Introduction

First of all, I want to talk about my program. There are 2 classes in this program. There is a class called Spotify and HashMap. In the Spotify class, the desired methods are written using some libraries. I tried to write my own HashMap commands in the HashMap class, but I was not very successful when I used it. So, I wrote all but the R<Name> method in the Spotify class and it works fine. I tested this by removing the lines containing R<Name> in the input.txt file.

Algorithm

public void I(String name):

This method first checks if the person with the given name already exists in the likedSongs map. If the person does exist, it prints an error message saying that the person can't be created. If the person does not exist, it creates a new HashSet to store the person's liked songs and adds it to the likedSongs map with the person's name as the key.

public void L(String name, String song):

This method first checks if the person with the given name exists in the likedSongs map. If the person does exist, it adds the given song to the person's liked songs set. If the person does not exist, it prints an error message saying that the song cannot be liked because the person is not created.

public void E(String name, String song):

This method first checks if the person with the given name exists in the likedSongs map and if the person's liked songs set contains the given song. If both conditions are true, it removes the song from the person's liked songs set and prints a message saying that the person no longer likes the song. If either condition is false, it prints an error message saying that the song cannot be erased.

public void D(String name):

This method first checks if the person with the given name exists in the likedSongs map. If the person does exist, it removes the person and their liked songs from the map. If the person does not exist, it prints an error message saying that the person is not in the list.

public void P(String name):

This method first checks if the person with the given name exists in the likedSongs map. If the person does exist, it gets the set of liked songs for the person and checks if the set is empty. If the set is empty, it prints a message saying that the person has no songs. If the set is not empty, it prints a message saying that the person likes the following songs and then iterates through the set of liked songs, printing each song. If the person does not exist, it prints an error message saying that the person is not in the list.

public void M(String name):

Checks if the user is in the list. If not, it writes the user's name to the screen and exits the method. If the user is in the list, it gets the list of songs that user likes (userLikedSongs). Starts a loop for all users. For each user, it gets the list of songs that user likes. If the user variable is the same as the searched username, it skips the loop and moves on to the next user. counts the common songs between userLikedSongs and songs and assigns them to the count variable. creates a Map called songCounts and adds user and count values to this Map. When the loop ends, it writes the match results to the screen using all values in songCounts.

public void O(String input):

Creates a BufferedReader to read the file. If the file cannot be opened, a FileNotFoundException is thrown and this is printed to the screen. Reads each line in the file and assigns it to the line variable. If the line cannot be read, the loop terminates. It splits the line variable by whitespace and assigns it to the lines array (just split it by 3).

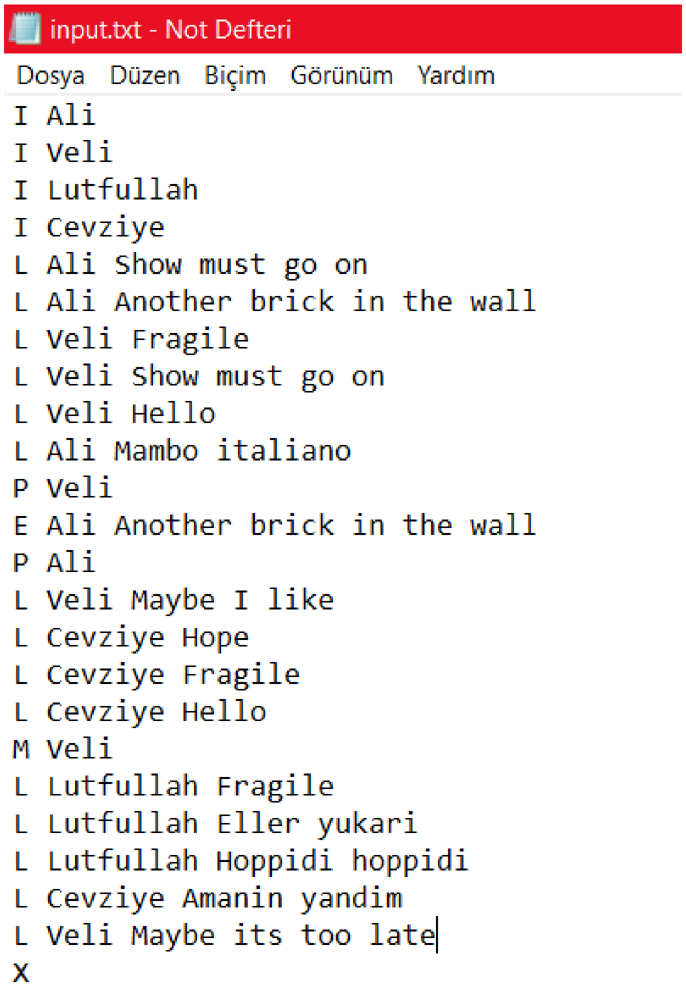
Starts a loop for each element in the lines array. Each element can be a command or an argument of a command. If the array of lines has 2 elements and the first element starts with "I", it calls the I(lines[1]) method. If the array of lines has 2 elements and the first element starts with "P", it calls the P(lines[1]) method. If the array of lines has 2 elements and the first element starts with "R", it calls the R(lines[1]) method. If the array of lines has 2 elements and the first element starts with "M", it calls the M(lines[1]) method. If the array of lines has 3 elements and the first element starts with "L", it calls the L(lines[1], lines[2]) method. If the array of lines has 3 elements and the first element starts with "E", it calls the E(lines[1], lines[2]) method. If the array of lines has 1 element and the first element starts with "X", it exits the method.

