Project 1: Deck Of Cards.docx

Objective

Q. Develop a Java program simulating a deck of cards, including functionalities like random card drawing, sorting, and (optional) shuffling.

Source Code

Card.java

```
package com.java.assingment.card_deck;

import java.util.*;

@Deprecated
enum SimpleSuit {
          HEART, SPADE, DIAMOND, CLUB
}

enum Suit {
          HEARTS(1), SPADES(3), DIAMONDS(2), CLUBS(4);

//H D   S C
          private final int value;

          private Suit(int value) {
                    this.value = value;
          }

          public int getValue() {
                    return value;
          }
}

//enum Rank {ACE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, TEN, JACK, QUEEN, KING}
enum Rank {
          ACE(1), TWO(2), THREE(3), FOUR(4), FIVE(5), SIX(6), SEVEN(7), EIGHT(8), NINE(9),
TEN(10), JACK(11), QUEEN(12),
          KING(13);

          private final int value;

          private Rank(int value) {
                    this.value = value;
          }

          public int getValue() {
                    return value;
          }
}

public class Card {
          private Suit suit;
          private 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;
        }
}

class Deck {
        private List<Card> cards;

        public Deck() {
                cards = new ArrayList<>();
                for (Suit suit : Suit.values()) {// 4
                        for (Rank rank : Rank.values()) {// 13
                                cards.add(new Card(suit, rank));
                        }
                }
        }

        public void shuffle() {
                Collections.shuffle(cards);
        }

        public Card drawCard() {
                if (cards.isEmpty()) {
                        throw new NoSuchElementException("Deck is empty");
                }
                return cards.remove(0);
        }

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

@Deprecated
class CardComparator implements Comparator<Card> {

        @Override
        public int compare(Card card1, Card card2) {
                // Compare by color first
                int colorComparison = Integer.compare(card1.getSuit().ordinal() % 2,
card2.getSuit().ordinal() % 2);
                if (colorComparison != 0) {
                        return colorComparison;
                }

                // Compare by suit within color
                int suitComparison = Integer.compare(card1.getSuit().ordinal(),
card2.getSuit().ordinal());
                if (suitComparison != 0) {
                        return suitComparison;
                }

                // Compare by rank value
                return Integer.compare(card1.getRank().getValue(),
card2.getRank().getValue());
        }
}
```

Main.java

```
package com.java.assingment.card_deck;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

class Main {

	public static void main(String[] args) {
		Deck deck = new Deck();

		// Bonus: shuffle before drawing
		deck.shuffle();

		List<Card> drawnCards = new ArrayList<>();
		for (int i = 0; i < 20; i++) {
			drawnCards.add(deck.drawCard());
		}


		// Compare suits based on assigned values
		Comparator<Card> cardComparator_4 = Comparator.comparing((Card card) ->
card.getSuit().getValue()).thenComparing(card -> card.getRank().getValue());

		Collections.sort(drawnCards, cardComparator_4);
		System.out.println("Drawn cards (sorted):");
		for (Card card : drawnCards) {
			System.out.println(card);
		}
	}
}
```

Output

```
Drawn cards (sorted):
TWO of HEARTS
NINE of HEARTS
TEN of HEARTS
JACK of HEARTS
KING of HEARTS
ACE of DIAMONDS
TWO of DIAMONDS
FOUR of DIAMONDS
SIX of DIAMONDS
SEVEN of DIAMONDS
JACK of DIAMONDS
ACE of SPADES
THREE of SPADES
FOUR of SPADES
EIGHT of SPADES
JACK of SPADES
QUEEN of SPADES
ACE of CLUBS
TEN of CLUBS
JACK of CLUBS
```