Junior Knights Homework: Simplified War

Rules of Simplified War

The game of war uses a standard deck of cards (4 suits: Clubs, Diamonds, Hearts, Spades, 13 kinds: 2, 3, …, 10, Jack, Queen, King, Ace). In the simplified game, we have n players play, where n divides evenly into 52. During a single round, each player plays the card on the top of their stack. Whichever player has the highest ranking card wins all of the cards played for that round and puts them on the bottom of their stack (any order for these is fine as long as they go on the bottom.)

We define the highest card as follows:

First, we break ties based on kind. The order of the kinds are as follows:

Ace > King > Queen > Jack > 10 > 9 > … > 2

If there are two cards of the same kind, we break the tie based on suit with the following ordering:

Spades > Hearts > Diamonds > Clubs

Notice that this is reverse alphabetical ordering from best suit to worst (or regular alpha ordering from worst suit to best!)

A whole game consists of playing rounds until all players except one run out of cards. When I player goes to 0 cards, they exit the game. When players exit the game, they will receive a score: the number of rounds they won. When this occurs, your program should print out a player's score upon exiting the game.

What You Will Be Doing

Given a description of each method in the Card and CollectionOfCards classes, write each method. The Card class should implement Comparable. Both classes have a testing main. When you've written your methods correctly, the testing main should output the desired results. Finally, write your main program in the War class. This class may have multiple static methods, if you feel they are necessary.

Listing of Static Variables, Instance Variables and Methods for Card

```
final public static String[] SUITS = {"Clubs", "Diamonds",
"Hearts", "Spades"};

final public static String[] KINDS = {"Two", "Three", "Four",
"Five", "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen",
"King", "Ace"};

private int suit;
private int kind;

// Creates a card with the suit mySuit and kind myKind as
// designated by the constant arrays SUITS and KINDS.
public Card(int mySuit, int myKind);


// Returns a String representation of a Card
public String toString();

// Returns a negative integer if this Card comes before (is
// lower than) other, 0 if they are equal, and a positive
// integer if it comes after (is higher than) other.
public int compareTo (Card other);
```

Listing of Methods for CollectionOfCards

```
private ArrayList<Card> myCards;
private int roundsWon;

// Creates an empty Collection of Cards.
public CollectionOfCards();

// Clears out this collection and fills it with all of the cards
// in the deck.
public void fillDeck();

// Adds c to the end of this collection.
public void addCard(Card c);

// Removes c from the top of this collection.
public Card removeFromTop();

// Returns a String representation of the collection.
public String toString();

// Adds a win of a round to this collection.
public void wonRound();
```