

MBT preparation

March 2012

Lets Start

- ◉ You need to have some packages installed and running to (easily) do the first assignment.
- ◉ Chalmers machines are well configured. 😊
- ◉ In most *nix machines you can easily use package manager to install all these tools.



- ◉ It is not necessary to run these commands in terminal, It is just to let you know what happens underneath.

Step 1 : Java

- ◉ Most of the *nix OS come with Java preinstalled. Find out if you have it installed, and which version by typing in the terminal:
`javac -version`
- ◉ If you don't have Java, get it from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- ◉ Make sure you install the full Java Development Kit (JDK), not just the runtime (JRE).
- ◉ You can download and install both Java EE or Java SE.

Step 1 : Java

- ◉ In the end, make sure you have Java version 5 or later.
- ◉ Now you must be able to compile and execute the calculator in terminal using following command :



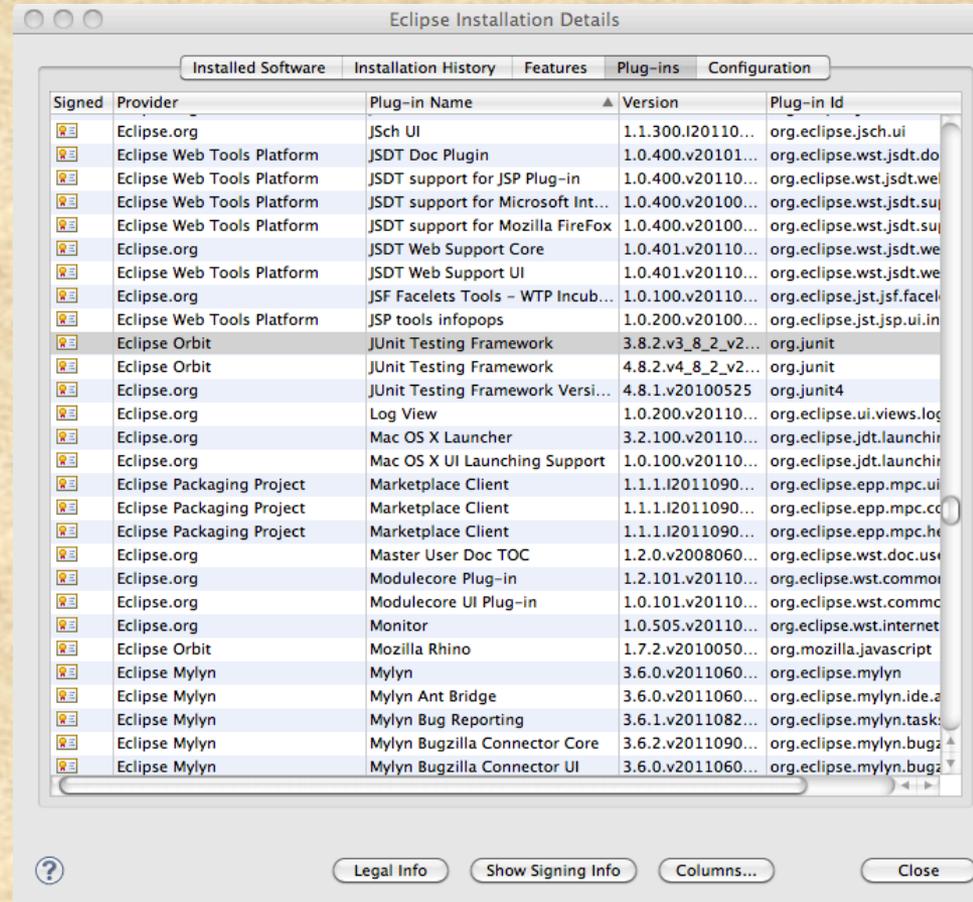
```
javac Calculator.java  
java Calculator 10+ 3=
```

Step 2 : Eclipse

- ◉ You will need an installation of Eclipse on your computer, version 3.5 or later. Version 3.6 or 3.7 include Eclipse Marketplace client, which makes things a bit easier.
- ◉ If you don't have it, get it from <http://www.eclipse.org/downloads>
- ◉ One of the following packages is recommended:
 - Eclipse IDE for Java Developers
 - Eclipse IDE for Java EE Developers
- ◉ Whatever package you choose, make sure it includes Java Development Tools (JDT) (check package details).
- ◉ Now using eclipse GUI, you are able import the calculator class (or create a new project/class).

