Assignment 3
White-box testing

Assignment Objectives:

- Generating test cases based on white box testing.
- Use JUnit for implementing the test cases**.[See Tutorial-JUnit in lab 02]**
- EclEmma for coverage analysis. **[See Tutorial EclEmma]**
- Testlink - test management tool. **[See Tutorial-Testlink in lab 02]**

Theoretical aspects

- Create test cases using White-box testing.
- Control Flow Graph
- Coverage criteria: statements, conditions/decisions, paths, loops
- References: [Myers]–chapter 4; [Naik]–chapter 4; [Young]–chapter 12; [Patton] –chapter 6,7

[Myers] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004
[Naik] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
[Young] M. Pezzand, M. Young, *Software Testing and Analysis: Process, Principles and Techniques*, John Wiley & Sons, 2008
[Patton] R. Patton, Software Testing, Sams Publishing, 2005
[TestLink1]: http://www.softwaretestinghelp.com/testlink-tutorial-1/
[TestLink tutorial]: At the Lecture homepage, http://www.cs.ubbcluj.ro/~avescan/

Assignment
[White-Box Testing]
Design and implement test cases based on source code, i.e. white-box testing. The test cases will be created for b) project requirement from the "Problem statement" in the first laboratory).
[TestLink]
a) Define the designed test cases (the ones that could be implemented and executed) in Testlink. See **Tutorial on Testlink**.
b) Define the requirements using Requirement Specification section. (if not created for Lab 02)
c) Associate the test cases and the requirements to your Test plans.

Remarks

1. Use JUnit platform for testing - JUnit 3.x/4.x;
2. Test case design must use:
   a. Control Flow Graph
   b. Cyclomatic complexity metric
   c. Individual Paths
3. For each project requirement, the test cases must be crested such that the folloswing coverage criteria will be used:
   - statement coverage;
   - condition/decision coverage;
   - paths coverage;
   - loop coverage.
4. All tables will be saved in the provided file **Lab03_WBT_Form.xls;**

5. Use EclEmma for coverage analysis.

- **Installation:**
  - **Eclipse (JUnit included).**
  - **EclEmma-Eclipse_Help_menu-Software_Updates-Available_Software-Add_Site (http://update.eclemma.org) and install it.**
  - **click-right. meniu-bar-CustomizePerspective-Commands tab -Coverage**
  - **See file *JUnit_EclEMMA.pdf* for instruction and example.**

6. Based on the report of the executed test cases, the source code will be debugged and then all the test cases will be re-executed.

7. Statistics for the executed test cases (total number of executed test cases, number of passed tests, number of failed tests), number of identified bugs (eliminated or not), and the statistics after the re-execution (after debugging).

Turn in:

[White-box testing]

1) The documentation will contain the reports **Lab03_WBT_Form.xls** in electronic format:
   a) Identification student data (name and group);
   b) Laboratory assignment title and date;
   c) Test cases tables for b) requirement using statement, condition/decision, paths, loop coverage.
   d) Statistic of the executed test cases. (last worksheet in the xls file from 1))**.**
2) Source code:
   a) Implementation of the test cases from 1)c) in JUnit.
   b) The source code - tested and debugged (and retested).

[TestLink]

1) The Testlink - TestPlan/Test (the name of the test plan will be formed using your scsId, scsId_TestPlan_WBT) Cases must be presented/delivered in class.
2) The Testlink generated documentation. See the Testlink Tutorial from Lab 02.

[EclEmma]

1)In the statistic table from the excel file, the coverage percentage of the tested method will be written.

Assignment and Delivery date:

1. Assignment date: laboratory 3
2. Delivery date (first): laboratory 4
3. Delivery date (last): laboratory 6 (-1 point for each week late delivery)