

Peer review on Robin Reijo's group's workshop 2

First and foremost I would like to point out:

- The application seems to be a bit from done, however this is not pointed out anywhere.
- The code lacks comments completely (except for code that has been commented out, which is a lot).
- There is an enormous amount of dead code throughout the entire application.
- Since the class Console is not divided into more than one method this becomes extremely long and hard to understand.
- Are the classes workshop2Test.java and DataCollector.java not used? They are not included in the class diagram but they are part of the project.
- There is a huge amount of locations where there are blank lines between every line of code which makes it much harder to read and unnecessarily long.
- Duplication of code is very common in the class Console and in the classes Boat and Member it seems as if there has been work done to make "better" methods of, for example the unique id of every member, which results in that there are several methods for accomplishing the exact same thing.

Running the application

The program does not have any error handling which means that any "wrong" input makes the program crash.

Another bug is that when a boat is removed, it is only the boat type that is removed from the verbose list, and the compact list still list the total number of boats as if there was no boat removed.

The program has an option in the menu named "save and exit", however I could not get it to save any information into the text files.

The length of the boats registered are never displayed and as they are stored as integers and in meters there is no possibility of entering, for example 3.5 meters, perhaps a double would be better or entering the length in cm.

When searching for members and selecting personal number nothing is shown.

You can only enter one word as name of member, meaning no surname is possible.

Disregarding the comments about bugs, the application works and runs. The messages to the user is clear but could use a little bit more work. The menu is easy enough to understand and makes navigation through the application easy.

Design and implementation:

In the code there is a lot of use of static variables and operations, aswell as importing classes and making static references to them making the imports unused.

Using limitless for-loops is not a very pretty solution for keeping things running.

Perhaps it would be a better option to update the files directly when a change has been made so that data is not lost, and now you need to save and exit for changes to be stored (if it was working) and that may not be the best solution.

Somewhat complicated and long code for reading and writing to/from files.

As is, the program uses several lists of string or integer to store names and ID's and so on. A better solution would be to make a list of type Member and Boat to avoid having several different lists.

The responsibilities of both Console, Member and Boat might be a bit too big and a class handling the registry might be a good idea.

Naming is generally good but sometimes some variables are named userInput1 and so on which might need a better name.

However, the program has a model-view-separation and the GRASP principles are kept in mind. The application is object oriented and the view contains no application logic. The design of the application is in line with model-view-separation and object oriented design when viewing the guidelines by Larman (1).

The unique ID assigned to each member has been solved in an easy and well working way, meaning every user has a unique ID.

The biggest weakness of the implementation is that there is no error handling and no saving of the registry to file. The huge amount of dead code, duplicate code and out-commented code needs addressing aswell.

The biggest upside to the implementation is the model-view-separation. The diagrams are helpful but as they contain a lot of dead code it makes it much harder to understand the application.

By: Sebastian Airisniemi (sa223gd)

Source: <https://github.com/RobinTestR/ForTheUMLCourse>

References

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062