//SUFIAN Ghazanfar//

Filename: TopStreamingArtists.java

import java.io.\*;  
import java.util.\*;

/\* The linked list implementation \*/  
class TopStreamingArtists {  
private Node head;  
/\* A node represents an artist \*/  
class Node {  
private String data;  
private Node next;  
public Node(String person) {  
data = person;  
next = null;  
}  
}

void sortedInsert(Node newNode) {  
Node current;

// Exception case for head node  
if (head == null || head.data.compareToIgnoreCase(newNode.data) > 0) {  
newNode.next = head;  
head = newNode;  
}  
else {  
current = head;  
while (current.next != null &&  
current.next.data.compareToIgnoreCase(newNode.data) < 0)  
current = current.next;  
newNode.next = current.next;  
current.next = newNode;  
}  
}

/\* Creates a new artist \*/  
Node newNode(String data) {  
Node node = new Node(data);  
return node;  
}

/\* Prints the linked list \*/  
void printList() {  
Node val = head;  
File out = new File("regional-global-daily-latest.csv");  
StringBuilder build = new StringBuilder();  
build.append("Artist");  
build.append('\n');

while (val != null) {  
System.out.print(val.data + ", ");  
build.append(val.data);  
build.append('\n');  
val = val.next;  
}

try (PrintWriter writer = new PrintWriter(out)) {  
writer.write(build.toString());  
} catch (FileNotFoundException error) {  
System.out.println(error);  
}  
}  
}

/\* The List TopStreamingArtists is composed of a series of artist names \*/  
class Hash {  
public void countFrequencies(List<String> list) {  
/\* Hash map to store the frequency of element \*/  
Map<String, Integer> map = new HashMap<String, Integer>();

for (String i : list) {  
Integer j = map.get(i);  
map.put(i, (j == null) ? 1 : j + 1);  
}

List<String> artistList = new ArrayList<String>();;  
StringBuilder sb = new StringBuilder();  
sb.append("Artist");  
sb.append(',');  
sb.append("Count");  
sb.append('\n');

TopStreamingArtists topStreamingArtists = new TopStreamingArtists();  
/\* Displays the occurrence of elements in the array list \*/  
for (Map.Entry<String, Integer> val : map.entrySet()) {  
artistList.add(val.getKey());  
sb.append(val.getKey());  
sb.append(',');  
sb.append(val.getValue());  
sb.append('\n');  
}

File csvOutputFile = new File("Amount of times Artist is noted.csv");  
try (PrintWriter writer= new PrintWriter(csvOutputFile)) {  
writer.write(sb.toString());  
} catch (FileNotFoundException error) {  
System.out.println(error);  
}

/\* Inserts into linked list then prints the results \*/  
for (String s : artistList) {  
topStreamingArtists.sortedInsert(topStreamingArtists.newNode(s));  
}  
System.out.println("Artist's names in Alphibetical Order");  
topStreamingArtists.printList();  
}  
}  
/\* Test class \*/  
public class Main {  
public static void main(String[] args) {  
String csvFile = "regional-global-daily-latest.csv";  
BufferedReader buff = null;  
String cut = "";  
String splitter = ",";  
List<String> names = new ArrayList<String>();  
try {  
buff = new BufferedReader(new FileReader(csvFile));  
while ((cut = buff.readLine()) != null) {  
// Cuts the segments with a comma  
String[] songInfo = cut.split(splitter);

// Edge case handling  
if (songInfo[2].replace("\"", "").trim().equals("000 Hours (with Justin Bieber)")) {  
names.add("Dance + Monkey");  
} else if (songInfo[2].replace("\"", "").trim().equals("next")) {  
names.add("Post Malone");  
} else if (songInfo[2].replace("\"", "").trim().equals("")) {  
} else if (!songInfo[2].replace("\"", "").trim().equals("Artist")) {  
names.add(songInfo[2].replace("\"", "").trim());  
}  
}

Hash count = new Hash();  
count.countFrequencies(names);  
} catch (FileNotFoundException error) {  
error.printStackTrace();  
} catch (IOException error) {  
error.printStackTrace();  
} finally {  
if (buff != null) {  
try {  
buff.close();  
} catch (IOException error) {  
error.printStackTrace();  
}  
}  
}  
}  
}