Step 3 : JUnit

- JUnit is an implementation of xUnit framework for Java language.
- Many Eclipse packages have JUnit preinstalled. Verify that by activating the menu About Eclipse -> Installation Details. You should see it in the plug-ins list.



Step 3 : JUnit

- If JUnit is not there, download it from www.junit.org and use the included installation instructions.
- Now you can use this command to test your test suite



```
javac -cp .:PATH_TO_JUnit/junit-4.10.jar  
Calculator.java CalculatorTest.java
```

```
java -cp .:PATH_TO_JUnit/junit-4.10.jar  
org.junit.runner.JUnitCore CalculatorTest
```

Step 4: EclEmma

Emma/EclEmma is a Java code coverage plugin for Eclipse.

Install it from Eclipse Software or marketplace:
Go to Eclipse menu

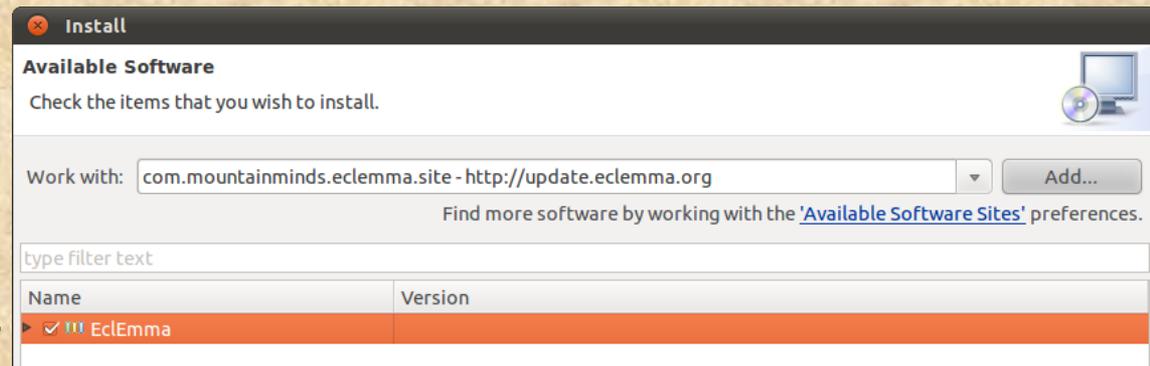
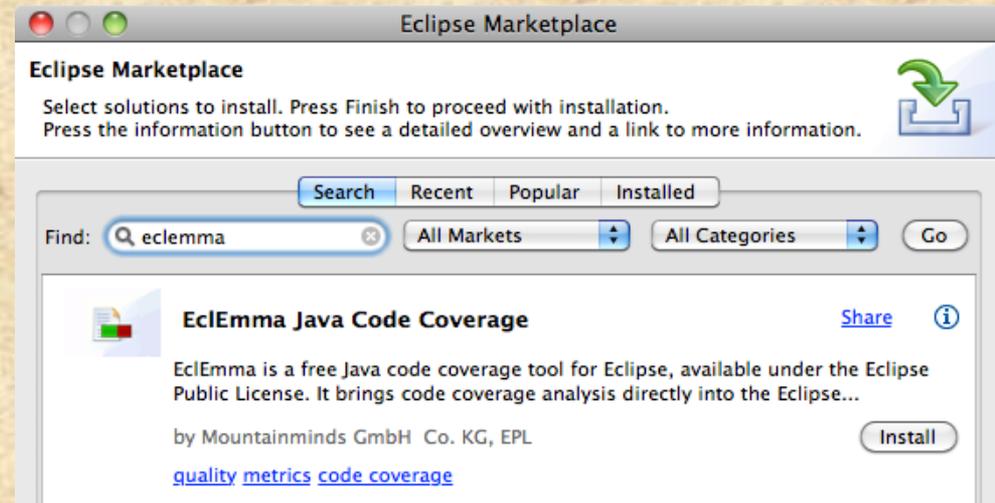
Help ->

