CTA Project Design

Mohammad Umar

CS 201

Professor Dylan Boliske

1. The Menu options will be
    1. Create a station
       (The user will be prompted to enter the name, longitude, latitude, wheelchair access, description and lines.)
    2. Modify an existing station
       (The user will be asked to input the details of the station they wish to modify. Then the new details will be taken as input from user and changed, respectively. The data could be the name, longitude, latitude, wheelchair access, description and lines.)
    3. Remove a station
       (The user will be further asked to input the name of the station he wishes to remove and that station will be removed)
    4. Search for a Station
       (The user will be asked to input the name and/or more details such as longitude, latitude, wheelchair access, description and lines. The program will print out the station that corresponds with the inputted data)
    5. Create a Path:
       (The user will be asked to input the names of the start and final station.)
    6. Search for a nearest station
       (The user will be asked to input the latitude and longitude.)
    7. Exit the program
       (Terminates the program.)

The program will display these options. The user will input the choice number. After inputting the choice, the program will run the respective code. These options will be displayed until the user wishes to exit (selects option 7).

2. Programmer Tasks:
    - Reading the file and Processing the Data

    To read the input file I will create a file object and then use the scanner to get the information from the file. The scanner will loop over all lines of the csv files and store the data alphabetically in the list so its easy to search. Further I will use the (.split()) function for string parsing to extract respective data and add it to the list by creating an object.

As the data is in alphabetical order it will process the data by station name and determine where to add each station.

- Storing Data:

The data will be stored in an array list. Each station would be one index in an array list I will store each station's name, latitude, longitude, location, wheelchair accessibility and the route it's on

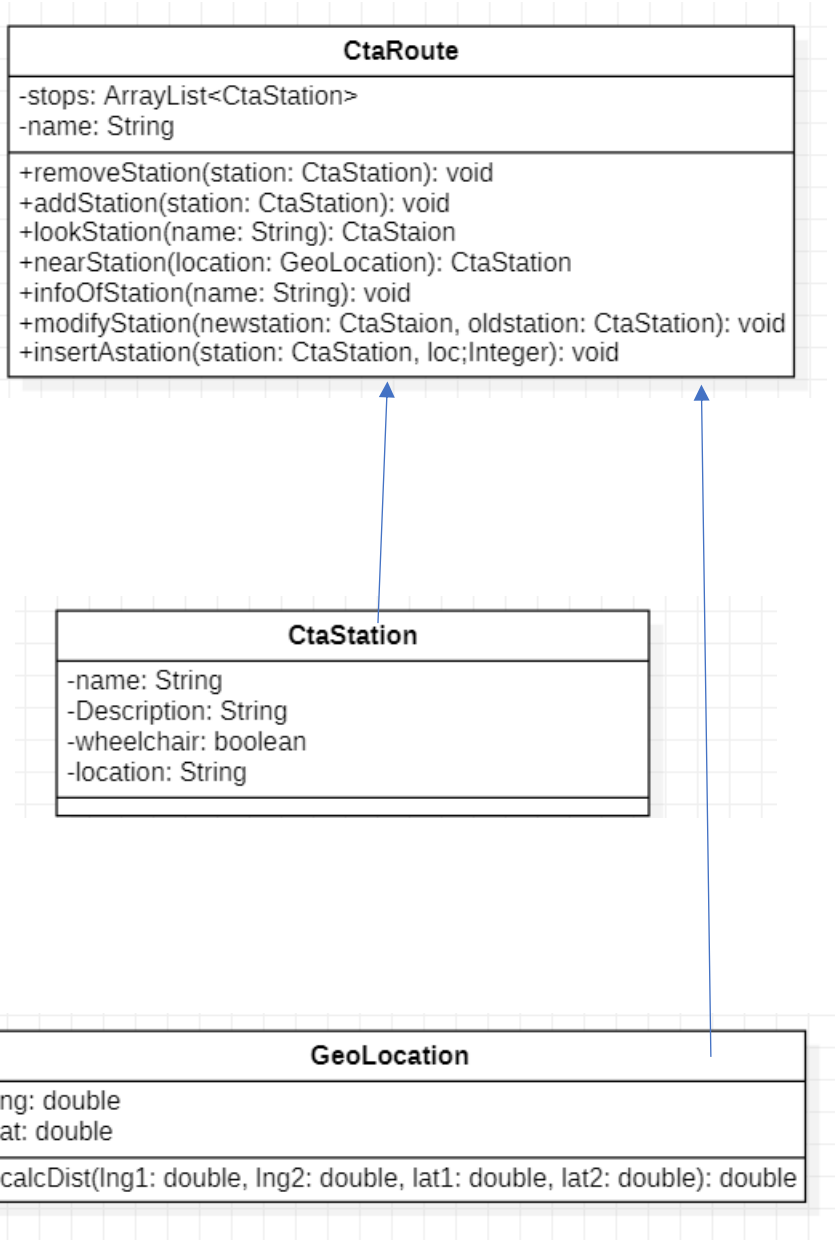
- Add/Delete Modify:
  For adding the data I will request the data from the station and then in the alphabetical order of the name I will place it in the array list.
  To delete the data I will ask the name of the station that the user wishes to delete and then remove it from the array list. I will maintain the order of the array list.
  To modify the data I will ask user for the name and then the details he wishes to change and I will append the data for that station and store that in the array list changing the old data.

