

Project 3: Classes and Class Functions

Spotify-like Music System Assignment

Objective:

Develop a system that simulates a music library similar to Spotify. Your task is to utilize object-oriented programming principles to create and manage songs, playlists, and a play queue. Throughout this project, your application should provide a seamless user experience, allowing music enthusiasts to interact with their favorite songs in meaningful ways.

Overview:

Song Object: Each song should have the following attributes:

- Title
- Album
- Artist
- Genre
- Liked status (True/False)
- Year of release
- Rating (out of 5)

Library Class: This class should maintain:

- A collection of song objects.
- Methods to add a song, like a song, change the rating of a song, and search for songs.

Playlist Class: This class should encompass:

- A collection of songs specific to the playlist.
- Methods to add and remove songs from the playlist.

Queue Class: This class should:

- Maintain a list of songs to be played.
- Allow songs to be queued and dequeued.
- Provide a method to play songs, printing them out in order until the queue is empty.

Examples of Input and Output:

Add a song to library

Input:

User selected option: 1

Enter song title: Mo Bamba

Enter song album: Mudboy

Enter song artist: Sheck Wes

Enter song genre: Rap

Enter song year: 2018

Enter song rating (1-5): 4

Output:

Song 'Mo Bamba' added to library!

Like a Song

Input:

User selected option: 2

Enter song title to like: Nonstop

Output:

Song 'Nonstop' has been liked!

Changing Song Rating

Input:

User selected option: 3
Enter song title: Sicko Mode
Enter new rating (1-5): 5

Output:
Rating for 'Sicko Mode' updated to 5.

Searching for Songs

Input:
User selected option: 4
Search by (title/album/artist/genre/year/rating):
artist
Enter artist name: Post Malone

Output:
Songs by Post Malone:
- Rockstar

Create a Playlist

Input:
User selected option: 5
Enter name for the new playlist: Rap Bangers

Output:
Playlist 'Rap Bangers' created!

Add a Song to a Playlist

Input:
User selected option: 6
Enter song title: God's Plan
Enter playlist name: Rap Bangers

Output:

Song 'God's Plan' added to 'Rap Bangers' playlist!

Remove a Song from a Playlist

Input:

User selected option: 7

Enter song title to remove: God's Plan

Enter playlist name: Rap Bangers

Output:

Song 'God's Plan' removed from 'Rap Bangers' playlist!

View All Playlists

Input:

User selected option: 8

Output:

Available Playlists:

1. Rap Bangers

2. Weekend Chill

Add a Song to Queue

Input:

User selected option: 9

Enter song title: Money Trees

Output:

Song 'Money Trees' added to the play queue!

Play the Queue

Input:

User selected option: 10

Output:

Now playing: Money Trees

...

Exit

Input:

User selected option: 11

Output:

Thank you for using the music library. Goodbye!

Spotify-like Music System Rubric (100 points)

Song Management & User Preferences (30 points)

- Correct creation and manipulation of song objects based on user input.
- Proper implementation of song liking and rating functionalities.
- Accurate search results based on various song attributes.

User Interface (20 points)

- An intuitive and easy-to-follow text-based interface.
- Clear feedback provided to the user regarding actions performed and their outcomes.

Organization (15 points)

- Logical code structure.
- Appropriate separation between data models (songs, playlists) and user interface.

Readability (15 points)

- Consistent naming conventions.
- Proper use of whitespace and indentation.

Comments (10 points)

- Essential sections of the code are commented on.
- Comments are concise and provide meaningful insights.
- Defend any techniques that are used outside the scope of this class.

Testing (10 points)

- Clearly defined test cases, accounting for typical interactions and edge cases.
- Test cases accurately validate system functionality.
- Annotations explaining the objective of each test case and the expected result.

Bonus: Additional Music Features (5 points)

- Integration of features beyond the basic requirements.

- Be Creative!

Bonus: Innovative Approach (5 points)

- Creative strategies or unique features that enhance user interaction or the overall music experience.