

Project 5 – Card Game

Due Date: April 30, 2009 @ 11:55 p.m.

In this project, you will use the Card class you developed in Project #4 to implement a card game. You might need to change the **toString** function of the Card class to meet the requirement of this class (See the output please). Our version of the Card Game will allow two players to take turns to play the game according to the following rules:

- At the beginning of the game, each player will draw 5 cards from the deck and place them side by side faced up in their table.
- Then, the players play in turn. In each turn, a player can remove a card if such a card is
 - One of the three cards in a **flush**, i.e. three cards of the same suit (You can remove any one of the three cards in the **flush**)
 - Having the same **rank** as some other card on the table.(You can remove any one of the cards making a rank pair)
- If there is no **flush** and no rank pair and the deck has some cards left, then the player removes the lowest rank card with lowest suit. Otherwise, the player cannot remove any card from his/her table and passes his/her turn to the other player.

After a card has been removed from the player's table, the player shifts the cards in his/her table to fill in the gaps and then, if some cards are still left in the deck, the player will draw a card from the deck and place it to the right of the cards in his/her table.

The game ends when the **deck is empty and neither player can remove any cards from his/her table**. The score of a player in the game is the sum of ranks of all the cards in the player's table. The player with the **lower** score wins the game.

Three classes are required to implement this project.

1. Card Class:

Use the Card class you implemented in the previous project.

2. DeckOfCards Class:

A **DeckOfCards** object represents a deck of cards in a card set.

The class **DeckOfCards** has two instance attributes:

- **deck**: an array of size 52 elements of type Card. This array stores the cards available in the deck.
- **nCards** – an integer which keeps track of the number of cards that have not been drawn from the deck of cards at a given time.

The class **DeckOfCards** must contain the following methods:

- **public DeckOfCards()** - a constructor that creates an array of 52 cards, and puts each of the 52 cards into the array. It also initializes **nCards** to the index of the last card in the deck, i.e. 52.
- **public void shuffle()** – a method to shuffle the elements in the array.
- **public Card draw()** – If there are some cards available in the deck, the method returns the first card available in the deck and updates the instance variable **nCards** properly. If all the cards have been drawn from the deck, the method should return the **null** reference.
- **public boolean isEmpty()** – returns true if there is no cards left in the deck and false otherwise.
- **public int cardsInDeck()** – returns the number of cards available in the deck.
- **public String toString()** - this method returns a String representation of the deck of cards. The string should be the content of the deck of cards in its current organization without including the drawn cards, printed out 4 cards per line.

3. Player Class:

The class Player has three instance attributes:

- **name**: a string that represents the player's name.
- **table**: an array of size 5 elements of type Card. This array stores the cards on the player's table.
- **nCards** – an integer which keeps track of the number of cards in the player's table.

Additionally, the class Player implements the following instance methods:

public Player(String name, DeckOfCards cardDeck) – the constructor of the class takes a String that represents the name of the player and a DeckOfCards object. The constructor will draw 5 cards from the DeckOfCards to fill the player's table. If the DeckOfCards does not have available as many cards, the constructor will fill the player's table with the cards left in the deck.

public String getName(): returns the name of the player.

public int getNumberOfCards(): returns the number of cards in the player's table.

public int numOfSameSuit(int suit): returns the number of cards in the player's table that have the same suit as the parameter.

public int numOfSameRank(int rank): returns the number of cards in the player's table that have the same rank as the parameter.

private boolean replace(int index, DeckOfCards cardDeck) – this method removes the card at the specified index from the player's table, if such an index is valid, i.e, it is the

index of one of the cards in the table. If so, the method shifts the cards on the right of the removed card one position over to the left to fill the gaps. Afterwards, if there are any cards left in the deck, the method draws a card from the deck and adds it to the right of the player's table. The method returns true if the card at the specified index has been removed and false otherwise.

