Peer-Review WS2-G2

General

- (+) Easy understanding menus.
- (-) No error handling. The program crashes when you type a letter in the menu.
- (-) Not so many comments.

Diagram

Class diagram

- (-) Both User and Admin has dependencies to every boat type.
- (-) Boat has no associations.
- (-) Every boat type has association to Boat.
- (-) A multiplicity at target end, but not the source end should be added.[1, Chapter 16.4]

Sequence diagram

Output: Compact list.

- (-) memberDisplay is a method with parameters. You should include it in the diagram.
- (-) printCompactlist method is not in the sequence diagram.
- (-) A loop should be added around getName, getMemberID, getNumberOfBoats, since you iterate over all member.

Input: AddBoat.

- (-) addBoat method is from User and not from Interface. And it has parameters. Otherwise it's fine!

Architecture

(+) Model, view and controller separation is achieved in the right way.
(+) The model is not dependent on the UI (view and controller)

OO-Design and GRASP

- (+) Great model view controller separation.

Code

- (-) Some variables have bad names, for example a reference to the Admin class is named “a”.
- (-) No error handling. You can type letters when you choose length and you can type numbers when choosing name.
- (-) Maybe use enums instead of classes for boat types.
- Why is there a file that is responsible for the number of kayaks etc? It is only written to the file but not used anywhere else in the code.

Our Thoughts

Overall we think it mostly fulfills the requirements, except the error handling part. The program should not crash when the user types e.g. letters where it should be integers. When this is fixed, we think it’s enough for passing grade 2.

Reference section:

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