

1. Overview of classes

Class:

playlist

Description:

It represents a playlist as an array of pointers to strings. It includes operations such as dynamically allocates an array with specified size, adding songs to the end of playlist, inserting a song to a specified index, erasing songs at a specified index, playing songs at a specified index, and printing the entire array.

Member variables:

max_list

The variable that stores an int value as the maximum size of the array.

songlist

The array of pointers to strings.

Member functions (operations)

- AddSong: Taking a string **song** as a parameter, adding it to the end of the playlist. No return value.
- InsertSong: Taking an int **_pos** and a string **song** as parameters. Inserting **song** to index **_pos**, and right shifting elements that have larger indexes. No return value.
- DeleteSong: Taking an int **_pos** as a parameter, erasing the song at index **_pos**. No return value.
- PlaySong: Taking an int **_pos** as a parameter, printing: “played” + “song at index **_pos**” to the output if the input is valid. No return value.
- PrintSong: No input parameter. Printing the entire playlist. No return value.

UML class Diagram

playlist
- max_list: int - songlist: *std::string
+ AddSong (song: std::string): void + InsertSong (_pos: int, song:: std::string): void + DeleteSong (_pos: int): void + PlaySong (_pos: int): void + PrintSong (): void

2. Constructors/Destructor/Operator overloading

- For the constructor of class playlist, I would like to dynamically allocate an array of pointers to strings. There should also be an output present when the constructor is called.
- For the destructor, I would like to completely deallocate the memory taken up by the array, and then set all elements to nullptr.
- I have created two additional operations: InsertSong and PrintSong. Users are allowed to use InsertSong to insert a certain song to a certain index of the playlist, just like how I would use my music players in real life. The PrintSong operation would allow users to print out the entire playlist, which enables them to navigate through the playlist in a more efficient manner.

3. Test Cases

test01: Testing AddSong, PlaySong and DeleteSong functionalities.

test02: Testing if My Heart Will Go On and Muskrat Love by Captain and Tennile could be added to the playlist. Testing DeleteSong functionality when trying to delete a song at the middle of the playlist (left-shifting the array).

test03: Testing if My Heart Will Go On by different artists could be added into the playlist. Further testing if Muskrat Love by other artists could be added into the playlist. Utilizing PlaySong operation to play songs in and out of range.

m 10 (creating a size-10 playlist)

.....

p 5

p 6 (checking if playing an out-of-range song would result in error)