EECS 2500: Linear Data Structures

Project 0: Sorting The Mail

Description:

In this project you will input an array of mailing addresses from a file and sort them into zip code order. Once sorted you should output the mailing addresses to an output file in the new sorted order. Each mailing address should be stored as an object (more on this later) in an array. Once you read in the mailing addresses you should then call a method to check the starting order. Once you have finished sorting the addresses you should call a second method to check the sorted list. You will also call methods in order to time the sort.

Details:

Your program will need to implement an interface to represent the mailing addresses. The interface is available with this assignment. The interface describes different methods to get fields from the class. In addition, it includes a complete constructor and an input method. These will help you set up your basic object for an address. In addition, there is a routine get zip digit which will return the given digit from the zip code. The units digit is digit 1, the tens digit is digit 2, etc…

Once you have implemented your mailing address class you should then work on opening the input file and reading in an array of the mailing addresses. The array will be no more than 10,000 elements long, You should read addresses until you reach the end of file or an exception occurs. I strongly suggest that you print out your array of addresses at this point and check if they are correct. The name of the input file should be read in from the system input. A sample data file with only a few addresses will be provided as an initial test. A further test set with a much larger number of addresses will be added before the due date.

To sort the items you will implement a radix sort. The radix sort will be discussed in class. The radix sort will require you to have 10 arrays that can hold mailing addresses. I suggest you set up a 2 dimensional array for this purpose. Again, we will discuss this in class.

Testing:

Testing of this project is to be aided by the two methods mentioned in the description. The first method, checkStartingOrder, will take your array as an argument and check the items in it. It assumes that a null object reference will mark the end of the data in the array. The second method, checkFinalOrder, will take your array as an argument and check the items in it for order. It will print error messages to the console if it finds objects out of order. These are simple test methods and are not able to test all conditions. To instantiate (create) the object that does this you will need to create a Project0Helper by using a statement such as:

```
Project0Helper help = new Project0Helper();
```

To call the methods you will have statements such as:

```
help.checkStartingOrder(mailingList);
help.checkFinalOrder(mailingList);
```

Where mailingList is an array containing MailAddressInterface objects.

Programming Style:

We will talk about programming style in class. It is expected that you will use class, method, and variables names that are descriptive. In addition, you are expected to use JavaDoc comments and provide your name at the top of each class.

Submission:

Your source files should be submitted online in BlackBoard. Your source files are the .java files for your project.   You may submit your project as many times as you wish before the due date. Only the last version will be graded.