

CSE 1322L - Lab 3

Introduction

In this lab, you will write a simple program which will allow the user to add, remove, list, and search entries in a phonebook. Each entry will contain the contact's name and their phone number.

Because we do not know how many contacts the user will want to store, we will use an Arraylist to store all contacts, as Arraylists can expand and contract automatically at runtime. However, since we do know how much information the program needs to hold each contact, contacts will be saved as arrays.

All names and phone numbers should be handled as strings.

Requirements

The features described below must be in your program:

- Outside of your main loop, create an arraylist to hold **arrays of strings** called "phonebook"
- In the main loop, implement the following menu options:
 1. **Add contact:** Prompts the user for a name and a phone number. Store both of them in an array of strings and then store the array in the arraylist. Note that the name must be stored in the first index and the phone must be stored in the second index. Your arrays should not have a size different than 2.
 2. **Remove contact:** Prompts the user for a name and then traverses the phonebook. If an entry is found whose name **matches exactly** what the user entered, remove said entry from the phonebook. If no contacts match, print instead "No contact with that name". Note that, in case more than one contact matches the user's input, only the first one should be deleted.
 3. **List contacts:** Lists all contacts in the phonebook, one per line, in the following format (where X is the contact's name and Y is their phone number):

Name: X | Phone: Y

4. **Search contacts:** Prompts the user for a keyword and then traverses the phonebook. Any contacts whose name **contains** the keyword should have their name and phone number printed out as above. If no contacts contain the keyword, print instead "No contacts contained the keyword".
5. **Quit:** Terminates the program

Deliverables

Lab3.java

Considerations

- You will need to make use of two string methods. You can find a list of them in the [String documentation](#).
 - Manually checking character by character if a string is contained in another string or if two strings match is possible, but will be laborious and could potentially introduce bugs to your code. You are encouraged to use the appropriate string methods.
- Remember that using the equality operator (==) to check if two strings have the same content will usually lead to incorrect results. Use the equality method instead.
- You will need to make use of two or three arraylist methods. You can find a list of them in the [Arraylist documentation](#).
- To specify that something is holding an array of a type, you must follow the type declaration with square brackets, as below:

```
int age;           // Normal integer
int[] ages;       // Array of integers
```

- Arrays need to have their sizes specified at initialization:

```
int[] ages = new int[10]    // An array of integers of size 10
```

- You can interact with a particular element in the array by using its index. Trying to access an index outside the bounds of the array will cause your program to crash.

```
int[] ages = new int[10];    // Create the array
ages[0] = 30;                // Writes 30 to position 0
System.out.println(ages[0]); // Reads the element at position 0
System.out.println(ages[10]); // Out of bounds. Will crash
```

- Remember that Arraylists need to specify the data type that they are holding at declaration time, inside the angle brackets (<>). Furthermore, Arraylists are not allowed to hold primitive types. As such, if you want to have your ArrayList hold a primitive type, you must instead use the wrapper class for that primitive, as below:

```
ArrayList<int> ages = new ArrayList<>();    // INCORRECT
ArrayList<Integer> ages = new ArrayList<>(); // CORRECT
```

Sample Output (user input in red)

[Phonebook]

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **3**

The phonebook is empty.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **1**

Enter the contact's name: **Alice**

Enter the contact's phone number: **404-111-0001**

Contact added.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **1**

Enter the contact's name: **Bob**

Enter the contact's phone number: **404-222-0002**

Contact added.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **1**

Enter the contact's name: **Charlie**

Enter the contact's phone number: **404-333-0003**

Contact added.

1. Add contact
2. Remove contact

3. List contacts
4. Search contacts
5. Quit

Enter option: **1**

Enter the contact's name: **Alice Cobb**

Enter the contact's phone number: **404-444-0004**

Contact added.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **1**

Enter the contact's name: **Charlie**

Enter the contact's phone number: **404-555-0005**

Contact added.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **3**

Listing all contacts...

Name: Alice | Phone: 404-111-0001

Name: Bob | Phone: 404-222-0002

Name: Charlie | Phone: 404-333-0003

Name: Alice Cobb | Phone: 404-444-0004

Name: Charlie | Phone: 404-555-0005

Done listing contacts.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **4**

Enter keyword to search: **ob**

Searching all contacts for keyword...

Name: Bob | Phone: 404-222-0002

Name: Alice Cobb | Phone: 404-444-0004

Done searching keyword.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **4**

Enter keyword to search: **Eve**

Searching all contacts for keyword...

No contacts contained the keyword.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **2**

Enter contact to remove: **Charlie**

Contact deleted.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **3**

Listing all contacts...

Name: Alice | Phone: 404-111-0001

Name: Bob | Phone: 404-222-0002

Name: Alice Cobb | Phone: 404-444-0004

Name: Charlie | Phone: 404-555-0005

Done listing contacts.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: **2**

Enter contact to remove: **Alice Cob**

No contact with that name.

1. Add contact
2. Remove contact
3. List contacts
4. Search contacts
5. Quit

Enter option: 5
Shutting off...