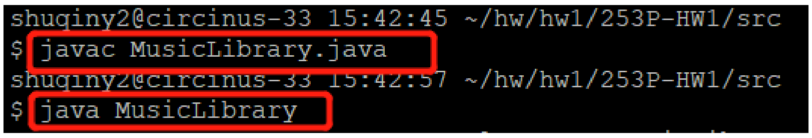
How to get access to the program?

```
git clone git@github.com:ShuqingYe1997/253P-HW1.git
cd 253P-HW1
cd src
javac MusicLibrary.java
java MusicLibrary
```



```
shuqiny2@circinus-33 15:42:45 ~/hw/hw1/253P-HW1/src
$ javac MusicLibrary.java
shuqiny2@circinus-33 15:42:57 ~/hw/hw1/253P-HW1/src
$ java MusicLibrary
```

Program started!

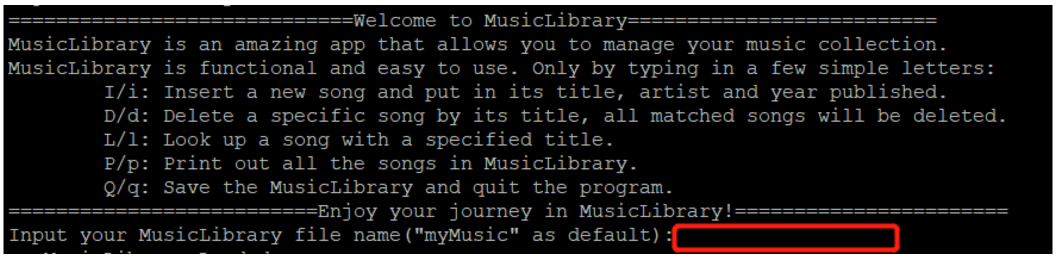
In this report you can see the following operations:

1. Load and print
2. Search
3. Insert from terminal/ file
4. Delete
5. Save and quit
6. Error handling

As well as important methods, major data structure and algorithms.

- Load and print

This is the welcome page of MusicLibrary. You can load your music library from your own file. If you type nothing, we'll automatically load myMusic.txt

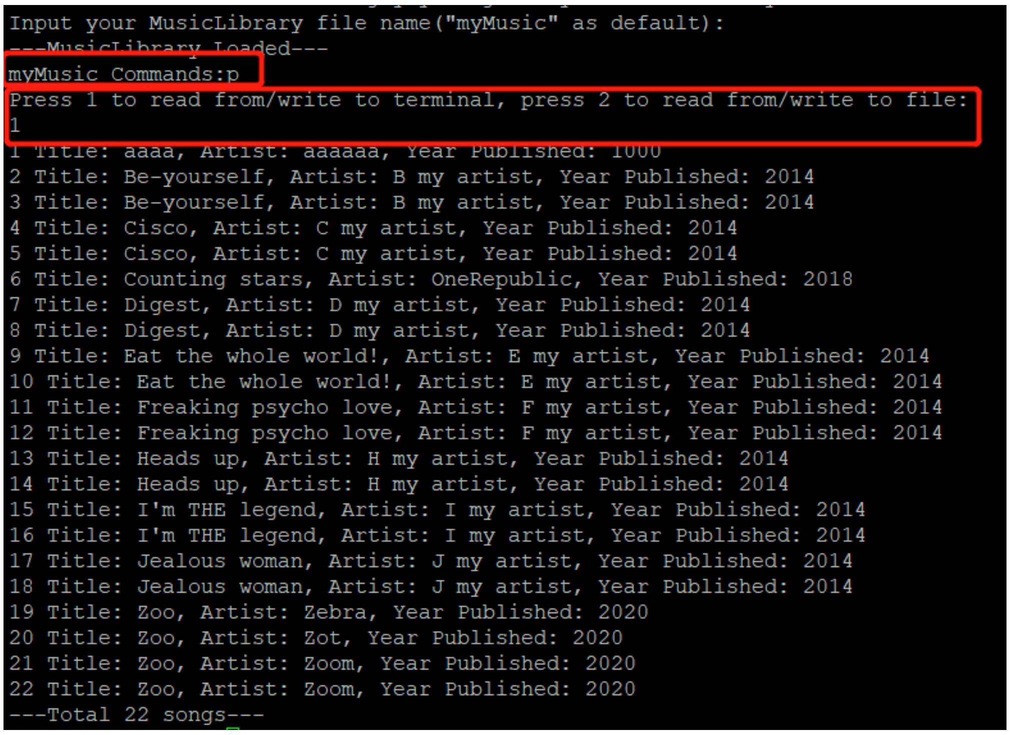


```
=============================Welcome to MusicLibrary=========================
MusicLibrary is an amazing app that allows you to manage your music collection.
MusicLibrary is functional and easy to use. Only by typing in a few simple letters:
        I/i: Insert a new song and put in its title, artist and year published.
        D/d: Delete a specific song by its title, all matched songs will be deleted.
        L/l: Look up a song with a specified title.
        P/p: Print out all the songs in MusicLibrary.
        Q/q: Save the MusicLibrary and quit the program.
=========================Enjoy your journey in MusicLibrary!=======================
Input your MusicLibrary file name("myMusic" as default):
```

