CSC8406 - Object Oriented Programming

PROJECT REPORT

Title - Music Library

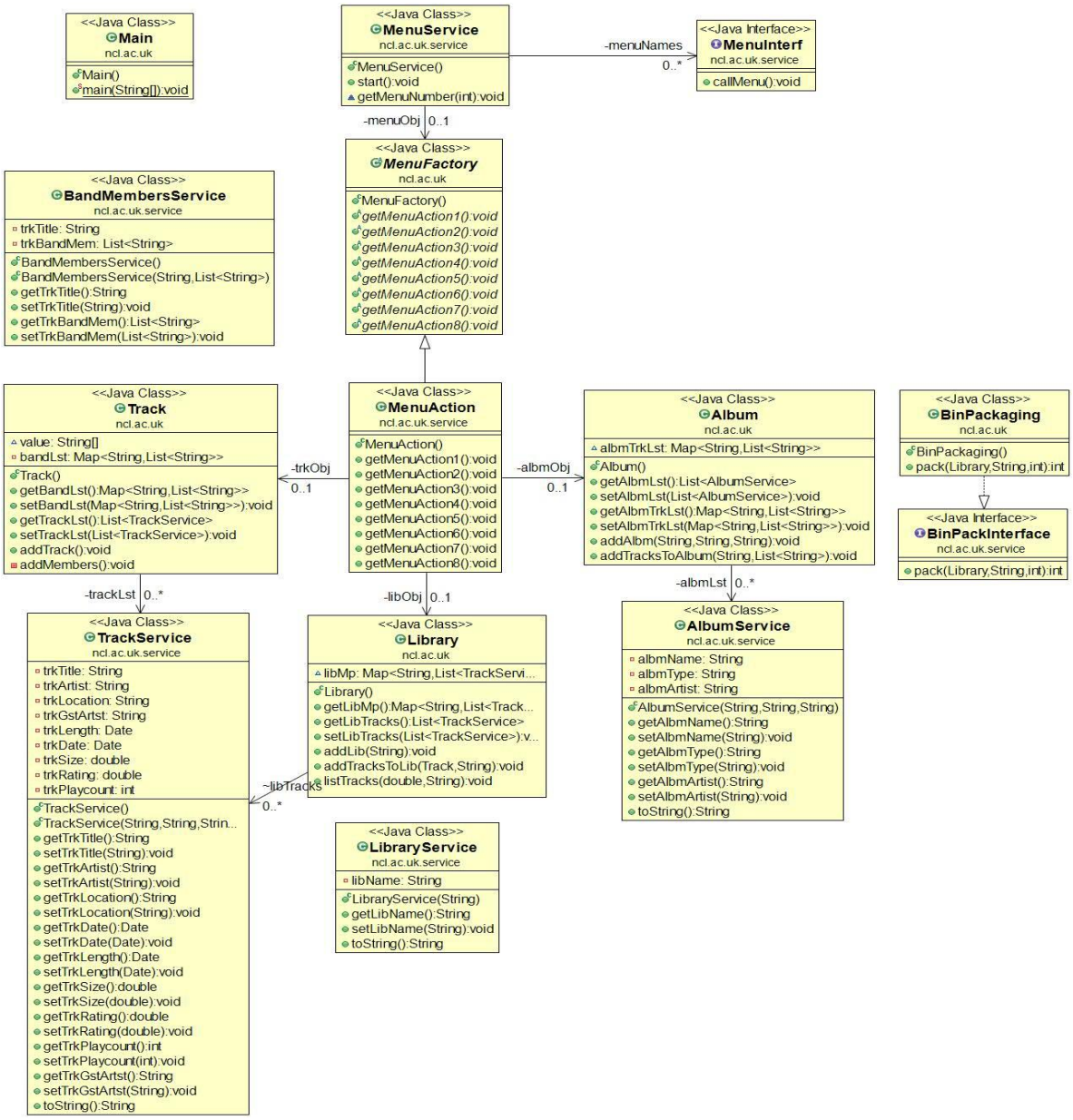
Table of Contents

1. Background

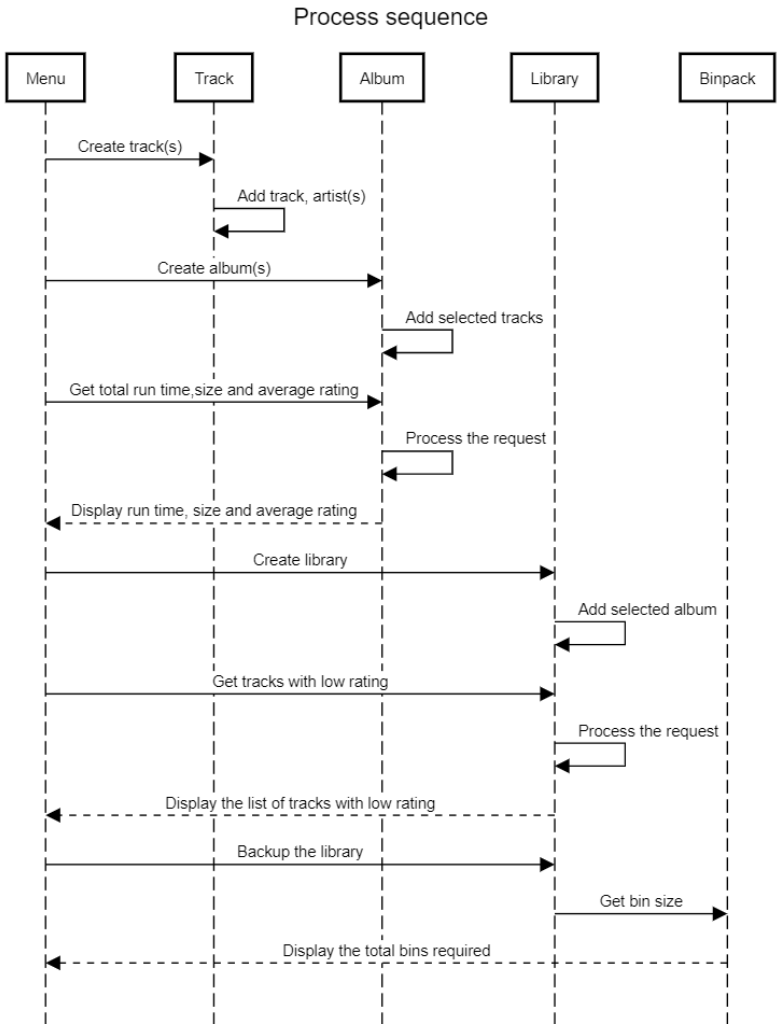
Music library refers to a virtual directory that holds music tracks and albums. Tracks are commonly referred to media files which can be of different extension. Each track is considered as an asset to the composer or an artist. Tracks are generally contributed by artist, guest artists and a group of members called band members. A group of tracks can combine to form an album. A composer/artist can have any number of albums where each album can hold any number of tracks.

In relation to this coursework it is understood to develop a music library that is capable of creating new tracks with its related attributes, adding band members/guest artist if applicable, creating an album with its attributes, adding tracks to the album, creating a library, adding tracks/ albums to the library, listing out the tracks under a low rating and finally to pack the library to storage system commonly called as bin packing. To develop this application on JAVA 1.8 JDK is used under Eclipse IDE and object-oriented programming techniques are exercised.

2. Class Diagram



3. Sequence Diagram



4. Discussion

Menu is the driver class which acts as a UI for user inputs. When main() method executed Menuservice class instance is created and invokes the start() instance method. The start() method displays the list of menu's and scanner object is ready to accept the input from the user. To reduce the several if-else conditional statements based on user input to invoke the appropriate method which makes the code more complex, therefore it is decided to use functional interface MenuInterf with single abstract method called callMenu().

MenuFactory class is an abstract class that provides the functionality to the user selected menu which is extended by a sub class MenuAction. MenuAction class contains 8 instance methods with a default constructor that calls super class constructor. This class is responsible for each menu action that user selects. Three instances (Track, Album, Library) classes are created to invoke its own member methods. To start with Track class is responsible to add tracks to the collection from the resource directory (Tracks.txt) where attributes are separated by comma. The schema is explained below.

