About the miniproject

- version 1.0 -

Your miniproject should reflect your ability to practically develop a Java application. You are required to develop either a standalone Java application or an applet of your choice.
The amount of time you should dedicate to this task should not exceed cca 40 hours (incl. work time in the laboratory).

1. Choice of Development Task

Your choice for the miniproject should be a reasonably complex (rather simple) program/applet.
Here are some suggestions:

- The mill game (Romanian: ţintar)
- Chinese checkers
- Support for chess – representing and validating moves and visualization of the board (the user must be able to select pieces and move then like on a real board, but he should not be allowed to execute illegal moves)
- Labyrinth traversals by a robot
- Simulation of a small airport with one runway, an airport building, airplane parking area for at most 4 equal-sized planes and a control tower
- Simulation of a radar screen for an airport and of the dialog flight controller – pilot for at most 20 simultaneous targets on the radar screen
- Simulation of the behavior of a digital automaton

2. Steps to Follow

We suggest you to follow these steps:

1. Make a cca 1 page document describing the essential elements of the problem.
2. Discover the necessary classes using CRC cards (Class Responsibilities Collaborators)
3. Draw a preliminary class diagram
4. Sketch the algorithms which must be implemented by your methods in pseudocode.
5. Implement, debug and test your solution
6. Finalize the documentation and create a .jar file containing your project deliverables

3. Miniproject Deliverables

When you present this project you must deliver a .jar file named according to the following rule:

<group_id>_<your_last_name>_<your_first_name>**.jar**

where:

- < group_id> is the number of the group you're a member. E.g. 3219
- < your_last_name > is your family name. E.g. Doe
- < your_first_name > is your first name. E.g. Jane

Thus, a student by the name of Jane Doe, a member of the group 3219 will deliver her project in a file named `3219_Doe_Jane.jar`

Contents of the .jar archive:

- the package(s) containing your program incl. source ocde and `javadoc` generated documentation.

- A document edited with Microsoft Word, and saved in `.rtf` format, containing the task specification (written during the first step) and a class diagram (and object diagrams if need) for your program

4. Deadline

You should submit/present your project during the last laboratory class.

5. Evaluation Criteria

When evaluating your work we will stick to the following criteria:

- Quality and clarity of your design
- Fidelity of the implementation
- Implementation correctness
- Implementation robustness/reliability
- Problem complexity
- Originality of the underlying idea
- Quality of the documentation
- Use of the recommended programming style

6. A Final Note

Copying the design/implementation of someone else or presenting downloaded / acquired by other means code as personal work will be penalized accordingly.