You can input P to print all the music in this library.

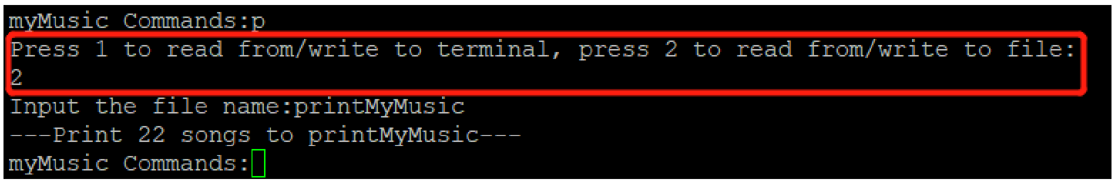
If you just want to print to the terminal, press 1, and then you can see the music listed alphabetically on your shell window.

Currently there are 22 songs in our library.



```
Input your MusicLibrary file name("myMusic" as default):
---MusicLibrary Loaded---
myMusic Commands:p
Press 1 to read from/write to terminal, press 2 to read from/write to file:
1
1 Title: aaaa, Artist: aaaaaa, Year Published: 1000
2 Title: Be-yourself, Artist: B my artist, Year Published: 2014
3 Title: Be-yourself, Artist: B my artist, Year Published: 2014
4 Title: Cisco, Artist: C my artist, Year Published: 2014
5 Title: Cisco, Artist: C my artist, Year Published: 2014
6 Title: Counting stars, Artist: OneRepublic, Year Published: 2018
7 Title: Digest, Artist: D my artist, Year Published: 2014
8 Title: Digest, Artist: D my artist, Year Published: 2014
9 Title: Eat the whole world!, Artist: E my artist, Year Published: 2014
10 Title: Eat the whole world!, Artist: E my artist, Year Published: 2014
11 Title: Freaking psycho love, Artist: F my artist, Year Published: 2014
12 Title: Freaking psycho love, Artist: F my artist, Year Published: 2014
13 Title: Heads up, Artist: H my artist, Year Published: 2014
14 Title: Heads up, Artist: H my artist, Year Published: 2014
15 Title: I'm THE legend, Artist: I my artist, Year Published: 2014
16 Title: I'm THE legend, Artist: I my artist, Year Published: 2014
17 Title: Jealous woman, Artist: J my artist, Year Published: 2014
18 Title: Jealous woman, Artist: J my artist, Year Published: 2014
19 Title: Zoo, Artist: Zebra, Year Published: 2020
20 Title: Zoo, Artist: Zot, Year Published: 2020
21 Title: Zoo, Artist: Zoom, Year Published: 2020
22 Title: Zoo, Artist: Zoom, Year Published: 2020
---Total 22 songs---
```

If you want to print to a certain file, press 2. Input your filename, for example, printMyMusic, then the results will be stored in printMyMusic.txt.



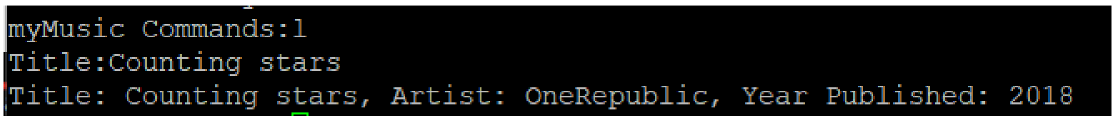
```
myMusic Commands:p
Press 1 to read from/write to terminal, press 2 to read from/write to file:
2
Input the file name:printMyMusic
---Print 22 songs to printMyMusic---
myMusic Commands:
```

- Look up songs by title

  Input L to start searching…

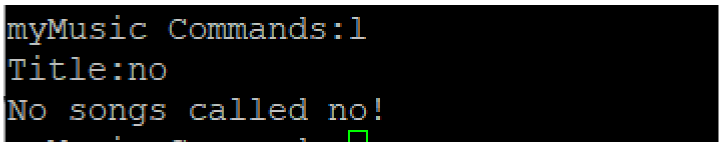
  For example, we want a song called "counting stars" (case doesn't matter)

  Here we go!



```
myMusic Commands:l
Title:Counting stars
Title: Counting stars, Artist: OneRepublic, Year Published: 2018
```

If you put in some songs that are not in the library, you will get an error.



