Junior Knights Homework: Simplified Uno

Rules of Simplified Uno

A Simplified Uno deck comes with 72 cards: Yellow, Red, Green and Blue are the four colors of cards and there are two copies of each number 1 through 9, for each color.

There are two players in the game.

The deck is shuffled and then each player is dealt 7 cards. The first player gets to discard any card they like into the discard pile. From that point on, the players alternate turns.

From this point on, a playable card is simply one that is either the same color or number as the top card (last card discarded) of the discard stack.

If the player has a "playable" card in his/her hand, then they are asked which card to play. If they choose one of the playable cards, it is removed from their hand and added to the top of the discard pile. Otherwise, if they choose a non-playable card, then they do NOT get to discard this card and they forfeit their turn.

The game continues until either player has discarded all of his/her cards or the initial deck is emptied. If a player gets rid of all their cards, they win at that point in time. If the initial deck is exhausted, the game finishes in a tie.

Run A Sample Version of the Game

Go to the list of homework assignments on the course webpage and go to the listing for Uno. Download the three following files:

UnoCard.class
CollectionOfUnoCards.class
Uno.class

Copy these files into a directory. (You can create a directory called uno and copy all three there.)

Then, open up a Command Prompt window. Change directories until you get to the one in which you copied the three files. (I'll help you with this in class.)

At the command prompt type:

```
java Uno
```

This should run the program so you can play =)

What You Will Be Doing

Given a description of each method in the UnoCard and CollectionOfUnoCards classes, write each method. Both classes have a testing main. When you've written your methods correctly, the testing main should output the desired results. Finally, the Uno class is given to you. So after you've tested both classes, just run the Uno class that is given to you!

Listing of Methods for UnoCard

```
// Creates a randomly generated UnoCard.
public UnoCard(Random r);

// Pre-conditions: c must be in between 0 and 3 inclusive,
//                 n must be in between 1 and 9, inclusive.
public UnoCard(int c, int n);

// Returns a String representation of a UnoCard. For example
// the green card with 7 on it should be represented as
// “Green 7”.
public String toString();

// Returns true iff other can be played on this card or
// vice versa.
public boolean canPlay(UnoCard other);
```

Listing of Methods for CollectionOfUnoCards

```
// Creates an empty Collections.
public CollectionOfUnoCards();

// Adds c to the end of this collection.
public boolean addCard(UnoCard c);

// Removes c from the end of this collection.
public UnoCard removeFromTop();

// Removes the card in the designated index and replaces it with
// the card that last at the time of the operation.
public UnoCard remove(int index);

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

// Shuffles the cards in this collection. (This can be achieved
// by repeatedly picking two cards at random and swapping them.
// A couple hundred times should suffice.
public void shuffle();
```

```
// Returns a String representation of the collection, as shown
// in the example output.
public String toString();

// Returns the number of cards in the collection.
public int getNumCards();
// Returns the top card in the collection without removing it.
public UnoCard getTopCard();

// Returns true if there exists at least one card in this
// collection that can be played on the card c.
public boolean canPlay(UnoCard c);

// Returns the card in location index of the collection.
public UnoCard getCard(int index);
```