Software Architectures

Assignment 2: Expression Problem Applied

Assistants: Humberto Rodriguez Avila, Kennedy Kambona
Email: {rhumbert, kkambona}@vub.be
Office: {10F719, 10F732}

Deadline: January 14th, 2018, 23:59

Description

For this assignment, you need to extend the *Cook* class of the project *used* in the first assignment[1]. The current implementation of this class can only prepare orders related to *eggs*. Using the *Visitor*[2] design pattern, you will extend the *Cook* class to support the preparation of multiple and different orders. The new version of the *Cook* class must support receiving a *list* of orders to prepare, where each order contains information about its *type* and *style*. For example: (Egg, SunnySideUp), (Bacon, American), (Waffle, Brussels). In the previous examples the first element is the *type* of the order, while the second one is the *style*. Implement and extend the required abstractions to support Egg, Bacon, and Waffle orders. Consider the following styles:

- **Bacon:** American, Candied, Applewood
- **Waffle:** Brussels, Liege

Implement and update the Unit Tests related with the new requirements.

Deliverables A short *report* (in English) explaining the new solution to the problem, and where you compare it to the original code. The report *file* should follow the naming schema `firstname_lastname_SA2.pdf`, and it should be handed in as a PDF file. For example: `Humberto_Rodriguez_SA2.pdf`.

Submit the *report* and *source code* of your solution as a single *ZIP* file on the Software Architectures course page[3] in PointCarré, by clicking on *Assignments (Opdrachten) > Assignment 2*.

Grading The assignment will be graded and can become subject of an additional defense.

[1] You can use your solution to assignment 1 or the original version of the *DDDOnionCookExample* available on PointCarré.

[2] See lecture

[3] Use the English variant "Software Architecture**s**", rather than the Dutch one "Software Architecture**n**".

Notes

- Team work is not allowed.
- The solution to this assignment can be implemented in *Scala* or in *Java*. In case you choose Java, you have translate the original solution to Java yourself.
- Copying – from other students, or from the internet – will be considered as plagiarism and be reported to the faculty.
- If you use any other resources besides those provided in the lectures and in this document, remember to cite them in your report.