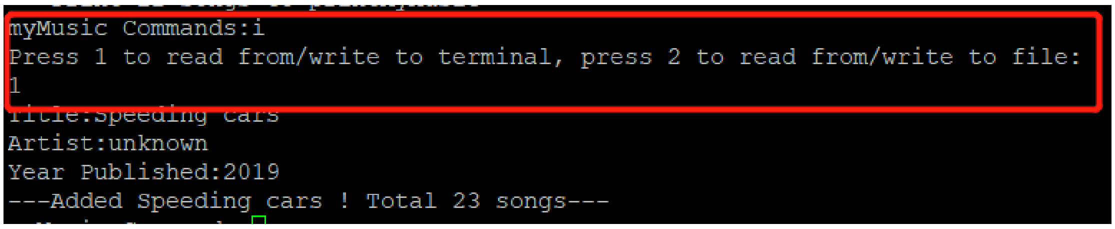
```
myMusic Commands:l
Title:no
No songs called no!
```

- Insert (from terminal/ file)

  Input i to insert new songs.

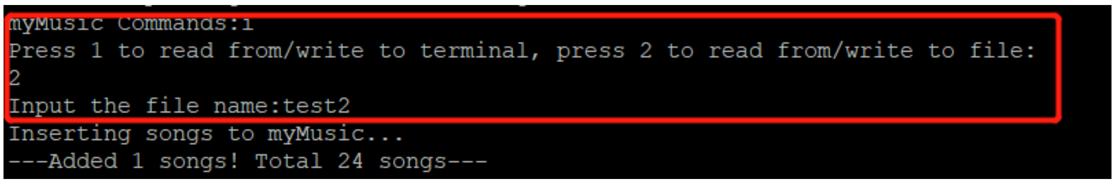
  Press 1, you can add songs from the terminal. For example, we input a song called “Speeding cars”, artist “unknown”, year published “2019”.

  Then “Speeding cars” is added. The total number of songs increase by 1.



```
myMusic Commands:i
Press 1 to read from/write to terminal, press 2 to read from/write to file:
1
Title:Speeding cars
Artist:unknown
Year Published:2019
---Added Speeding cars ! Total 23 songs---
```

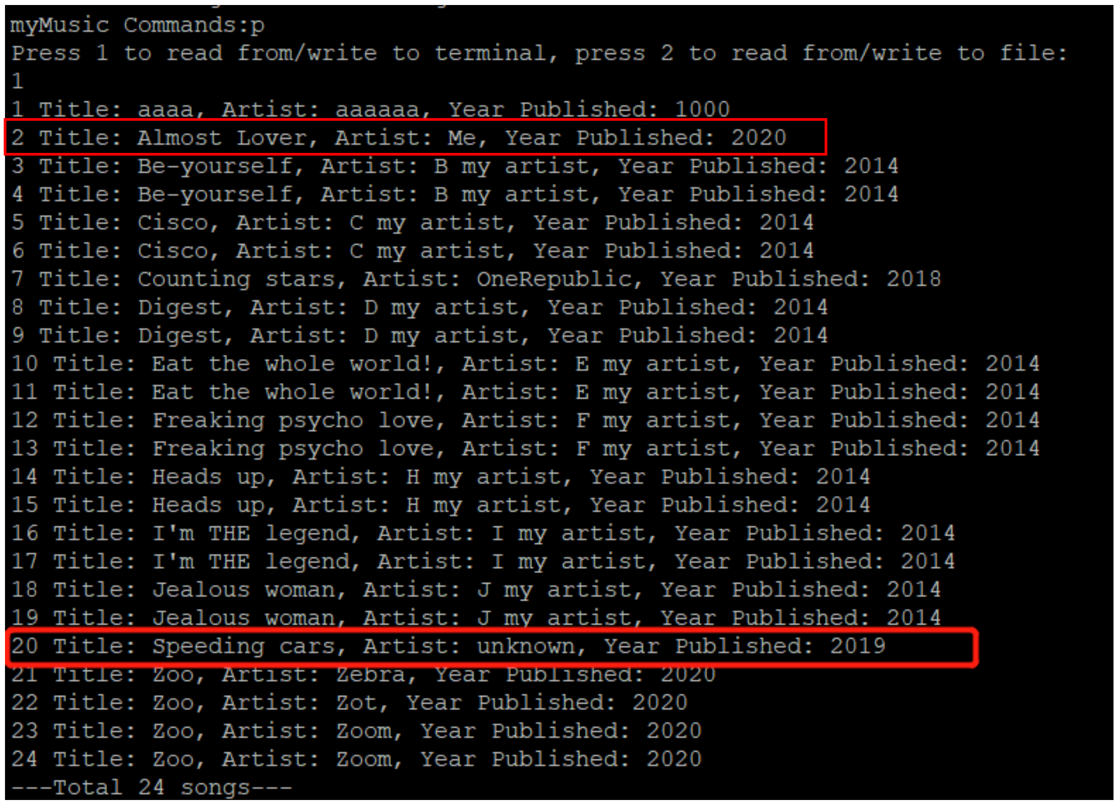
You can also insert songs from a file. Here is an example of adding songs from test2.txt. There is one song called “Almost Lover” in test2.txt



```
myMusic Commands:i
Press 1 to read from/write to terminal, press 2 to read from/write to file:
2
Input the file name:test2
Inserting songs to myMusic...
---Added 1 songs! Total 24 songs---
```

