Peer Review Workshop 3

For the domain model made by Markus Alshraydeh (ma223ik) and Daniel Hammerin (dh222gh)

Daniel Vedin (dv222bk)
2015-10-27

Application

I could not get your code to run. I am not a java developer so I had some hard time even getting the code to compile at all in Eclipse. When everything seemed to work and I got the interface up and running, an error was spotted in Subject.java:

```
Syntax error on token(s), misplaced construct(s)
          Syntax error on token ":", EnhancedForStatementHeaderInit
expected after this token
```

This is for line 23. Also on the subject on this class, the “unregister” method is empty.

Since I cannot run the code I cannot comment on how it works. I can comment on the code itself though.

Implementation and Design

I found the class diagram very confusing to read, mainly because everything is so cramped up together. The lack of package folders in the diagram also hurts readability in this case. I also see a lot of different arrows which appears to try to show the same type of connection. For example, I do not understand why an extension of an interface gets a green striped arrow (PlayGame > IObserver) and other extensions get a black arrow (Dealer > Player)

I also feel that the create arrows do not really impact the diagram in a good way. Sometimes they show dependency and sometimes they are redundant since the classes are connected in some other way which trumps simple dependency. A good example of this is Game > Player.

InputChoice is incorrectly placed in the diagram. It should be connected to IView and PlayGame should also have a dependency towards it. Also on the subject of InputChoice, I can understand the decision of using single letters that correspond to the actual entered input from the user. However, what happens if these letters change? I would have preferred describing words such as “Start” and “Hit” here.

Missing dependency between Dealer and RulesFactory.

Missing dependency between ISubject and IObsever.

Controller > View dependency

The controller view dependency is taken care of, but take note of my suggestions above considering enum naming. Also note that the error message you are printing in “SwedishView” is written in English and not Swedish.

Strategy Pattern

The implementation of the observer pattern is well implemented as Larman describes the pattern [1, p447].

I do however have some comments on this subject.

The SoftSeventeen class is wrong. What it does right now is check if the hand contains an ace and makes the dealer hit if the total hand value is below 17. What the code does not consider is that the ace could be counted as 1. This means that if the dealer has the following hand: (Ace(1), Ten(10), Six(6)), the dealer will take another card. The whole point of the Soft 17 rule is to catch hands such as (Ace(11), Three(3), Three(3)) , since taking another card here makes sense.

I also feel that the “IPlayerWinsOnEqual” is wrong. The whole point of this was to create an interface and then two rules using that interface, one where the player wins on equal and one where the dealer wins. Naming the interface the same as the rule is basically just creating an unnecessary file since no other class will ever use it. I advise you to add the missing class and to rename the interface.

Duplicate code

The duplicated code problem is not fixed. The code still resides in the dealer class.

Observer Pattern

Here is another case of you creating an interface for a single class to use. No other class but “Subject” will ever use the “ISubject” interface. I feel you could have skipped the Subject.java file altogether and just made Player extend ISubject directly.

Other than that, it seems to me that the implementation of the observer pattern is well implemented as Larman describes the pattern [1, p465].

Grade

Based on all the problems I’ve described above, I cannot say you have passed grade 2. Fix the above however and you will pass grade 2.

References:

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