

JAVA Assignment -4

Name- Abhinav Kumar

PRN- 21070126006

Branch- AIML-A1

Github- <https://github.com/Abhinav-kr-2807/JAVA>

Problem Statement- 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.

```
Code- 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:

```
1 "C:\Program Files\Java\jdk-19\bin\java.exe" "-  
  javaagent:C:\Program Files\JetBrains\IntelliJ IDEA  
  Community Edition 2023.1\lib\idea_rt.jar=50828:C\  
  Program Files\JetBrains\IntelliJ IDEA Community  
  Edition 2023.1\bin" -Dfile.encoding=UTF-8 -Dsun.  
  stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -  
  classpath "C:\Users\AK-Lenovo\IdeaProjects\Deck of  
  Cards\out\production\Deck of Cards" CardDeckGame  
2  
3 --- Menu ---  
4 1. Print the deck  
5 2. Sort the deck  
6 3. Check if two cards are the same  
7 4. Find a card  
8 5. Deal cards  
9 6. Exit  
10 Enter your choice (1-6): 1  
11  
12 --- Deck ---  
13 Ace of Spades  
14 2 of Spades  
15 3 of Spades  
16 4 of Spades  
17 5 of Spades  
18 6 of Spades  
19 7 of Spades  
20 8 of Spades  
21 9 of Spades  
22 10 of Spades  
23 Jack of Spades  
24 Queen of Spades  
25 King of Spades  
26 Ace of Hearts  
27 2 of Hearts  
28 3 of Hearts  
29 4 of Hearts  
30 5 of Hearts  
31 6 of Hearts  
32 7 of Hearts  
33 8 of Hearts  
34 9 of Hearts
```

```
35 10 of Hearts
36 Jack of Hearts
37 Queen of Hearts
38 King of Hearts
39 Ace of Diamonds
40 2 of Diamonds
41 3 of Diamonds
42 4 of Diamonds
43 5 of Diamonds
44 6 of Diamonds
45 7 of Diamonds
46 8 of Diamonds
47 9 of Diamonds
48 10 of Diamonds
49 Jack of Diamonds
50 Queen of Diamonds
51 King of Diamonds
52 Ace of Clubs
53 2 of Clubs
54 3 of Clubs
55 4 of Clubs
56 5 of Clubs
57 6 of Clubs
58 7 of Clubs
59 8 of Clubs
60 9 of Clubs
61 10 of Clubs
62 Jack of Clubs
63 Queen of Clubs
64 King of Clubs
65
66 --- Menu ---
67 1. Print the deck
68 2. Sort the deck
69 3. Check if two cards are the same
70 4. Find a card
71 5. Deal cards
72 6. Exit
73 Enter your choice (1-6): 3
74
75 Enter the first card:
```

```
76 Enter rank (0-12): 6
77 Enter suit (0-3): 2
78 Enter the second card:
79 Enter rank (0-12): 9
80 Enter suit (0-3): 1
81 The two cards are different
82
83 --- Menu ---
84 1. Print the deck
85 2. Sort the deck
86 3. Check if two cards are the same
87 4. Find a card
88 5. Deal cards
89 6. Exit
90 Enter your choice (1-6):
```