**FOOP HW2- POOCasino: Jacks or Better**

B01902039 施楷文

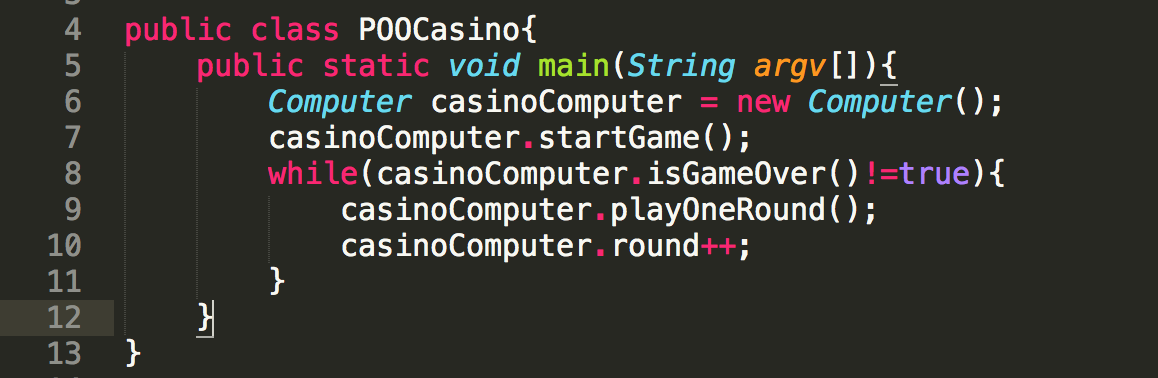