This method can be used by the methods **removeLowest** and **remove** to implement the functionality of those methods according to the requirements (See below). Therefore it is private. You must invoke this method inside the **remove** and **removeLowest** to get the full credit.

public boolean remove(int index, DeckOfCards cardDeck) – this method performs the following actions only if the card in the player's table at position *index* is one of the cards from a flush or rank pair:

- removes the card.
- shifts all cards on the right of the removed one position over to the left to fill the gaps.
- if there are some cards available in the deck, draws a card from the deck and adds such a card to the right of the table.
- returns true.

The method returns false if the specified card in the player's table is neither a part of a Flush nor a part of a rank pair, or the index is not a valid index for a card in the player's table.

public int canRemove(): returns the number of cards that can be removed from the table, i.e. the method counts the number of cards in the player's table that are either part of a Flush or of some rank pair.

public boolean removeLowest(DeckOfCards cardDeck) – this method removes the lowest ranked card with lower suit from the player's table when there is not a single card which is a part of some Flush or rank pair, then shifts the cards to fill gaps. Afterwards, if there are cards left in the deck, the method draws a card from the deck and adds it to the right of the player's table. If there is a card in the player's table which is a part of some Flush or rank pair, the method does nothing and return false.

public int score() – this method calculates and returns the sum of ranks of all cards in the player's table.

public Card[] table() – this method returns a copy of the player's table. The method must create an array of cards whose length is equal to the number of cards in the player's table and populate this array with the same cards in the player's table. Finally, the method will return such an array.

public String toString(): returns a String representation of the player in which the name and the player's table are included.

Finally, you should implement a driver class **CardGame.java** to allow two players to play our Card game. The driver class should prompt the players for their names. Next, the players will start the game and play the Card game in alternating turns according to the rules explained above until neither player can remove more cards from their table. Next, the score of each player is computed and the program should declare as winner the player with lower score.

A run of the project should look like this:

```
Please enter the name of player #1:
Please enter a name with at least one character.
Please enter the name of player #1:James
Please enter the name of player #2:
Please enter a name with at least one character.
Please enter the name of player #2:Jack

James's turn:
James: K:Spades, 3:Diamonds, 5:Hearts, 4:Spades, 5:Spades
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: -2
The index must be a number between 0 and 4.
Please indicate the index of the card you want to remove: 10
The index must be a number between 0 and 4.
Please indicate the index of the card you want to remove: 1
Please try another card in the table.
Please indicate the index of the card you want to remove: 0
James: 3:Diamonds, 5:Hearts, 4:Spades, 5:Spades, 4:Club

Jack's turn:
Jack: A:Club, 3:Club, 7:Spades, 2:Spades, 2:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: A:Club, 3:Club, 7:Spades, 2:Spades, 9:Club

James's turn:
James: 3:Diamonds, 5:Hearts, 4:Spades, 5:Spades, 4:Club
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 1
James: 3:Diamonds, 4:Spades, 5:Spades, 4:Club, 9:Diamonds

Jack's turn:
Jack: A:Club, 3:Club, 7:Spades, 2:Spades, 9:Club
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 0
Jack: 3:Club, 7:Spades, 2:Spades, 9:Club, 8:Club

James's turn:
James: 3:Diamonds, 4:Spades, 5:Spades, 4:Club, 9:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 1
James: 3:Diamonds, 5:Spades, 4:Club, 9:Diamonds, 8:Hearts

Jack's turn:
Jack: 3:Club, 7:Spades, 2:Spades, 9:Club, 8:Club
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 3
Jack: 3:Club, 7:Spades, 2:Spades, 8:Club, K:Hearts

James's turn:
James: 3:Diamonds, 5:Spades, 4:Club, 9:Diamonds, 8:Hearts
```

You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 5:Spades, 4:Club, 9:Diamonds, 8:Hearts, A:Diamonds

Jack's turn:
Jack: 3:Club, 7:Spades, 2:Spades, 8:Club, K:Hearts
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
Jack: 3:Club, 7:Spades, 8:Club, K:Hearts, 6:Spades

James's turn:
James: 5:Spades, 4:Club, 9:Diamonds, 8:Hearts, A:Diamonds
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 5:Spades, 9:Diamonds, 8:Hearts, A:Diamonds, 10:Diamonds

