A client has asked to design an online poker system. Your task is to implement a Java Library that will serve as the backend of this system.

I have designed the game of poker based on the following assumptions:

- I describe the game only by the help of two methods i.e. shuffle() and dealOneCard() and hence only these methods shall be found in classes.
- dealOneCard() will only return a card till the 52^{nd} call, at the 53^{rd} call it will not return any card as all have been seen already.
- A shuffle() call can be followed by many such shuffle calls, and every time it will generate a different order of cards.

I have knowledge of Fisher-Yates shuffle algorithm and it performance as compared to other random shuffling algorithm. (Source: Codinghorror.com) hence will be implementing it for the purpose of shuffle() method.

Requirements:

- I have a fixed set of card weights and pre-defined card suit (we are only considering a deck of playing cards). Hence in my object oriented implementation using Java platform I use enum for card weight and card suit. Enum is a set of predefined constants for example colors.
- We have 52 cards and each card has some specific weight and belongs to one of 4 suits, hence we need a class for a card.
- A deck is a set of 52 cards hence a class for deck of cards that holds cards in an list

Fisher-Yates Implementation: As codinghorror describes, this shuffle algorithm works in place by swapping a value at the randomly selected index with the value at the last unselected index. That is at every instance the size of list to be shuffled decreases from the higher end.