Marina Semenova April 24, 2019

Deck of Cards

Client Code

1. **What is the identifier for the new instance of the DeckofCards class?**

   The identifier is deck1.

   a. **Is any data being passed to the constructor?**

      No data is being passed to the constructor.

2. **Explain what happens when deck1.deal() is placed in a print statement. Mention a few things that happen (not just one). Refer to the code for the other classes.**

   deck1.deal() calls the deal() method in the DeckofCards class, which returns a Card from the array myDeck. When the returned Card object is in a print statement, the toString() method of the Card class is invoked, which prints the face and the suit of the Card.

Card Class

1. **What is being assigned in the constructor?**

   The face and the suit of the card are being assigned in the Card constructor.

2. **Explain what the toString() method does.**

   When a Card object is in a print statement, the toString() method of the Card class is invoked, which prints the face and the suit of the Card.

DeckofCards class

1. **Explain the line of code: myDeck[n] = new Card( )**

   This line creates n objects of the class Card and stores the references in myDeck. Each card is assigned a face and a suit in the process.

   a. **What do you think is being stored in myDeck[ ]?**

      References of each Card object are stored in myDeck[].

   b. **What two features does each element in myDeck[ ] have?**

      Each Card object has a face and a suit.

   c. **How does use of array improve the functionality of the program?**

      It removed the need to make 52 separate instances of the Card class. This enables organization and increased readability of the code.

2. **Why does the constructor have a for loop?**

   Because there need to be 52 objects created and 52 references assigned. This is done through iteration.

3. **Explain**

   a. **the purpose of [n % 13] in the faces[ ] array.**

      This iterates 0-12 4 times to assign all the faces to each suit. The faces array contains the 13 faces that are assigned and n%13 controls the index.

   b. **the purpose of [n / 13] in the suits array.**

n/13 goes 0-4 but it only increases every 13 iterations. n/3 is the index being accessed to assign 1 of the 4 suits to each Card. Each suit must also be assigned 13 faces, thus the index must only be increased after 13 iterations.

4. **What is the return type for the deal() method?**

   The return type of the deal() method is Card.

   a. **What is the purpose of the if statement?**

      If the deal() method is called more times than there are cards then return null because otherwise, the method would be accessing nonexistent elements of the myDeck[] array.

Shuffle method

A shuffle method could be created by iterating through the length of the deck and each index could be swapped with a random index determined by Math.random(). This would require a for loop, a temporary variable, and a random index.

Implementation

```
public void shuffle()
   {
      for (int x = 0; x < 52; x++){
         int randomIndex = (int)((Math.random() * 52));
         Card temp = myDeck[x];
         myDeck[x] = myDeck[randomIndex];
         myDeck[randomIndex] = temp;
      }
      myCardNum = 0;
   }
```