1DV607 Workshop 2, Peer review

Domain model author: Marcus Rosén

Peer review author: Michael Racette Olsen

Running the application

Following the provided instructions the jar file executed as expected, the program provided the user with adequate information to understand and navigate the menus. All the required functionalities were implemented and seem to work without problems.

Minor notations

In terms of invalid input, some parts of the program lacks describing/helpful error messages. For example if the user enters a number outside of the valid range when choosing member or boat, the view prints “Index: <input-1>, Size: <length>”. If the author want to keep the message format, I suggest changing the Index to match the input and length to reflect the visible lists length and not the implementation.

Architecture

The author has structured his project into three packages *model, view* and *controller*. As far as I could tell the implementation of the classes followed through with the chosen pattern. No dependencies from model to view nor model to controller were found.

Also the domain rules have been properly placed, for example the rules for validating input is contained within the model. Making the rules consistent independent of the View and Controller.

Quality of implementation / source code

The author has followed standard java naming conventions and added Javadoc comments to his methods providing information about parameters and returned data.[i]

However some of the longer methods and generated code clauses could benefit from some comments.

Quality of design

The classes have been properly divided depending on their responsibilities and are coupled together through data coupling and subclass coupling.

Class members have been encapsulated within get and set methods.

Quality of diagrams

Class diagram

The provided class diagram contains information about the *classifier*, *class name* and *class path.* The model-view separation can be seen in *class path.* This could however be more apparent by structuring the classes into frames. (Model, View and Controller).

At a first impression the class diagram look rather clustered, this is mainly caused by the alignment of the association and dependency arrows. Also the relations between classes is not clear.

Suggestion for improvement

- Try to align the associations with straight lines and avoid crossing over boxes.
- Add a frame for View, Model and Controller and structure the classes into them.
- Add important methods and attributes.

Sequence diagrams

When it comes to the sequence diagrams, they give a much better impression. They contain relevant information in a nicely structured fashion, giving a much better understanding of the domain model compared to the class diagram.

Do you think the design/implementation has passed the grade 2 criteria?

The design and implementation look really good as well as the sequence diagrams, however the class diagram is a bit lacking. By spending some time and improving the class diagram, I definitely believe the author satisfies the requirements for grade 2.

---

[i] http://www.oracle.com/technetwork/java/codeconventions-135099.html