

Daniel Vedin

Peer review by Roy Nilsson, Mikael Eriksson and Oskar Klintrot

Try to compile/use the source code provided. Can you get it up and running? Is anything problematic?

The code is up and running.

Test the runnable version of the application in a realistic way. Note any problems/bugs.

No bugs, everything works as expected.

Does the implementation and diagrams conform (do they show the same thing)? Are there any missing relations? Relations in the wrong direction?

Class diagram: The only flaw is that the dependency between `Dealer` and `Card` is not shown, otherwise it seems to be up to date with the code.

Sequence diagram: In `Dealer.Stand` method it is an `foreach` loop used that is unnecessary because it uses already in `showhand()`.

Is the dependency between controller and view handled? How? Good? Bad?

Dependency is removed and implemented in the `Iview` instead with `enum`, It's good because it removes dependencies.

Is the Strategy Pattern used correctly for the rule variant Soft17?

The Strategy pattern is used correctly regarding the soft 17 rule since the strategy is in an own class which shares a common interface (1, p. 447). However when testing the rule doesn't work as it should. If we hand the dealer e.g. three of clubs, three of hearts and ace of hearts he does not continue the game. We think this is because he use the method `GetValue` to calculate the score. `GetValue` does not provide the score of the given card, only the cards place in the order of the deck. For example, Two = 0, Three = 1 and so on, which means Ace is given the value 12 but has the score value of 11.

Is the Strategy Pattern used correctly for the variations of who wins the game?

The application follows the strategy pattern according to section 26.7 in the book (1, p.447).

Is the duplicate code removed from everywhere and put in a place that does not add any dependencies (What class already knows about cards and the deck)? Are interfaces updated to reflect the change?

The duplicated code could be changed in the different game rules also. Use your `NewCard()` method in the different game rules.

Is the Observer Pattern correctly implemented?

There can only be one observer at the time, there's no option for adding several observers (1, p.469). The cards doesn't pause when dealer starts the game, only when a new card is given.

Miscellaneous

`System.Threading.Thread.Sleep(500);` is in the controller, low-level code should be handled in the view. Larman says that "the View is the UI Layer, and the Controllers are the workflow objects in the Application layer" (1, p.209).

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

Only minor changes can be done so we think the implementation has passed the grade.

References

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