```
Blinding Lights,The Weeknd,,08/01/2020,3:21,8.9,c:/users/music/Weekend,3.59,0
```

The first value contains the Track name, followed by Artist name and if guest artist is present will be available else empty value, then date of the track and length, rating for the track, location of the track, size of the track finally play count.

Once the tracks are added automatically track members are also added to the track from the resource directory (TrackMembers.txt) where attributes are separated by comma. The schema is described below.

```
Blinding Lights,Tesfaye
```

Foremost is the track name followed by the band members name. Once all the values are added to the collection, instance method addTrack() return control back to the menu interface.

The Album class addAlbum() is responsible to create new album with a name, genre and type once the album is created immediately it adds the selected number of tracks to album using addTracksToAlbum().

The library class addLibrary() is responsible to create new library. Once the library is created addTracksToLib() is invoked to add selected number of tracks to the created library. ListTracks() method displays the number of tracks available in the collection with respect to the library name.

Bin packing problem is NP hard problem which is used for identifying the best solution to storage. This application uses first fit ascending approach where the file sizes in the library are sorted to an ascending order and accepts the bin size from the user and fits the file into bin and find out how many bins are required to fit all the files. The entire process is handled by pack() method overridden from BinPackInterface

Test cases are written using jUnit 4.1 and is place under the package ncl.ac.uk.test which contains 3 test cases to check the main method which invokes the Menuservice and on user selection for Tracks and members are mapped to the appropriate collections are verified.

5. Test Results

```
----------Music Library Menu----------
1. Add Track
2. Get Tracks and Members
3. Add New Album
4. Add Tracks to Album
5. Get Run time, Size, Avg Rating of Album
6. Create Library and add tracks
7. List tracks with low rating from library
8. Backup the library
9. Close
Enter the option number: 1
Tracks created successfully..
Members are added to the corresponding tracks..

Enter the option number: 2
---Track Name------Members---
After Hours     Tesfaye
                Quenneville
                Balshe
                Montagnese
                Mario Winans
Alone Again     Abel Tesfaye
                Jason
                Quenneville
                Carlo
                Montagnese
                Adam Feeney
Too Late        Tesfaye
                Quenneville
                Montagnese
                Eric
```

```
              Burton Frederic
Blinding Lights Tesfaye
                Balshe
                Quenneville
                Sandberg

Track Title   Artist Name  Guest Artist       Date   Track Length       Rating
       Location     File Size    Play count
Alone Again  The Weeknd               21/11/2019   04:10  3.8
      c:/users/music/Weekend          3.12         0
Too Late     The Weeknd   Lil Uzi Vert 26/12/2019   03:59  5.3
      c:/users/music/Weekend          4.02         0
Blinding Lights     The Weeknd              08/01/2020   03:21  8.9
      c:/users/music/Weekend          3.59         0
After Hours  The Weeknd   Chromatics   12/02/2020   06:01  7.3
      c:/users/music/Weekend          5.6          0

Enter the option number: 3
Album Name    Type   Artist
-----------------------------------
After Hours  Rock   The Weeknd

Album created successfully..

Enter the option number: 4
Album Name ----- Track Name
After Hours  Alone Again
             Blinding Lights

Added the above tracks to album

Enter the option number: 5
Album running time: 06:31
Album file size:    6.71 MB
Average rating :    6.35

Enter the option number: 6
--Library Name--
American Music
Added tracks to library -American Music..

Enter the option number: 7
Track Name    Length       Rating       Size   Play Count
----------
Alone Again  04:10        3.8          3.12 MB      0
Too Late     03:59        5.3          4.02 MB      0
Enter the option number: 8

Enter the bin capacity: 10
Bin size is 10
Total bins required are: 2
```

6. Outcome

From this module and coursework it is identified that there is a tremendous up trend in the learning curve and gained relevant knowledge about object oriented programming on how to use collection framework, constructors, abstract classes, interfaces, inheritance, encapsulation and how to achieve runtime polymorphism also helped to visualize things easier. Simultaneously, practiced some Java 1.8 features towards functional programming like lambda expression and streams API. This had laid a strong foundation towards Java programming paradigm.

7. Run Instructions

1. This program is developed under Eclipse EE environment using JAVA 1.8
2. To execute the program under command line, use the command “java Main”
3. Programs are pre-compiled, therefore compilation is not necessary.
4. Once the program executes a series of menu will be loaded on the screen
5. Based on the user selection further navigation and execution will be carried out.
6. It is understood that user guides through ascending order of the menu because each menu operation are closely bonded with next set of actions.
7. Exceptions are handled therefore if user tries to key in illegal characters other than menu number the application will throw an error.
8. At menu number 8 the application further extends to accept bin capacity from user. This input should also be in integer, any character input may result in exception at run time.
9. To exit the application user can press 9 at the menu, this will immediately terminate the application.
10. Test cases are written under test package therefore change in code or resource files may lead to violation of test cases.

Note:

This application’s source code and its complete documentation are made publicly available under github repository link https://github.com/ashokjjk/MusicLibrary after **19th June 2020**.

Source code reference

Problem: Basic I

Definition	Class / Method Name
Define a Java class to store information about artists, including their name and membership to bands if appropriate.	Track.java
Define a Java class to store information about music tracks. Include methods to: o create a track;	Track.java -> addTrack()
set and get the title, artist, date, length, rating, location and size of the track;	TrackService.java
add a guest artist to the track;	TrackService.java
get a list of all individuals on the track, including band members and guest artists;	BandMemberService.java, Track.java -> addMembers()
add to the play count of a track; get the play count of a track.	Track.java -> addTrack()

Problem: Basic II

Definition	Class / Method Name
Define a Java class to represent albums. Provide methods to: o create an album;	Album.java -> addAlbm()
define the list of tracks for the album;	Album.java -> addTracksToAlbum()
get the overall running time of the album;	MenuAction.java -> getMenuAction5()
get the overall file size of the album;	
get the average rating of tracks on the album.	

Problem: Basic III

Definition	Class / Method Name
Define a Java class to represent a music library. Provide methods to: o create a library;	Library.java -> addLib()
add tracks and/or albums to the library;	Library.java -> addTracksToLib()
create a list of tracks from the library with the lowest rating.	MenuAction.java -> getMenuAction7()

Problem: Extensions II

Definition	Class / Method Name
Write a method for a library that works out how to back up all of the music tracks onto a small number of discs. Different types of disc (CD, DVD) have different capacities in terms of total file size, so the capacity of the disc should be a parameter to the method. This is an example of the general problem known as bin-packing problem.	BinPackaging.java -> pack()