CORE JAVA PROJECT

Music Player App

Submitted by

Name: Miss. Savita Jitendra Mali

Under the Guidance of Trainer

Name: Mrs. Indrakka Malli Mam

Aim

To create a simple Music Player App

Languages Use

Java

Software Requirements

Eclipse for Java

Introduction

We create the app like song playlist app. Here this going to be core java and oop's and we are going to combine this concept and create a simple song playlist app. Hence create a album add our favorite music or song to that album. If we create a app of song playlist we must need a class of song.

Music is one of the best way to relieve pressure in stressful modern society life. This project can play current song, previous song, next song and delete songs

INITIAL SETUP:
(IN JAVA):

Maven Project Name: Music Player App.
Package Name : com.music
Class Name : 1. MusicMain
2. Song
3. Album

1. Music Main

In main music class we use the **ArrayList** and **LinkedList** to add song in the list. Take the input from the user and define some boolean then we have list iterator for playlist and if the size of list is 0 print that don't have any song and the else part print current playing song and at the same time print all the available option that means print menu. So while not quit value as not quit yet take user input and use switch case and we have 0, 1, 2, 3, 4, 5, 6 all the different cases.

Suppose you press the following number :

0 = playlist is complete and we just quit.
1 = To play next song.
2 = To play previous song.
3 = To replay the current song.
4 = List of all songs.

5 = Print all available option.

6 = Delete current song.

❖ Class Main Music Coding

```
package com.music;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.LinkedList;

import java.util.ListIterator;

import java.util.Scanner;


public class MusicMain

{

     private static ArrayList<Album> albums = new
ArrayList<>();

 public static void main(String[] args)

 {

     Album album = new Album("Album1","Darshan
Raval");

     album.addSong("Asal mein",3.25);

     album.addSong("Tera Zikr",3.54);

     album.addSong("Ek Tarfa",4.29);

     album.addSong("Piya Re", 3.04);

     album.addSong("Rabba Mehar Kari", 3.04);
```

```
        albums.add(album);


        album = new Album("Album2","Arjit Singh");
        album.addSong("Apna Bana Le ", 3.25);
        album.addSong("Kesariya", 2.53);
        album.addSong("Tera Fitur", 5.08);
        album.addSong("Agar Tum Sath Ho", 5.42);
        album.addSong("Hawayein", 4.52);
        albums.add(album);


        LinkedList<Song>PlayList_1 = new LinkedList<>();
        albums.get(0).addToPlayList("Asal mein",
PlayList_1);
        albums.get(0).addToPlayList("Tera Zikr",
PlayList_1);
        albums.get(1).addToPlayList("Apna Bana Le",
PlayList_1);
        albums.get(1).addToPlayList("Hawayein",
PlayList_1);
        albums.get(1).addToPlayList("Agar Tum Sath
Ho",PlayList_1);
         play(PlayList_1);
    }
    private static void play(LinkedList<Song>PlayList)
    {
        Scanner sc = new Scanner(System.in);
```

```
        boolean quit = false;

        boolean forward = true;

        ListIterator<Song>listIterator =
PlayList.listIterator();

        if(PlayList.size() == 0)

        {

          System.out.println("This PlayList have No
Song");

        }

        else

        {

           System.out.println("Now
Playing"+listIterator.next().toString() );

           PrintMenu();

        }

        while(!quit)

        {

            int action = sc.nextInt();

            sc.nextLine();

            switch (action)

            {

                case 0:

                    System.out.println("PlayList
Complete");

                    quit = true;
```

```
                break;
            case 1:
                if(!forward)
                {
                    if(listIterator.hasNext())
                    {
                        listIterator.next();
                    }
                    forward = true;
                }
                if(listIterator.hasNext())
                {
                    System.out.println("Now
Playing"+listIterator.next().toString());
                }
                else
                {
                    System.out.println("No song
available, reached to the end of the list");
                    forward = false;
                }
                break;
            case 2:
                if(forward)
```

```
                {
                    if(listIterator.hasPrevious())
                    {
                        listIterator.previous();
                    }
                    forward = false;
                }
                 if(listIterator.hasPrevious())
                 {
                     System.out.println("NowPlaying"+
listIterator.previous().toString());
                 }
                 else
                 {
                     System.out.println("we are the
first song");
                     forward = false;
                 }
                 break;
             case 3:
               if(forward)
               {
                  if(listIterator.hasPrevious())
                  {
```

```
                    System.out.println("NowPlaying"+
listIterator.previous().toString());

                    forward = false;

                 }

                else

                {

                   System.out.println("we are at the
start of the list");

                }

              }

              else

              {

                 if(listIterator.hasNext())

                 {

                    System.out.println("Now
Playing"+listIterator.next().toString());

                    forward = true;

                 }

                 else

                 {

                      System.out.println("We have
reached to the end of list");

                 }

              }

             break;
```

```
                case 4:
                    PrintList(PlayList);
                    break;
                case 5:
                    PrintMenu();
                    break;
                case 6:
                    if(PlayList.size()>0)
                    {
                        listIterator.remove();
                        if(listIterator.hasNext())
                        {
                          System.out.println("Now
Playing"+listIterator.next().toString());
                        }
                        Else
                       {
                          if(listIterator.hasPrevious())
                          {
                                  System.out.println("Now
Playing"+listIterator.previous());
                           }
                       }
                }
```

```
            }
        }
}
private static void PrintMenu()
{
    System.out.println("Available option\nPress");
    System.out.println("0- to quit\n"+
    "1 - To play next song\n"+
    "2 - To play previous song\n"+
    "3 - To replay the current song\n"+
    "4 - List of all songs\n"+
    "5 - Print all avilable option\n"+
    "6 - Delete current song");
}
private static void
PrintList(LinkedList<Song>PlayList)
{
    Iterator<Song> iterator = PlayList.iterator();

System.out.println("**********************************
****");
while(iterator.hasNext())
{
    System.out.println(iterator.next());
```

```
}

System.out.println("**********************************
***");

 }

}
```

