Object Oriented Programming

Report for Hw2

資工一 B00902064 宋昊恩

1. Implementation:

I use the similar way as teacher's and TAs' suggestion to divide the required work into several classes, such as `Card.java`, `Checker.java`, `Computer.java`, `Player.java`, `POOCasino.java`, `Shuffler.java`.

The `POOCasino.java` is used to create a computer of Class Computer to hold up the whole game.

The `Computer.java` is used to create a shared checker of Class Checker, shuffler of Class Shuffler and a player of Class Player. Besides, it will keep holding the game until the player want to quit the game. Every round, it will ask the player to get the bet, give the player five cards and ask him whether he want to repick some of them or not. After that, it will ask the player to judge the prize that he should get.

The `Player.java` is used to serve the certain player in my designation. It will first get the checker and shuffler which are given from the computer. The functions in it include `getBet()`, `showCard()`, `getCard()`, `repick()`, and judge the prize it should give the human player.

The `Checker.java` is used only for analyze the biggest card type that the player has on hand. I use different functions to judge different card type. And, if none of them work, I will return “others” as result.

The `Shuffler.java` is used to shuffle all the cards and it will return a card of Class Card randomly every time you call the function `pickCard()`.

Every human player can play this game by simply start up the `POOCasino.class`.

2. Correctness:

First, I used various functions to test the card set whether it belongs the certain card type or not. In this case, it is much easier and clearer to achieve the requirement without making any blunder mistake.

Second, I check various card type many times for correctness. Furthermore, I did a lot of precaution to prevent user to insert illegal command.

I believe the methods that I mentioned above can effectively increase the correctness of my work.

3. Bonus part:

I do some optimization on the user interface, such as, my program can tolerate more stupid operations. For example, when you ask for more than 5 dollars as bet or you require to gamble in debt, you will get refused and ask you to have a second thought. Or, when the computer ask you which card you want to keep on hand, you do some stupid or insane requirement, you will be refused, too. By the way, I even add a further option for user to choose, if you don't want to keep any of your cards on hand.

I add lots of notes and create the `javadoc` file for easier recognition.

4. Some of my thought:

Homework 2 is quite interesting to me, so I start writing this work pretty early.

Honestly speaking, I don't know how to divide this homework into object properly at the beginning. Initially, I want to cancel the object `Class Player,` because I am confused about the function of object `Class Computer` and `Class Player.` However, I later found out that if this work requires to accommodate more than one player, then, the `Class Player` is obviously needed.

The second thing which confused me is that I don't know whether I should share the shuffler or checker or not. In fact, I can give every player both of them simultaneously. I found a great solution for it later on. I create the shared checker and shuffler when computer is created. Then, I will pass them to each player as the parameter. And, this saves a lot of resources.

The third, and also, the last problem that I met is I don't know how to import a .class file properly. I check a lot of information from the website. Finally, I know that I must implement the real path of the package. Furthermore, I must use `-classpath` to fulfill the the requirement of compilation and execution. I also learned how to link to multi-classpath and so on.

I did learn a lot from this work! Thanks!

P.S.

I sent an email to ask when the submission place on CEIBA will open, but no one answered. On top of that, I forgot the deadline of Hw2 was on Thursday and not on Friday. This is the reason why I began and finished this work very early but I still have to use a medal...

5. Reference:

The whole program is implemented by my own self. I search for the concept of “import”, “package”, “randomization”, etc. from the google website.