Jack's turn:
Jack: 3:Club, 7:Spades, 8:Club, K:Hearts, 6:Spades
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
Jack: 7:Spades, 8:Club, K:Hearts, 6:Spades, 2:Hearts

James's turn:
James: 5:Spades, 9:Diamonds, 8:Hearts, A:Diamonds, 10:Diamonds
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 3
James: 5:Spades, 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club

Jack's turn:
Jack: 7:Spades, 8:Club, K:Hearts, 6:Spades, 2:Hearts
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
Jack: 7:Spades, 8:Club, K:Hearts, 6:Spades, K:Club

James's turn:
James: 5:Spades, 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club, 2:Club

Jack's turn:
Jack: 7:Spades, 8:Club, K:Hearts, 6:Spades, K:Club
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 2
Jack: 7:Spades, 8:Club, 6:Spades, K:Club, 3:Spades

James's turn:
James: 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club, 2:Club
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club, 10:Hearts

Jack's turn:
Jack: 7:Spades, 8:Club, 6:Spades, K:Club, 3:Spades
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 0
Jack: 8:Club, 6:Spades, K:Club, 3:Spades, Q:Spades

James's turn:
James: 9:Diamonds, 8:Hearts, 10:Diamonds, 7:Club, 10:Hearts
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 2
James: 9:Diamonds, 8:Hearts, 7:Club, 10:Hearts, J:Spades

Jack's turn:
Jack: 8:Club, 6:Spades, K:Club, 3:Spades, Q:Spades
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 8:Club, 6:Spades, K:Club, 3:Spades, 6:Hearts

James's turn:

James: 9:Diamonds, 8:Hearts, 7:Club, 10:Hearts, J:Spades
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Diamonds, 8:Hearts, 10:Hearts, J:Spades, 5:Club

Jack's turn:
Jack: 8:Club, 6:Spades, K:Club, 3:Spades, 6:Hearts
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 1
Jack: 8:Club, K:Club, 3:Spades, 6:Hearts, 6:Club

James's turn:
James: 9:Diamonds, 8:Hearts, 10:Hearts, J:Spades, 5:Club
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Diamonds, 8:Hearts, 10:Hearts, J:Spades, 9:Spades

Jack's turn:
Jack: 8:Club, K:Club, 3:Spades, 6:Hearts, 6:Club
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 1
Jack: 8:Club, 3:Spades, 6:Hearts, 6:Club, 9:Hearts

James's turn:
James: 9:Diamonds, 8:Hearts, 10:Hearts, J:Spades, 9:Spades
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 0
James: 8:Hearts, 10:Hearts, J:Spades, 9:Spades, 8:Spades

Jack's turn:
Jack: 8:Club, 3:Spades, 6:Hearts, 6:Club, 9:Hearts
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 2
Jack: 8:Club, 3:Spades, 6:Club, 9:Hearts, 4:Hearts

James's turn:
James: 8:Hearts, 10:Hearts, J:Spades, 9:Spades, 8:Spades
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 2
James: 8:Hearts, 10:Hearts, 9:Spades, 8:Spades, A:Spades

Jack's turn:
Jack: 8:Club, 3:Spades, 6:Club, 9:Hearts, 4:Hearts
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
Jack: 8:Club, 6:Club, 9:Hearts, 4:Hearts, J:Hearts

James's turn:
James: 8:Hearts, 10:Hearts, 9:Spades, 8:Spades, A:Spades
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 4
James: 8:Hearts, 10:Hearts, 9:Spades, 8:Spades, 3:Hearts

Jack's turn:
Jack: 8:Club, 6:Club, 9:Hearts, 4:Hearts, J:Hearts
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 2
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, A:Hearts

James's turn:
James: 8:Hearts, 10:Hearts, 9:Spades, 8:Spades, 3:Hearts
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 1
James: 8:Hearts, 9:Spades, 8:Spades, 3:Hearts, 10:Spades