Then we print out the library to see the results.

Almost Lover and Speeding cars are already there!

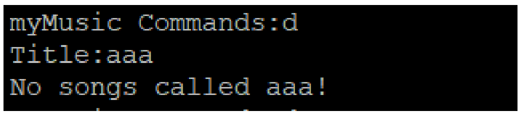


```
myMusic Commands:p
Press 1 to read from/write to terminal, press 2 to read from/write to file:
1
1 Title: aaaa, Artist: aaaaaa, Year Published: 1000
2 Title: Almost Lover, Artist: Me, Year Published: 2020
3 Title: Be-yourself, Artist: B my artist, Year Published: 2014
4 Title: Be-yourself, Artist: B my artist, Year Published: 2014
5 Title: Cisco, Artist: C my artist, Year Published: 2014
6 Title: Cisco, Artist: C my artist, Year Published: 2014
7 Title: Counting stars, Artist: OneRepublic, Year Published: 2018
8 Title: Digest, Artist: D my artist, Year Published: 2014
9 Title: Digest, Artist: D my artist, Year Published: 2014
10 Title: Eat the whole world!, Artist: E my artist, Year Published: 2014
11 Title: Eat the whole world!, Artist: E my artist, Year Published: 2014
12 Title: Freaking psycho love, Artist: F my artist, Year Published: 2014
13 Title: Freaking psycho love, Artist: F my artist, Year Published: 2014
14 Title: Heads up, Artist: H my artist, Year Published: 2014
15 Title: Heads up, Artist: H my artist, Year Published: 2014
16 Title: I'm THE legend, Artist: I my artist, Year Published: 2014
17 Title: I'm THE legend, Artist: I my artist, Year Published: 2014
18 Title: Jealous woman, Artist: J my artist, Year Published: 2014
19 Title: Jealous woman, Artist: J my artist, Year Published: 2014
20 Title: Speeding cars, Artist: unknown, Year Published: 2019
21 Title: Zoo, Artist: Zebra, Year Published: 2020
22 Title: Zoo, Artist: Zot, Year Published: 2020
23 Title: Zoo, Artist: Zoom, Year Published: 2020
24 Title: Zoo, Artist: Zoom, Year Published: 2020
---Total 24 songs---
```

- Deletion

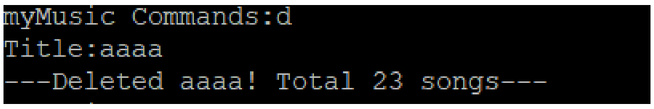
Input D to delete songs.

If you type in some songs that are not in the library. For example, aaa is not one of our song names.



