

Ömer Faruk Çelik-2021400084-23.03.2023-this code finds the path for the people who chooses two stations and show them how to go from one station to another station using metrolines and visualize them on canvas

First of all, I created some lists to store the data from coordinates.txt. I created an allMetroLines list and filled it with metro lines' names. I also created other lists sorted by allMetroLines list, where if M4's index in allMetroLines is 4, then the fourth element of lists corresponds to the M4. I applied a similar logic for breakpoints.

Before filling the lists with data from coordinates.txt, I created a list and filled it, then added the list to a big list that includes every metro line. My aim was to ensure that allMetroLines index and other lists' index would be the same.

After taking input from the user, I checked whether my map includes the names of cities given by the user. If not, I gave a message and the canvas would not pop up. I determined the location of the cities given by the users according to metroStopsNames. The first location variable corresponds to its metroline's index and the second location variable corresponds to the index of the station in the metroline. I also created a list that includes breakpoint locations.

Moving on to the algorithm, if the current location is on the same metro line as the target, we quit the while loop and move to another loop which will be explained later. If they are not on the same metro line, I check if there is a breakpoint by incrementing the second location variable by one on the metroline where it currently is. If it coincides with a breakpoint, I rearrange the location according to the new metro line location and increase the second variable location by one according to the new metro line. I then add the old location and the new location to the oldBreakPoints list together. If it reaches the end of the metroline and does not encounter a breakpoint, I rearrange the second variable location as 0 and continue from the beginning of the list. If it reaches the old location, it means there is no breakpoint on the metroline, so I remove the locations from the oldBreakPoint list, rearrange the location according to oldMetroLine, increase the second variable location by one, and continue. This continues until it reaches the same metro line as the target. If it cannot find the target, the loop never ends. If the loop rotates too many times, meaning it cannot find the target, I quit the loop and the canvas would not pop up.

When it reaches the same line as the target, it quits the loop. Then it reads the location in the oldBreakPoints list and adds the stations from one location until another location to the solution list. At the end, it goes to another loop created for the same line and adds the stations from the location until the target points. Finally, it prints the solution.

Moving on to visualization, I first set the background and the canvas information, such as metro line stops and names. I created a new solution list and loop for the movement based on solution lists. Every loop, I drew the metro lines and other elements. I drew the passed location as a little orange circle and the current location as a bigger circle, based on the solution lists. I cleared the canvas every loop and recreated it using double buffering and the show method.





<terminated> Main [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (25 Mar 2023 18:33:53 – 18:33:57) [pid: 6824]

Uskudar

Bahariye

These two stations are not connected

Problems Debug Shell Console ×



Tavsantepe	
FevziCakmak	
Yayalar	
Kurtkoy	
SabihaGokcenHavalimani	

