(1)R03527017 Yen Cheng, Wang 王嚴徵

(2)Design of class structures and reason

For re-writing of hw1(*Game1*), I design 5 basic parent classes(Shown following) and 1 extended class from "***Player***" which are Player0~Player3.

Class name	Function
Card	Restore the database of 54 cards and method of getter for all cards
Shuffle	Shuffle cards and distribute correct number of cards for 4 players
Computer	Check the hand cards for all players and show the winner
Drawcards	Dealing with the card drawing process for all players and the finishing of the program
Player	Restore the database of hand cards and method "dropcard()" for Game1, Game2 and "dropcard3()" for Game3 to deal with the pairing up process for all players. Also, used as parent class for each player to reduce write same method repeatedly
Player_	Extend from "Player" and generate 4 class - Player0, Player1, Player2, Player3 for restore information of hand cards and method of process of pairing up.

For *Game2*, I design extended class of ***Shuffle2***, ***Computer2*** and one new class, ***Drawcards2*** to deal with the different rule of *Game2*. "***Shuffle2***" can deal with the new ways to distribute cards, "***Computer2***" override some methods to show the taken cards when the winner appears. "***Drawcards2***" is design to make compatibility of the program for the interaction of extended class and the original class.

For *Game3*, I design extended class of ***Shuffle3***, ***Computer3*** and one new class, ***Drawcards3*** and the reason for the design is same with *Game2*.

For the main class, I use a while loop for the player to choose the Game that they want to see. Thus, in some of the classes, I use "restter" to reset the database and variables to run the loop without problems.

(3)How to play the program of 1 original game and the 2 variants

The original game(Game 1) is the one same as in Homework 1, with 2 joker cards and they can not be paired up with any other cards. Both of the 2 variants have 4 players(Player0, Player1, Player2, Player3) and 1 deck of cards, on top of that, all the 3 games show no difference for the card-drawing process and all the games will finish when the first winner appears.

The rule of first variants(Game 2) is to take 1 card from the deck consists of 52 cards(no additional joker cards) at the beginning of every game which means that there are only 51 cards and the taken cards will be shown when the first winner appears. All players acquire 13 cards except the Player3 who has only 12 cards. For

example, if the taken card is C3, the potential joker cards will be D3, H3, S3 as only 2 of them could be paired up and the left card can not be gotten rid of from the hand cards of players.

The rule of second variants(Game3) is to add one more SA to one of the 4 players decided by the computer which means there are 5 Aces (CA, DA, HA, SA, SA) and 53 cards for each game. The player who get the additional SA will have 14 cards while others only have 13 cards. The rule of pairing up Ace cards is shown following and the original rule can be applied to other cards. The SA can only be paired up with the other SA, and the CA, DA, HA can be paired up with each others. Therefore, the potential joker cards is the one of CA, DA, HA. This game actually show more powerful effects than just adding one more card with unchanging pairing up rule. This game will make the player much more difficult to get rid all of their hand cards of they have any Ace cards as the troublesome pairing up rules for Ace cards.

*Available ways for pairing up Ace cards (*These figures are from 菁英國際設計)

*Prohibited ways for pairing up Ace cards

When players open the program the computer will show the message to ask players type 1~3 to choose which games they would like to see.

```
Please type int 1,2,3 to show Game1,Game2,Game3. Otherwise, type 0 to exit.
0
Ok!
Goodbye! The program will stop!
```

Players can type 1,2,3 for many times if they want to see more lots of rounds of games. If they want to close the program they just have to type 0 to end the program. The complete process will be shown in part (5).

(4)How to test the correctness of program?

I usually think that “System.out.print()” is a good way to check the correctness as it can show which method is wrong when running the program or when the program get stuck and I can revise it. For example, I forgot to reset some variables when I design the loop for players to choose Games to see at ease. The

abovementioned way become a perfect solution for checking which variables I need to reset in the corresponding method. In this homework, it is also necessary to check the output of the program to assure the correctness of the game as sometimes my logic design has some defects so the poker game will show some mistakes for the output. Sometimes, I can revise my design through checking the output for several times. Running the program for many times is important as well as some mistakes will only appear for some special cases.

