B01902044 Steven YuChih Lin
FOOP 2015 fall HW4
Instructor: Huan-Tien Lin.
2016/1/5

Player's Strategy

The strategy that I implemented for my player is based on the table below:

(1) Splittable Hand

	2	3	4	5	6	7	8	9	10	A
2,2	H	P	P	P	P	P	H	H	H	H
3,3	H	H	P	P	P	P	H	H	H	H
4,4	H	H	H	Dh	Dh	H	H	H	H	H
5,5										
6,6	P	P	P	P	P	H	H	H	H	H
7,7	P	P	P	P	P	P	H	H	Rs	Rh
8,8	P	P	P	P	P	P	P	P	P	P
9,9	P	P	P	P	P	S	P	P	S	S
10,10										
A,A	P	P	P	P	P	P	P	P	P	P

(for 5,5 and 10,10 hand, refer to hard total of 10 and 20 points respectively)

(2) Hand with Soft Total

	2	3	4	5	6	7	8	9	10	A
13	H	H	Dh	Dh	Dh	H	H	H	H	H
14	H	H	Dh	Dh	Dh	H	H	H	H	H
15	H	H	Dh	Dh	Dh	H	H	H	H	H
16	H	H	Dh	Dh	Dh	H	H	H	H	H
17	Dh	Dh	Dh	Dh	Dh	H	H	H	H	H
18	S	Ds	Ds	Ds	Ds	S	S	H	H	H
19	S	S	S	S	Ds	S	S	S	S	S
20+	S	S	S	S	S	S	S	S	S	S

(3) Hand with Hard Total

	2	3	4	5	6	7	8	9	10	A
4-7	H	H	H	H	H	H	H	H	H	H
8	H	H	H	Dh	Dh	H	H	H	H	H
9	Dh	Dh	Dh	Dh	Dh	H	H	H	H	H
10	Dh	Dh	Dh	Dh	Dh	Dh	Dh	Dh	H	H
11	Dh	Dh	Dh	Dh	Dh	Dh	Dh	Dh	Dh	Dh
12	H	H	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	Rh
16	S	S	S	S	S	H	H	H	Rh	Rh
17	S	S	S	S	S	S	S	S	S	Rs
18+	S	S	S	S	S	S	S	S	S	S

This strategy is a slightly modified version of the one on this website.
with

```
H - Hit.                      Dh - Double if allowed; otherwise, hit.
S - Stand.                    Ds - Double if allowed; otherwise, stand.
P - Split.                    Rh - Surrender if allowed; otherwise, hit.
                              Rs - Surrender if allowed; otherwise, stand.
```

Class Structures: Design and Reasons

All of my class definitions are as follows.

class `Deck`: representing a deck (of cards.)
class `PlayerB01902044`: my player class with the above strategy; a subclass of class `Player`.
class `POOCasino`: the core of this game, where the logic resides.
class `CompositeHand`: a more powerful class to store the hand information; supporting `face_up_hand` and `face_down_hand` as well as other useful utility actions.

Concise, and Self-explanatory.

— The fundamental concept for the defining of my own class structure.

Some examples of my design:

a. `flipUp()` function in `CompositeHand` class, which flips up all face-down cards in a `CompositeHand` instance.

b. `canSplit()` function in `CompositeHand` class checks whether this particular instance can be split or not in order to determine whether the casino should further ask the player if he/she would like to split his/her hand.

c. `selfEvaluate()` function in `PlayerB01902044` class, a function that returns the total value of the player's hand and also sets some boolean flags after evaluating (`isSoft`, `isBlackJack`, `canSplit`.)

