Individual assignments - General description

For this year's 2ID90 course students have to hand in two individual assignments. The main intention of these assignments is to give additional opportunities to have hands-on experience with the material. Apart from this we hope that theory instuctions will become more lively and better visited. To further stimulate that, the individual assignments count for part of the final grade.

The two individual assignments have the following common characteristics:

- **individual**: all students hand in their own result;
- **short**: turnaround time of approximately a week;
- **small**: to give an indication, the program should take less than a 150 lines of code (including mandatory documentation) ;
- **automatically graded**: Momotor will grade your work.

Next to the detailed descriptions of the individual assignments, there are a few general remarks.

- You are required to write clear documentation for your code artifacts: loops, variables, methods, method parameters, ...
- This documentation should form approximately half of your code.
- Based on the *quality* and *quantity* of this documentation your automatically computed grade may be *reduced.*

Approach

You are advised to follow an approach similar to the one described below.

1. download, unpack and open a given netbeans project;
2. work at your algorithm, documentation and/or tests;
3. perform JUnit tests;
4. if not satisfied with test results, continue at 2;
5. hand in into Momotor;
6. if not satisfied with the computed grade, continue at 2.

JUnit

JUnit is a framework to write repeatable tests. For these individual assignments we recommend you to write your own tests (examples are given). Good choices of tests help you to make correct implementations more quickly.

Momotor

Momotor is a plugin in Canvas. Given correct input Momotor will perform the following tasks:

- Run its tests on the algorithm; this may actually take some time.
- Determine and show a grade, based on the (hidden!) results of the tests. The test results are hidden on purpose. We ask you to write a simple algorithm, and challenge you to think it through completely to find possible mistakes in your implementation. Of course, you can write your own tests in JUnit.
- Output an error message.