

Homework 9: Card Set Object

Create a “card set” object, along with some of its basic methods. This `CardSet` object represents a collection of `Card` objects: it could be as small as zero `Cards`, or as big as millions. This same object could be used to represent a deck, a discard pile, or a player’s hand.

For this assignment, you’ll have to use the `Card` object that you made during this week’s lab. You’ll have to make a `CardSet` object, which contains several cards.

Pretend that you are making this object for a colleague who is making a card game. Your object will be based on an `ArrayList<Card>`, which is an array-list that holds several `Card` objects. However, your colleague doesn’t know or care that your object contains the `ArrayList`. The methods you create are all geared toward cards, and should be as intuitive as possible. However, internally they will all rely on the `ArrayList`, which should be private.

Since this is another object, you should use the formal Javadoc-style comments.

For this homework to be completed, you need to implement the following public functions:

- `CardSet()`, which makes a new empty `CardSet`.
- `String toString()`, which returns a `String` that is a comma-separated list of the `Cards`, inside parentheses.
- `void add(Card card)`, which adds the given `Card` to the end of the `CardSet`.
- `Card discard(int whichCard)`, that removes the `Card` at the given index and returns it.
- `int getSize()`, that returns the number of `Cards` in the hand.
- `void shuffle()`, that shuffles the `Cards` using `Collections.shuffle()`.

You should also implement the following public static methods:

- `CardSet makeStandardDeck()`, which returns a brand-new `CardSet` that contains 52 `Cards`: one of each type. (This function will probably be the last thing you write. It may use some of the above methods.)
- A `main()` method for unit testing.

Of course, you may also make any number of private functions, if you feel that they will be useful.

Be sure that your `CardSet` object compiles and runs with the `TestCards` program on the web page. Your output should look similar to the following:

```
Empty hand is: ()
Hand of size 5: (2 of Clubs, Queen of Hearts, 10 of Spades, Queen of
Diamonds, 2 of Clubs)
Discarded the 2 of Clubs and the 2 of Clubs.
```

Hand of size 3: (Queen of Hearts, 10 of Spades, Queen of Diamonds)
Same hand, shuffled: (Queen of Diamonds, Queen of Hearts, 10 of Spades)
Here's a full deck:

(Ace of Clubs, 2 of Clubs, 3 of Clubs, 4 of Clubs, 5 of Clubs, 6 of Clubs, 7 of Clubs, 8 of Clubs, 9 of Clubs, 10 of Clubs, Jack of Clubs, Queen of Clubs, King of Clubs, Ace of Diamonds, 2 of Diamonds, 3 of Diamonds, 4 of Diamonds, 5 of Diamonds, 6 of Diamonds, 7 of Diamonds, 8 of Diamonds, 9 of Diamonds, 10 of Diamonds, Jack of Diamonds, Queen of Diamonds, King of Diamonds, Ace of Spades, 2 of Spades, 3 of Spades, 4 of Spades, 5 of Spades, 6 of Spades, 7 of Spades, 8 of Spades, 9 of Spades, 10 of Spades, Jack of Spades, Queen of Spades, King of Spades, Ace of Hearts, 2 of Hearts, 3 of Hearts, 4 of Hearts, 5 of Hearts, 6 of Hearts, 7 of Hearts, 8 of Hearts, 9 of Hearts, 10 of Hearts, Jack of Hearts, Queen of Hearts, King of Hearts)

Keep in mind that the first two cards are random, and shuffling is random, so yours will differ.

Extra Credit You may earn a modest amount of extra credit, by adding any of the following methods to your `CardSet` (and demonstrating them, using your own version of `TestCards`).

- A function that returns the rank of the highest card.
- A function to tell if the set is a flush (i.e. contains all cards of the same suit).
- A function to tell if there are duplicate cards (same rank and suit) in the hand. Such a function might return an array of two integers, which hold the indices of the cards, or null if there are no duplicates.
- A function to tell if there is a pair (2 cards with the same rank), that returns the rank of the pair if it exists, or 0 if it doesn't.
- Other functions to detect other hands in poker (e.g. full house, straight, two pair, royal flush, etc.). These will be trickier than the functions mentioned above.

To get full credit, these functions should not alter the `CardSet` in any way! Be sure to tell us exactly which extra functions you implemented so you get credited properly.