2. Class Song

- **Class Song** : In the class Song first we need to initialize **Title** of the song and second is **Duration** of the song.
- **Default Constructor :** create the **Default Constructor** to initialize the attributes of the object with their default value.
- **Parameterized Constructor** : create a **Parameterized Constructor** to create user instance of objects with user defined states.
- **ToString Method()** : it will return the whatever property class contain.

❖ Class Song Coding

```
package com.music;

public class Song

 {

      String title;

      double duration;

      public Song()
```

```
        {
            super();
        }


        public Song(String title, double duration)
        {
            super();
            this.title = title;
            this.duration = duration;
        }
        public String getTitle()
        {
            return title;
        }
        public double getDuration()
        {
            return duration;
        }
        @Override
        public String toString()
       {
            return "Song [title=" + title + ",
duration=" +          duration + "]";
```

```
        }
}
```

3. Class Album

Class Album : The property of class album is album **Name**, **Artist** and **ArrayList** of song because that is what in album list.

- **Default Constructor :** Create the **Default Constructor** to initialize the attributes of the object with their default value.
- **Parameterized Constructor** : Create a **Parameterized Constructor** to create user instance of objects with user defined states.
- **FindSong Method():** In this we pass the title of the song if the song is already exist in the list it will return **Title** of song else it will return null.
- **AddSong Method():** We have need some functionality to add songs inside the album. In this we pass the **Title** and **Duration** of the song. Now if you are going to add song to an album we have to check if that song is already exist in our list or not. The song that we are going to add in list is null that is does'nt exist in our list of song then only we have add the song.
- **AddToPlayList Method():** We have to nees functionality to add song to the playlist and this song must exist in our ArrayList already then we can only add our playlist. We can only add

those songs to playlist that already exist in our album because we cannot add those songs to playlist which does’nt even exist in our album. First we need to check for the track number so pass the track number and the second thing is going to be linked list of playlist that means in what playlist we are going to add the song. The linked list is basically data structure which will have a reference of next element and previous element. While creating the song playlist app we need to keep track what’s the next song is & previous song is.

- We have two same methods but different parameters inside it. User can either add songs with track number & playlist and either title & playlist.

❖ Class Album Coding

```
package com.music;

import java.util.ArrayList;
import java.util.LinkedList;

public class Album
{

    private String name;
    private String artist;
    private ArrayList<Song>songs;

    public Album()
```

```
    {
        super();
        // TODO Auto-generated constructor stub
    }


     public Album(String name, String artist)
     {
        super();
        this.name = name;
        this.artist = artist;
        this.songs = new ArrayList<Song>();
     }

     public Song findSong(String title)
     {
        for(Song checkedSong : songs)
        {
            if(checkedSong.getTitle().equals(title))
return checkedSong;
        }
        return null;
      }

     public boolean addSong(String title,double
duration)//Add song in the album
     {
        if(findSong(title)==null)
```

```
        {
            songs.add(new Song(title,duration));
            //System.out.println(title+"Successfully
added to the list");
            return true;
        }
        else
        {
            //System.out.println("Song with
name"+title+"already exist in the list");
            return false;
        }
    }
    public boolean addToPlayList(int
trackNumber,LinkedList<Song>PlayList)
    {//Add song which is present in the album to the
playlist
        int index = trackNumber-1;
        if(index > 0 && index <= this.songs.size())
        {
            PlayList.add(this.songs.get(index));
            return true;
        }
        //System.out.println("This album does not
have song with trackNumber"+trackNumber);
        return false;
    }
    public boolean addToPlayList(String
title,LinkedList<Song>PlayList)
    {
```

```
        for(Song checkedSong : this.songs)
        {
            if(checkedSong.getTitle().equals(title))
            {
                PlayList.add(checkedSong);
                return true;
            }
        }
        //System.out.println(title+"There is no such
song in album");
        return false;
    }

}
```