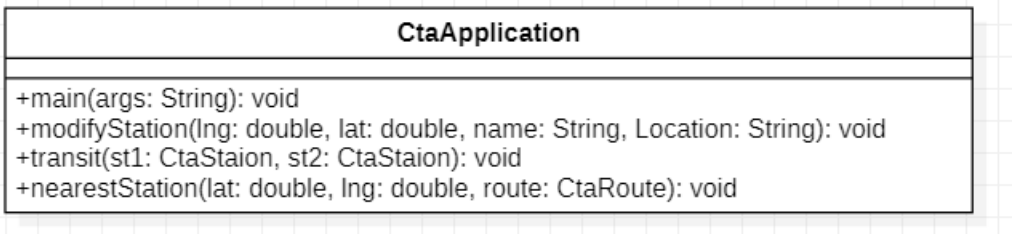
  I will use the .add method and .remove method for array lists.
- Searching:

To search the data, I will use a do while loop and the Boolean variable to look for the data which the clients want to search. I will search through every element until the data is found.(is Equals method will also be used to compare the data.

The Classes I will use are GeoLocation, CtaRoute , CtaStation, CtaApplication. I may add or modify the classes as per the need when I start to code in detail.

3. This is the initial class setup I have planned. I might have missed some things which I may realize when I get in detail and I will add that further in future. This is a basic prototype for the classes. There might be new or more relations b/w the existing classes. The basic methods(getter setters equals and to string are default)


CtaRoute
-stops: ArrayList<CtaStation>
-name: String
+removeStation(station: CtaStation): void
+addStation(station: CtaStation): void
+lookStation(name: String): CtaStaion
+nearStation(location: GeoLocation): CtaStation
+infoOfStation(name: String): void
+modifyStation(newstation: CtaStaion, oldstation: CtaStation): void
+insertAstation(station: CtaStation, loc;Integer): void
CtaStation
-name: String
-Description: String
-wheelchair: boolean
-location: String
GeoLocation
-lng: double
-lat: double
+calcDist(lng1: double, lng2: double, lat1: double, lat2: double): double


CtaApplication
+main(args: String): void
+modifyStation(lng: double, lat: double, name: String, Location: String): void
+transit(st1: CtaStaion, st2: CtaStaion): void
+nearestStation(lat: double, lng: double, route: CtaRoute): void

Test Plan

Option	Expected Input	Behavior
1.Create Station	User will be asked for a new CTAStation (Name, location, wheelchair access, lat,lng) and then the routes this station appears on	Adds a new station
2.Modify Station	User will be asked for a new CTAStation (Name, location, wheelchair access, lat,lng) and then the routes this station appears on	Change the old station with the new station
3.Remove Station	User will be asked to enter the name of the station and the route it appears on	Deletes the station
4.Search for a Station	User will be asked for the name of the CTAStation	Search for the station by the inputted name and display the details of that station if its found else returns a message to notify the user that the station was not found.
5.Create a path	User will be asked for the name of the initial CTAStation and the final CTAStation	If on the same line it displays out the station between them in right order If stations on different lines, search for the station the lines have in common and display the stations on the first line up till the common station
6.Search for nearest Station	User will be asked to enter the latitude and longitude	It calculates the distances between all the stations and the station which has the shortest distance is displayed with all details.
7.Exit	User will just input 7	Terminates the program with a goodbye message.