CCPS506 Lab 2 – Classes and Methods in Smalltalk

Preamble

After completing the first lab, you should be well acquainted with the Pharo environment. In this lab we'll go a step further and create a Smalltalk class that implements several simple methods. The methods are related to the card game of Poker, but no prior knowledge of Poker is required.

Lab Description

Start by creating a new class category through the System Browser called CCPS506, and a new class called **Lab2**. Check the lecture slides for a reference on how to do this. Your **Lab2** class will implement several different methods, each of which is described below. As long as each method adheres to the specifications below, you have complete freedom. Use whichever Smalltalk language constructs you find most useful.

1) `pokerValidate:`

- This method takes an **Array** as an argument and returns a **Boolean**.
- The input array should contain **five integers** whose values are between **1-13**.
- If it does, return true. If not, return false.

2) `pokerThreeOfAKind:`

- This method takes an array as an argument and returns a Boolean.
- If the input array is not valid (call the pokerValidate method), return false.
- This method should return true if three of the integers in the array are the same, with the other two being different. Return false if this is not the case.
- If the hand is a full house, pokerThreeOfAKind should return false! For example, the array **`#(1 2 3 3 3)`** should return true, but **`#(2 2 3 3 3)`** should return false. This is an easy mistake to make.

3) `pokerStraight:`

- This method takes an array as an argument and returns a Boolean.
- If the input array is not valid (call the pokerValidate method), return false.
- The integers represent card ranks. 1 is Ace, 2-10 are the numbered cards, and 11, 12, 13 are face cards Jack, Queen, King respectively.
- If the cards are valid, check if they form a straight (five cards in a row)
- When it comes to straights, Ace can be both high and low! **`#(1 3 4 2 5)`** is a straight (Ace to 5), as is **`#(1 13 10 11 12)`** (10, Jack, Queen, King, Ace).
- The order of the cards doesn't matter: **`#(4 8 5 7 6)`** is a straight (4 to 8).
- If the integer ranks form a straight, return true. Otherwise, return false.

4) **`pokerFullHouse:`**

- This method takes an array as an argument and returns a Boolean.
- If the input array is not valid (call the pokerValidate method), return false.
- This method will return true if the array could represent a Full House (two integers of one rank, three of another) and false otherwise.

5) **`pokerTwoPair:`**

- This method takes an array as an argument and returns a Boolean.
- If the input array is not valid (call the pokerValidate method), return false.
- This method returns true if the array is a two-pair hand, and false otherwise.
- A two-pair hand contains two cards of one rank, two cards of another rank, and one last card whose rank is *different* from either pair.
- If the hand is a full house, pokerTwoPair should return false! This is an easy mistake to make.

Testing

When your **Lab2** class and the above methods are implemented correctly, you should be able to test them in the Playground as follows:

```
| poker |
poker := Lab2 new.
Transcript show: (poker pokerValidate: #(1 2 3 4 5 6 7 8)).
Transcript show: (poker pokerValidate: #(1 2.9 '34')).
Transcript show: (poker pokerValidate: #(9 8 7 6 5)).
Transcript show: (poker pokerThreeOfAKind: #(13 13 4 13 8)).
Transcript show: (poker pokerThreeOfAKind: #(13 13 4 13 4)).
Transcript show: (poker pokerThreeOfAKind: #(9 9 9 13 9)).
Transcript show: (poker pokerThreeOfAKind: #(1 2 3 2 2)).
Transcript show: (poker pokerStraight: #(1 13 11 12 10)).
Transcript show: (poker pokerStraight: #(8 6 4 7 5)).
Transcript show: (poker pokerFullHouse: #(13 13 4 13 4)).
Transcript show: (poker pokerTwoPair: #(1 1 1 2 2)).
Transcript show: (poker pokerTwoPair: #(3 1 1 2 2)).
```

Submission

Labs are to be completed and submitted *individually*. Submit the following items on D2L, under the submission for Lab 2:

1) An archive containing your entire Pharo image directory. To find your image directory, select the image in the launcher that contains your Lab2, and look at the bottom of the launcher screen to see the "Location" directory. Archive (zip/rar/7z/whatever) ***everything*** in this folder – not just the image. Submit this archive file.