CECS 274 – Project 3

Create a program that reads in a list of contact information from a text file and stores it in a Linked List of Contacts. Allow the user to add, remove, modify, or search for Contacts.

Contact class:

- Data Members (Strings): last name, first name, phone number, address, city, zip code.
- Methods:
    1. Constructor - pass in first and last name (all other fields are blank).
    2. Constructor - pass in all data and set them.
    3. get and sets - for all data members.
    4. toString - all data on one line in the same format as the file.
    5. equals - compares by first and last name.
    6. compareTo - compare by last name, if the last names are the same then compare by first name).

Node and Linked List class (similar to the lecture notes, modified to use Contacts):

- Methods to add:
    1. remove – pass in the first and last name, use the Contact's equals method to find the Contact to remove.
    2. searchName - pass in the full name, compare using the Contact's equals method, return the first matching Contact.
    3. searchName - pass in a last name, returns an ArrayList of matching Contacts.
    4. searchZip - pass in a zip code, returns an ArrayList of matching Contacts.
    5. sort - use a Bubble Sort that uses the Contact's compareTo() function to sort the list. Do not pass in the list to the function and don't call get().
    6. toString - return the list of Contacts sorted and enumerated as a String.
    7. toFile - return the list of Contacts as a String suitable for writing the contents of the linked list to the file.

Main class – create a Linked List of Contacts that is populated with "contacts.txt".

- Allow the user to choose from the following functions:
    1. Add - prompt the user for the Contact's data, then add the Contact to the list.
    2. Remove
        a. by full name - if found, then remove the Contact from the list.
        b. by index - display the enumerated list, and remove the Contact.
    3. Search
        c. by last name - display all Contacts with the same last name.
        d. by zip code - display all Contacts with the same zip code.
    4. Update - prompt the user to enter Contact's full name, find the Contact using the search by full name. If it exists, ask the user to choose which field they want to update, along with the new information, then update the Contact.
    5. Display - print an enumerated list of the sorted Contacts.
    6. Quit - write the list of Contacts back to the file using the toFile function.

Notes:

- Create a main menu and any necessary sub-menus so the user can choose from the different possible functions. Loop until the user decides to quit the program.
- Check all user input for validity.
- Javadoc all classes, data members, methods, @params, and @returns.