

Peer review of Sebastian Airisnemis Workshop 3

by Linda Ott Olander

I downloaded NetBeans with JDK and created a new project, and added the files to that. When I try to run the program, I get the message "Project does not have a main class set". And there are no main classes found according to NetBeans, so I'm unable to test the program.

Does the implementation and diagrams conform

I've been reviewing the code and the diagram, and they seem to conform.

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

No, the controller class PlayGame still seems to get input from the view, so it is still dependent on the view. I think this needs to be changed. Perhaps the input could be stored in the model.

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

Yes, the Soft17 class implements the interface IHitStrategy. According to Larman [1, section 26.7], the strategy pattern should be implemented by creating multiple classes, each with a polymorphic method. Each algorithm should be defined in a separate class, with a common interface. This is how this project is implemented so the Strategy pattern is used correctly.

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

There is an interface called IWinner with a method IsWinner(). This interface is implemented according to [1, section 26.7] by classes IsDealerWinner and IsPlayerWinner. But the classes seem to be exactly the same so I don't know if the strategy pattern is used correctly.

What class already knows about cards and the deck?

According to the old class diagram, there is no class that knows about both the card and the deck. The Player class knows about the Card, and the Dealer which is inherited from the Player, knows about the Deck. So I suppose the Player could be said to know both about the Player and the Dealer, but it's not crystal clear.

Is the duplicate code removed from everywhere and put in a place that does not add any dependencies?

The duplicate code (Card c = m_deck.GetCard(); etc) is removed from the Hit() method in the Dealer class.

Is the Observer Pattern correctly implemented?

In the Observer pattern, one should define a "subscriber" or "listener" interface. [1, section 13.7]. Subscribers implement this interface, and the publisher registers subscribers who are interested in an event and notifies them when an event occurs.

There seems to be an observer pattern implemented in the code, as the class PlayGame implements the interface BlackJackObserver. I don't think it's completely implemented.

Is the class diagram updated to reflect the changes?

Yes.

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

No, I think it needs some additions in the code to fill the requirements.

References:

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