## Description:

Develop a MelodyMix Manager system in Java, aimed at managing playlists of music tracks. The system provides essential functionalities for organizing music collections efficiently. Key features include:

1. Storage for Songs

2. Add Songs to Playlist

3. Remove Songs from Playlist

4. Print Playlist Information

## Structure

1. Song:

   - Represents an individual music track within the MelodyMix Manager system.
   - Contains attributes for storing information about the song, including its title and artist.

2. MelodyMixManager:

   - Acts as the core component responsible for managing playlists of music tracks.
   - Utilizes a List data structure to store playlists, each comprising multiple songs.
   - Provides methods for adding new songs to playlists, removing songs from playlists, and printing playlist information.

3. LibraryTester:

   - Functions as the testing class for the MelodyMix Manager system.
   - Contains a main method to validate the functionalities of the MelodyMixManager class.

class song:

string - title

string - artist

class PlailistManager:

List storage

void addSong(Song song)

boolean removeSong(Song song)

void printPlaylists()

## Class Song

The class Book should have several fields, including title and artist. This class can be implemented in the following way:

```
package musiclibrary;

import java.util.ArrayList;
import java.util.List;

public class Song {
    private String title, artist;

    public String getTitle() {
```

```java
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getArtist() {
        return artist;
    }

    public void setArtist(String artist) {
        this.artist = artist;
    }
}
```

## Class `PlaylistManager`

Acts as the core component responsible for managing playlists of music tracks. Utilizes a List data structure to store playlists, each comprising multiple songs. Provides methods for adding new songs to playlists, removing songs from playlists, and printing playlist information.

```java
package musiclibrary;

import java.util.ArrayList;
import java.util.List;

public class PlaylistManager {
    private List playlists = new ArrayList<>();

    public void createPlaylist(Song song) {
        playlists.add(song);
    }

    public boolean removeSongFromPlaylist(Song song) {
        boolean removed = false;
        for (int i = 0; i < playlists.size(); i++) {
            Song s = playlists.get(i);
            if (s.getTitle().equals(song.getTitle()) && s.getArtist().equals(song.getArtist())) {
                playlists.remove(i);
                removed = true;
                break;
            }
        }
        return removed;
    }

    public void printPlaylists() {
        if (playlists.isEmpty()) {
            System.out.println("No playlists available.");
        } else {
            for (Song s : playlists) {
                System.out.println(s.getArtist() + ", " + s.getTitle());
                System.out.println();
            }
        }
    }
}
```

## LibraryTester class

Functions as the testing class for the MelodyMix Manager system. Contains a main method to validate the functionalities of the MelodyMixManager class.

```java
package musiclibrary;

public class LibraryTester {
```

```java
    public static void main(String[] args) {
        Song song1 = new Song();
        song1.setTitle("God's Menu");
        song1.setArtist("Stray Kids");

        Song song2 = new Song();
        song2.setTitle("1, 2, 3, 4, 5");
        song2.setArtist("Han");

        PlaylistManager playlistManager = new PlaylistManager();
        playlistManager.createPlaylist(song1);
        playlistManager.createPlaylist(song2);
        playlistManager.removeSongFromPlaylist(song1);
        playlistManager.printPlaylists();
    }
}
```
We print the state of the music library to check if all the methods are working properly.