CS 2101 Lab 3: Database of People

Due: 28 September 2016

Design and implement a database of people. Your program should use a linked list so that it can handle an unlimited number of entries. The NSA (National Security Agency) almost certainly has a [redacted] with much more [redacted] and we are all [redacted] (although I doubt that they use [redacted].

Specifically, you should:

- Design and implement a **Database** class with a *separate* **Entry** class. Unlike the linked list example in class, neither of these classes will be parameterized. An **Entry** object should contain the following instance variables:
  - **firstName** (String)
  - **lastName** (String)
  - **age** (int)
  - **isStudent** (boolean)
  - **next** (a reference to the next **Entry** object)

- Ask the user how many entries they would like in the database, ensuring that it is greater than zero, and create a database with that many entries.

- Ask the user for a last name and search the list for a person with that last name, printing the data for that entry if it is found. You will find the Java **String** method **compareTo** useful here. If the name is not found, your program should indicate that (see sample output below).

- Ask the user for two ages and print the entries of people whose ages are in the range specified by the two ages, ***inclusive***. The output must be in sorted order by last name. If there are no entries in that age range, your program should indicate that (see sample output below).

- Print the entire entry for all people in the database who are students. If there are no students, your program should indicate that (see sample output below).

In order to do the next to last item, you will want to make your linked list ordered alphabetically by last name, which means that when you insert an entry you will have to find the right place in the list to insert it so that the list is still in alphabetical order. Do not worry about alphabetizing by first name if there are two people in the database with the same last name (which is likely).

There are arrays of first and last names in the provided **TestDatabase** class that you can use to generate random names. You will also need to generate ages as integers from 1 to 120. Whether the person is a student should also be randomly generated. Note that the **Random** class has a method **nextBoolean()**.

Your program should be appropriately modularized. For this lab, I am providing skeletons of the three classes you need to write on BlackBoard.

SAMPLE OUTPUT 1:

```
Enter the number of database entries you would like (must be > 0): 0
Enter the number of database entries you would like (must be > 0): -4
Enter the number of database entries you would like (must be > 0): 10

Database:
Entry: Svetlana Borisov, 32,  not a student
Entry: Gwyneth Christakos, 58,  not a student
Entry: Niejls Kourakis, 35,  not a student
Entry: Itzhak Kourakis, 52,  not a student
Entry: Terik Kuznetsov, 38,  not a student
Entry: Artemisia Kuznetsov, 110,  student
Entry: Artemisia Mckesson, 22,  not a student
Entry: Gwyneth Mckesson, 88,  not a student
Entry: Sonya Newton, 86,  student
Entry: Terik Yang, 33,  not a student


Name to search for: Kourakis
Entry: Niejls Kourakis, 35,  not a student
Entry: Itzhak Kourakis, 52,  not a student

Print names between two ages, inclusive.
Enter starting age: -1
Enter ending age: 900
All ages must be from 1 to 120 inclusive.
All ages must be from 1 to 120 inclusive.
Enter starting age: 56
Enter ending age: 23
Starting age must be less than ending age.
Enter starting age: 900
Enter ending age: -1
All ages must be from 1 to 120 inclusive.
All ages must be from 1 to 120 inclusive.
Starting age must be less than ending age.
Enter starting age: 30
Enter ending age: 60

People in the age range 30-60:
Entry: Svetlana Borisov, 32,  not a student
```

Note that it is fine if you get the same error message twice (if both ages are out of range).

Entry: Gwyneth Christakos, 58, not a student
Entry: Niejls Kourakis, 35, not a student
Entry: Itzhak Kourakis, 52, not a student
Entry: Terik Kuznetsov, 38, not a student
Entry: Terik Yang, 33, not a student

Students:
Entry: Artemisia Kuznetsov, 110, student
Entry: Sonya Newton, 86, student

Here is sample output showing what your program should output if the name is not found, or there are no people in the specified age range, or there are no students in the database:

SAMPLE OUTPUT 2:

Enter the number of database entries you would like (must be > 0): 2

Database:
Entry: Nevin Elfasi, 30, not a student
Entry: Svetlana Voltaire, 81, not a student

Name to search for: Smith
Smith is not in the database.

Print names between two ages, inclusive.
Enter starting age: 10
Enter ending age: 20

People in the age range 10-20:
No entries in that age range.

Students:
There are no students in the database.

Submitting Your Program

Please put your source files — **TestDatabase.java**, **Database.java**, and **Entry.java** — in a single folder, compress that folder, and upload it to Blackboard in the **Lab Submissions** section. Also, print out a hardcopy of those files and submit it at the beginning of class on the due date.