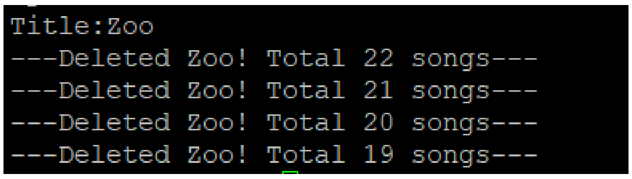
```
myMusic Commands:d
Title:aaa
No songs called aaa!
```

Delete one song called aaaa:



```
myMusic Commands:d
Title:aaaa
---Deleted aaaa! Total 23 songs---
```

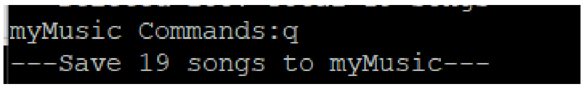
Delete all songs called Zoo:



```
Title:Zoo
---Deleted Zoo! Total 22 songs---
---Deleted Zoo! Total 21 songs---
---Deleted Zoo! Total 20 songs---
---Deleted Zoo! Total 19 songs---
```

- Save and quit

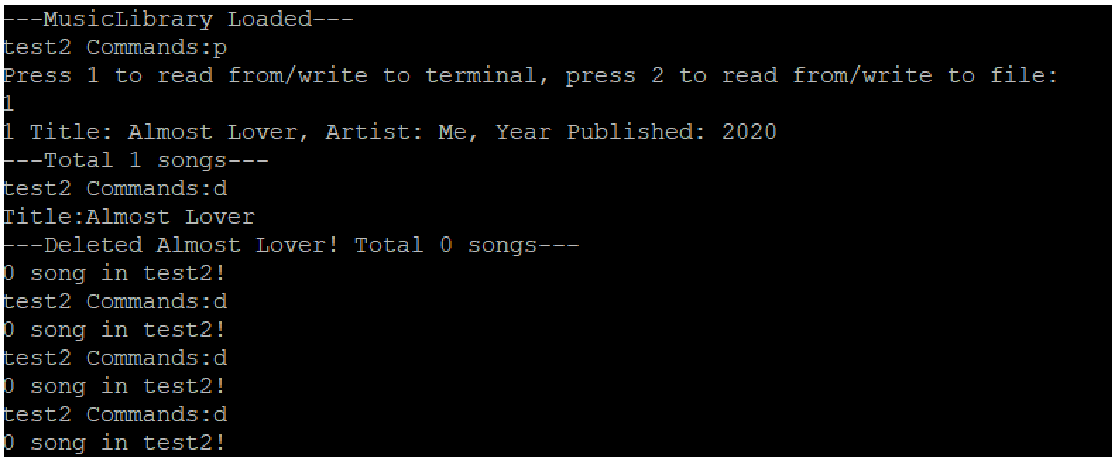
  Input Q, the music library will be automatically saved, and the program will quit.



```
myMusic Commands:q
---Save 19 songs to myMusic---
```

- Error handling

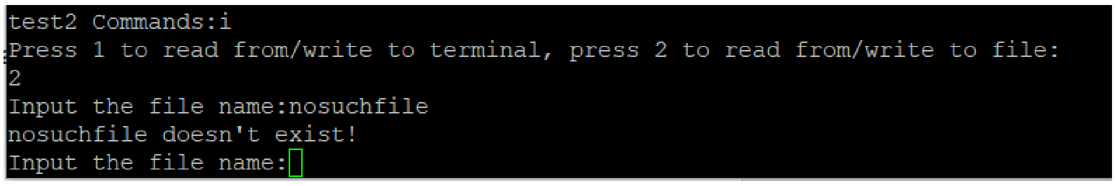
  You can't delete from an empty library.



```
---MusicLibrary Loaded---
test2 Commands:p
Press 1 to read from/write to terminal, press 2 to read from/write to file:
1
1 Title: Almost Lover, Artist: Me, Year Published: 2020
---Total 1 songs---
test2 Commands:d
Title:Almost Lover
---Deleted Almost Lover! Total 0 songs---
0 song in test2!
test2 Commands:d
0 song in test2!
test2 Commands:d
0 song in test2!
test2 Commands:d
0 song in test2!
```

  Nor add new songs to a full library.

You can’t load library or songs from a non-exist file.



```
test2 Commands:i
Press 1 to read from/write to terminal, press 2 to read from/write to file:
2
Input the file name:nosuchfile
nosuchfile doesn't exist!
Input the file name:
```

You have to put in a valid filename.

