PLAYLIST

Create a program that implements a playlist of songs.

To start you off, implement the following classes:

1. Album

- It has three fields, two Strings called name and artist, and an ArrayList that holds objects of type Song called songs.

- A constructor that accepts two Strings (name of the album and artist). It initialises the fields and instantiates songs.

- And three methods, they are:

  - addSong(), has two parameters of type String (title of song), double (duration of song) and returns a boolean. Returns true if the song was added successfully or false otherwise.

  - findSong(), has one parameter of type String (title of song) and returns a Song. Returns the Song if it exists, null if it doesn't exists.

  - addToPlayList(), has two parameters of type int (track number of song in album) and LinkedList (the playlist) that holds objects of type Song, and returns a boolean. Returns true if it exists and it was added successfully using the track number, or false otherwise.

  - addToPlayList(), has two parameters of type String (title of song) and LinkedList (the playlist) that holds objects of type Song, and returns a boolean. Returns true if it exists and it was added successfully using the name of the song, or false otherwise.

2. Song

- It has two fields, a String called title and a double called duration.

- A constructor that accepts a String (title of the song) and a double (duration of the song). It initialises title and duration.

- And two methods, they are:

  - getTitle(), getter for title.

  - toString(), Songs overriding toString method. Returns a String in the following format: "title: duration"

-> SAMPLE INPUT

```
ArrayList<Album> albums = new ArrayList<>();
```

```
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(3, playList);
albums.get(1).addToPlayList(2, playList);
albums.get(1).addToPlayList(24, playList);  // There is no track 24
```

TIP: In Album, use the findSong() method in addSong() and addToPlayList(String, LinkedList) to check if a song exists before proceeding.

TIP: Be extremely careful with the spelling of the names of the fields, constructors and methods.

TIP: Be extremely careful about spaces and spelling in the returned String from the toString() method.

NOTE: All fields are private.

NOTE: All constructors are public.

NOTE: All methods are public (except for findSong() which is private).

NOTE: There are no static members.

NOTE: Do not add a main method to the solution code.

NOTE: If you get an error from the Evaluate class, it's most likely the constructor. Check if you've added a constructor or if the constructor has the right arguments.