Name:Aghara Hem

Title: Develop a Java program simulating a deck of cards.

Code:

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import java.util.Random;

// Card class

class Card {

    public enum Suit {SPADE, CLUB, HEART, DIAMOND}

    public enum Rank {ACE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, TEN, JACK, QUEEN, KING}

    private final Suit suit;

    private final Rank rank;

    public Card(Suit suit, Rank rank) {

        this.suit = suit;

        this.rank = rank;

    }

    public Suit getSuit() { return suit; }

    public Rank getRank() { return rank; }

    @Override

    public String toString() { return rank + " of " + suit; }

}

// Deck class

class Deck {

    private List<Card> cards;

    public Deck() {

        cards = new ArrayList<>();

        for (Card.Suit suit : Card.Suit.values()) {

            for (Card.Rank rank : Card.Rank.values()) {

                cards.add(new Card(suit, rank));

            }

        }

    }

    public void shuffleDeck() { Collections.shuffle(cards, new Random()); }

    public Card drawCard() {

        if (cards.size() == 0) throw new IllegalStateException("No cards left in the deck");

        return cards.remove(cards.size() - 1);

    }

    public int getDeckSize() { return cards.size(); }

    public List<Card> drawMultipleCards(int numberOfCards) {

        List<Card> drawnCards = new ArrayList<>();

        for (int i = 0; i < numberOfCards; i++) {

            drawnCards.add(drawCard());

        }

        return drawnCards;

    }

}

// Comparator class for sorting

class CardComparator implements Comparator<Card> {

    @Override

    public int compare(Card c1, Card c2) {

        int colorComparison = getColorPriority(c1.getSuit()) - getColorPriority(c2.getSuit());

        if (colorComparison != 0) return colorComparison;

        int suitComparison = c1.getSuit().compareTo(c2.getSuit());

        if (suitComparison != 0) return suitComparison;

        return c1.getRank().compareTo(c2.getRank());

    }

    private int getColorPriority(Card.Suit suit) {

        if (suit == Card.Suit.HEART || suit == Card.Suit.DIAMOND) return 1; // Red

        else return 0; // Black

    }

}

// Main class

public class DeckOfCards {

    public static void main(String[] args) {

        Deck deck = new Deck();

        // Shuffle the deck (Bonus)

        deck.shuffleDeck();

        // Draw 20 random cards from the deck

        List<Card> drawnCards = deck.drawMultipleCards(20);

        // Print the drawn cards

        System.out.println("Drawn Cards:");

        for (Card card : drawnCards) {

            System.out.println(card);

        }

        // Sort the cards using custom comparator

        CardComparator cardComparator = new CardComparator();

        Collections.sort(drawnCards, cardComparator);

        // Print sorted cards

        System.out.println("\nSorted Cards:");

        for (Card card : drawnCards) {

            System.out.println(card);

        }

    }

}

Output: