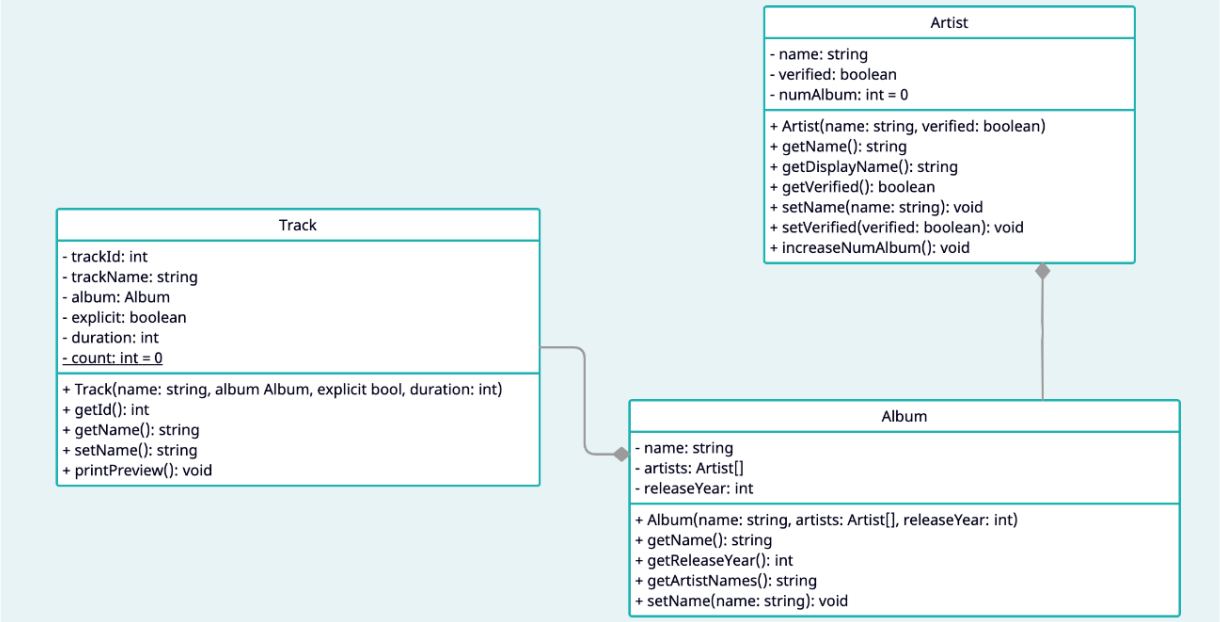
Consider the below UML diagram and write Java classes accourding to the following requirements.



Artist class stores artist name, verification status, and number of their albums started with 0. Its method contains default getter for the name and verification status also with addition methods as follows:

- *increaseNumAlbum* which will increase *numAlbum* value by 1
- *getDisplayName* will return the name of artist plus if they have verified, append the **(/)** as the check mark after their name (for example **Christina Grimmie (/)**), otherwise return only their namealone.

Album class stores the album name, release year, artists (can be multiple artist) and tracks. Everytime album has constructed, it should increase number of albums for every artist by calling *increaseNumAlbum* for eachartist. It has the method of setter for name and getter for name, release year and method *getArtistName* will return the display name of every artist seprated by comma and one space e.g. **Idina Menzel (/), AURORA**.

Track class stores unique track ID, name, album, explicit and duration (in seconds) of the track. It will have static integer, *count*, for auto increasement value used in assging track id, start from 0. It have getter for their name and ID, also with one addition method, *printPreview* which will print the track information as follow example:

```
(E) The Boy Who Shattred Time
4:03
Warsongs
League of Legends (/), MitiS (/)
```

1st line: Print track name. If the track is explicit, add suffix (E) before the track name

2nd line: Print the duration of track in format mm:ss (converted from number in seconds)

3rd line: Print the album name

4th line: Print the display name of all artists of the album

The driver class is provided in the preload answer box already, please do not modify the

```
public class Driver {
	public static void main(String[] args) {
		Artist artist1 = new Artist("Artist 1", false);
		Artist artist2 = new Artist("Artist 2", false);
		Artist artist3 = new Artist("Artist 3", true);
	    System.out.printf("%s %s %s\n", artist1.getName(), artist1.getDisplayName(), artist1.getNumAlbum());
		System.out.printf("%s %s %s\n", artist3.getName(), artist3.getDisplayName(), artist3.getNumAlbum());
```

```
        artist1.setName("Artist 1 Edited");
        artist2.setVerified(true);

        System.out.printf("%s %s %s\n", artist1.getName(), artist1.getDisplayName(),
artist1.getNumAlbum());
        System.out.printf("%s %s %s\n", artist2.getName(), artist2.getDisplayName(),
artist2.getNumAlbum());

        Album album1 = new Album("Album 1", new Artist[] { artist1, artist2 }, 2017);
        Album album2 = new Album("Album 2", new Artist[] { artist1, artist3 }, 2018);

        System.out.printf("%s %s %s\n", album1.getName(), album1.getReleaseYear(),
album1.getArtistNames());
        System.out.printf("%d %d %d\n", artist1.getNumAlbum(), artist2.getNumAlbum(),
artist3.getNumAlbum());

        Track track1 = new Track("Track 1", album1, true, 200);
        Track track2 = new Track("Track 2", album2, false, 210);

        System.out.printf("%s %s\n", track1.getId(), track1.getName());
        System.out.println(track1.getId());
        track1.printPreview();
        System.out.println(track2.getId());
        track2.printPreview();

        artist1.setVerified(true);
        artist3.setName("Artist 3 Edited");
        album1.setName("Album 1 Edited");
        track2.setName("Track 2 Edited");

        System.out.printf("%s %s %s\n", album1.getName(), album1.getReleaseYear(),
album1.getArtistNames());
        track2.printPreview();


    }
}
```