to add it?

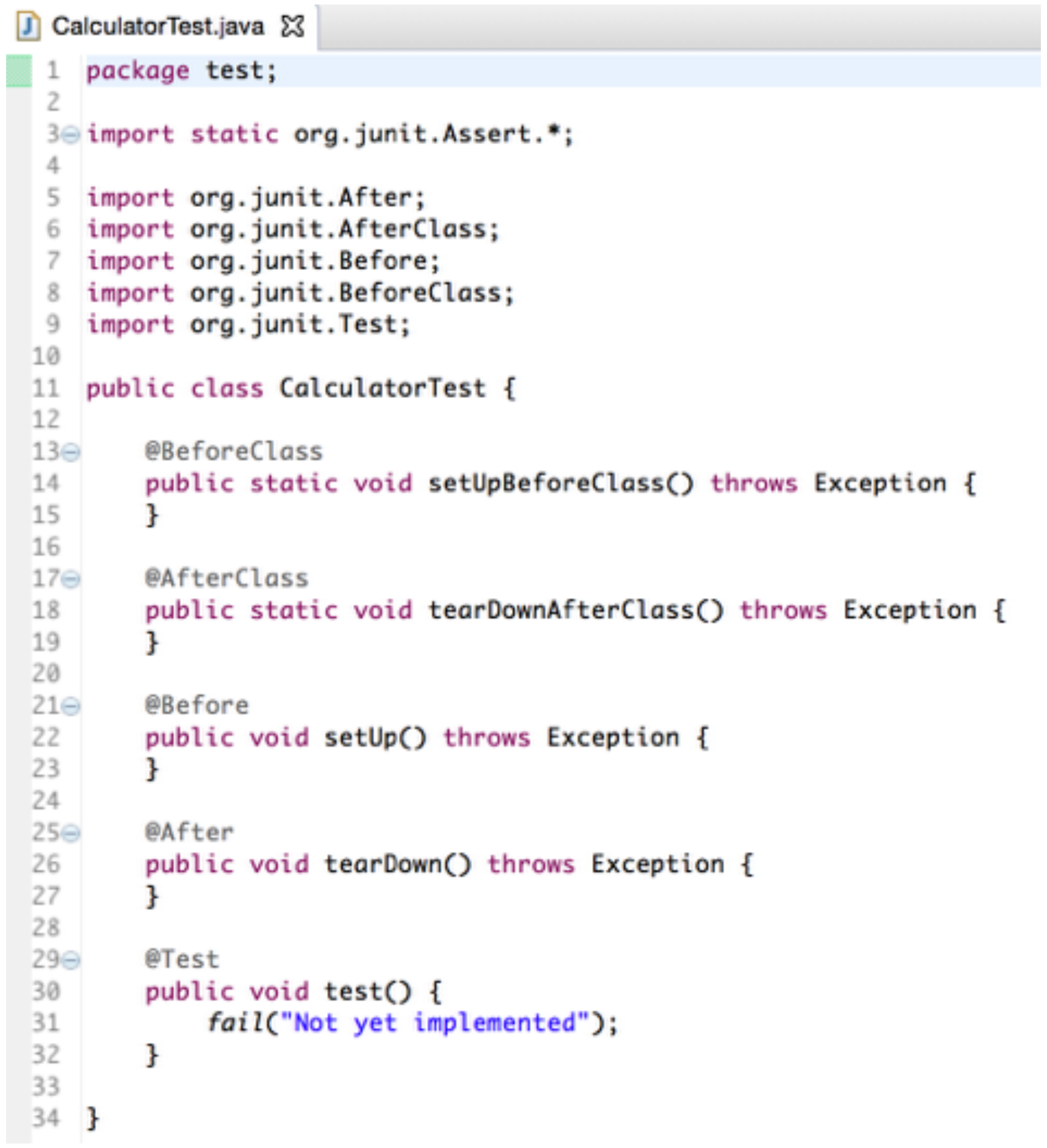
Not now

Open the build path property page

[✓ Perform the following action:]

