

## Summary 15 - JUnit

- What is this?

Unit testing means testing a "unit", here, a method. One way to test a method that returns a value is to:

1. Call the method, sending input as its arguments.
2. Print the value returned by the method.
3. Look at the printed value to see if it is the right output for the given input.

JUnit is a unit testing *framework*, that is, a collection of classes to be used in another program. It allows us to semi-automate the above process. Here is how:

1. We store input/output pairs (test cases) in a particular notation (see example).
2. We run JUnit.
3. It shows us which tests passed, and for the failures, where the failure occurred and other information.

By using JUnit, anyone else who knows JUnit can run the tests! If someone changes our code, they can run the tests and see if they broke something!

JUnit was written by Erich Gamma and Kent Beck. It is open-source software now used by **millions** of Java developers.

To write JUnit tests in Eclipse:

1. Right-click the class for which you want to write tests and select New ~ JUnit Test Case.
2. In the dialog box that appears, select the New JUnit 4 Test radio button and check whichever boxes you want, where:
  - setUp (which uses the @Before annotation) is run before each test
  - setUpBeforeClass (which uses the @BeforeClass annotation) is run once before all the tests
  - tearDown and tearDownAfter are similar, for after instead of before

3. In the next dialog box that appears, check the methods that you want to test. That causes a test stub to be created. Notice that it has an @Test annotation.

All you have to do to make tests is put

@Test

in front of the name of the test method. Use separate methods (each with @Test) for separate tests.

For tests, write any code you want, but the most common test is to use *assertEquals* as in the example below. It simply checks whether the first argument (expected output) equals the second argument (actual output, from calling the method being tested).

- Example

```
public class CensorTest {
    @Test
    public void test1() {
        Censor censor_s = new Censor('s');
        assertEquals("Mi**i**ippi",
            censor_s.transform("Mississippi"));
    }

    @Test
    public void test2() {
        Censor censorDefault = new Censor();
        assertEquals("H*lllo th*r*",
            censorDefault.transform("Hello there"));
    }
}
```

- For further study:

- *JUnit FAQ* at [junit.sourceforge.net/doc/faq/faq.htm](http://junit.sourceforge.net/doc/faq/faq.htm)
- Documentation of the org.junit.Assert class
- This summary's *author*: David Mutchler