ECE 250 - Project 0
Playlist as Array
Design Document
Harshit Manchanda, UW UserID:h2mancha
Jan 17th, 2020

1. Overview of Classes

Class 1:
Song

Description:
Represents a song as an object of this, stored inside the Playlist array and provides operations like setting the title and artist for that object and extracting the information when required.

Member Variables:
1. Song Title
2. Song Artist

For example, a song object inside the array will hold the values Mamma Mia as the title and Abba as the artist.

Member Functions (operations):
1. setTitleArtist: sets the title and artist for that object passed as parameters.
2. getTitle: extracts the title from the song and then returns that string.
3. getArtist: extracts the artist from the song and then returns that string.

Class 2:
PlaylistManager

Description:
Represents a playlist of songs as an array and provides operations associated with songs ie. adding a song, playing a song, erasing a song, etc.

Member Variables:
1. A dynamic Song type array that stores the songs
2. Size of the playlist
3. Number of songs present in the playlist

For example, the playlist for -
m 4
i Mamma Mia;Abba
i We Will Rock You;Queen
i Daniel;Elton John

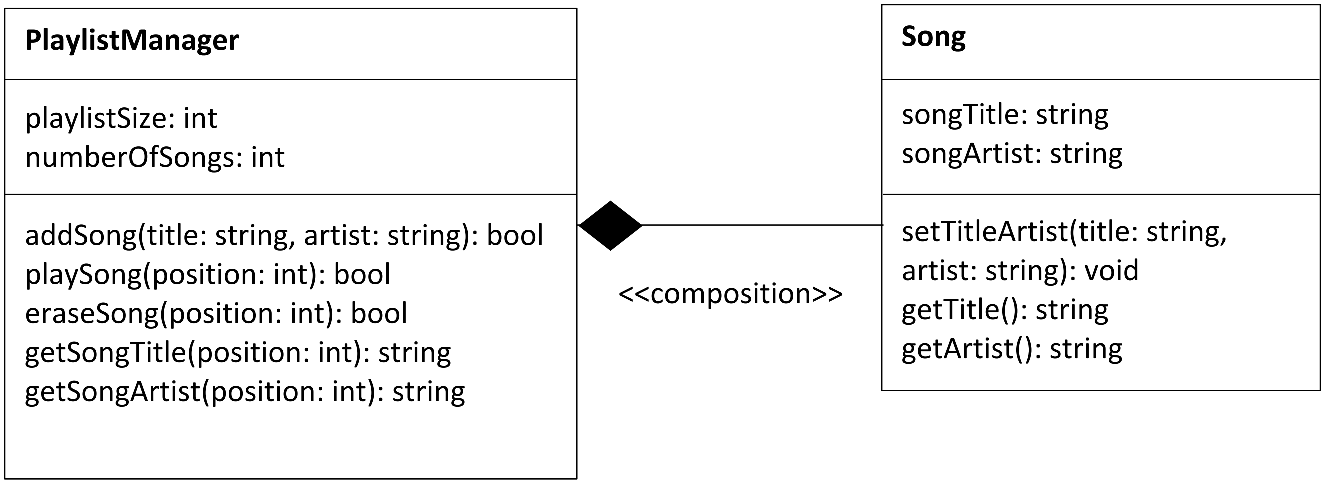
will be represented as an array:

song object1	song object2	song object3	song object4
Mamma Mia	We Will Rock You	Daniel	
Abba	Queen	Elton John	

Member Functions (operations):

1. addSong: adds the title and artist passed as parameters to the song object in the array only if the playlist isn't full or the song doesn't exist in it.
2. playSong: plays the song at the position passed as a parameter if there is a song at that position.
3. eraseSong: deletes the song at the position passed as a parameter and shifts the rest of the songs to the left.
4. getSongTitle: extracts and returns the title of the song object at the position passed as a parameter.
5. getSongArtist: extracts and returns the artist of the song object at the position passed as a parameter.

UML Class Diagram -


PlaylistManager
playlistSize: int
numberOfSongs: int
addSong(title: string, artist: string): bool
playSong(position: int): bool
eraseSong(position: int): bool
getSongTitle(position: int): string
getSongArtist(position: int): string
<<composition>>
Song
songTitle: string
songArtist: string
setTitleArtist(title: string, artist: string): void
getTitle(): string
getArtist(): string

2. Constructors/Destructors

Class Song (Constructor):
The constructor for this class just assigns the title and artist string for the object as empty strings because there will be no need to put any value in them unless the song needs to be added.

Class Song (Destructor):
There is no need to create a destructor for this class as strings are the only member variables and their memory will be automatically deallocated by the compiler.

Class PlaylistManager (Constructor):
The constructor takes in the size as a parameter from the input file (given with the first command) and creates a dynamic playlist array of type Song. Since the first command "m" will always be present, there is no need for a default constructor.

Class PlaylistManager (Destructor):
The destructor is necessary for this class since the playlist is a dynamic array and the memory isn't freed on its own at the end of the program so there is a need to manually deallocate the memory. For this, the destructor deletes the array and changes the value of the playlistSize and numberOfSongs to 0.

3. Test Cases

Test 1: Create a playlist for a given size

Test 2: Add songs in the playlist
Try to add different songs, same songs, songs when the playlist is full.

Test 3: Play songs from the playlist
Try to play songs which are present in the playlist and not present in the playlist by specifying the index bigger than the number of songs.

Test 4: Erase songs from the playlist
Try to erase random songs from the playlist either within the bounds or out of bounds.

<u>Test File Example 1:</u>

```
m 100
i We Will Rock You;Queen
i Donwtown;Petula Clark
i Mamma Mia;Abba
i Rocket Man;Elton John
i Daniel;Elton John
```

```
p 3
p 7
i Somebody to love;Queen
i Mamma Mia;Abba
e 2
p 2
e 10
i We Will Rock You;Queen
```

Test File Example 2:

```
m 4
i Crocodile Rock;Elton John
i Night and Day;Frank Sinatra
p 5
p 2
e 1
i Venus;Bananarama
i We Will Rock You;Queen
i Daniel;Elton John
i Mamma Mia;Abba
i Somebody to love;Queen
e 4
e 3
i We Will Rock You;Queen
i Daniel;Elton John
p 3
p 4
```

References:

For test cases - https://github.com/ece23/ECE250-testCases

For help in code – Tutorial Polynomial project