Computing and Algorithms II: Programming Assignment 2

CS-102 **Fall 2008**

In this programming assignment, you will demonstrate your knowledge of dynamic data structures (in particular, linked lists) by revising the airport abbreviation code program from Programming Assignment 1 to incorporate a dynamic data structure.

New Internal Requirements

For this assignment, you should replace your Database class with another implementation which uses *two instances of a linked list* to organize the information within the database.

- You should define a class `Node` with three members: a *key*, a *value*, and a *link*. The key should be a `String` object. The value should be a `Airport` object. The link should be a reference to another `Node`.
- You should define a class `LinkedList`, which manages a linked list of `Node` objects. The `Node` objects should be organized within the list in sorted order, sorted according to the key values.
- Your `Database` class should use *two* separate instantiations of your `LinkedList` class. One list will use the abbreviation as the key value; the other list will use the name as the key value. That is, one list will be sorted by abbreviation, while the other list will be sorted by name.

Note that this design implies that your program should not contain excessive duplication of code or space. Adding an `Airport` object to both lists should be accomplished by two calls to the add method of the `LinkedList` class on different `LinkedList` objects, rather than two calls to different add methods. Also: adding an airport to the database should result in the creation of one `Airport` object, which is then inserted into both lists (using different nodes, of course).

You may use any variation on linked lists which you desire; doubly-linked, dummy head nodes, head/end pointers, *etc.*. You may implement additional classes as desired in order to manage the list. (You are cautioned *not* to use generic classes without discussing the issues with the instructor first.)

New Functional Requirements

Your program will be an extension of Programming Assignment 1, and thus should operate in the same manner as that assignment, unless otherwise specified herein. In particular, this means that any errors present in your submissions for Programming Assignment 1 should be fixed for this assignment.

The following new requirements should be implemented as well:

- The print database command should print the contents of the database sorted either by name or by abbreviation, as specified by the user. (Since your lists are required to be stored in sorted order, this should be no problem.) You may implement this either as separate commands from the main menu or with an explicit user query from the old print command.

- A new command should be implemented which allows the user to insert a new airport record from the command line. If selected, the program should prompt the user for all the necessary information and insert the selected airport into the database appropriately (*i.e.* into both lists).

- A new command should be implemented which allows the user to delete an airport record. If selected, the program should prompt the user for the abbreviation of the airport to be deleted. If the specified abbreviation is in the database, the user should be asked to confirm the deletion, and appropriate action taken based on the response. If no airport matching the description can be found in the database, the user should be informed of that fact.

Submitting Your Program

Before 11:59:59 p.m., Monday, 3 November 2008 (5th Monday), you must send a MIME-encoded email message to `jhuggins@kettering.edu` containing all source code files for your program, including a class named `Prog2` containing a `main` method.

In addition, you must deliver to the instructor a printout of your program files at the start of class on Tuesday, 4 November 2008 (5th Tuesday).

Notes

1. *Plan for the future!* If your submission for Programming Assignment 1 was well designed, the number of changes to classes outside of the Database class should be minimal. In Programming Assignment 3, you will be asked to replace the Database class with an implementation of a different dynamic data structure; again, changes outside of the Database class should be minimal. Design your program with this in mind.

2. Keep in mind that the midterm will be held on 7 November (5th Friday). Finishing this program (and understanding it) will be excellent preparation for the exam ...

3. It is technically possible for your linked list classes to be declared using generic types and still support maintaining the list in sorted order. But there are some tricky details. If you would like to attempt it, see the instructor for additional information on how to make relative ordering work with generic types.