Add JUnit 4 library to the build path

# JUnit Tutorial (cont'd)

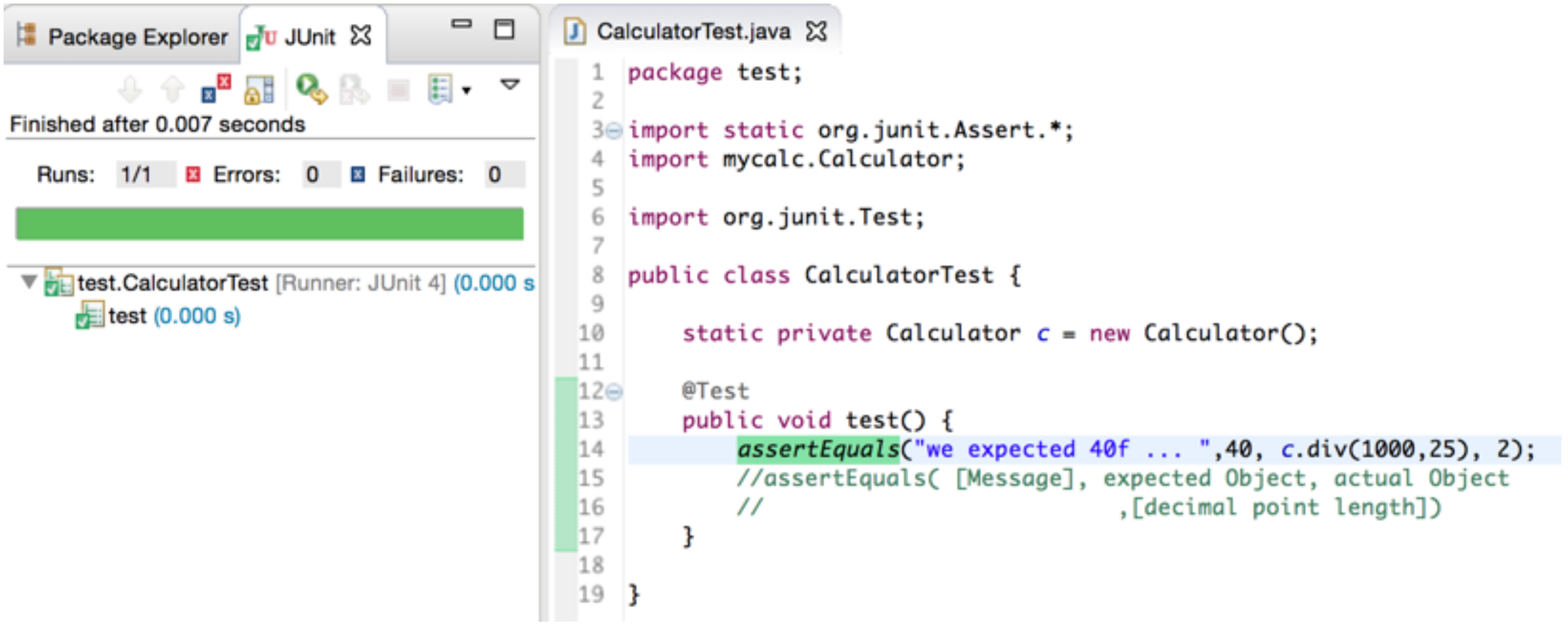


CalculatorTest.java

```java
package test;

import static org.junit.Assert.*;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class CalculatorTest {

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
    }

    @AfterClass
    public static void tearDownAfterClass() throws Exception {
    }

    @Before
    public void setUp() throws Exception {
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void test() {
        fail("Not yet implemented");
    }

}
```

Generated testing code, but not work.

