

Faculty Of Engineering

Presented by

Mohamed Brary



PHONE BOOK

Dr \ Noha Adly



Java project

Telephone Book

Project

It is required to write a program that will create and maintain a phone directory. Each entry of the directory will include the name – last and first, the address, city, and the phone number. Entries should be sorted according to last name then first then phone number. Your program should allow the user to execute the following commands:

1- LOAD : This command reads in a file name and loads the phone directory from this file. The file is a text file that is comma delimited in which each entry is found on a separate line. Each entry must contain the last name, first name, street address, city and phone number. An example of one line in the file is as follows:

Mahmoud, Ahmed, 26 Elhoreya Street, Alexandria, 4876321

2- QUERY : The system should process a request by the user to look up information about a specific entry. The user must supply the person's last name, and the system should provide the first name, address, city and phone number for all users of that last name.

3- ADD : The system should prompt the user field by field for the data of a single record and add it to the directory.

4- DELETE : The user should be prompted for the name – last and first and a record associated with that name should be deleted from the directory.

5- MODIFY : The user should be prompted for a last name. Then the user should be prompted field by field to modify the information for one of the records (selected by user) that have that last name.

6- PRINT : Print the entire directory, in sorted order.

7- SAVE : Save the directory by writing it out to an external file in a format similar to that explained in the Load command.

8- QUIT : Exit (without saving the directory in the external file).

Language used : java .
IDE used : eclipse 3.2

Classes:

1. Contact.
2. Group.
3. User.
4. Method.
5. Main.

Contact:

1. This class creates object Contact and contains all methods that deals with this object.
2. Object Contact represents a simple phone contact.
3. Object Contact attributes :
 - a. Last name of the contact.
 - b. First name of the contact.
 - c. Address of the contact.
 - d. City of the contact.
 - e. Phone number of the contact
4. Main methods:

ToString() Methods:

Put the data of the contact into a string ,every method of them put the data in a different way.

getContactData() :

- Prompt the user to fill the contact details .
- Last name ,first name and phone number must be provided by the user but address and city are optional and if they aren't given they will be set as "??".
- This method validates the input and removes any extra spaces in the entry.

insertContact() :

Takes a contact list and puts in it the contact placed in its sorted place.

Group:

1. This class creates object Group and contains all methods that deals with this object.
2. Object Group represents a group of contacts.
3. Object Group attributes :
 - i. User name : the user which this group of contacts belongs to.
 - ii. Group name : name of the group.
 - iii. Contact list : list of contacts at this group
 - iv. Path : path of the file *.txt that contains all the contacts.
4. Main Methods:

newGroup() :

- Prompts the user for the group name ,and checks if it is existed.
- Creates new file *.txt with the name provided by the user.

saveGroup() :

- Writes the contact list in the file created before.
- This method have 2 options : writes the contacts in a certain code to keep the privacy of the user as in (option Save and Save as) , or writes the contacts simply to be printed or so ever as in (Export file option) .

loadGroup() :

- Reads the contact list from a file created before by this project and put them in an array to be ready to operate on them.
- This method reads the coded contacts and decodes them .

addContact() :

- Prompts the user for number of contacts he\she wants to add to his\her group of contacts.
- Expands the array to be ready to receive more contacts.
- Prompts the user to fill the contacts details and checks the input.
- Checks if the contact with the same last name, first name and phone number already exists and rejects it.
- Puts the valid contact in its sorted place in the list.
- The user can go back to menu at any time by writing the escape word " *back " in last name field .

getModDelData () :

- Prompts the user for the last name of the wanted contact.
- This method deals with 5 cases:
 1. Case 1 :
If there isn't any contact in the group.
 2. Case 2 :
If there isn't any contact with this last name in the group.
 3. Case 3 :
If there is just one contact with this last name it asks the user for confirmation to operate on this contact.
 4. Case 4 :
If there are less than 10 contacts with this last name , it prints them and asks the user to choose from them.
 5. Case 5 :
If there are more than 10 contacts with this last name , it gives the user 2 choices even to view all or to search on result by first name then it does one of the cases (2 , 3).
- In case of deletion the user can choose to delete all the contacts resulted from the search or delete just one contact.
- This method uses binary search algorithm in searching.
- The user can go back to menu at any time by writing the escape word " *back " in last name field .

`print()` :

Prints all the contacts in the group on the screen , giving the user three forms to choose from them.

`query()` :

- Searches on the group by last name (binary), first name or phone number (linear) and prints the result on the screen.



