Ejercicios Arrays of Objects

Programación II

Resuelva los siguientes ejercicios extraídos del capítulo 14 del libro ThinkJava:

55) The goal of this exercise is to implement the shuffling and sorting algorithms from this chapter.

1. Download the code from this chapter from http://thinkapjava.com/code/Card2.java and import it into your development environment. I have provided outlines for the methods you will write, so the program should compile. But when it runs it prints messages indicating that the empty methods are not working. When you fill them in correctly, the messages should go away.
2. If you did Exercise 12.3, you already wrote randomInt. Otherwise, write it now and add code to test it.
3. Write a method called swapCards that takes a deck (array of cards) and two indices, and that switches the cards at those two locations.

   HINT: it should switch references, not the contents of the objects. This is faster; also, it correctly handles the case where cards are aliased.

4. Write a method called shuffleDeck that uses the algorithm in Sec- tion 14.2. You might want to use the randomInt method from Exer- cise 12.3.
5. Write a method called indexLowestCard that uses the compareCard method to find the lowest card in a given range of the deck (from lowIndex to highIndex, including both).
6. Write a method called sortDeck that arranges a deck of cards from lowest to highest.
7. Using the pseudocode in Section 14.6, write the method called merge. Be sure to test it before trying to use it as part of a mergeSort.
8. Write the simple version of mergeSort, the one that divides the deck in half, uses sortDeck to sort the two halves, and uses merge to create a new, fully-sorted deck.
9. Write the fully recursive version of mergeSort. Remember that sortDeck is a modifier and mergeSort is a function, which means that they get invoked differently:

```
sortDeck(deck);          // modifies existing deck
deck = mergeSort(deck);     // replaces old deck with new
```