# JUnit Tutorial (cont'd)



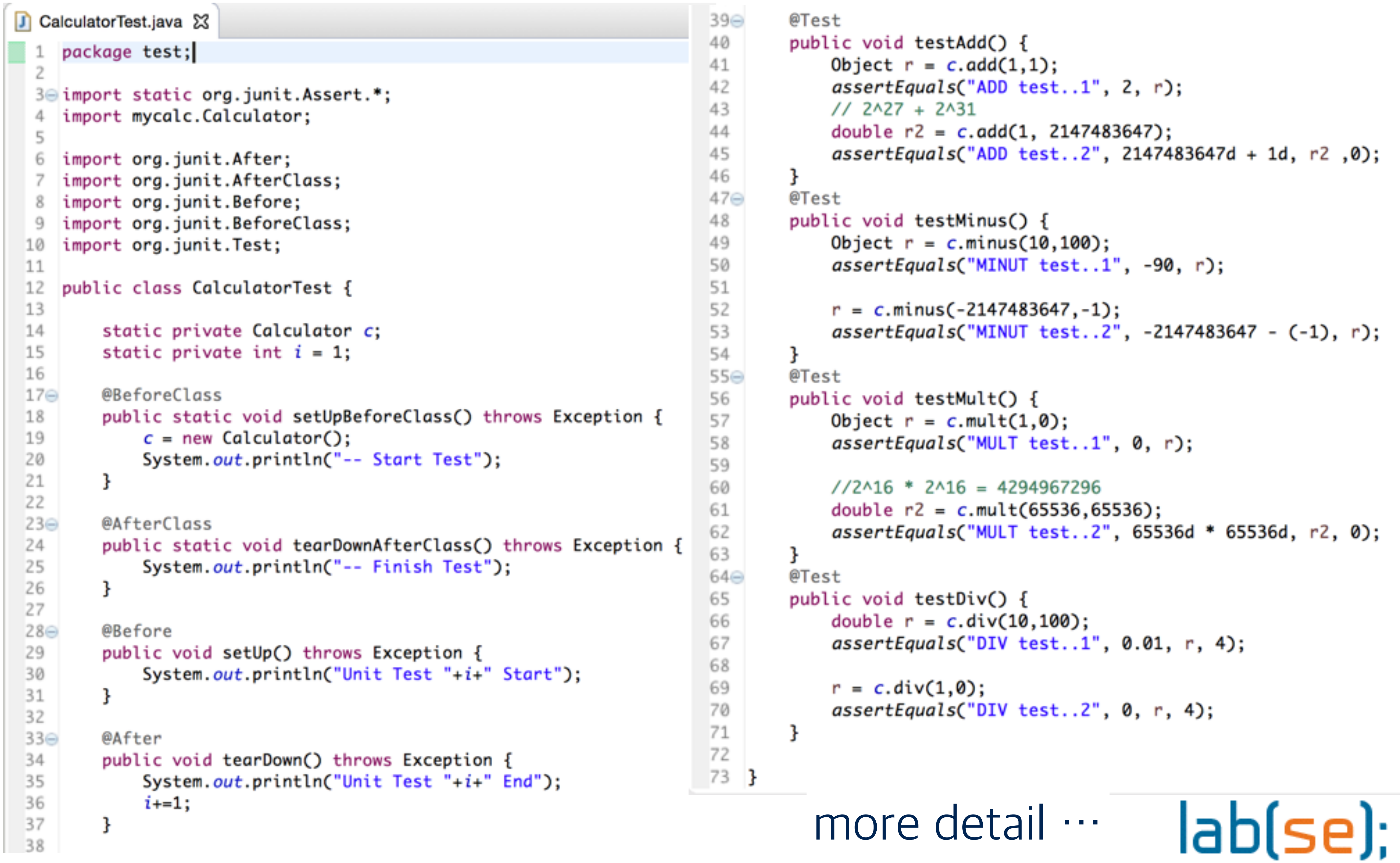
```
Finished after 0.007 seconds

Runs: 1/1    Errors: 0    Failures: 0

test.CalculatorTest [Runner: JUnit 4] (0.000 s
    test (0.000 s)
```

```java
package test;

import static org.junit.Assert.*;
import mycalc.Calculator;

import org.junit.Test;

public class CalculatorTest {

    static private Calculator c = new Calculator();

    @Test
    public void test() {
        assertEquals("we expected 40f ... ",40, c.div(1000,25), 2);
        //assertEquals( [Message], expected Object, actual Object
        //                            ,[decimal point length])
    }

}
```

