# Lab 1. Unit testing, tests generation

## Lab goals:

1. Unit tests creation in order to test software components functionality.
2. Tests generation using JTest, C++ Test, Dot test or similar programs.

## Background

Software testing is a process for identifying software under development correctness, completeness, security and quality. Testing process consists of software execution in order to detect defects in it. The testing, in this aspect, cannot fully state that software is bug free. Testing, in this work, is software execution with some data and checking if it delivers the expected values.

For example we have the simple Java program:

|  |
| --- |
| public class Program  {  public static void main([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky)[] args)  {  int a = 5;  int b = 6;    [System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).out.println("a + b = " + add(a, b));  }    public static int add(int a, int b)  {  return a - b;  };  } |

The method “add” should add two numbers and return the addition result. But during the implementation a plus sign was misplaced with a minus sign.

During the testing, a developer executes the main method by replacing constant values and verifying if the result is ok. This is called ad-hoc testing, and the tests are not retained for further use.

Next step would be to crate for testing this programs corrected the following testing program:

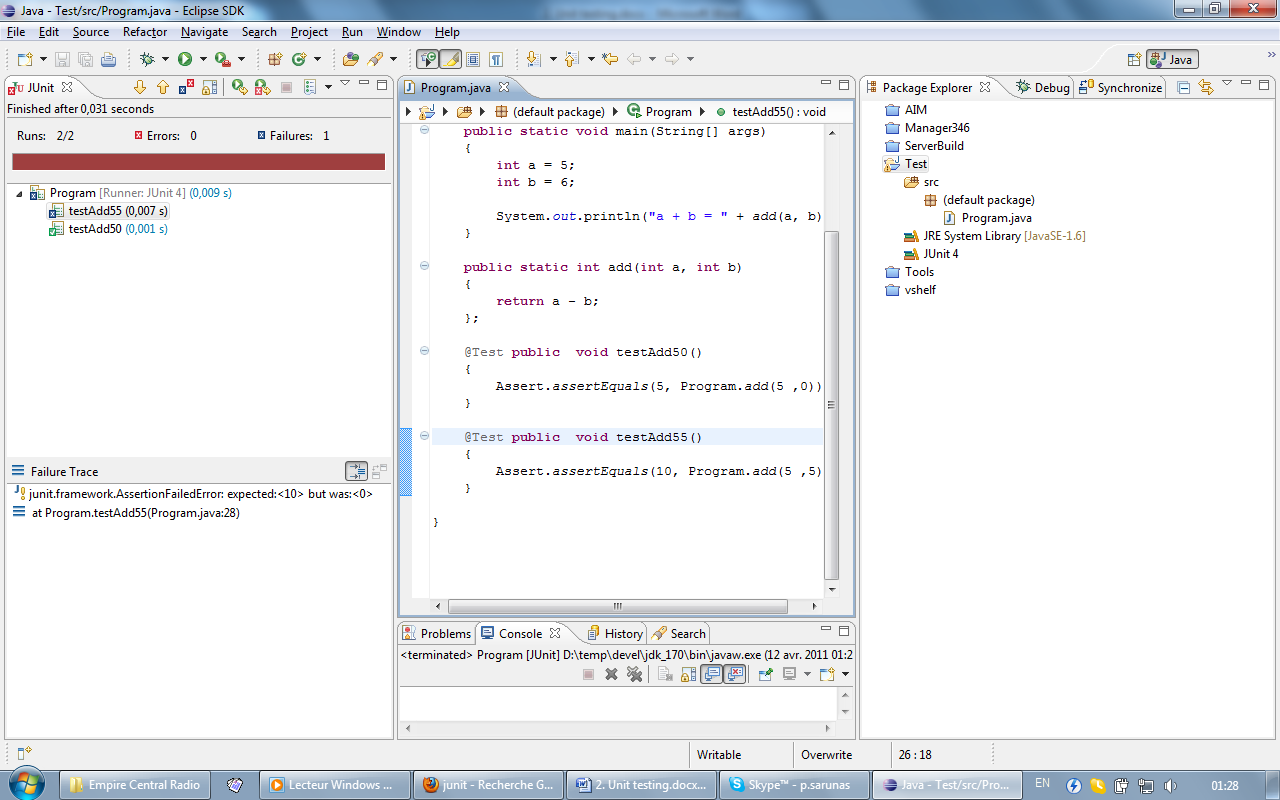
|  |
| --- |
| public class Tests extends Program  {  public static void test1()  {  if (add(5, 0) != 5)  {  [System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).out.println("Error");  }  }    public static void test2()  {  if (add(5, 5) != 10)  {  [System](http://www.google.com/search?hl=en&q=allinurl%3Asystem+java.sun.com&btnI=I%27m%20Feeling%20Lucky).out.println("Error");  }  }  public static void main([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky)[] args)  {  test1();  test2();  }  } |

In comparison methods we can see that the program compares the calculated values with in advance predefined expected values. If retuned result is equal to expected value, the defect is found in the program. If the program does not print error messages, we can assume that program is working correctly. In this case, the first test will not be able to detect the bug, thus the second test is able to detect the bug within the code.

In this example the test was created as a simple test program. Thus, it is not suitable for large sets of tests. In this case the unit testing frameworks could be used and all the checking and failures reporting is performed automatically by testing framework. One of such frameworks is JUint. The previous unit tests could be rewritten using JUint tests like this:

|  |
| --- |
| public class Tests  {  @Test public void testAdd50()  {  Assert. assertEquals (5, Program.add(5 ,0),  “failed on 5 + 0”);  }    @Test public void testAdd55()  {  Assert. assertEquals (10, Program.add(5 ,5),  “failed on 5 + 5”);  }  } |

In this case the main method is missing, and special comparison methods are used. The unit testing framework automatically discovers tests and executes them. The framework can show other criteria such as passed tests count, code coverage. The testing results are presented in the following picture.



There is a tests report on the left, in the middle there is test code and on the right is the software project under test.

In this example, the several unit tests were written for a small program. This type of testing is called unit testing - then the smallest part of software is tested. In this work several unit tests will be created and also automated tool for generating unit tests will be used.

## Lab work tasks:

1. Create software unit tests in order to evaluate software quality for chosen software.
2. Try to create tests for a whole software application (100% code coverage)
3. Evaluate software tests coverage.

## Lab work defence:

1. The student should be ready to answer to questions relating to the work.

## The questions for students to look into (defence preparation):

1. What parts of software could be unit tested?
2. What successful test means?
3. What code coverage means, what are other coverage metrics?
4. What is unit testing framework, what is its purpose?
5. Can unit testing framework be used for other types of testing?

## References:

1. Canna, J. Testing, fun? Really?: Using unit and functional tests in the development process. 2001 [cited 2006-08-25]; Available from: <http://www-128.ibm.com/developerworks/library/j-test.html>.