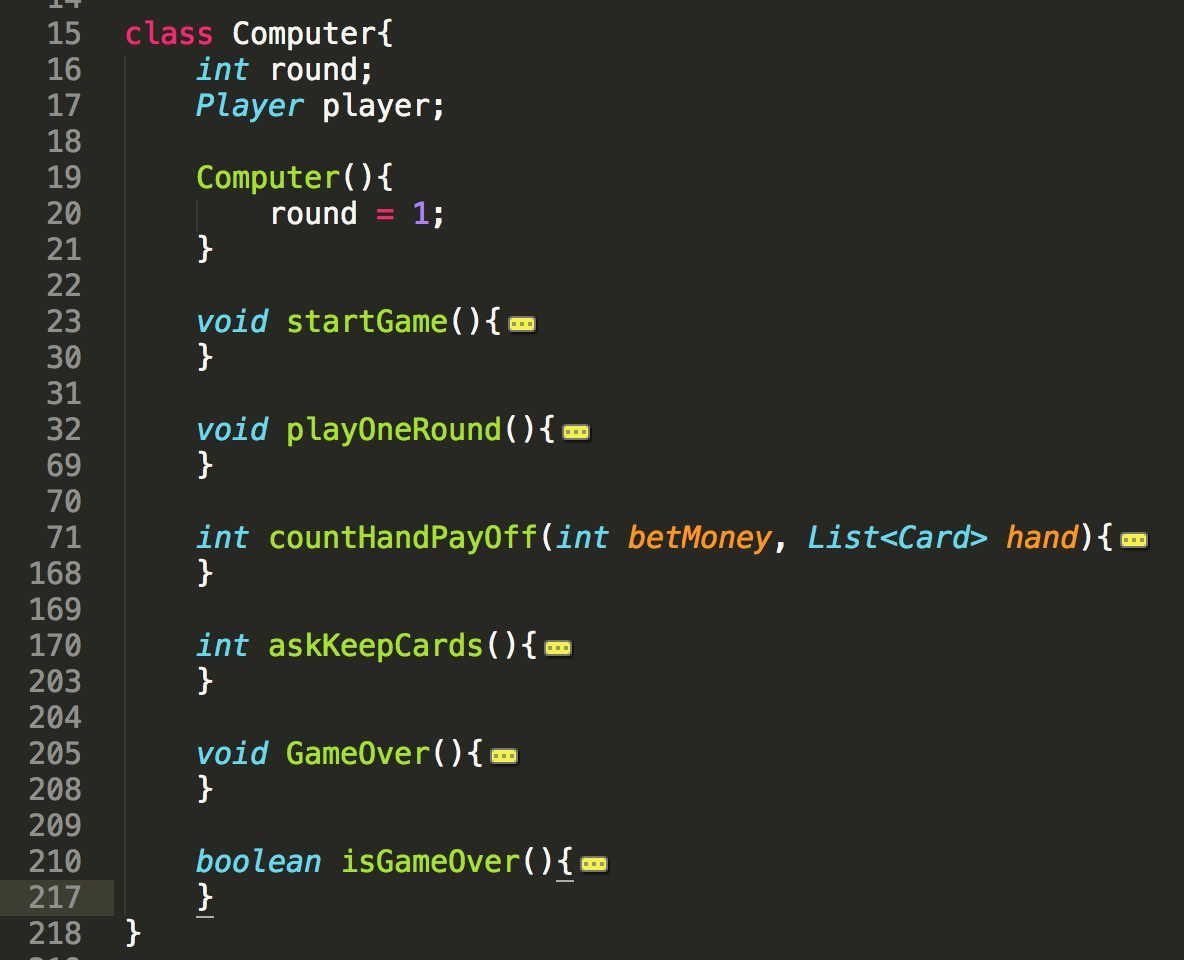
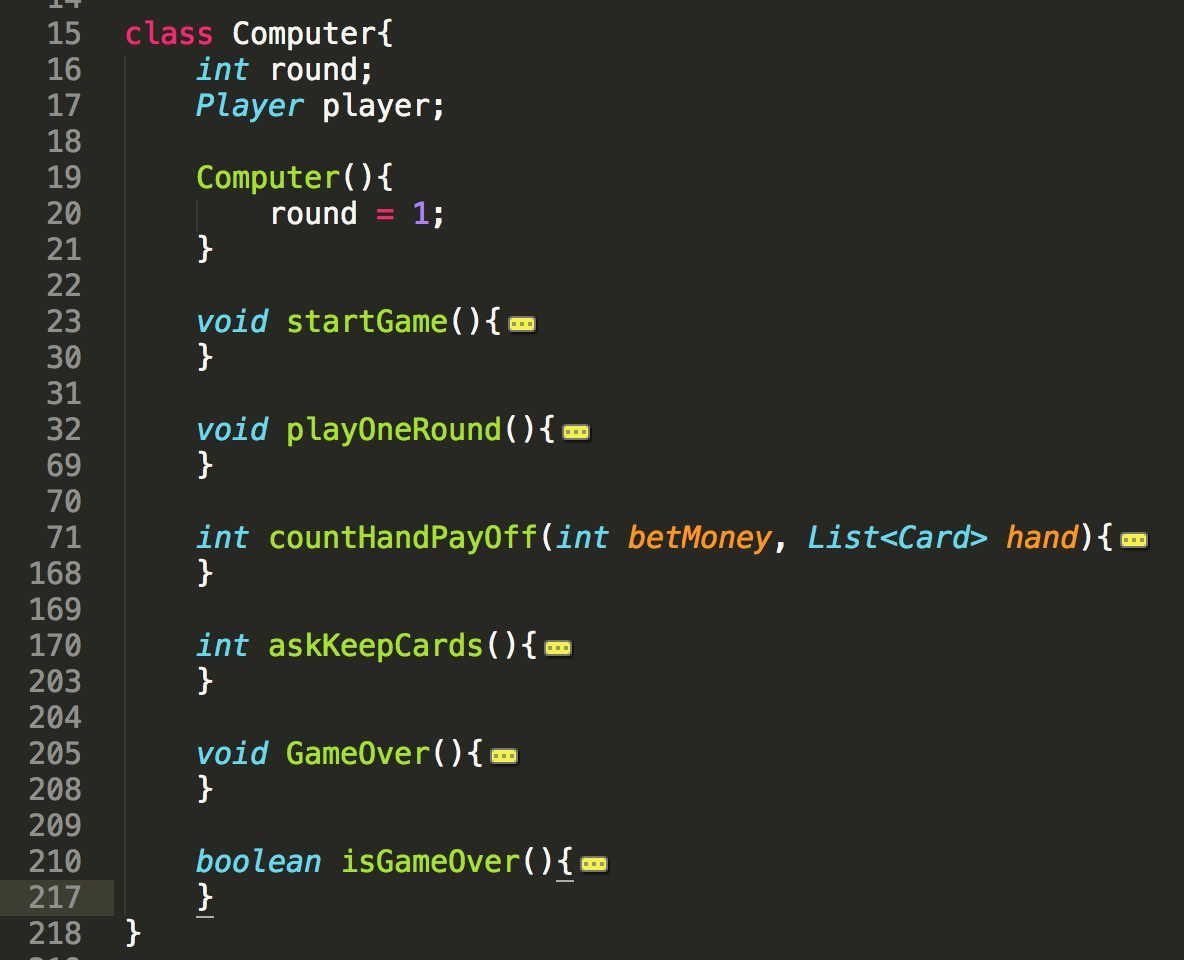
