DPIT121 – Lab Exercise 6

Due: Week 8 lab

In lab 6, you will continue on another iteration focusing on Persistence layer of your software to Save and load your data into / from text file as well as binary file. This acts as a simple database for your software. You will use a proper relational database in assignment III.

Download and study lecture codes from week 6. Also, study Lab 3 Solution from "Sample labs 4-6 and Solutions" folder which is similar to this lab.

Marking Guide: 0.5 mark for each item.

1) Study InheritanceFilingSerialization.java and InheritanceFilingSerialization2.java from week 6 to learn how to read and write parent and children classes (Inheritance) as well as classes with simple and complex aggregation/composition in the binary file by using java Serialization. Make all of your classes (except the program and UI) Serializable (i.e., to implement Serializable interface).

2- Add static methods to your InsurancePolicy to read and write a HashMap of policies from/to a binary file by using Serialization:

HashMap<Integer, InsurancePolicy> load (String fileName)

Boolean save (HashMap<Integer, InsurancePolicy>, String fileName)

3- Add static methods to User class to read and write a HashMap of users from/to a binary file by using Serialization (The same as 2).

4- Add these methods to your InsuranceCompany to read and write an InsuranceCompany object from/to a binary file by using Serialization. As you are using Serializable you only need to save and load one insuranceCompany object to save and load all the users and policies. Note that these methods are NOT static.

Boolean load (String fileName) // add a default constructor (a constructor with no parameter to create an empty object) to your insuranceCompany. In your test code create an empty InsuranceCompany object with default constructor and then call object.load to fill the information from file to the object

Boolean save (String fileName) // to save the InsurancePolicy object (this) into the file.

5- Study InheritanceFiling.java from week 6 lecture code to learn how to read and write parent and children objects (Inheritance) in the text file. Add toDelimitedString() method to ~~MyDate~~, ~~Car~~, ~~InsurancePolicy~~, ~~ThirdPartyInsuranec~~, and ~~ComprehensivePolicy~~ with proper object identifications.

6- Add static methods to your InsurancePolicy to read and write a HashMap of policies from/to a text file by using toDelimitedString():

HashMap<Integer, InsurancePolicy> loadTextFile (String fileName)

Boolean saveTextFile (HashMap<Integer, InsurancePolicy>, String fileName)

7- Add toDelimitedString() to User class. Then add static methods to User class to read and write a HashMap of users from/to a text file by using toDelimitedString() (The same as 6). Here for each user you can assume that number of polices is stored as a field and then a list of policies (comma separated) in each record. Check the Employer and list of projects in supplementary lab 3 part 4&5 sample codes.

8- Add toDelimitedString() to the InsuranceCompany (to convert the whole company including all users and all polices to a list of comma separated records). Then add these methods to your InsuranceCompany to read and write an Insurance Company object from/to a text file by using toDelimitedString().

Boolean loadTextFile (String fileName)

Boolean saveTextFile (String fileName)

Note that another option is to have separate files for company, list of users, and list of policies. In user file, instead of having all the policies' information in the record only put the policy IDs and in the InsuranceCompany store a list of user IDs and not full users' information. In the loading process create a hashmap of all users and a hashmap of all policies from their own files and then use addUser and addPolicy methods in the company to populate all users and policies in the memory. You may do this in your assignment II. For this lab, you don't need to do it.

9) Add test code to test all of these methods for both binary and text file. You need to submit these data files in addition to your Java files. Show that after saving, clearing your hashmaps and loading them again, the data is correctly stored and loaded. Hint : Check lab 6 test ideas.java

10) Add options to load and save binary and text files to your UI.

Submit Text and Binary files with your Java files.