CISC 3130 Ari Mermelstein

Homework #2:

In this homework assignment, you will implement a Contact List class. This assignment (and some subsequent ones) will use the Contact class from homework 1.

Your ContactList should use an ArrayList (from the java collections hierarchy) as the underlying structure. Your ContactList class must implement the Iterable<Contact> interface(!). The Contacts stored in your ArrayList should be stored in sorted order by the compareTo() method of the Contact class.

Your ContactList should have the following functionality:

Constructors:

- A constructor that creates an empty ContactList.
- A constructor that uses an array of Contacts in order to initialize the ContactList,

  ContactList(Contact [ ] contacts);

  HINT: You can use the Arrays.asList() method described in class.

Operations:

- A method that searches for a particular contact by last name, and returns a reference to the Contact. If there is no such Contact, a sentinel value (e.g. a default valued Contact or null) should be returned. Since the list is in sorted order, this method should be a binary search of the list.

  Contact findByLastName(String last);

- A method that searches for a Contact by phone number, and returns a reference to the Contact. If there is no such Contact, a sentinel value should be returned.

  Contact findByPhoneNumber(String phone);

- A method that searches for and returns a ContactList containing all Contacts with a last name starting with a particular letter. If there are no such Contacts, you should return the empty ContactList.

  ContactList findAllByLastInitial(char ch);

- A method that searches for and returns a ContactList containing all Contacts that live in a particular city. If there are no such Contacts, you should return the empty ContactList.

  ContactList findAllByCity(String city);

- A method that allows you to add a Contact to the ContactList **in sorted order**. You should only add a Contact to the list if it is not there already. Use a modified binary search to determine this. The method returns **true** if the add is successful and **false** if it is unsuccessful.

  boolean add(Contact c);

- A method that returns the size of the ContactList.

  int size();

- A method that allows you to remove and return a Contact from the ContactList.

  Contact remove(Object obj);

- A method that allows the client to get a Contact from the ContactList by index. An IndexOutOfBoundsException should be thrown if that index doesn't exist.

  Contact get(int index);

- An overridden equals() method. Let's define one ContactList being equal to another if they contain the same Contacts in the same order.

  boolean equals(Object obj);

- An overridden toString() method that creates a representation for a ContactList.

  String toString();

- An iterator() method that allows you to iterate through a ContactList. (You can implement your own or use the ArrayList's own iterator).

Iterator<Contact> iterator();

.You must also write a test program for the ContactList class.