



# Introduction to Computer Science 1

WS2006/07

Prof. Schiele, Prof. Kangasharju  
<http://www.mis.informatik.tu-darmstadt.de/ics1>

JUnit 4 Primer

29.01.06

## 1 Introduction

This tutorial gives a short overview of JUnit 4 [1]. It is based on [2]. To work through this tutorial, you should be familiar with eclipse and programming in Java. Knowledge about quality management is not necessary.

### 1.1 What is JUnit?

JUnit is a framework that allows a developer to write unit tests easily and run collections of tests -so called test suites - conveniently. JUnit offers different methods (such as `assertTrue()`, `assertFalse()` and `assertEquals()`) which enable a developer to easily check conditions in his code. It also offers many annotations (annotations require Java 5), whereby one can declare methods as test, setup, teardown or ignored.

To test a class, one normally implements the following methods:

- A setup-method that creates all needed objects and establishes the initial state for the test.
- Test-methods that contain the actual tests.
- A teardown-method to clean up.

It is then possible to run the tests automatically in a separate window and to view the results and reasons for failures, if any. How JUnit 4 works is shown with examples in section 2.

### 1.2 Why use JUnit?

#### Test Driven Development:

TDD is a design methodology which is getting increasingly popular and is often recommended with good cause. It prescribes that tests should be written before the code, which has to pass these tests. TDD is one of the key elements of XP (eXtreme Programming [3]) and has been adopted by most agile approaches to software-development. JUnit is best suited for TDD and supports the developer in getting a clear picture of what a method or a class has to fulfill. To this end, ignore annotations are very helpful because they make it possible to temporarily disable certain tests for parts that haven't been implemented yet.

#### Regression Testing

Writing a regression suite without the help of tools is practically impossible. Regression tests are performed after every change or addition to a program and ensure that there are no unintentional side effects that may endanger the correctness of other functions. In big projects, it is also not feasible to run such a testsuite over and over by hand.

## 2 Installation and Usage in Eclipse

Eclipse 3.2 already comes with JUnit 4.1. The only thing you have to do is: create a new Project. Right click on Project and select New → JUnit Test Case. In the following dialogue choose New JUnit 4 test. If a warning is shown that JUnit 4 is not on the build path, just click Click here and select OK to solve the problem. Afterwards, input `JUnitHelloWorld` as class name and insert the following code:

```
import org.junit.Test;
import static org.junit.Assert.*;

public class JUnitHelloWorldEclipse3_2 {
    @Test
    public void testHelloWorld() {
        String s = "HelloWorld";
        assertEquals("Just a test to see if everything works ...", "HelloWorld", s);
    }
}
```

Now click on the class inside the Package Explorer and select Run As → JUnit Test to run the test.

## 3 Language elements of JUnit 4

A short overview of the language elements of a JUnit testsuite.

### 3.1 Annotations

#### @BeforeClass:

Methods that are called before any test runs. Test parameters can be set and objects can be instantiated here if they do not change in later tests.

#### @AfterClass:

Methods that are called at the end of a testsuite after all tests have been run. Resources that have been used during the tests (like network connections or streams) can be freed here.

#### @Before:

Methods that are executed before every test.

#### @After:

Methods that are executed after every test.

#### @Test:

The actual test methods.

#### @Ignore:

Methods, testing features that are still to be implemented, can be disabled temporarily.

### 3.2 Test methods

#### assertTrue:

Checks if the given expression evaluates to true. If the expression is not true an error is thrown. An error message can be specified.

#### assertFalse:

Checks if given expression evaluates to false. If the expression is not true an error is thrown. An error message can be specified.

#### assertEquals:

Compares two values with each other. Throws an error if the values are not identical. It is implemented for several types.

#### fail:

Always fails. Can be used to indicate tests yet to be implemented.

### 3.3 Helper methods

It is possible to declare and use normal auxiliary methods in the class.

## 4 Example suite of a banking program

Some JUnit elements are demonstrated on the basis of a money class of a bank program.

```
import org.junit.Test;
import static org.junit.Assert.*;
public class MoneyTest{
    private Money am1, am2;

    /**
     * Methods with the annotation 'BeforeClass' are executed once before
     * the first of the series of tests . External resources that are used
     * by all tests should be initialised here .
     */
    @Before()
    protected void setUp() {
        am1 = new Money(12, "EUR");
        am2 = new Money(14, "EUR");
    }

    /**
     * Methods with the annotation 'Test' are the actual test methods .
     */
    @Test()
    public void testSimpleAdd() {
        Money expected = new Money(26, "EUR");
        Money result = am1.add(am2);
```

```
        assertTrue(expected.equals(result));
    }

    @Test()
    public void testEquals() {
        assertTrue(!am1.equals(null));
        assertEquals(am1, new Money(12, "EUR"));
        assertTrue(!am1.equals(am2));
    }
}
```

## References

- [1] K. Beck, E. Gamma and others, "JUnit Testing Framework," <http://www.junit.org>; accessed Jan 26, 2007.
- [2] A. Anjorin, "JUnit 4 Tutorial" TU Darmstadt, FB Informatik, FG Metamodellierung, 2006, <http://www.mm.informatik.tu-darmstadt.de/courses/helpdesk/junit4.pdf>; accessed Jan 26, 2007.
- [3] "Extreme Programming."(2007, January 21). In Wikipedia, The Free Encyclopedia. Retrieved 14:08, January 26, 2007, from [http://en.wikipedia.org/w/index.php?title=Extreme\\_Programming&oldid=102295329](http://en.wikipedia.org/w/index.php?title=Extreme_Programming&oldid=102295329)