HW5

Josh Valentino

CSC 3630

Professor David Lash

10/29/2021

Object:

The object I created for this assignment is named courseObject. The variables it contains are String Course, String Section, String Title, int NStudents, int NSeats, String Instructor, and int Year. Each of these values match to the input text file, courses.txt. I picked this object to keep track of the courses data while aggregating and displaying sorted lists of courses.

Data Structures:

1. ArrayList<String> inputDataRows. An array list of strings named inputDataRows. This holds each row of the input text file as Strings.
2. ArrayList<courseObject> dataRowsAsObjects. An array list of objects named DataRowsAsObjects. This holds the data from inputDataRows with each row of data being input to an object.
3. HashMap<String, ArrayList<courseObject>> courseHashMap. A hash map with a string key and each value being an array list of objects. The hash map is named courseHashMap. This structure was selected because on average lookup operation of O(1), meaning it is generally fast to search for a value given the key.
4. TreeMap<String, ArrayList<courseObject>> courseTreeMap. A tree map with a string key and each value being an array list of objects. The tree map is named courseTreeMap. This structure is used in the showAggregateCourses function because TreeMaps are automatically sorted in the way needed for the proper ordered output. For the purpose of the function, a new tree map is made by inputting the data from hash map, courseHashMap.
5. String[] duplicateInstructors. An array of strings named duplicateInstructors, initialized in the function, showAggregateCourses. The purpose of this array is to hold the individual string values for each instructor in the current courseObject.
6. String[] distinctInstructors. An array of strings named duplicateInstructors, initialized in the function, showAggregateCourses. The purpose of this array is to hold the distinct string values for each instructor in the current courseObject. This array is sorted alphabetically before being printed out.

Test Cases:

**#1: CSC1700**

**Graphical user interface, text, website

Description automatically generated**

**#2: CSC1010**

**Graphical user interface, text

Description automatically generated**

**#3: CSC3610**

**Text

Description automatically generated**

**#4: Q**

**Text

Description automatically generated**

**#5: Multiple Entries in Order: CSC1700, CSC1010, CSC3610, Pizza, Hello, CSC4500, Q**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**