The Result of Duel with Classmate

Configuration in `make run` **rule:** `(nRound = 2000, nChip = 5566)`

Output of first 3 rounds.

```
/---------- After Round1 ----------/
Dealer's hand: [H6] [C3] [D9]

Player 0's hand: [H7] [H9]
Player 1's hand: [D4] [S5] [HJ]
Player 2's hand: [C6] [DA] [S6]
Player 3's hand: [H5] [SK] [H3]

Players' bets are: 1 556 2 556

Player 0: having 5565.0 chips. (5565.0)
Player 1: B01902032 (6122.0)
Player 2: having 5564.0 chips. (5564.0)
Player 3: B01902032 (5566.0)

/---------- After Round2 ----------/
Dealer's hand: [H2] [D5] [C4] [HA] [C9]

Player 0's hand: [S7] [ST]
Player 1's hand: [D8] [HK]
Player 1's hand: [C8] [CK]
Player 2's hand: [CJ] [DJ]
Player 3's hand: [C7] [D3] [H4] [D7]

Players' bets are: 1 612 1 556

Player 0: having 5564.0 chips. (5564.0)
Player 1: B01902032 (4898.0)
Player 2: having 5563.0 chips. (5563.0)
Player 3: B01902032 (5566.0)

/---------- After Round3 ----------/
Dealer's hand: [D6] [CQ] [DK]

Player 0's hand: [S8] [HT]
Player 1's hand: [CT] [DQ]
Player 2's hand: [S4] [SA] [SJ] [C2] [HQ]
Player 3's hand: [SQ] [H8]

Players' bets are: 1 489 1 556

Player 0: having 5565.0 chips. (5565.0)
Player 1: B01902032 (5387.0)
Player 2: having 5562.0 chips. (5562.0)
Player 3: B01902032 (6122.0)
```

Output of last 3 rounds.

```
/---------- After Round1998 ----------/
Dealer's hand: [HK] [D5] [D5]

Player 0's hand: [C5] [S7]
Player 1's hand: [D7] [S9] [C6]
Player 2's hand: [C8] [H7]
Player 3's hand: [D6] [S3] [C5] [SJ]

Players' bets are: 1 1 1 1

Player 0: having 4889.0 chips. (4889.0)
Player 1: B01902032 (2565.0)
Player 2: having 4871.0 chips. (4871.0)
Player 3: B01902032 (2394.0)

/---------- After Round1999 ----------/
Dealer's hand: [C9] [H6] [H4]

Player 0's hand: [DA] [HK]
Player 1's hand: [D9] [S5] [H3]
Player 2's hand: [S7] [D8] [H2] [ST]
Player 3's hand: [CK] [C3] [HA]

Players' bets are: 1 1 1 1

Player 0: having 4890.0 chips. (4890.0)
Player 1: B01902032 (2564.0)
Player 2: having 4870.0 chips. (4870.0)
Player 3: B01902032 (2393.0)

/---------- After Round2000 ----------/
Dealer's hand: [CQ] [D2] [CA] [S2] [C8]

Player 0's hand: [S6] [C2] [D7] [H9]
Player 1's hand: [DJ] [H5] [HQ]
Player 2's hand: [D3] [D4] [S9] [SK]
Player 3's hand: [CJ] [DT]

Players' bets are: 1 1 1 1

Player 0: having 4889.0 chips. (4889.0)
Player 1: B01902032 (2563.0)
Player 2: having 4869.0 chips. (4869.0)
Player 3: B01902032 (2394.0)
```

Bonus Part

Javadoc and Comments

I wrote proper comments for each method and instance variable of my classes according to the spec of Javadoc, providing sufficient explanation for all parameters as well as return values. Also the purposes of all methods are stated briefly.
How to generate the document?
> simply use `$make doc-gen` will do!
> to know more, the command for doc-gen is `javadoc -private -d doc/ src/*.java`
(Notice: private classes, fields, and actions are also included because of the `-private` flag.)

Spec Enhancements

There are some rules for this game that are not specified in `hw4.pdf` so well, and therefore, the following includes some of them for TAs to know my Casino logic.

a. Double-down is not allowed after the player has split his/her hand.
b. Use 1-deck at a time. Re-open and shuffle a deck once the previous deck runs out.
c. When splitting, the bet for the second hand is exactly the same as the other hand of the player.

(Notice: the b. item above will probably result in identical cards on the current table.)