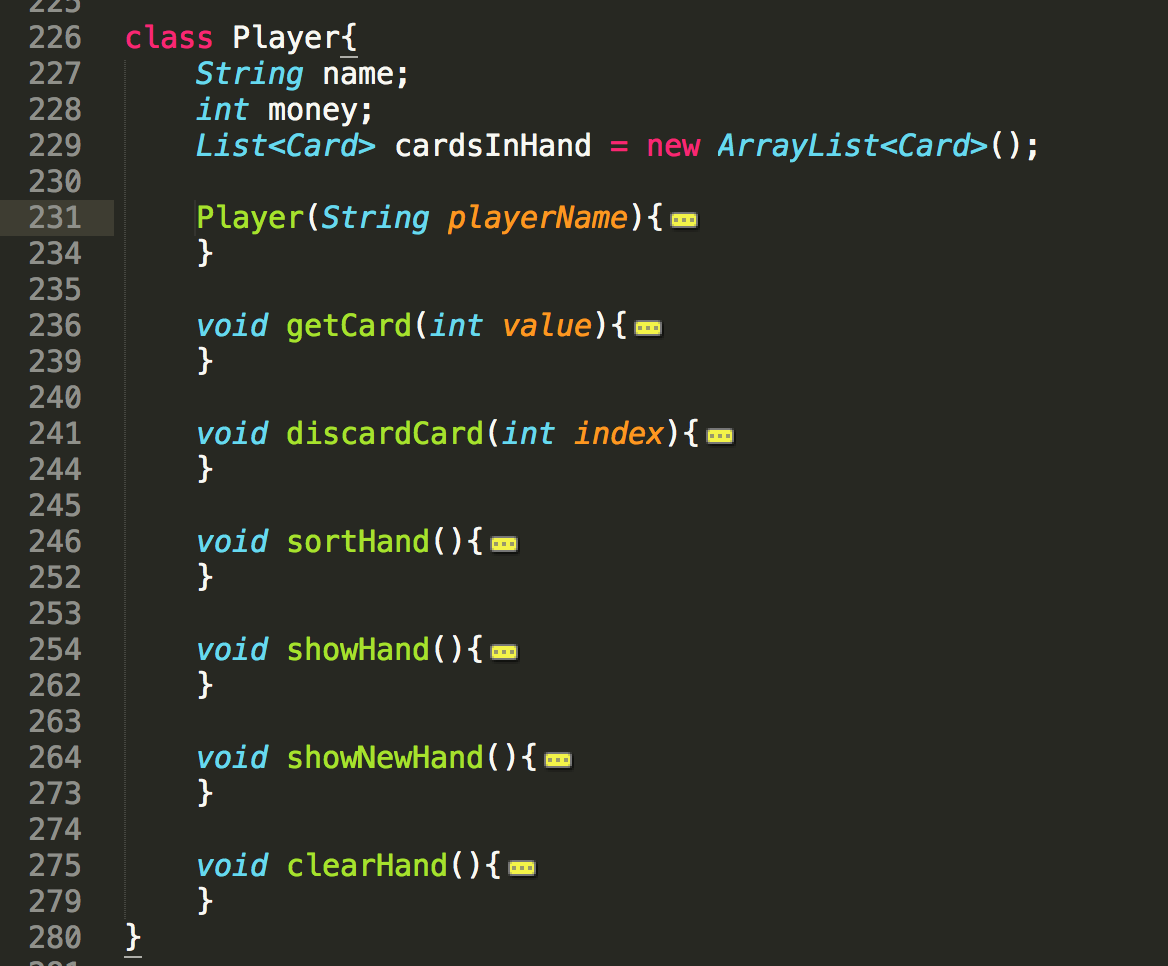
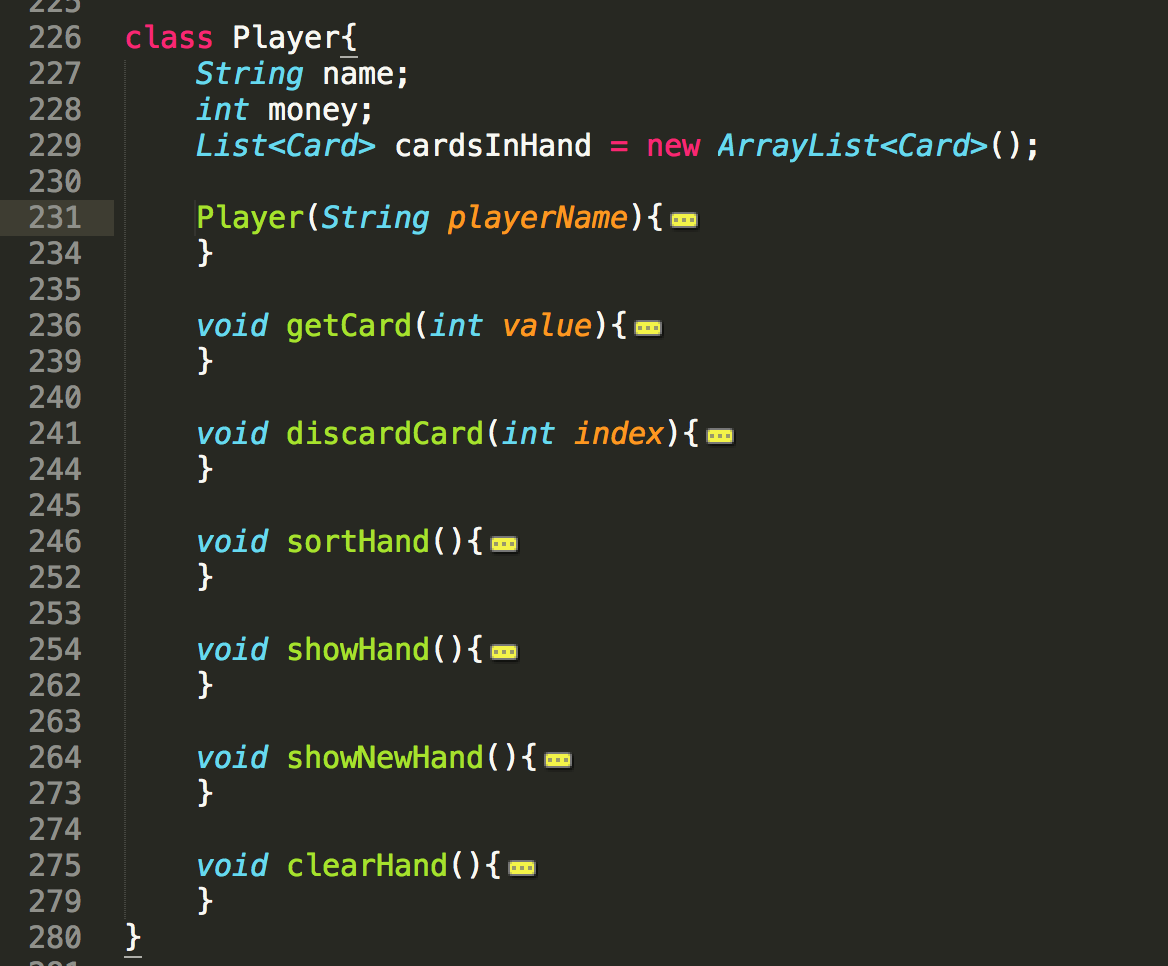
* Outline:

