**Module 2. HomeWork 1.**

**Task 1.**

1. Develop methods that calculate maximal value of numbers (long) where the numbers are provided by:
   * java - array of N elements, filled with different random long numbers (no duplicates);
   * java.util.ArrayList<Long>, filled with the same numbers (their Long wrappers);
   * java.util.LinkedList<Long> of the same numbers;
   * java.util.Vector<Long> of the same numbers;
   * java.util.HashSet<Long> of the same numbers;

To supply random numbers to the methods use the results of separate methods providing the numbers in the correspondent required form (i.e. converting the array of longs to the collection type needed):

generateNumbers (int N) -> long[N] or ArrayList<Long>, … (with the same numbers).

1. Develop an application that executes the method for each input supplying variant written above (with different input numbers each time ) M times, measuring the time (as precise as possible) spent to find the maximum value each time, and printing the average (mean) time for each input supplying variant.

M and N are command line arguments found in args[0] and args[1] of the main()-method).

1. Provide a conclusion about the performance estimations and comparisons depending on the containers types used.

**Task 2.**

1. Create the two arrays of long numbers, one having the length n1, another having the length n2, and have them initialized with random numbers.
2. Sort each of the two arrays in increasing order.
3. Write a method that takes the two sorted arrays and returns the third array (having the length n1 + n2), containing all the elements of the given two arrays sorted in increasing order.
4. Try to find data structures, algorithms and implementations best suited for the task execution efficiency/performance. Motivate your choice in comments.
5. Develop the program solving the same task in C++.
6. Perform a comparison of the performance estimation for your C++ and Java solutions (in terms of the execution times estimated in average). To do that, provide a program means for comparison estimations.

**Remarks.**

The results of the homework should be uploaded in Moodle prior the deadline assigned in form of archived IntellyJ Idea project, containing codes and tests with appropriate comments and docs (to keep the size of the archive small, javadocs-generation itself is not necessary needed).

For C++ staff the source file(s) of the program should be provided with the estimations results produced. It should be placed in separate directory of the Idea project named **natives**. The runnable .exe file that prints its results into console must be provided.