(5)Sample output

Game1

```
java -classpath src hw3
Welcome to HW3!
Please type int 1,2,3 to show Game1,Game2,Game3. Otherwise, type 0 to exit.
1
```

```
Ok!
Hello,welcome to Game1!
Deal Cards:
Player0: R0 B0 D5 S5 D6 H6 S6 H7 H8 H9 C10 H10 DQ DK
Player1: D3 H3 D4 S4 C5 C6 C7 D9 S9 S10 HQ CK HK DA
Player2: D2 C4 H5 S7 D8 D10 CJ HJ SQ SK CA HA SA
Player3: C2 H2 S2 C3 S3 H4 D7 C8 S8 C9 DJ SJ CQ
Drop Cards:
Player0: R0 B0 S6 H7 H8 H9 DQ DK
Player1: C5 C6 C7 S10 HQ DA
Player2: D2 C4 H5 S7 D8 D10 SQ SK SA
Player3: S2 H4 D7 C9 CQ
Game Start
Player0 draws a card from Player1. S10
Player0: R0 B0 S6 H7 H8 H9 S10 DQ DK
Player1: C5 C6 C7 HQ DA
Player1 draws a card from Player2. D10
Player1: C5 C6 C7 D10 HQ DA
Player2: D2 C4 H5 S7 D8 SQ SK SA
Player2 draws a card from Player3. D7
Player2: D2 C4 H5 D8 SQ SK SA
Player3: S2 H4 C9 CQ
Player3 draws a card from Player0. DQ
Player3: S2 H4 C9
Player0: R0 B0 S6 H7 H8 H9 S10 DK
Player0 draws a card from Player1. C6
Player0: R0 B0 H7 H8 H9 S10 DK
Player1: C5 C7 D10 HQ DA
Player1 draws a card from Player2. C4
Player1: C4 C5 C7 D10 HQ DA
Player2: D2 H5 D8 SQ SK SA
Player2 draws a card from Player3. H4
Player2: D2 H4 H5 D8 SQ SK SA
Player3: S2 C9
Player3 draws a card from Player0. H7
Player3: S2 H7 C9
Player0: R0 B0 H8 H9 S10 DK
Player0 draws a card from Player1. D10
Player0: R0 B0 H8 H9 DK
Player1: C4 C5 C7 HQ DA
Player1 draws a card from Player2. H5
Player1: C4 C7 HQ DA
Player2: D2 H4 D8 SQ SK SA
Player2 draws a card from Player3. S2
Player2: H4 D8 SQ SK SA
Player3: H7 C9
Player3 draws a card from Player0. R0
Player3: R0 H7 C9
Player0: B0 H8 H9 DK
```

```
Player0 draws a card from Player1. C4
Player0: B0 C4 H8 H9 DK
Player1: C7 HQ DA
Player1 draws a card from Player2. SA
Player1: C7 HQ
Player2: H4 D8 SQ SK
Player2 draws a card from Player3. C9
Player2: H4 D8 C9 SQ SK
Player3: R0 H7
Player3 draws a card from Player0. DK
Player3: R0 H7 DK
Player0: B0 C4 H8 H9
Player0 draws a card from Player1. HQ
Player0: B0 C4 H8 H9 HQ
Player1: C7
Player1 draws a card from Player2. C9
Player1: C7 C9
Player2: H4 D8 SQ SK
Player2 draws a card from Player3. R0
Player2: R0 H4 D8 SQ SK
Player3: H7 DK
Player3 draws a card from Player0. HQ
Player3: H7 HQ DK
Player0: B0 C4 H8 H9
Player0 draws a card from Player1. C9
Player0: B0 C4 H8
Player1: C7
Player1 draws a card from Player2. H4
Player1: H4 C7
Player2: R0 D8 SQ SK
Player2 draws a card from Player3. H7
Player2: R0 H7 D8 SQ SK
Player3: HQ DK
Player3 draws a card from Player0. C4
Player3: C4 HQ DK
Player0: B0 H8
Player0 draws a card from Player1. H4
Player0: B0 H4 H8
Player1: C7
Player1 draws a card from Player2. H7
Player1:
Player2: R0 D8 SQ SK
Player1 wins
Basic game over
Game1 is over!
```

```
Please type int 1,2,3 to show Game1,Game2,Game3. Otherwise, type 0 to exit.
0
Ok!
Goodbye! The program will stop!
```