Simplest unit testing code(right) and result(left)

lab[se];

# JUnit Tutorial (cont'd)

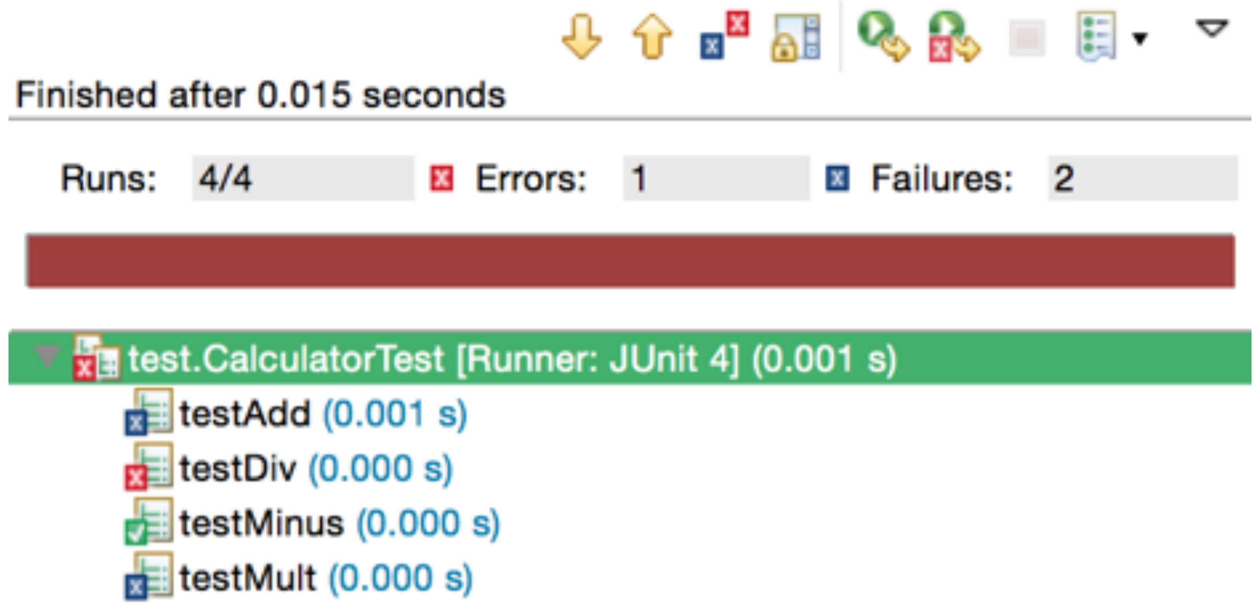

CalculatorTest.java

```java
package test;

import static org.junit.Assert.*;
import mycalc.Calculator;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class CalculatorTest {

    static private Calculator c;
    static private int i = 1;

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
        c = new Calculator();
        System.out.println("-- Start Test");
    }

    @AfterClass
    public static void tearDownAfterClass() throws Exception {
        System.out.println("-- Finish Test");
    }

    @Before
    public void setUp() throws Exception {
        System.out.println("Unit Test "+i+" Start");
    }

    @After
    public void tearDown() throws Exception {
        System.out.println("Unit Test "+i+" End");
        i+=1;
    }

    @Test
    public void testAdd() {
        Object r = c.add(1,1);
        assertEquals("ADD test..1", 2, r);
        // 2^27 + 2^31
        double r2 = c.add(1, 2147483647);
        assertEquals("ADD test..2", 2147483647d + 1d, r2 ,0);
    }
    @Test
    public void testMinus() {
        Object r = c.minus(10,100);
        assertEquals("MINUT test..1", -90, r);

        r = c.minus(-2147483647,-1);
        assertEquals("MINUT test..2", -2147483647 - (-1), r);
    }
    @Test
    public void testMult() {
        Object r = c.mult(1,0);
        assertEquals("MULT test..1", 0, r);

        //2^16 * 2^16 = 4294967296
        double r2 = c.mult(65536,65536);
        assertEquals("MULT test..2", 65536d * 65536d, r2, 0);
    }
    @Test
    public void testDiv() {
        double r = c.div(10,100);
        assertEquals("DIV test..1", 0.01, r, 4);

        r = c.div(1,0);
        assertEquals("DIV test..2", 0, r, 4);
    }

}
```