User:

1. This class creates object User and contains all methods that deals with this object.
2. Object User represents a group of groups of contacts belongs to the same person.
3. Object User attributes :
 - User name : the user name which these groups of contacts belongs to.
 - Pass word : user's password to keep his privacy.
 - Contact list : list of contacts at this group
 - Path : path of the folder that contains the user groups.
4. Main Methods:

newUser() :

- Prompts the user for his name ,and checks if it is existed.
- Creates new folder with the name provided by the user which will contain all the txt files presenting this user groups.
- Prompts the user for his password to raise the level of privacy and to protect his personal info.
- Creates a file txt named "password" contains the user password written in certain code.

loadUser() :

- Prints all users created before and asks the user to choose from them.
- Then ask him\her for the password and compares it with the saved one , if it is correct the user will be given an access to all groups in this user.

getAllUsers() :

- Prints to the screen all users created before.

getAllGroups() :

- Prints to the screen all groups created before by the loaded user.

Method :

This class contains all the static methods used in this project.

Main Methods:

search() : it searches for the last name , last name & first name or last name & first name & phone number in the contact list by binary search .

** The next 2 methods handle all the input from the user in the whole project.

getInput() :

This method takes 5 kinds of inputs from the user, and checks it and makes sure it is a valid input.

Case 1 :

If the input must be a string and the user must enter something.

Case 2 :

If the input must be a number.

Case 3 :

If the input must be "y "or "n" .

Case 4 :

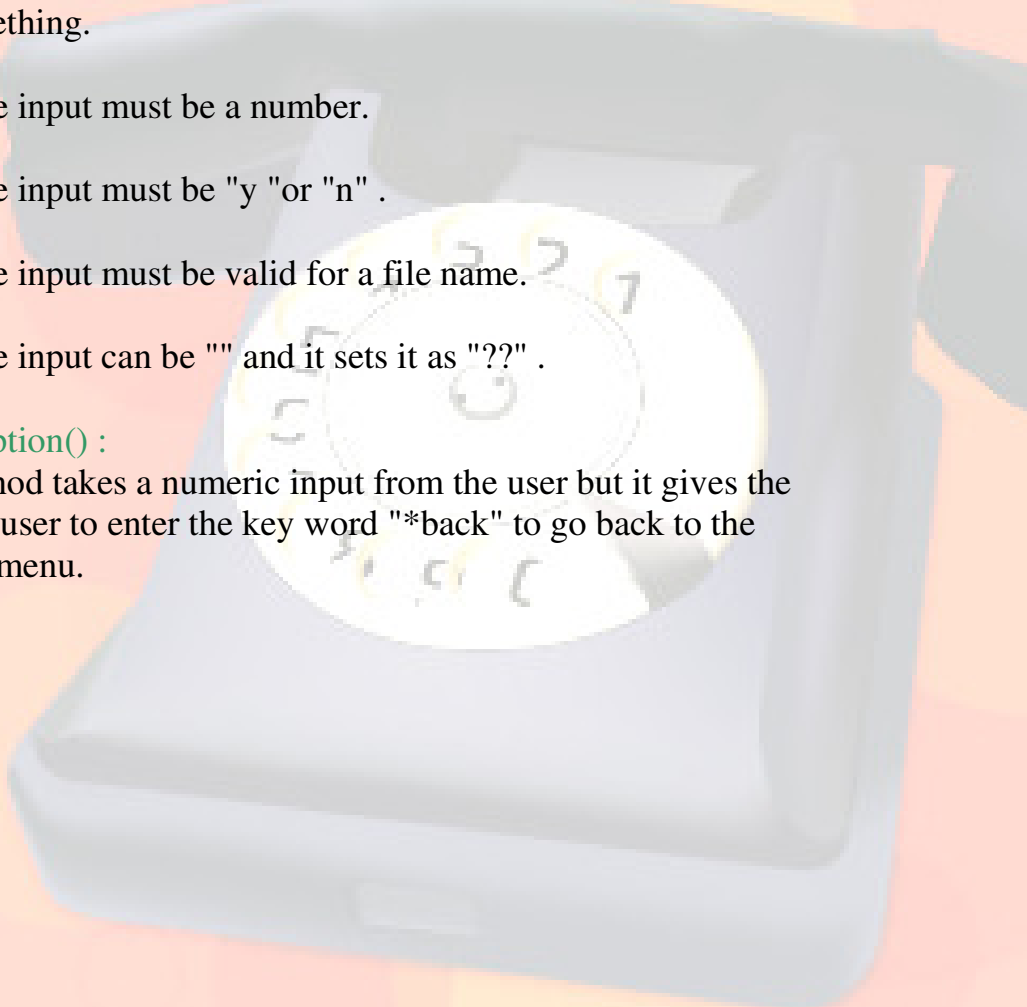
If the input must be valid for a file name.

