Assignment 2
Black-box testing

Assignment Objectives:

- Generating test cases based on black box testing.
- Use JUnit for implementing the test cases. **[See Tutorial-JUnit]**
- Testlink - test management tool. **[See Tutorial-Testlink]**

Theoretical aspects

- Create test cases using Black-box testing.
- Equivalence partitioning
- Boundary-value analysis
- References: [Myers]-chapter 4; [Naik]-chapter 6; [Patton]–chapter 4 and 5; [Frentiu]-chapter 3

[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
[Patton] R. Patton, Software Testing, Sams Publishing, 2005
[Frentiu] M. Frentiu, Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
[TestLink1]: http://www.softwaretestinghelp.com/testlink-tutorial-1/
[TestLink tutorial]: http://www.cs.ubbcluj.ro/~avescan/?q=node/189

Assignment

[Black-Box Testing]

Design and implement test cases based on specification, i.e. black-box testing. The test cases will be created for project requirement a) from the "Problem statement" in the first laboratory.

[TestLink]

During Lab02-first hour, please make your account in Testlink such that I will be able to assign the correct project to you. If not, you will not be able to do your assignment! See **Tutorial on 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.

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. Equivalence partitioning
   b. Boundary value analysis
3. For a) project requirement, the test cases design must include:
   - table for Equivalence Classes (EC) and the corresponding table with the created test cases based on EC;
   - table for Boundary Value Analysis (BVA) and the corresponding table with the create test case based on BVA;
   - table with all the created test cases based on EC and BVA (no redundancy). This table will be at use in the next laboratory to compare with White-box testing technique.
4. All tables will be saved in the provided file **Lab02_BBT_Form.xls**
5. Based on the report of the executed test cases, the source code will be debbugged and then all

the test cases will be reexecuted.
6. 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 reexecution (after debugging).

Turn in:
[Black-box testing]
1) The documentation will contain the report **Lab02_BBT_Form.xls** in electronic format:
   a) Identification student data (name and group);
   b) Laboratory assignment title and date;
   c) Test cases tables for requirement a), using EC and BVA. **[See Tutorial-JUnit]**
   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).
   b) The source code - tested and debugged (and retested).

[TestLink]
1) The Testlink - TestPlan/Test Cases must be presented/delivered in class.
2) The Testlink generated documentation. See the Testlink Tutorial.

Assignment and Delivery date:

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