more detail …

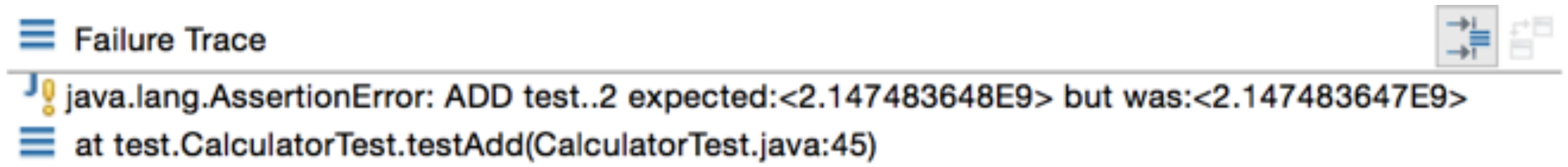
lab(se);

# JUnit Tutorial (cont'd)



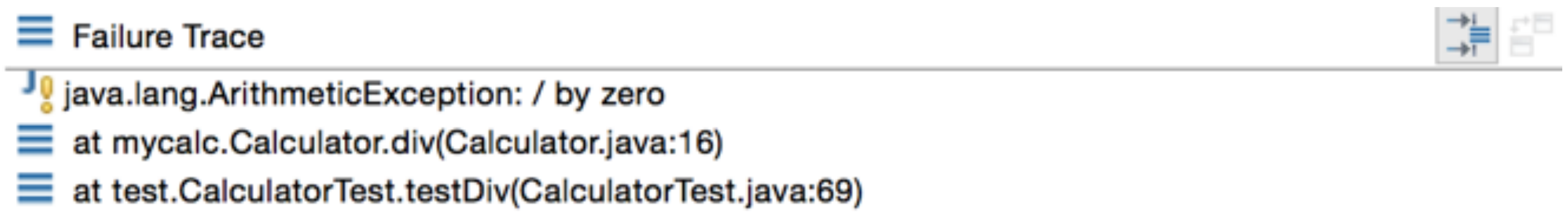
```
Finished after 0.015 seconds
Runs: 4/4    Errors: 1    Failures: 2
test.CalculatorTest [Runner: JUnit 4] (0.001 s)
    testAdd (0.001 s)
    testDiv (0.000 s)
    testMinus (0.000 s)
    testMult (0.000 s)
```

Result



```
Failure Trace
java.lang.AssertionError: ADD test..2 expected:<2.147483648E9> but was:<2.147483647E9>
at test.CalculatorTest.testAdd(CalculatorTest.java:45)
```

