Object-Oriented Programming

Appendix to the Project Statement

Academic Year 2018-2019

In order to answer early questions about the project and ease your task, we provide you the following guidelines for both implementing and submitting the project.

Solution display In order to display your solution in a terminal, we suggest you to use the following display convention. Upon solving the problem, your program should write 32 lines in the terminal, each line consisting in a position in the puzzle, a type of element and the orientation of said element. For instance, here are the first four lines of a possible solution.

```
Position 1 - Element 11 - Orientation 0
Position 2 - Element 1 - Orientation 2
Position 3 - Element 1 - Orientation 0
Position 4 - Element 2 - Orientation 1
```

Position numbers, element types (and their associated numbers) and orientation numbers are depicted in PDF figures you can get by downloading the `project_basis.zip` archive on the exercise sessions website.[1] If you apply this display convention, we also advise you to strictly provide the same display as in our example (i.e., uppercase letters and spacing should be the same).

You are free to choose another type of display if you wish to, as long as you provide us some details on how to interpret it (see submission rules).

Program execution Your `main()` method should be located in a `SoccerBall` class, which can be simply executed in a terminal by running the following command.

```
java SoccerBall
```

Libraries and side features You are free to use any relevant class from the Java library or even add additional features to your solution. However, no extra point will be given for any feature that wasn't part of the initial statement.

Evaluation criteria The evaluation of the project will take into account the fact that the program compiles successfully and is functionally correct, but most importantly, your project will be graded with respect to how it puts object-oriented mechanisms to (good) use. The overall structure and readability of your solution will be taken into account as well. Note also that the performance of your solution will only be taken into account if your program is too slow to compute a valid assembly.

To avoid bad surprises with compilation, we suggest you to check that your project compiles and works properly on the university computers (a.k.a. *"Network 8"*).

[1] `http://www.run.montefiore.ulg.ac.be/~grailet/INFO0062.php`

Submission rules Your solution must be submitted as a ZIP archive containing your `.java` source files. Your archive should not contain any subfolder, except if you use packages (in which case, the structure of your archive must match your packages).

However, if you use other conventions to display your solution in the terminal, you can provide one additional `README` file, as a text file (`.txt`) or PDF file, which describes how your output is formatted and how we can use it to verify that your assembly is valid. You can also provide a `README` file if your program implements a particular feature which requires some commentary.

Any file other than `.java` and the optional `README` will be ignored (this includes project files that are typically produced by IDEs such as Eclipse).

Your archive must also comply to the naming conventions described below. If you do the project alone, name your archive

```
oop_lastname_firstname.zip
```

where `lastname` and `firstname` have been respectively replaced by your last and first names (only the first letter of each name being in uppercase). If you do your project with a classmate, name your archive

```
oop_lastname1_lastname2.zip
```

where `lastname1` and `lastname2` correspond to your respective last names, given in alphabetical order (only the first letter of each name being in uppercase).

In both cases, your archive must be sent for April 22th to `oop@montefiore.ulg.ac.be`. If you do the project on your own, the title of the e-mail must be

```
OOP - 2019 - lastname firstname
```

with the same replacements as in the name of the archive. Likewise, if you do the project with a classmate, the title must be the following.

```
OOP - 2019 - lastname1 lastname2
```