EECS 233 Programming Assignment #1
100 points
Due February 14, 2020 before 11:59pm

In this programming assignment, you will provide an abstract data type PhoneBook that can be used to represent sequences of phone book entries regardless of the implementation. Your PhoneBook will be a sequence of objects of class Person. You will also need to define class Person that is a tuple (String personID, String phoneNum).

The ADT PhoneBook must support the following operations:

- int size() - return the current size of the sequence.
- void insert(int i, Person person) – Insert a new component before the i-th component of the sequence (using 0 for the index of the first component). (Provide a reasonable action for the case when the sequence has fewer than i components, e.g., insert the new element at the end of the sequence)
- Person remove(int i) - Remove the i-th component of the sequence and return the object removed (Provide a reasonable action for the case when the sequence has fewer than i components, e.g., do nothing and return null since the element to be removed does not exist already).
- Person lookup(int i) – Return the i-th component of the sequence (raise an exception if the sequence has fewer than i components).

Provide two implementations of this ADT: a class PhBArrayList and a class PhBLinkedList. The first one must use an **array** to store the sequence of persons and the second a **singly linked list**. Notes:

- **Using built-in Java classes, such as ArrayList, LinkedList, or iterator, is not allowed.**
- Do not forget the issue of managing the "real estate" (i.e., the fixed size of the array) in the case of the PhBArrayList implementation. In other words, you will need to internally re-allocate the array as it grows or shrinks too much.

You will also need to augment your ADT with an PhBIterator ADT (interface) and implementation(s) that allow efficient traversal of PhoneBook sequences without violating the information hiding principle (as discussed in class).

Write a demo application (name it "Demo") utilizing this ADT. Your Demo class should include the following methods:

- A static method "void removeDuplicates(…) that accepts as arguments instances of either PhBArrayList or PhBLinkedList class and removes people that appear in both lists from one of the lists, so that the lists contain no people in common. **Note: this method should not be specific to either class that implements PhoneBook: the same method should work for instances of either class.**
- The main method where you would:
  - Create an instance of PhBArrayList sequence and an instance of PhBLinkedList sequence, insert several elements into both using the insert(int i, Person person) method. Make sure to include some elements with the same personID into both lists. Since your demo application does not care at which position you insert elements into the lists, please specify the positions that minimize the execution time for each list type. Explain in the comments your choice of the positions at which you insert your elements.
  - Print both sequences in a readable format;
  - Invoke the removeDuplicates method to remove items with common names from both lists
  - Print both resulting sequences in a readable format.

Submission and Grading:

Please use Java version 8 or later. You can check your version by typing "java -version" on the command line, and if the output starts with "1.8" or higher, you are OK. You can use any Java IDE you want, but in the end make sure your code compiles and executes from the command line (using "javac <list of your .java files separated by a space>" command to compile, and "java Demo" command to run, your application). See TAs if you need help with your environment.

The submissions will be evaluated on completeness, correctness, and clarity. Please provide sufficient comments in your source code to help the TAs read it. Please ***generate a single zip file*** *containing all your *.java files needed for this assignment and *.class files that your compiler produced. Name your zip file <your_case_id>_P1.zip (e.g., for me, it would be mxr136_P1.zip). Submit your zip file electronically to Canvas***.**

Grading:

- PhoneBook and PhBIterator ADT specification: 15 points
- PhBArrayList implementation: 20 points
- PhBLinkedList implementation: 20 points
- Demo application: 15 points
- Proper encapsulation/information hiding: 10 points
- Design and style: 10 points
    i. Style: 3 points. Are variable names descriptive and convey their purpose? Of course simple concepts like a loop variable do not require descriptive names; e.g., it is perfectly fine, even preferable, to use i for a loop variable. Is formatting clear and consistent?
    ii. Comments: 3 points. Comments should aid the reader to understand the code. Comments that restate what is already clear from the code are redundant and not helpful. Nor are comments that are not consistent with the code.
    iii. Design: 4 points. Are there lines that never execute? Inelegant constructions? Convoluted or unnecessarily inefficient ways to achieve some result?
- Runtime analysis: 10 points. For each method implementation, state in the comments its worst-case running time, using big-O or big-Theta notation.