Data Structures 1

Homework Assignment #1

Objectives

* Demonstrate ability to utilize the java RandomAccessFile class and its methods
* Demonstrate understanding of how an index can be used to access data
* Demonstrate how more than one index can be used to access the same data
* Be able to explain why it is not efficient to access data that has been stored in a RandomAccessFile via an index.

1. Develop code to implement a PharmaceuticalCo class that will support the following specifications

* Maintain CompanyCode (String), CompanyName, PhoneNumber
* Set up a Constructor : PharmaceuticalCo(String companyCode, String name, String PhoneNumber)
* Set up a Constructor: PharmaceuticalCo(Scanner filename) that will read the next three strings from a text file and construct an instance of PharmaceuticalCo
* Set up a Constructor: PharmaceuticalCo(RandomAccessFile raFile, Long location) that will set up an instance of PharmaceuticalCo based on data read starting at a particular location in the file
* Set up getters
* Define a setPhoneNumber (String newNumber) that will modify the phoneNumber assuming it is valid
* Define writeToFile(RandomAccessFile raFile. Long location) that will store the data maintained by a given instance of PharmaceuticalCo to the disk file at location, ‘location’.
  + companyCode - fixed length field of 4 characters
  + company name – fixed length field of 25 characters
  + telephone number – fixed length field of 10 characters.
* Define compareTo() , equals() and toString() methods

1. Develop code to implement a CompanyCodeIndex class that implements Comparable and Serializable and will support the following specifications.

* Maintain CompanyCode (String), Location in RandomAccessFile (Long), isActive (boolean)
* Set up a constructor: CompanyCodeIndex(String companyCode, Long location)
* Set up getters
* Define compareTo() , equals() and toString() methods

1. Develop code to implement a CompanyNameIndex class that implements Comparable and Serializable and will support the following specifications

* Maintain CompanyName(String) , Location in RandomAccessFile (Long), isActive(boolean)
* Set up a constructor: CompanyNameIndex(String companyName, Long location)
* Set up getters
* Define compareTo(), equals() and toString() methods

1. Develop a PharmacyList class that implements Serializable and implements the following specifications

* Maintain a set of CompanyCodeIndex references in an ArrayList
* Maintain a set of CompanyNameIndex references in an ArrayList
* Maintain a reference to the RandomAccessFile that contains the company data
* Implement a method – addCompany(String companyCode, String companyName, String phoneNumber)
  + Determine the companyCode and companyName has not been entered already in the system. If it has, throw DuplicateDataException
  + Instantiate an instance of PharmaceuticalCo
  + Write out the instance to a Random Access File – appending it to the end of the file
  + Set up two separate index records CompanyCodeIndex and CompanyNameIndex , each referencing the same location in the RandomAccessFile
* Implement a method – PharmaceuticalCo findCompanyCode(String companyCode)
  + Search through the ArrayList of CompanyCodeIndex references
  + If not found, through NotFoundException
  + If found, but no longer active , throw NotFoundException
  + If found, retrieve location of that record, instantiate a PharmaceuticalCo record from the RandomAccessFile and return the record
* Implement a method - PharmaceuticalCo findCompanyName(String companyName)
  + Search through the ArrayList of CompanyNameIndex references
  + If not found, through NotFoundException
  + If found, retrieve location of that record, instantiate a PharmaceuticalCo record from the RandomAccessFile and return the record
* Implement a method – void modifyCompanyPhone(String companyCode, String newNumber)
  + Search through the ArrayList of CompanyCodeIndex references
  + If not found, through NotFoundException
  + If found, but no longer active , throw NotFoundException
  + If found, retrieve location of that record, instantiate a PharmaceuticalCo record from the RandomAccessFile, invoke the PharmaceuticalCo setPhoneNumber(newNumber)
  + Invoke the PharmaceuticalCo.writeToFile() to correct location within the file , replacing the current record of that company

Develop a user interface

* Instantiate a new PharmacyList class
  + Read in data from the pharmacyCompanies.txt file, invoking the PharmacyList.addCompany () method for each set of data you have read in from the text file.
* Present the users with a menu and write code that will process each user request
  + Add a pharmaceutical company
  + Remove a company
    - Mark the record as ‘deleted’ in the index records that reference this company
  + Modify company phone number
  + Display company information
    - Given company code
    - Given company name
  + List information about each company on file, in alphabetical order
    - **Question: how inefficient is it to provide this information? Why?**
  + End the application
* When the application ends store PharmacyList to file using Object Serialization – extra credit

Sample Data

**Question: Why did I choose to place the company name as the last item on each line?**

ABL 5162846831 Abbott Labs

ADW 7188548385 Adwe Labs

ALC 8008625266 Alcon

ALG 8004338871 Allergan

BMS 8003322056 Bristol Meyers Squibb

GAL 6004097702 Galderma

JJM 8005263967 Johnson & Johnson

NOV 8627782100 Novartis

PGP 2129806400 Proctor & Gamble

PZR 9736605000 Pfizer