CS 2401 Assignment #4

Due Date: Tuesday, February 28, 2017 11:59PM.

Objective: The goal of this assignment is to practice linked list.

Background: The national Security Agency (NSA) maintains databases of incidents, field reports, publicly available blogs, and news articles to track suspicious activities of persons of interest (PoI). By analyzing the data, intelligence analysts build connection profiles for the PoIs. Once connection profiles are built, each of the profile is saved in a file. NSA hires you to develop a software that reads a connection profile and allows the analyst to make necessary modifications.

Assignment: The NSA will provide any profile connection in the form of a text file. Every three lines in the file is reserved for the record of a person. These three lines respectively represent a long integer ID of a person, the name of the person, and an integer ranging from 0 to 5 indicating the threat level posed by a person. A sample connection profile file is provided below, which gives a connection profile between two PoIs, Dan Carte and Osama Laden. The full connection is: Dan Carte → Prio Notim → Hons Nohish → Diran Egrac → Osama Laden.

```
389114
Dan Carte
5
399012
Prio Notim
0
685015
Hons Nohish
3
179318
Diran Egrac
2
284139
Osama Laden
5
```

Your task is to read the file content, construct a singly linked list to store the records in a sequence they appear in the file, and allow flexibility to the analysts so that she/he can perform the following operations.

Operation 1. Print the linked list content on the terminal (ID, name, and threat level of the people in a sequence they appear in the linked list)
Operation 2. Search a person in the linked list with ID or name
Operation 3. Insert a new person in a particular location index
Operation 4. Swap two people in two specific location indices
Operation 5. Remove a record containing a specific ID

Operation 6. Remove all records with a certain threat level
Operation 7. Write the content of the linked list in an output file that has the same format as the input file.
Operation 8. Quit program.

Your program must start by prompting the user to provide the filename of the input connection profile. Then the program should load the information in a linked list and prompt the user to perform one of the eight operations listed above. Operations 2 to 7 will require more prompting to the user for operation-specific information.

Other requirements:

- You must write three classes: `POI`, `POIList`, and `AnalyzePOI`.
  - `POI.java`: The `POI` class must hold the variables for the ID (`long`), name (`String`), and threat level (`int`). All these variables must be `private`. `POI` must also have a variable of `POI` type that can be used to point to the next `POI` object. This variable should be `public`.
  - `POIList.java`: This class must contain a data variable named `head` of `POI` type, which should contain the head address of the constructed linked list. Each of the operations from 1 to 7 must be implemented in a method of this class.
  - `AnalyzePOI.java`: This is the only class that will contain a `main` method. This class will be responsible for prompting the user for the input filename, creating the linked list (a `POIList` object), and prompting for the seven listed operations. `AnalyzePOI` must call appropriate methods associated with the `POIList` object to perform operations 1 to 7. Prompts for operation specific-information must be implemented in the `POIList` class, not in `AnalyzePOI`.
- As stated earlier, your program must start by prompting the user to provide the filename of the input connection profile. You must handle the following exceptions. Do not quit the program if you have these exceptions, rather give an error message and prompt the user again for a correct input file.
  - If the file does not exist
  - If an expected ID in the file is not a long integer
  - If threat level is not in integer
  - If threat level is not an integer between 0 to 5 (threat level can be 0, or 5, or any integer between 0 and 5)
- Make sure that your program is not vulnerable to user's incorrect input regarding the operations.

Deliverables: You must submit three Java files for this assignment (`POI.java`, `POIList.java`, and `AnalyzePOI.java`) using Blackboard. You have to demo your programs within one week after the due date. Your demo will be based on your last submission in the Blackboard. Your TA will instruct you with further details.