Playlist - Inner Class

Description:

For this exercise you will need your `Album` and `Song` classes from the **Playlist** Coding Exercise.

The `Album` class will be modified to use an **inner class** called `SongList` which will replace the `ArrayList` that was used to hold the list of songs for an album.

The `SongList` inner class will use an `ArrayList` to hold the track list for the album. To do this a member variable called **songs** will need to be declared and a constructor will need to be implemented to initialize the field.

In addition, the inner class will need three(3) methods:

1) **add** - accepts a parameter of type Song. This method returns a **boolean** value which will be **false** if the song is found to already be in the songs list. If not, the song will be added to the list and a value of **true** will be returned.

2) **findSong** - accepts the title of a song as its only parameter. If the song with that title is found in the list then the song is returned from the method. If not, a value of **null** is returned.

3) **findSong** - overridden method which accepts a track number for a song. The track number is of type **int**. If a song is found with that track number the song is returned, otherwise **null** is returned.

The member variable, constructor and all methods may be marked with **private** access.

IMPORTANT - If you are using an IDE to write this code you may have your inner class marked with private access. However, for the purposes of this code evaluation please mark the class itself **public static**.

There are only three methods in the album class.

1) **addSong(String title, double duration)**

2) **addToPlayList(int trackNumber, LinkedList<Song> playList)**

3) **addToPlayList(String title, LinkedList<Song> playList)**

Change the code in these three methods so that it uses the `SongList` inner class rather than the `ArrayList` field.

Example input:

```
Album album = new Album("Stormbringer", "Deep Purple");
album.addSong("Stormbringer", 4.6);
album.addSong("Love don't mean a thing", 4.22);
album.addSong("Holy man", 4.3);
album.addSong("Hold on", 5.6);
album.addSong("Lady double dealer", 3.21);
album.addSong("You can't do it right", 6.23);
album.addSong("High ball shooter", 4.27);
album.addSong("The gypsy", 4.2);
album.addSong("Soldier of fortune", 3.13);
albums.add(album);

album = new Album("For those about to rock", "AC/DC");
album.addSong("For those about to rock", 5.44);
album.addSong("I put the finger on you", 3.25);
album.addSong("Lets go", 3.45);
album.addSong("Inject the venom", 3.33);
album.addSong("Snowballed", 4.51);
album.addSong("Evil walks", 3.45);
album.addSong("C.O.D.", 5.25);
album.addSong("Breaking the rules", 5.32);
album.addSong("Night of the long knives", 5.12);
albums.add(album);

LinkedList<Song> playList = new LinkedList<Song>();
albums.get(0).addToPlayList("You can't do it right", playList);
albums.get(0).addToPlayList("Holy man", playList);
albums.get(0).addToPlayList("Speed king", playList);  // Does not exist
albums.get(0).addToPlayList(9, playList);
albums.get(1).addToPlayList(8, playList);
albums.get(1).addToPlayList(3, playList);
albums.get(1).addToPlayList(2, playList);
albums.get(1).addToPlayList(24, playList);  // There is no track 24
```

Example output:

```
The song Speed king is not in this album
This album does not have a track 24
```