Game2

```
Please type int 1,2,3 to show Game1,Game2,Game3. Otherwise, type 0 to exit.
2
```

```
Ok!
Hello,welcome to Game2!
Deal Cards:
Player0: C2 S2 D3 C4 D4 C5 C7 D7 D9 D10 CQ HQ DK
Player1: D2 H2 H5 C6 H7 C8 H9 DJ SJ CK HK SK CA
Player2: C3 S3 S4 D5 S5 D6 H8 S9 C10 H10 S10 DQ SQ
Player3: H3 H6 S6 S7 D8 S8 C9 CJ HJ DA HA SA
Drop Cards:
Player0: D3 C5 D9 D10 DK
Player1: H5 C6 H7 C8 H9 SK CA
Player2: S4 D6 H8 S9 S10
Player3: H3 S7 C9 SA
Game Start
Player0 draws a card from Player1. H5
Player0: D3 D9 D10 DK
Player1: C6 H7 C8 H9 SK CA
Player1 draws a card from Player2. S10
Player1: C6 H7 C8 H9 S10 SK CA
Player2: S4 D6 H8 S9
Player2 draws a card from Player3. C9
Player2: S4 D6 H8
Player3: H3 S7 SA
Player3 draws a card from Player0. D10
Player3: H3 S7 D10 SA
Player0: D3 D9 DK
Player0 draws a card from Player1. S10
Player0: D3 D9 S10 DK
Player1: C6 H7 C8 H9 SK CA
```

```
Player1 draws a card from Player2. H8
Player1: C6 H7 H9 SK CA
Player2: S4 D6
Player2 draws a card from Player3. S7
Player2: S4 D6 S7
Player3: H3 D10 SA
Player3 draws a card from Player0. D3
Player3: D10 SA
Player0: D9 S10 DK
Player0 draws a card from Player1. H9
Player0: S10 DK
Player1: C6 H7 SK CA
Player1 draws a card from Player2. D6
Player1: H7 SK CA
Player2: S4 S7
Player2 draws a card from Player3. SA
Player2: S4 S7 SA
Player3: D10
Player3 draws a card from Player0. S10
Player3:
Player0: DK
Player3  wins
Basic game over
Game2 is over!
```

Game3

```
Please type int 1,2,3 to show Game1,Game2,Game3. Otherwise, type 0 to exit.
3
```

```
Ok!
Hello,welcome to Game3!
Deal Cards:
Player0: C3 C4 D4 H4 S5 H6 S6 S7 D8 S8 C9 D9 S9
Player1: D3 H3 S4 C5 H5 C6 C7 D7 H7 H9 H10 SJ CK
Player2: D2 S2 S3 D6 C8 H8 C10 D10 DJ CQ DQ DK CA SA
Player3: C2 H2 D5 S10 CJ HJ HQ SQ HK SK DA HA SA
Drop Cards:
Player0: C3 H4 S5 S7 S9
Player1: S4 C6 H7 H9 H10 SJ CK
Player2: S3 D6 DJ DK CA SA
Player3: D5 S10 SA
Game Start
Player0 draws a card from Player1. S4
Player0: C3 S5 S7 S9
Player1: C6 H7 H9 H10 SJ CK
Player1 draws a card from Player2. CA
Player1: C6 H7 H9 H10 SJ CK CA
Player2: S3 D6 DJ DK SA
Player2 draws a card from Player3. D5
Player2: S3 D5 D6 DJ DK SA
Player3: S10 SA
Player3 draws a card from Player0. S9
Player3: S9 S10 SA
Player0: C3 S5 S7
Player0 draws a card from Player1. H7
Player0: C3 S5
Player1: C6 H9 H10 SJ CK CA
Player1 draws a card from Player2. D6
Player1: H9 H10 SJ CK CA
Player2: S3 D5 DJ DK SA
Player2 draws a card from Player3. SA
Player2: S3 D5 DJ DK
Player3: S9 S10
```

```
Player3 draws a card from Player0. C3
Player3: C3 S9 S10
Player0: S5
Player0 draws a card from Player1. H9
Player0: S5 H9
Player1: H10 SJ CK CA
Player1 draws a card from Player2. DJ
Player1: H10 CK CA
Player2: S3 D5 DK
Player2 draws a card from Player3. S9
Player2: S3 D5 S9 DK
Player3: C3 S10
Player3 draws a card from Player0. H9
Player3: C3 H9 S10
Player0: S5
Player0 draws a card from Player1. CK
Player0: S5 CK
Player1: D5 CA
Player1 draws a card from Player2. S3
Player1: S3 D5 CA
Player2: DK
Player2 draws a card from Player3. C3
Player2: C3 DK
Player3:
Player3  wins
Basic game over
Game3 is over!
```

(6)Other personal design for bonus

I design a more convenient way of typing 1~3 to let the player to see different games and end the program at ease. For the "***Player***" class, I use abstract class and method to make overriding process more careful. I use getter and setter for all classes to deal with all the private data member as well. I restore total database in "Card" class so all the Games can use same "Card" class instead of overriding.