Analysis

Results

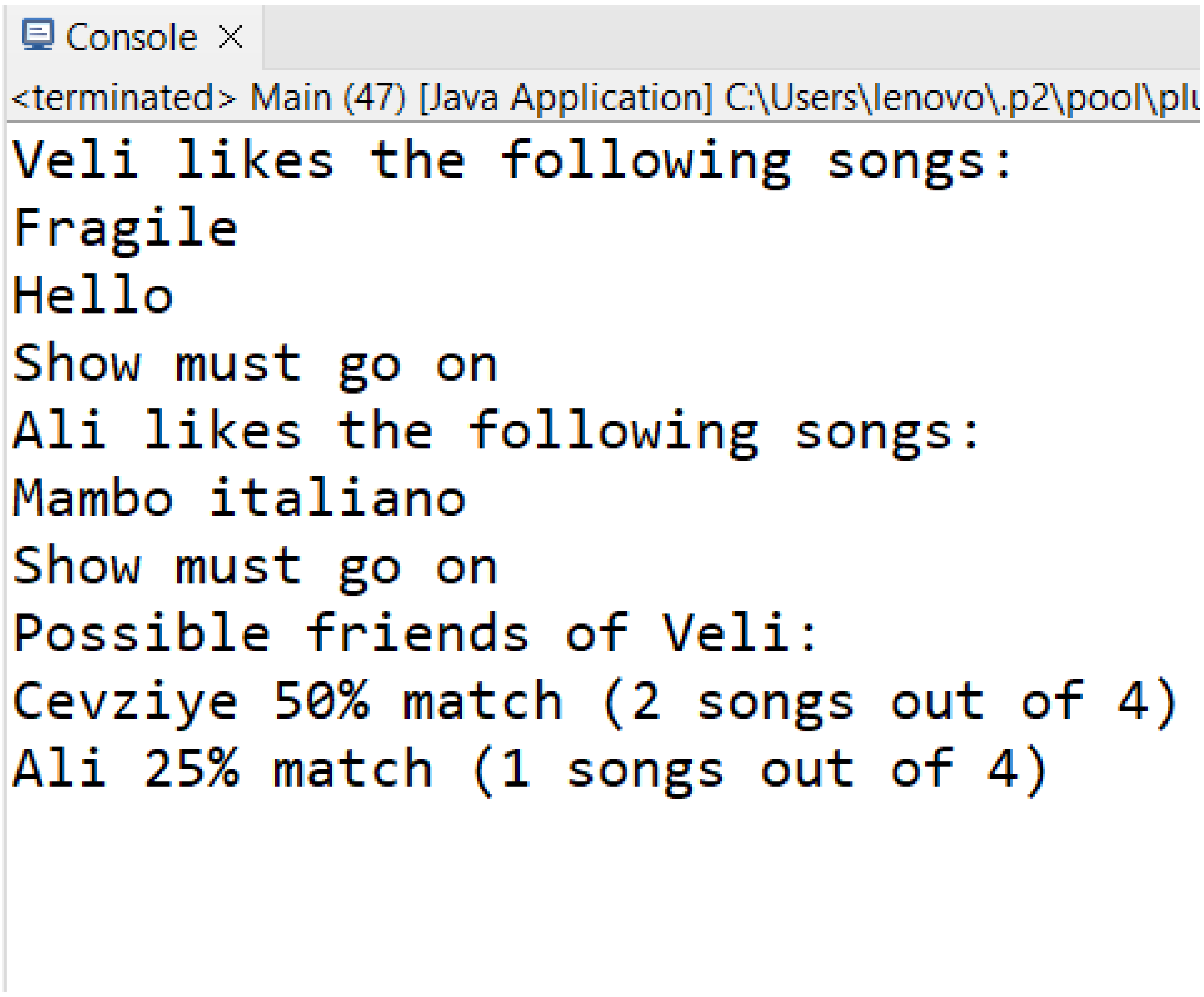
The desired methods were written and tested with the input.txt I specified below. The output is as follows:

!!! R method was extracted from input file.



```
I Ali
I Veli
I Lutfullah
I Cevziye
L Ali Show must go on
L Ali Another brick in the wall
L Veli Fragile
L Veli Show must go on
L Veli Hello
L Ali Mambo italiano
P Veli
E Ali Another brick in the wall
P Ali
L Veli Maybe I like
L Cevziye Hope
L Cevziye Fragile
L Cevziye Hello
M Veli
L Lutfullah Fragile
L Lutfullah Eller yukari
L Lutfullah Hoppidi hoppidi
L Cevziye Amanin yandim
L Veli Maybe its too late
X
```

Output:



```
<terminated> Main (47) [Java Application] C:\Users\lenovo\.p2\pool\plu
Veli likes the following songs:
Fragile
Hello
Show must go on
Ali likes the following songs:
Mambo italiano
Show must go on
Possible friends of Veli:
Cevziye 50% match (2 songs out of 4)
Ali 25% match (1 songs out of 4)
```