- Important methods
  - ✧ **read_command** can print a prompt, then read a character from the user using next().charAt(0). Skip any whitespace characters (space, tab, newline).
  - ✧ **evaluate_command** can take a command character and decide which command it is, then do the appropriate action.
  - ✧ **load_MusicLibrary** and **store_MusicLibrary** load or store a named MusicLibrary file into the memory MusicLibrary.
  - ✧ **print_MusicLibrary** that may be used for both storing the MusicLibrary to a file or for printing the MusicLibrary on the terminal for the P command. I passed in a bollean to indicate if I want print to terminal(true) or print to file print(false).
  - ✧ **isOverflow and isUnderflow** return a bollean value to see whether the library size if lager than MAX_SIZE or smaller than 0.
  - ✧ **crunch_up_from_index(i) and crunch_down_from_index(i)** Are auxiliary functions that each use a for loop to copy items up or down within the array from a given index. They will be used to insert or remove an item from the music list. <u>The time complexity of insert or remove is O(N) if N is the number of songs in the song list.</u>
  - ✧ **find_index_of_song_with_title** returns the location of the song with the specified name (used by remove and lookup). This function uses binary search and return the index of where it found the song or where the song should be if it were in this array list. <u>The time complexity is O(log(N))</u>
  - ✧ **remove_song_from_MusicLibrary_by_title** removes a song with the specified name. You will find it with int i = find_index_of_song_with_name(), then crunch_up_from_index(i).
  - ✧ **add_song_to_MusicLibrary** takes a song and puts it in the MusicLibrary in memory in the proper location. You could use i = find_index_of_song_with_name(...), then crunch_down_from_index(i), then buf[i] = song.
  - ✧ **write_song** and **read_song** handle writing and reading songs from a specified file.

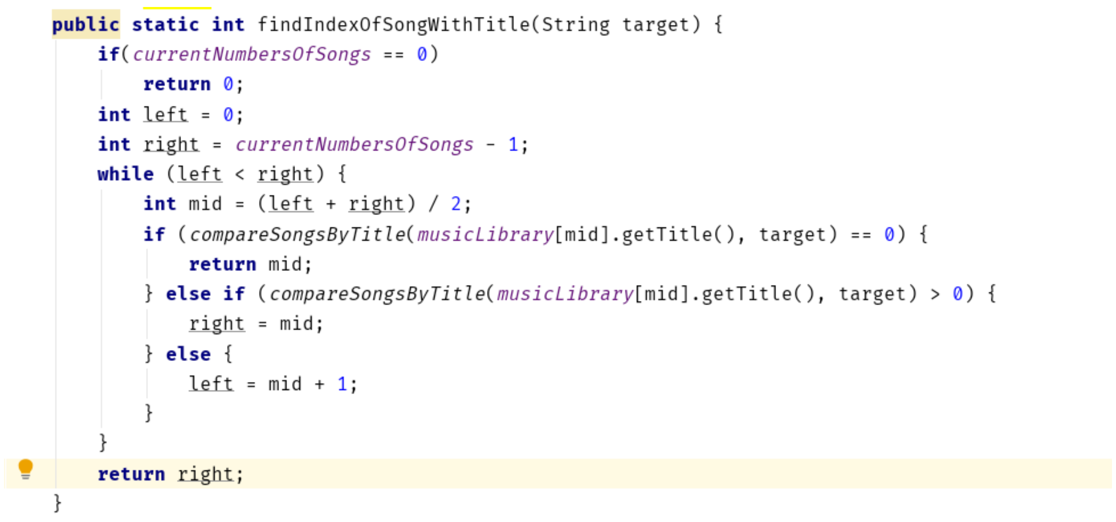
- Data structure

```
class Song {
      private String title;
      private String artist;
      private int yearPublished;
        public constructor();
        public getter();
        public setter();
        public toString();
 }
```

```
public class MusicLibrary {
public static int currentNumbersOfSongs = 0;
public static final int maxSize = 1024;
public static Song[] musicLibrary = new Song[maxSize];
...
other functions...
}
```

- Algorithms

✧ Search: Binary search. Time complexity: O(log(N))



```
public static int findIndexOfSongWithTitle(String target) {
    if(currentNumbersOfSongs == 0)
        return 0;
    int left = 0;
    int right = currentNumbersOfSongs - 1;
    while (left < right) {
        int mid = (left + right) / 2;
        if (compareSongsByTitle(musicLibrary[mid].getTitle(), target) == 0) {
            return mid;
        } else if (compareSongsByTitle(musicLibrary[mid].getTitle(), target) > 0) {
            right = mid;
        } else {
            left = mid + 1;
        }
    }
    return right;
}
```

✧ Insert: use i = find_index_of_song_with_name(...), then crunch_down_from_index(i), then buf[i] = song. Time complexity: O(N)

✧ Delete: int i = find_index_of_song_with_name(), then crunch_up_from_index(i)