Eclipse Marketplace

• *Or*

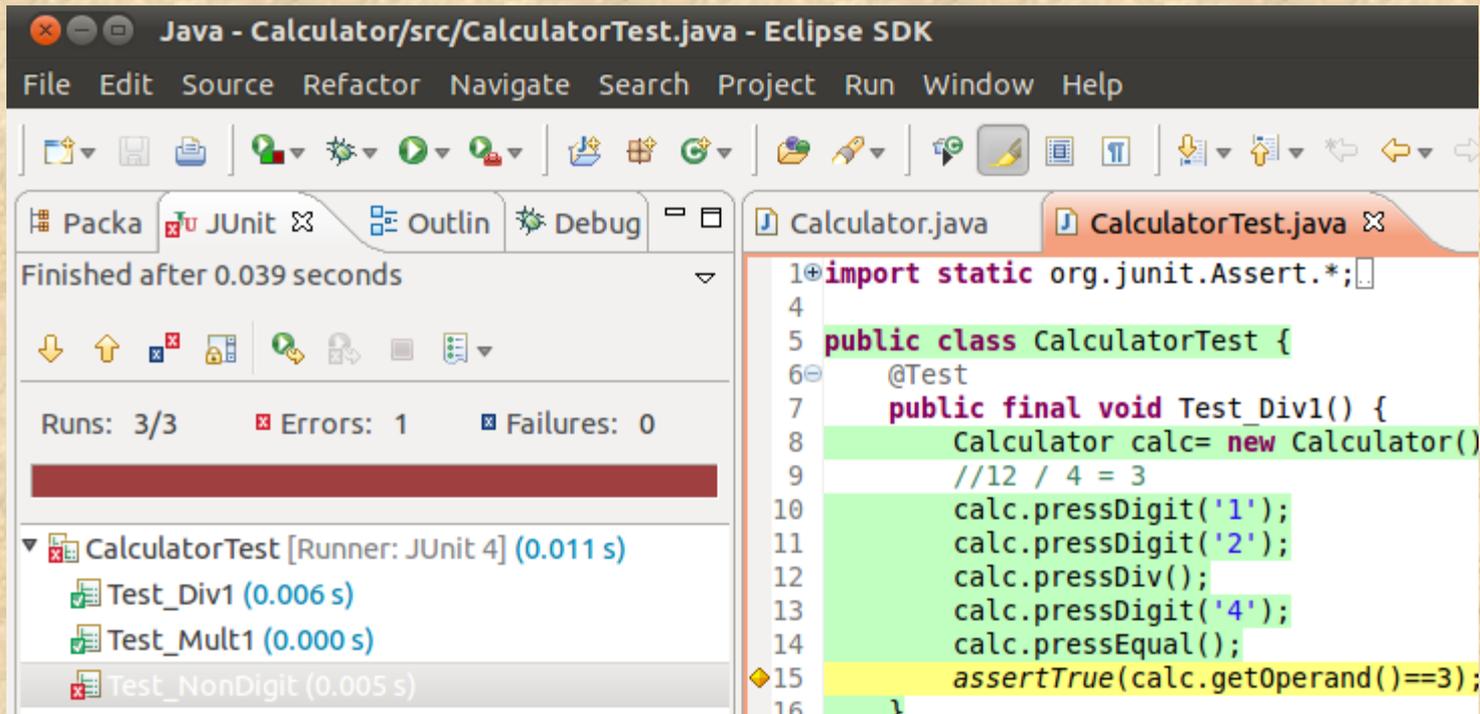
Help → Install New software

Search for EclEmma



Step 4: EclEmma

- Now you can use Coverage drop-down toolbar button in your current workbench perspective



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu bar is a toolbar with various icons. The main editor area displays two tabs: Calculator.java and CalculatorTest.java. The CalculatorTest.java tab is active, showing the following code:

