coe318 Lab 5:
Cards and BlackJack

Objectives

- Implement a `Card` class.
- Implement a `CardPile` class.
- Implement a `SimpleUI` class.
- Fix the `BlackjackGame` class.
- Use an `ArrayList`
- Perform user I/O.
- **Duration: two weeks.**

The game of Simplified Blackjack

Blackjack is the most popular casino gambling game in the world. (Note: the rules of the game favour the casino. The Casino is guaranteed to win in the long run.) Here's how it works:

1. Both you and the House (the Casino) are dealt two cards: one is face up and the other face down. So you can see only one of the House's cards and it can see only one of yours. But both you and the House can discretely peek at your face down cards.
2. Each card has a score as follows:
    - An Ace has a score of 1. (This differs from real Blackjack where the Ace can have a score of either 1 or 11 at the player's discretion. The simplified version for this lab is easier to program.)
    - A Jack, Queen or King has a score of 10.
    - All other cards have a score equal to their rank. (For example, the 4 of Hearts or the 4 of any suit have a rank of 4 and a score of 4.)
3. The House will obtain additional cards until its score is 17 or more.
4. You are asked if you want another card. If you answer yes, you get another face down card and you will asked again. This continues until you say "no". This ends the game.
5. All cards are now turned face-up and the scores of you and the House are calculated.
    - You lose if your score is more than 21 (no matter what the House's score is).
    - You lose if your score is the same as the House's.
    - You win if:
        - You don't go over 21 and the House does go over 21

- Both your scores are 21 or under and your score is more than the House's.

The `Card` class

The `Card` class will consist of instance variables, a constructor and the methods.

A template for the class is available http://www.ee.ryerson.ca/~courses/coe318/lab5/Card.java

The template contains javadocs for all the methods. You should generate the html representation of the class to have an easier representation of the class's API (Application Programmer Interface). Many of the methods are only stubs and you have to modify them so that they implement the API.

The template will compile and it has a `main` method that will run but it prouduces the wrong output.

Step 1: Create a Netbeans project and Card class

1. Create a Netbeans project called `Blackjack`.
2. Create a Java file (class library type) called `Card` with package `coe318.lab5`. (All java files in this lab should have that package declaration.)
3. Determine your instance variables and implement the constructor.
4. Implement the other methods.
5. **Important**: Do **not** continue to the next classes until this class works!

Step 2: Implement the remaining classes

Templates for the remaining classes are available:

- http://www.ee.ryerson.ca/~courses/coe318/lab5/CardPile.java

You should get the `CardPile` class to work before proceding. (It has its own `main` method.)

You next have to create the `BlackjackGame` and `SimpleUI` classes starting with the templates provided. You also have to create an **interface** (`UserInterface`) copying the code given below. To create an interface in Netbeans, create a new file and have an interface (not a class) created. We have not covered interfaces yet in lectures. However, the only thing you need to know for now is that this file is essential and must not be modified.

- http://www.ee.ryerson.ca/~courses/coe318/lab5/BlackjackGame.java
- http://www.ee.ryerson.ca/~courses/coe318/lab5/SimpleUI.java
- http://www.ee.ryerson.ca/~courses/coe318/lab5/UserInterface.java

Note: All three of these templates have to be loaded in order for them to compile without error.

Step 3: Submit your lab

1. Submit your lab by zipping it to a file called `lab5.zip`
2. Then use the command `submit coe318 lab5 lab5.zip` to complete the submission.

Notes:

1. The Card class compareTo(Card c) method should return a negative, zero or positive value depending on whether "this" is less than, equal to or greater than the other card. For 2 cards of unequal rank, the one with the higher rank is "bigger". If the ranks are the same, the suit is considered; the suit orders (from lowest to highest) are Clubs, Diamonds, Hearts, Spades.
2. The Card equals(Card c) method should only return true if the cards have the same suit and rank.

Sample sessions

The following are 2 sample sessions. In one the House wins, in the other you win.

Note: The format of the output is not defined. Within `SimpleUI` you can format the output of `display(), hitMe()` and `gaameOver()` any way you want.

```
House holds:
     ?
     7 of Hearts

You hold:
     Queen of Hearts
     Jack of Clubs

House holds:
     ?
     7 of Hearts
     6 of Spades

You hold:
     Queen of Hearts
     Jack of Clubs

Another card? n
House holds:
     ?
     7 of Hearts
     6 of Spades
     King of Spades

You hold:
     Queen of Hearts
     Jack of Clubs
```

```
Game over
House holds:
     Ace of Diamonds
     7 of Hearts
     6 of Spades
     King of Spades

You hold:
     Queen of Hearts
     Jack of Clubs

Your score: 20, House score: 24
You win
```

```
House holds:
     ?
     4 of Clubs

You hold:
     King of Spades
     3 of Spades

House holds:
     ?
     4 of Clubs
     Ace of Diamonds

You hold:
     King of Spades
     3 of Spades

Another card? y
House holds:
     ?
     4 of Clubs
     Ace of Diamonds

You hold:
     King of Spades
     3 of Spades
     9 of Clubs

House holds:
     ?
     4 of Clubs
     Ace of Diamonds
     Queen of Clubs
```

```
You hold:
     King of Spades
     3 of Spades
     9 of Clubs

Another card? n
Game over
House holds:
     Jack of Spades
     4 of Clubs
     Ace of Diamonds
     Queen of Clubs

You hold:
     King of Spades
     3 of Spades
     9 of Clubs

Your score: 22, House score: 25
The House wins
```