```
package a_elevens;

import java.util.List;
import java.util.ArrayList;

public class Deck {

        private List<Card> cards;
    private List<String> suits;


    public Deck(Deck another){
        this.cards = another.cards;
        this.suits = another.suits;
    }


        public Deck() {
                cards = new ArrayList<>();
        suits = new ArrayList<>();
        suits.add("Hearts");
        suits.add("Spades");
        suits.add("Diamond");
        suits.add("Clubs");
        for(int i = 0; i < 4; i++){
            for (int j = 2; j < 11; j++) {
                                cards.add(new Card(String.valueOf(j),
suits.get(i), j));
                    }
            cards.add(new Card("Jack", suits.get(i), 11));
            cards.add(new Card("Queen", suits.get(i), 12));
            cards.add(new Card("King", suits.get(i), 13));
            cards.add(new Card("Ace", suits.get(i), 1));
        }
                shuffle();
        }
        public boolean isEmpty() {
                return cards.isEmpty();
        }
        public int get_size() {
                return cards.size();
        }
        public void shuffle() {

                for (int k = cards.size() - 1; k > 0; k--) {
                        int howMany = k + 1;
                        int start = 0;
                        int randPos = (int) (Math.random() * howMany) + start;
                        Card temp = cards.get(k);
                        cards.set(k, cards.get(randPos));
                        cards.set(randPos, temp);
                }
        }
        public Card get_one_card() {

                if (isEmpty()) {
                        return null;
                }

                Card c = cards.get(0);
        cards.remove(0);
                return c;
        }
        @Override
        public String toString() {

        String deck_string = "";

        for (Card card : cards) {

            deck_string = deck_string + ", " + card.toString();
        }
        return deck_string;
        }
    String getCheckCapacity() {
        throw new UnsupportedOperationException("not used in the program");
    }

    Card remove() {
        throw new UnsupportedOperationException("not used in the program");
    }
}
```