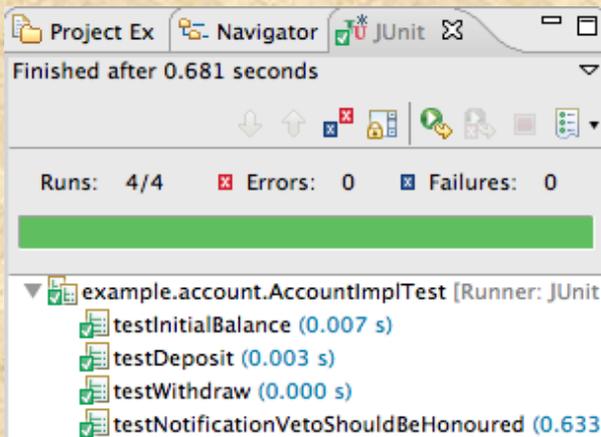
```
1 import static org.junit.Assert.*;
4
5 public class CalculatorTest {
6     @Test
7     public final void Test_Div1() {
8         Calculator calc= new Calculator()
9         //12 / 4 = 3
10        calc.pressDigit('1');
11        calc.pressDigit('2');
12        calc.pressDiv();
13        calc.pressDigit('4');
14        calc.pressEqual();
15        assertTrue(calc.getOperand()==3);
16    }
```

The left sidebar shows the Package Explorer and the JUnit toolbar. Below the toolbar, it indicates "Finished after 0.039 seconds". The Run toolbar shows "Runs: 3/3", "Errors: 1", and "Failures: 0". The Test Results view shows the following test results:

- CalculatorTest [Runner: JUnit 4] (0.011 s)
 - Test_Div1 (0.006 s) [Pass]
 - Test_Mult1 (0.000 s) [Pass]
 - Test_NonDigit (0.005 s) [Fail]

Step 5: Verify the environment

- From the menu *Run -> Coverage As* choose *JUnit Test*.
- Two new views should appear, *JUnit* and *Coverage*, showing test results and code coverage information.
Besides, the source code should be highlighted.



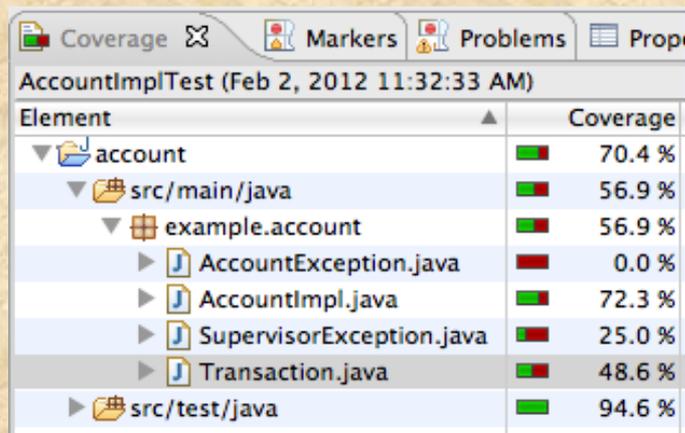
Project Ex Navigator JUnit

Finished after 0.681 seconds

Runs: 4/4 Errors: 0 Failures: 0

example.account.AccountImplTest [Runner: JUnit]

- testInitialBalance (0.007 s)
- testDeposit (0.003 s)
- testWithdraw (0.000 s)
- testNotificationVetoShouldBeHonoured (0.633 s)



Coverage Markers Problems Prop

AccountImplTest (Feb 2, 2012 11:32:33 AM)

Element	Coverage
account	70.4 %
src/main/java	56.9 %
example.account	56.9 %
AccountException.java	0.0 %
AccountImpl.java	72.3 %
SupervisorException.java	25.0 %
Transaction.java	48.6 %
src/test/java	94.6 %

```
this.type = type;
this.amount = amount;
}

public int getAmount() {
    return amount;
}

public String getType() {
    return type;
}
```

Need more information ?

- ◉ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- ◉ <http://openjdk.java.net>
- ◉ <http://www.eclipse.org>
- ◉ <http://www.junit.org>
- ◉ <http://junit.sourceforge.net>
- ◉ <http://emma.sourceforge.net/downloads.html>
- ◉ <http://www.eclemma.org>
- ◉ Still have questions ? hamide #@# student.chalmers.se