Case 5 :

If the input can be "" and it sets it as "??".

getInputBackOption() :

This method takes a numeric input from the user but it gives the option to user to enter the key word "*back" to go back to the previous menu.



** The next 3 methods deals with arrays :

shrinkArray() : copies the array and decrease its length ignoring the last unwanted rows.

expandArray() : copies the array and increase its length to be able to house more elements .

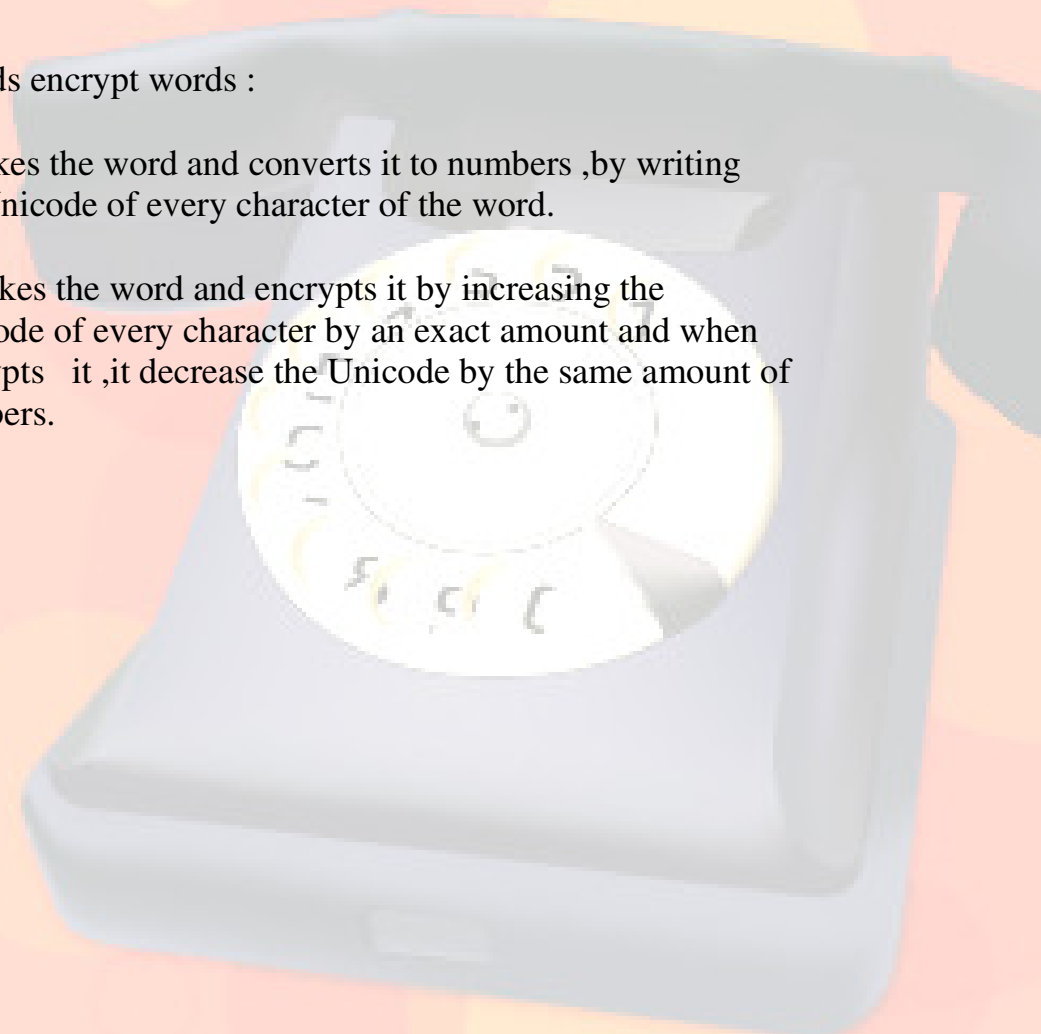
deleteRow() : deletes the wanted row from the array.

deleteRows() : deletes number of following rows from the arrays.

** The next 2 methods encrypt words :

encode() : it takes the word and converts it to numbers ,by writing the Unicode of every character of the word.

encrypt() : it takes the word and encrypts it by increasing the Unicode of every character by an exact amount and when decrypts it ,it decrease the Unicode by the same amount of numbers.



