

Playlist Management System

Description

The Playlist Management System leverages the Composite Design Pattern to handle hierarchical structures within a media player application. This system allows users to create and manage playlists, which can contain both individual songs and nested playlists. Users can interact with the system to add songs, create new playlists, and navigate through existing playlists. The system also demonstrates the use of logging to trace actions such as adding songs, creating playlists, and playing songs.

Design Pattern-Composite Pattern

This pattern is used to treat both individual songs and playlists uniformly. It allows the media player to manage and manipulate complex hierarchical structures (playlists containing other playlists) as if they were single objects.

File Structure

1. **Component.java**: Abstract class defining the common interface for both Song and Playlist.
2. **Song.java**: Concrete class representing a single song with attributes for name and artist. Implements the play method to play the song.
3. **Playlist.java**: Concrete class representing a playlist that can contain other Component objects (both Song and nested Playlist). Includes methods to add, remove, and play components, as well as to find a playlist by name.
4. **MediaPlayer.java**: Main application class that handles user interactions. Provides options for creating playlists, adding songs, playing the next song, finding playlists, and displaying the current playlist. Uses logging to track actions.