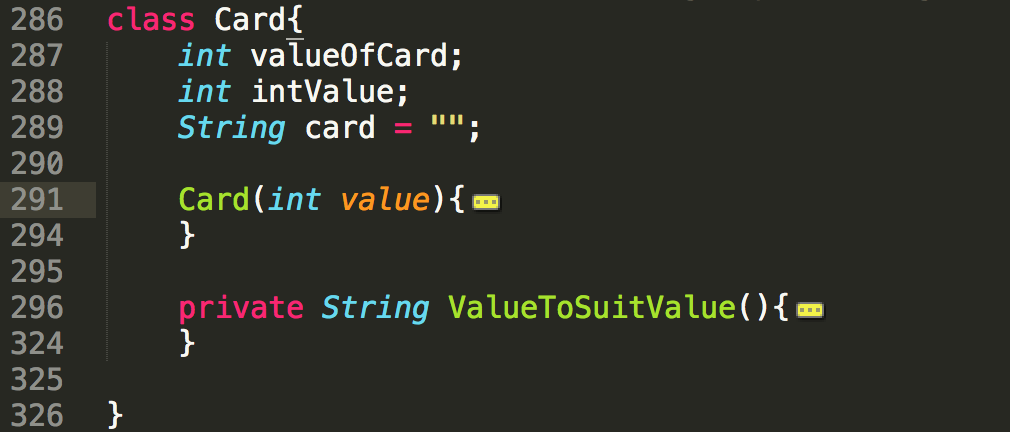
1. Introduction
2. How to play with my program
3. How I tested the correctness of my program
4. The output from three rounds of the game
5. Introduction:

