PROGRAMMING IN JAVA LAB-4

//

PRN-21070126002

Name- Aadarsh Nayyer

Batch-AIML A1

Problem: Write a menu-driven Java Program for the following:
There are 52 cards in a deck, each of which belongs to one of four suits and one of 13 ranks. Represent a deck of cards as an array of
Objects (*you may use the Vector class)

1. **Use integers to encode the ranks and suits.**
2. **Have suitable default & parameterized constructors.**
3. **all data members to have private access.**
4. **The class 'Card' to have the following methods:**

createDeck(), printCard(), printDeck (),sameCard(),compareCard(), sortCard(), findCard() which searches through an array or vector of Cards to see whether it contains a certain card, dealCards() function: to print 5 random cards from the existing deck.

//

```
import java.util.Scanner;

import java.util.Vector;

import java.util.Random;


class Card {

   private int rank;

   private int suit;


   public Card() {

      this.rank = 0;

      this.suit = 0;

   }


   public Card(int rank, int suit) {

      this.rank = rank;

      this.suit = suit;

   }
```

```
public int getRank() {
   return rank;
}

public int getSuit() {
   return suit;
}

public void printCard() {
   String[] suits = {"Spades", "Hearts", "Diamonds", "Clubs"};
   String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};
   System.out.println(ranks[this.rank] + " of " + suits[this.suit]);
}

public static Vector<Card> createDeck() {
   Vector<Card> deck = new Vector<Card>();
   for (int suit = 0; suit < 4; suit++) {
      for (int rank = 0; rank < 13; rank++) {
         deck.add(new Card(rank, suit));
      }
   }
   return deck;
}

public static void printDeck(Vector<Card> deck) {
   for (Card card : deck) {
      card.printCard();
   }
}

public boolean sameCard(Card other) {
```

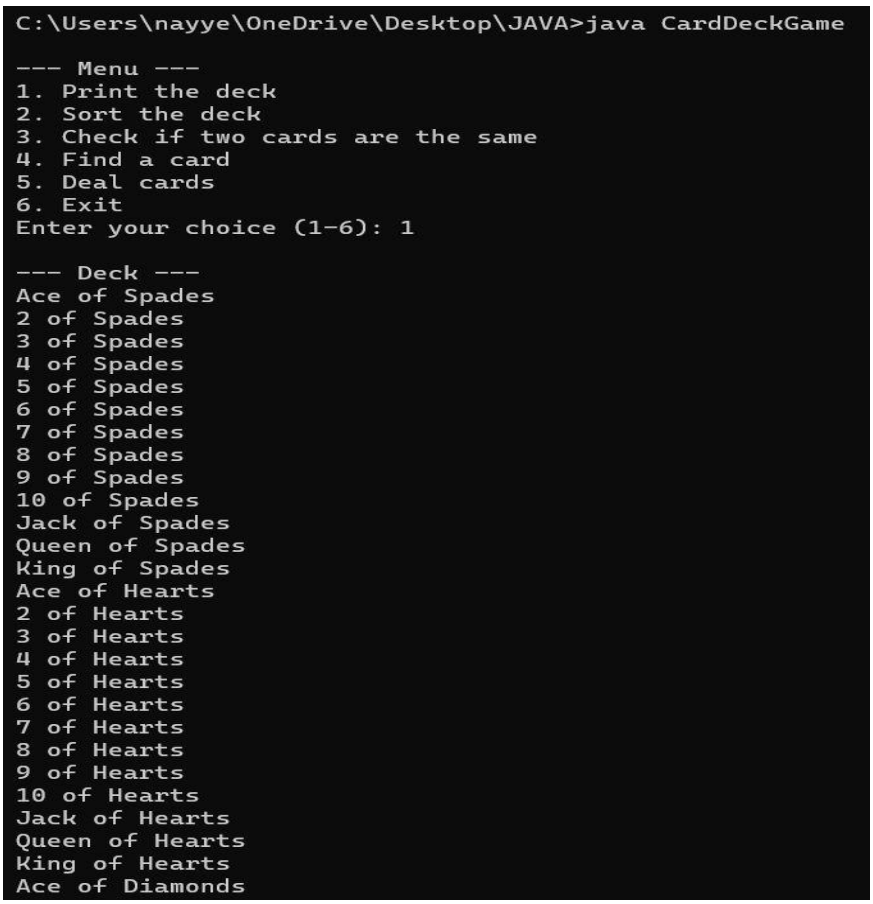
```
    return (this.rank == other.rank && this.suit == other.suit);
}

public int compareCard(Card other) {
    if (this.rank < other.rank) {
        return -1;
    } else if (this.rank > other.rank) {
        return 1;
    } else {
        if (this.suit < other.suit) {
            return -1;
        } else if (this.suit > other.suit) {
            return 1;
        } else {
            return 0;
        }
    }
}

public static void sortDeck(Vector<Card> deck) {
    deck.sort((c1, c2) -> c1.compareCard(c2));
}

public static void findCard(Vector<Card> deck, Card card) {
    for (int i = 0; i < deck.size(); i++) {
        if (deck.get(i).sameCard(card)) {
            System.out.println("Card found at index " + i);
            return;
        }
    }
    System.out.println("Card not found");
```

```
    }

    public static void dealCards(Vector<Card> deck, int numCards) {
        Random rand = new Random();
        for (int i = 0; i < numCards; i++) {
            int index = rand.nextInt(deck.size());
            Card card = deck.get(index);
            card.printCard();
            deck.remove(index);
        }
    }
}

public class CardDeckGame {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Vector<Card> deck = Card.createDeck();

        while (true) {
            System.out.println("\n--- Menu ---");
            System.out.println("1. Print the deck");
            System.out.println("2. Sort the deck");
            System.out.println("3. Check if two cards are the same");
            System.out.println("4. Find a card");
            System.out.println("5. Deal cards");
            System.out.println("6. Exit");
            System.out.print("Enter your choice (1-6): ");
            int choice = input.nextInt();

            if (choice == 1) {
                System.out.println("\n--- Deck ---");
```

```
    Card.printDeck(deck);

}else if (choice == 2) {
    Card.sortDeck(deck);
    System.out.println("\n--- Sorted deck ---");
    Card.printDeck(deck);

} else if (choice == 3) {
    System.out.println("\nEnter the first card:");
    Card card1 = readCard(input);
    System.out.println("Enter the second card:");
    Card card2 = readCard(input);
    if (card1.sameCard(card2)) {
        System.out.println("The two cards are the same");
    } else {
        System.out.println("The two cards are different");
    }

} else if (choice == 4) {
    System.out.println("\nEnter a card to search for:");
    Card card = readCard(input);
    Card.findCard(deck, card);

} else if (choice == 5) {
    System.out.println("\nDealing cards...");
    Card.dealCards(deck, 5);

} else if (choice == 6) {
    System.out.println("Thank You!");
    break;
```

```
        } else {
            System.out.println("Invalid choice, please try again");
        }
    }
}

public static Card readCard(Scanner input) {
    System.out.print("Enter rank (0-12): ");
    int rank = input.nextInt();
    System.out.print("Enter suit (0-3): ");
    int suit = input.nextInt();
    return new Card(rank, suit);
}
```

OUTPUT



```
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 5

Dealing cards...
2 of Clubs
King of Hearts
Queen of Diamonds
Jack of Hearts
9 of Clubs
```

```
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 4

Enter a card to search for:
Enter rank (0-12): 10
Enter suit (0-3): 0
Card found at index 10

--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 6
Thank You!
```

GITHUB LINK: https://github.com/aadarsh1810/JAVA/tree/main/Assignment-4