Object Oriented Analysis and Design

Workshop2: Design

Haofei Yan(hy222ap), Huan Rong(hr222dx)

11 October 2015

Peer View for Ludvig Magnusson(lm222ix)'s team

Test runnable version

Bugs

1. When we modify a boat or a member, it would be good to have the possibility to not modify something. IE: I want to modify the name of the user. I wouldn't have to enter the personal number again.

2. The naming function didn't check whether the name was number or alphabet. And the personal number didn't check right or wrong. [1]

3. When we typed in name with blank, it showed "unknown command". But this member has added into database without personal number.

4. A weird thing happened when we continued to add a boat to a member, whose name was wrong with blank as above.

```
AddMember
Full name:
haofei yan
Personal Number:
1234
Unknown command.
VerboseList

ID:2 - a - 1234 - Boats: 1
a's Boats:
ID: 3 - Type: eee - Length: 4.0
ID:3 - b - 1234 - Boats: 1
b's Boats:
ID: 2 - Type: www - Length: 3.0
ID:4 - haofei - yan - Boats: 0
haofei's Boats:
AddMember
Full name:
haofei yan
Personal Number:
123456
Unknown command.
```

```
addboat
For which member you like to add a boat
0 ('a', '1234')
1 ('b', '1234')
2 ('haofei', 'yan')
3 ('haofei', 'yan')
2
Boat type:
ooooooooo
Length:
5
VerboseList

ID:2 - a - 1234 - Boats: 1
a's Boats:
ID: 3 - Type: eee - Length: 4.0
ID:3 - b - 1234 - Boats: 1
b's Boats:
ID: 2 - Type: www - Length: 3.0
ID:4 - haofei - yan - Boats: 1
haofei's Boats:
ID: 4 - Type: ooooooooo - Length: 5.0
ID:5 - haofei - yan - Boats: 1
haofei's Boats:
ID: 5 - Type: ooooooooo - Length: 5.0
```

The ship would be added to both users with same name.

5. When we failed to add member, the next member ID would be occupied by this failed action. For example, from beginning we added members successfully 4 times, and failed 1 time, then the next ID would be 6 instead of 5.

6. About add boat function, we have to choose member from Number Zero. It would have exception if we typed in the wrong data and the program stopped.

Architecture

1. There is a model view separation. [1] But in controller, there is too many System.out.println("err message") where it should be only in the UI.
2. The console should be in charge of output and input. Usually a method of the console is called by the controller, and the console return the entry. The same for output. A console's method should be called with in parameter if some data should be showed. Console should have no error if controller and model are deleted.

   Controller should just transmit data between the view and the model. IE: controller ask the view to know which option has been selected, the view return the selection, the controller call the model to perform the option selected. Send to the view if some data have been send back by the model and have to be shown.

   The model shouldn't have any error if the controller and view are deleted. It should just perform what is asked by the controller, or send the data asked by the controller. You can have as in your code like a hierarchy with the model. [3]

Code

The code has high-standard quality. Naming is good. There is no duplication. Everything seems perfect for me. The requirement of a unique member id is correctly done with database count.

Design

I think everything meets the requirement of Peer Review Instructions Workshop 2 Design. [4]

Class diagram

The implementation and diagrams conform. And you don't need to add every single attribute or operation to class diagram. [3]

Sequence diagram Create Member

Use the correct function's name, the same as in the code. It should really represent the final code. [5]

Sequence diagram Verbose List

1. The diagram doesn't correspond to the code. The database returns string to ClubRegistry firstly. Secondly, the ClubRegistry returns string to UI. Finally, UI showed the string to User.
2. Use the correct function's name, the same as in the code. It should really represent the final code. [5]

Questions

As a developer would the diagrams help you and why/why not?

Yes, it helps. But there are some problems with sequence diagrams. Class diagram contains too much information.

What are the strong points of the design/implementation, what do you think is really good and why?

It's nice to write code with java.sql.*. This library supplies many methods and you won't lose data when some errs happened with database.

What are the weaknesses of the design/implementation, what do you think should be changed and why?

The design of view and controller need to fix. They have too much work there.

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

Yes, if the view and controller part can be fixed.

Bibliography

1. https://en.wikipedia.org/wiki/Personal_identity_number_(Sweden)

2. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, Ch 13.7

3. The peer view Guillaume Fumeaux sent to me. I think his summary is right and clear to understand.

4.https://docs.google.com/document/d/1VDoSeyT_qwuXiQtbI_oSLadr1QQRMf5zGl_R6-MCMNo/edit

5. http://www.visual-paradigm.com/VPGallery/diagrams/Sequence.html