Jack's turn:
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, A:Hearts
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 4:Diamonds

James's turn:
James: 8:Hearts, 9:Spades, 8:Spades, 3:Hearts, 10:Spades
You can remove 4 card(s) from the table.
Please indicate the index of the card you want to remove: 4
James: 8:Hearts, 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds

Jack's turn:
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 4:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 6:Diamonds

James's turn:
James: 8:Hearts, 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 0
James: 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds, 7:Hearts

Jack's turn:
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 6:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, Q:Club

James's turn:
James: 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds, 7:Hearts
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 0
Please try another card in the table
Please indicate the index of the card you want to remove: 4
James: 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds, J:Club

Jack's turn:
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, Q:Club
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 8:Diamonds

James's turn:
James: 9:Spades, 8:Spades, 3:Hearts, 7:Diamonds, J:Club
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, 5:Diamonds

Jack's turn:
Jack: 8:Club, 6:Club, 4:Hearts, J:Hearts, 8:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 0
Jack: 6:Club, 4:Hearts, J:Hearts, 8:Diamonds, Q:Hearts

James's turn:
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, 5:Diamonds
You can remove 0 card(s) from the table.
Removing the lowest card from the player's table
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, J:Diamonds

Jack's turn:
Jack: 6:Club, 4:Hearts, J:Hearts, 8:Diamonds, Q:Hearts
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 4
Jack: 6:Club, 4:Hearts, J:Hearts, 8:Diamonds, K:Diamonds

James's turn:
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, J:Diamonds
You can remove 2 card(s) from the table.
Please indicate the index of the card you want to remove: 4
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, 10:Club

Jack's turn:
Jack: 6:Club, 4:Hearts, J:Hearts, 8:Diamonds, K:Diamonds
You can remove 0 card(s) from the table.

```

Removing the lowest card from the player's table
Jack: 6:Club, J:Hearts, 8:Diamonds, K:Diamonds, Q:Diamonds

James's turn:
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, 10:Club
You can remove 0 card(s) from the table.
Pass. No card can be removed for the player's table and the deck is empty
James: 9:Spades, 8:Spades, 7:Diamonds, J:Club, 10:Club

Jack's turn:
Jack: 6:Club, J:Hearts, 8:Diamonds, K:Diamonds, Q:Diamonds
You can remove 3 card(s) from the table.
Please indicate the index of the card you want to remove: 3
Jack: 6:Club, J:Hearts, 8:Diamonds, Q:Diamonds

The game has terminated.
Player #1 ( James ) score: 45
Player #2 ( Jack ) score: 37
Congratulations Jack. You won!!!

```

Deliverables

- Card.java
- DeckOfCards.java
- Player.java
- CardGame.java

Hints

CardGame.java

- Use private static methods to implement the different parts of the game in the driver class CardGame.java.
 - A private static method to allow the user to enter the name of a player of the game. The method should validate that the name of the player has at least one character. Otherwise, an error message will be displayed and the method will prompt the user for the player's name again. This process continues until the user provides a valid name for a player.
 - A private static method to allow the user to enter a valid index of a card in a player's table. The method will display an error if the index provided by the user is not a valid index and keep prompting the user for a card index until the user provides such.
 - A private method that allows a player to remove a card from his/her table which belongs to either a flush or a rank pair. The method should prompt the user for the index of the card (s)he wants to remove, instructs the player to remove the specified card. If the card was not removed, the method will display an error and prompt the player for the index of another card in the player's table. This process continues until the player removes one of the card belonging to either a flush or a rank pair.
- You may use an array of two Players to store the players of the game.
- Use an [int](#) variable to determine which player will play next. The variable should take two values 0 for the first player and 1 for the second player. After a player has completed

his/her turn, increase the value of the variable by 1 and compute the remainder of dividing this result by 2.

```
int turn = 0;
System.out.println("X's turn: ");    // X plays
...
turn=(turn + 1) % 2;
System.out.println("Y's turn: ");    // Y plays
...
turn =(turn + 1) % 2;
```