Seminar Objectives

- Generating test cases based on black box testing.

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

Equivalence partitioning classes (EC)

No. EP	Condition	Valid EC	In-valid EC

Test cases based on EC

No. Test case	ECs	Input values	Expected result	Actual result

Assignment

1. Specify the given problems (data and preconditions, results and postconditions).
2. Create test cases based on the specification (using equivalence partitioning and boundary value analysis).
3. Implement the test cases using JUnit.

Problems

1) Verify if a number is prime.
2) Compute the maximal sequence of prime numbers from an array of natural numbers. An array X with n components is given.
  a) First example: 1 subalgorithm - Second problem from Seminar 01 about Inspection;
  b) Second example: 2 subalgorithms - First problem from Seminar 01 about Inspection;
    b.1) A subalagoritms that computes the longest sequence of prime numbers starting at position i; n elements in the vector x.
    b.2) A subalgorithm that computes the longest sequence of prime numbers from the array x with n values.

Remark: For Seminar 02 -bbt - 2)a) and 2)b.2) are the same but in Seminar 03 the source code is different.

3) Array x with n integer numbers.
a. Delete from X all negative elements.
b. Delete from X all repeated elements.
c. For a given value a find the smallest value greater than a.
d. Find all positions with the maximum value.
e. Find the longest ascending ordered subsequence.