testAdd() Failure Trace



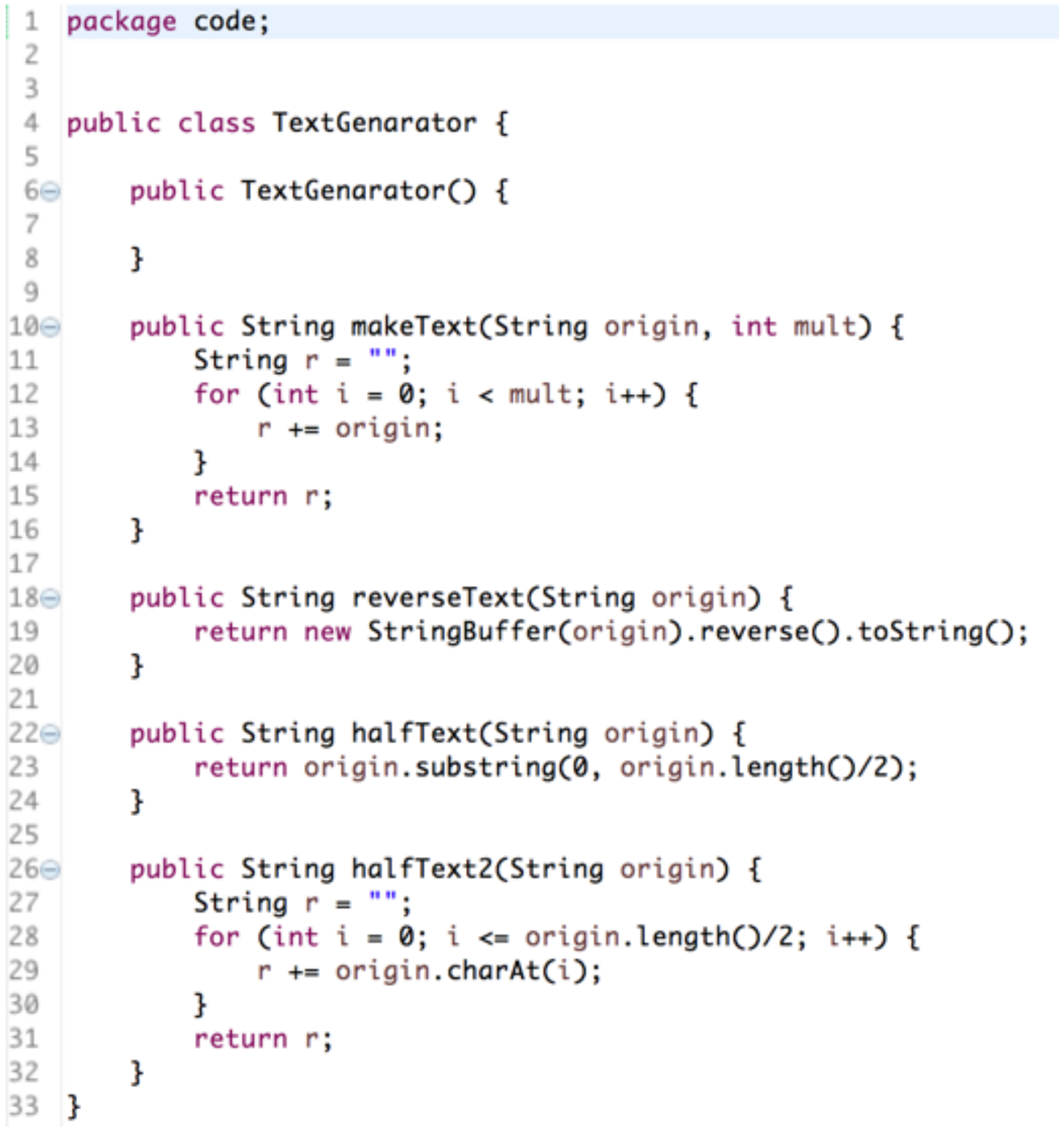
```
Failure Trace
java.lang.ArithmeticException: / by zero
at mycalc.Calculator.div(Calculator.java:16)
at test.CalculatorTest.testDiv(CalculatorTest.java:69)
```

testDiv() Failure Trace

lab[se];

# Exercise.

- Write test code for each method. ( makeText(), reverseText(), halfText() )
- Check **halfTest()** is equal to **halfText2()**.



```java
package code;


public class TextGenarator {

    public TextGenarator() {

    }

    public String makeText(String origin, int mult) {
        String r = "";
        for (int i = 0; i < mult; i++) {
            r += origin;
        }
        return r;
    }

    public String reverseText(String origin) {
        return new StringBuffer(origin).reverse().toString();
    }

    public String halfText(String origin) {
        return origin.substring(0, origin.length()/2);
    }

    public String halfText2(String origin) {
        String r = "";
        for (int i = 0; i <= origin.length()/2; i++) {
            r += origin.charAt(i);
        }
        return r;
    }
}
```

download code : http://pasted.co/82ced221

## Submission

1. Take screenshots your code and results

2. Compress your screenshots to 20xx03xxxx_YourName.zip

3. Send mail to ng0301@gmail.com