Project 0: ASEAN Phonebook

Objectives

This project aims to:

- Utilize the array list data structure in storing contact data,
- Implement incomplete functions based on documentation strings,
- Test validity of implemented functions by running test files.

Overview

You are given a half-finished ASEAN Phonebook program where it has several classes and methods with missing implementation. Your job in this project is to complete the implementation of the ASEAN Phonebook program whose behavior and functionalities similarly follow everything illustrated in the document attached with this one.

The purpose of this document is to guide you on which functions to finish first and how each of these functions should behave.

Project File Structure

Firstly, download the `Project0_ArrayList.zip` file and extract the contents. Once this is done, it should have the following file structure:

```
|- Project0
|--- Contact.py
|--- ContactArray.py
|--- Main.py
|--- TestCountryPrint.py
|--- TestDelete.py
|--- TestInsert.py
|--- TestPrint.py
```

Figure 1. Project File Structure for Project 0 – Phonebook.

You can use the test files to check the validity of your ASEAN Phonebook program. Note that this is only surface-level testing, and your instructors will have additional hidden test cases on their own.

Storing Contact to Phonebook

When storing a contact into our ASEAN Phonebook, it is required that the new contact must be "alphabetically" ordered primarily by their last names. If two contacts share the last name, their first names should then be checked. Assume that no more than two contacts will share the same first name to simplify this process.

To accomplish this functionality of storing new contacts, you will need to complete the following functions:

- `ContactList.__findIndexInsertion()`
    - `Contact.compareNames()`
- `ContactList.insert()`
    - `ContactList.__adjustPhonebook()`
    - `ContactList.__increasePhonebookSize()`
- `ContactList.getContactAtIndex()`

The purpose of `ContactList.__findIndexInsertion()` is to find the insertion point of a new contact. It can find a node insertion point with the help of a helper function, `Contact.compareNames()`. The definition and the purpose of all these methods are thoroughly described inside their respective Python files.

The last method, `ContactList.getContactAtIndex()` is to verify whether you were able to successfully insert your contacts at the correct node.

As a tip, complete the `ContactList.__findIndexInsertion()` function and its helper functions first. The `ContactList.insert()` will depend on this function in inserting new contacts into the Phonebook. Moreover, `ContactList.insert()` will also depend on `__adjustPhonebook()` method to adjust the phonebook whenever a new contact is inserted into the array. The `ContactList.__increasePhonebookSize()` is a precautionary measure that increases the size of the phonebook whenever the storage size is full of contacts.

To test that your Phonebook program can safely store contacts, run the `TestInsert.py` file. For time completion, it is expected that it will take you 4 – 8 days to complete this functionality.

Printing the Phonebook

Printing the phonebook means printing all the contacts present in the current Phonebook. To accomplish this functionality, you will need to complete the following functions:

- `ContactList.__str__()`
- `ContactList.getContactBySurname()`

The `ContactList.__str__()` has one parameter aside from `self`, which is `f`. The `by` parameter is optional and can be ignored unless you want to accomplish the bonus points. The purpose of the `f` variable is to act as the filter when selecting country codes (this refers to ***Search by Country*** functionality). This is useful especially when the user wants to print contacts belong only to certain countries.

The challenge here is to make sure that the `ContactList.__str__()` can print not only the entire contact, but also with a filter all in the same function.

As a tare, make sure to finish printing all contacts first as the filtering part with country codes is just a complicated way of printing the ASEAN Phonebook.

To test that your Phonebook program can print the entire Phonebook, run the `TestPrint.py` file.

To test that your Phonebook program can print contacts based on a given country code filter, run the `TestCountryPrint.py` file.

For time completion, it is expected that it will take you 4 – 8 days to complete this functionality.

Deleting Contacts

Deleting contacts requires supplying a student number to the function. To accomplish this task, you must complete and/or adjust the following function(s)

- `ContactList.deleteContact()`
  - `ContactList.__adjustPhonebook()`

`ContactList.deleteContact()` should return the contact stored inside the phonebook. It is a required functionality as it will be part of the next problem. Make sure to decrease the size if something is being deleted from the ASEAN Phonebook, and adjust the phonebook accordingly using `ContactList.__adjustPhonebook()`.

You will have to use the same function in adjusting the phonebook whether you are deleting a contact or inserting a new contact.

To test the functionality of deleting a contact, run the `TestDelete.py` file.

For time completion, it is expected that it will take you 1 – 3 days to complete this functionality.

Editing Contact Information

Editing contacts are straightforward. However, this becomes complicated when we are editing a contact's surname as we have to re-arrange the Phonebook whenever this happens.

As such, editing contact information, specifically a contact's surname, will require you in completing the functionalities from the previous sections. This time, there will be no tips and test files. It will be up to you to determine which functionalities must be used when editing a contact's surname.

For time completion, it is expected that it will take you 2 – 6 days to complete this functionality.

Menu Program

The main program requires you to read the attached document, `Phonebook Functionality.pdf`, and understand its contents.

The main program must be implemented under Main.py file, together with the already defined functions: `showMenu()`, `convertChoices()`, and `receiveContactInfo()`. There is also a `MENUS` variable containing the set of menus that can be used together with the `showMenu()` function.

The output of Main.py should follow exactly as what is required (or at least 90% similar) in `Phonebook Functionality.pdf`.

Sample Output

Refer to the attached file, `Phonebook Functionality.pdf`, for the specific outputs of every functionality.

Limitations and Warnings

You are forbidden in using any lambda, and other built-in functions in Python except for the following: `len()`, `string.format()` , `input()`, and type casting functions like `int()`, `str()`, `etc.` Violating this note will result in significant sanctions imposed by your instructor.

You are also forbidden in creating additional functions/methods inside the `Contact`, `ContactArray`, and `Main` python files. The signature or definition of the function(s) and/or methods should also not be modified. This includes but not limited to:

- Function name
- Function parameter
- Function return type
- Class/Function documentation string
- Type hints

Violating this note will result in significant sanctions imposed by your instructor.

Any changes to the Test files, whether accidental or intentional, will result to a score of 0 on this project without additional considerations. These files should not be modified and should only be used as a measurement for progress of your development.

For any student with similar code submissions, ignoring variable/function/class name changes, white spacing, etc., your score will be divided upon how many other students is found to share such code similarity. Since you decide to submit the same output, then it is reasonable that the score will be divided upon each, and everyone involved in such practice.

Submission

You will submit a compressed file containing all the Python files required for this project. It should also include a text file describing the name of the students partaking in your group. Present your output FACE TO FACE on or before one (1) week before the midterm exam week. No due date extension, and no ***failed buzzer beater submission*** will be entertained by your instructors.

The name of the compressed file is:

GroupName_ProjectZeroArray.rar

For example:

JOSS_ProjectZeroArray.rar

Failure to follow this instruction will result in point deduction.