“Jacks or Better” is a single-player video poker game. At first, the player will be asked to enter his name and will be given 1000 P-dollars. Then, in each round:

* 1. The computer opens a new desk and shuffle it.
  2. The player chooses how much to bet (1~5 P dollars).
  3. The computer then distributes 5 cards from the desk to the player.
  4. The player can choose to keep 0~5 cards on hand. Then the other cards would be discarded and replace with the new cards from the shuffled desk.
  5. The computer then determines the best hand to describe the final 5 cards.
  6. The computer then pays the user payoff P-dollars, determined by the P-dollars bet by the user and the payoff table.

1. How to play with my program?
2. Open your command line.
3. **$make all** in the game directory.
4. **$make run** in the game directory.
5. The game starts, just the follow the instructions on the command line window to play the game.
6. Structure and cooperation of the classes

* Main class- POOCasino: Call the computer class to start the game, play every round of game, and stop when the game is over.
* Computer class: The computer class have to maintain the game status & the player (in this game) status, start/play/stop the game, and the most important job is to communicate with the player.
* Player class: The class contains the player status, what cards does he have, and also some methods manipulating cards in hand.
* Card class: The card class contains the status of one card, including card value in several forms (integer: 0~51, 0~13, string such as “C10” “SA”)



* Shuffler class: I use the class provide by TAs.

1. How I tested my program?

Since I followed the structure from the homework spec, the only parts that needed to be tested are “desk-shuffling” and “how to determine the best hand to describe the final 5 words”. In the latter, I modified the code to give the player various kinds of cards including those I considered critical. After a few rounds of testing, there’s no mistakes anymore. And in desk-shuffling I use the shuffler provided by TAs, and the result of it seems really random after I tried to print hundred lines of desks.

1. The output of 3 rounds of my program:

