A Terms of cooperation

Obtaining a credit

Each task has a **first deadline** specified in its title. A student may obtain maximum score for solution to an assignment in case he/she presents it during the laboratory class on the first deadline at the latest. Assignments are, in general, evaluated for maximum 10 points. Solutions to some specific assignments may be assessed for more points, which is discussed in the text of the particular tasks.

The **second and very last opportunity to present a solution to an assignment** is the second deadline – i.e. laboratory class next to the first deadline class. Defending a solution to an assignment during a **second deadline** – i.e. laboratory class next to the first deadline class – entitles a student to obtain 50% of the maximum original score at the most.

Evaluation of student solutions

While presenting a solution to an assignment a student should **demonstrate that he/she is aware of its internal architecture** (the applied technologies and combining them or used algorithms).

During the presentation a student should be prepared to answer some control question asked by the person who evaluates the solution. In case the person assessing the solution is not convinced of student's authorship, the **student gets 0 points for the solution and loses any further opportunity to defend the solution to the given assignment**.

The above statement implies that a student **may fail in getting a credit** in case he/she attempts to defend a solution to a mandatory task (see below) and the person evaluating the solution comes to conclusion that the student has not authored the presented solution.

Robustes of student solutions

The presented solutions should validate the input data provided by the user. **The undesired behaviours which entail reducing of awarded numer of points during assessment include i.a.:**

- no verification whether a user passed a valid file path or whether file exists;
- missing validation of the input format – in case the format is known in advance (e.g. format of postal code);
- missing verification, whether value of a reference variable/field was set (i.e. avoiding of NullPointerException);
- missing verification whether value of array index does not exceed the actual size of the array/list (ArrayIndexOutOfBoundsException).

Evaluation

- 3.0 – 25÷35 points
- 3.5 – 36÷55 points
- 4.0 – 56÷85 points
- 4.5 – 86÷120 points
- 5.0 – 121+ points

Minimum requirements for getting credit (mandatory tasks)

To get a credit you need to receive at least 5 points for each of the assignments concerning the following topics:

1. Non-blocking I/O **(assignment 1)**
2. Client/Server based on NIO channels **(assignment 2)**
3. Web Applications – introduction **(assignment 3)**
4. either (1) Remote Method Invocation **(assignment 8)** – or alternatively (2) Java Messaging Service **(assignment 9)**
5. Web Services – introduction **(assignment 10)**

1 Non-blocking I/O – shared mapped file (11th March, 12th March, 13th March)

Write simple two applications (it may be one application which operates in two modes) which communicate with each other by saving and reading data from a shared file mapped with into memory.

One of the application should save some data – e.g. two integer values. The other application should retrieve the data and perform some operation – e.g. adding the values read – and then print out the result to the console.

NOTE: Your solution should be based on non-blocking I/O implementation.