Main Features

```
*****
Welcome to my Phone Book !
I hope you enjoy using it ..
From author: Mohamed Brary ..
*****
```

This program represent a digital phone book , you can create groups of contacts , add, modify ,delete , search for contacts .

```
There are 3 Users:
    User 1 :Mohamed
    User 2 :Privacy
    User 3 :test
If you want to go back to menu write "**back" .
Please, write the number of user:2

If you dont know the password write "**back" .
Password : blah blah
Invalid password try again .
Password : PassWord_Keep_Privacy
The user "Privacy" has been loaded .....
```

Most important feature of this program is the high level of privacy provided to user.



Even source files of the program are encrypted as shown in the picture to maintain the level of privacy .. the two files indicate to the same contacts.

Other main feature is user interface in other words making the user comfortable is a priority.

```
There are 5 groups:
  Group 1 :9
  Group 2 :Kebeeeeer aweeeeeeeeee
  Group 3 :Kotofotos
  Group 4 :Test Encrypted
  Group 5 :Too much
If you want to go back to menu write "*back" .
Please, write the number of group that you want to load :
```

The program does most of the work and just asks the user for confirmation and most of the input from the user is just numbers .

```
Please, enter the name of contact you want to delete :
  Last name : Brary

There are too much contacts with this last name :
  1- View all.
  2- Search on these contacts by first name.
Choose from those commands (Write its number):
```

The program provides the user with various options to choose from as that our target is to please the user without tiring or confusing him\her .

```
Please, enter number of contacts you want to add to the group :100
If you want to go back enter "*back" in "Last name" field.
  " * fields are optional "
Please, enter the following for contact 1 :
  Last name : *back
```

In almost all cases the user can go back to the previous menu by entering the key word "*back" .

User guide

```
*****
Welcome to my Phone Book !
I hope you enjoy using it ..
From author: Mohamed Brary ..
*****
```

This program represent a digital phone book , you can create groups of contacts , add, modify ,delete , search for contacts .

```
User options menu:
*****
      1- New   : Creates a new user.
      2- Open  : if you already have a user.
      3- Dir   : To see all users.
      4- Quit  : Exiting program.

*****

Choose from those commands (Write its number 1 ,2 ,3 or 4):
```

First menu "User menu" :

- Every user of the hosting computer can create a personal user on this phone book keeping all groups of contacts he has.
- Every user has a password to keep his privacy.
- In this menu you got two options: create new user or open if you already have one.
- You can't delete an existing user but you can do it manually by going to "C:\ My PhoneBook" and delete the folder named after the user you want to delete.

Group options menu:

- 1- New : Creates a new group of contacts.
- 2- Load : Opens an existed group of contacts.
- 3- Import : Imports a group of contacts made by other program.
- 4- Delete : Deletes an existed group of contacts.
- 5- Dir : To see all groups.
- 6- Quit : Exiting program.

Choose from those commands (Write its number):

Second menu "Group menu":

- Every user can have more than one group of contacts for example : " Family , Work , Friends ".
- You can create new group or load group you made before.
- You can import a file text contains contacts in this form:
"last name, first name, street address, city, phone number"
for example:
Mahmoud,Mohamed,26 Elhoreya Street,Alex, 4876321.
- In case of importing file you will have to enter the full path of the file as in the following form:
"C:\Folder name\File name.txt"
- If the imported file contains duplicated or invalid contacts the program can fix that by skipping those contacts and continuing loading the rest of the file.

Contact options menu:

- 1- Add : To add contacts.
- 2- Query : To search for an existing contact.
- 3- Delete : To delete an existing contact.
- 4- Modify : To modify an existing contact.
- 5- Print : To print all contacts sorted in the group.
- 6- Save : Saves changes made in the group.
- 7- Save as: Saves the group with other name.
- 8- Export : Saves the group to be ready for print.
- 9- Back : To go back to the User options menu.
- 10- Quit : Exiting without saving the changes.

Choose from those commands (Write its number):

Third menu "Contacts menu" :

- There are eight options to choose from , all operating on the loaded group of contacts.
- You can go back to "Group menu" or quit the program at any time, and in case you don't want to save the changes you made don't worry just go without saving.
- If you aren't sure about the changes you made don't take the risk and save and overwrite the old one, just choose "Save as" option to make a copy of the group with other name, and continue editing the main group.
- You can export the group into a file text contains contacts in this form:
"last name, first name, street address, city, phone number"
to be ready to be printed or whatsoever and it will be created at
: "C:\file name.txt"