Alexander Dean

COS 285

Program 1

6 September 2017

Report: Program 1

The purpose of Program 1 was to develop two Java classes. One of these classes is for a Song object. It has String members for the artist, title, and lyrics of the song. It contains basic accessor methods for each member. In this case, it would be pointless to have mutator methods, for the songs will never be modified. The second class reads and parses an input file. The program expects an input file with the following format:

ARTIST="Mildred Hill and Patty Hill"

TITLE="Happy Birthday To You"

LYRICS=" Happy birthday to you

Happy birthday to you

Happy birthday dear \_\_\_\_\_

Happy birthday to you

"

Figure : Example format of input file

The program parses this input file and creates a new Song for each similar entry. This song is temporarily added to an ArrayList object. An ArrayList is used here because it is easier and faster to add an unknown number of entries, and it offers a method to convert it to a “primitive” array. Once the end of the file is reached, the program converts the ArrayList to an array and sorts it.

I worked on both classes alone, so all of the code was written by myself or provided in the assignment. As of now, I am not aware of any bugs or glitches in the code that would negatively affect its performance or functionality. The only quirk I noticed thus far is that, in some Alabama songs, the Lyrics fields of the input file also contains the names of the songwriters. This could possibly be accounted for in the parsing of the input file, but any such parsing would also cut out environmental lyrics such as “(loud party noises)” or “(bass solo)” if they appeared at the beginning of the song. It does not negatively impact performance or functionality, and I think it’s a nice touch to know who wrote the song.

The output of both classes can be found below.

testing getArtist: Professor B

testing getTitle: Small Steps

testing getLyrics:

Write your programs in small steps

small steps, small steps

Write your programs in small steps

Test and debug every step of the way.

testing toString:

Song 1: Professor B, "Small Steps"

Song 2: Brian Dill, "Ode to Bobby B"

Song 3: Professor B, "Debugger Love"

testing compareTo:

Song1 vs Song2 = 14

Song2 vs Song1 = -14

Song1 vs Song3 = 15

Song3 vs Song1 = -15

Song1 vs Song1 = 0

Figure : Output of Song.java

Total songs = 10514, first songs:

Aerosmith, "Adam's Apple"

Aerosmith, "Ain't Enough"

Aerosmith, "Ain't that a Bitch"

Aerosmith, "All Your Love"

Aerosmith, "Amazing"

Aerosmith, "Angel"

Aerosmith, "Angel's Eye"

Aerosmith, "Animal Crackers"

Aerosmith, "Attitude Adjustment"

Aerosmith, "Avant Garden"

Figure : Output of SongCollection.java

Additionally, this assignment also required the program to function with the GUI that was provided. This was solely done through the Run configurations in Eclipse. A screenshot of the functional GUI is shown below.

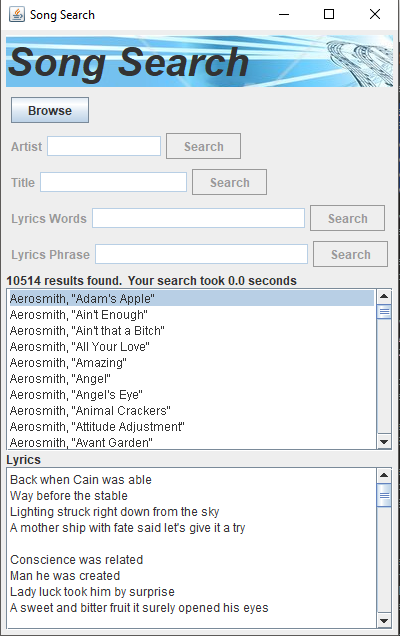


Figure : Screenshot of functional SongSearchGUI