REPOSITORY: https://github.com/kartezix/CW1AlexaKralewska

<?xml version="1.0" encoding="UTF-8"?>  
<module type="JAVA\_MODULE" version="4">  
 <component name="NewModuleRootManager" inherit-compiler-output="true">  
 <exclude-output />  
 <content url="file://$MODULE\_DIR$">  
 <sourceFolder url="file://$MODULE\_DIR$/src" isTestSource="false" />  
 </content>  
 <orderEntry type="inheritedJdk" />  
 <orderEntry type="sourceFolder" forTests="false" />  
 </component>  
</module>

///introduction to the "app".  
public class FFSMusicConsole {  
 public static void main(String[] args)  
 {  
 System.*out*.println("======================");  
 System.*out*.println("Welcome to FFS Music Player by Alexa Kralewska!");  
 System.*out*.println("======================");  
  
 MusicApp app = new MusicApp();  
 app.doMainMenu();  
 }  
}

import helpers.InputReader;  
///Alexa Kralewska.  
import java.util.ArrayList;  
  
public class MusicApp {  
 private final ArrayList<Song> songs;  
  
 public MusicApp() {  
 songs = new ArrayList<>();  
 introduction();  
 addSomeSongs();  
 doMainMenu();  
  
 }  
//allows user input name to be greeted by.  
 public void introduction() {  
 String intro = InputReader.*getString*("Please input your name:");  
 System.*out*.println("\n Hello " + intro);  
  
  
 }  
//song list to be used as data to complete given actions, such as "add", "remove" or "delete".  
 private void addSomeSongs() {  
 Song song = new Song("AntiSystem's", "jon-YAKITORY, Ado", 4220317);  
 songs.add(song);  
  
 song = new Song("Aching Horns", "OLDCODEX", 1489307);  
 songs.add(song);  
  
 song = new Song("Ghost Rule", "DECO\*27", 39333615);  
 songs.add(song);  
  
 song = new Song("Young Girl A", "siinamota", 8791842);  
 songs.add(song);  
  
 song = new Song("Somewhere I Belong", "Linkin Park", 254143516);  
 songs.add(song);  
  
 song = new Song("Dog Days", "MAN WITH A MISSION", 7325389);  
 songs.add(song);  
  
 song = new Song("Jesus", "GACKT", 2331740);  
 songs.add(song);  
  
 song = new Song("BLACK MEMORY", "THE ORAL CIGARETTES", 19580349);  
 songs.add(song);  
  
 song = new Song("Renagade", "STEREO DIVE FOUNDATION", 396025);  
 songs.add(song);  
  
 song = new Song("BAD SAD AND MAD", "bibi", 2709784);  
 songs.add(song);  
 }  
///exit console + methods for options.  
 public void doMainMenu() {  
 boolean wantToQuit = false;  
  
 while (!wantToQuit) {  
 displayMenu();  
 int choice = InputReader.*getInt*("Please enter your choice : ");  
  
 switch (choice) {  
 case 1 -> addSong();  
 case 2 -> deleteSong();  
 case 3 -> printSongs(0);  
 case 4 -> topHits();  
 case 5 -> wantToQuit = true;  
 }  
 }  
 }  
  
//display of options available.  
 private void displayMenu() {  
 System.*out*.println("\n 1. Add Song");  
 System.*out*.println(" 2. Delete Song");  
 System.*out*.println(" 3. Print Songs");  
 System.*out*.println(" 4. Print Songs via Highest Playcount");  
 System.*out*.println(" 5. Quit\n");  
  
 }  
//method for adding songs.  
 private void addSong() {  
 System.*out*.println(" \nAdding a Song\n");  
 String title = InputReader.*getString*("Please enter the song title >");  
 String name = InputReader.*getString*("Please enter the artist name >");  
 int count = InputReader.*getInt*("Please enter the play count > ");  
  
 Song song = new Song(title, name, count);  
 songs.add(song);  
 }  
//method for deletion of songs + error system for unassigned values.  
 private void deleteSong() {  
 Song song = findSong();  
 if (song != null) {  
 songs.remove(song);  
 } else  
 System.*out*.println("Could not remove non-listed song.");  
 }  
  
//instruction: remove by title, but can be given a different element to be removed via, e.g.: ID.  
 private Song findSong() {  
 String removeSong = InputReader.*getString*("Insert title to remove song:");  
 for (Song song : songs) {  
 if (song.getTitle().contains(removeSong))  
 return song;  
 }  
 return null;  
 }  
//id++ serves to number songs. the code prints it along with print list + highest count play list.  
 private void printSongs(int count) {  
 System.*out*.println("\nPrinting All Songs\n");  
 int id = 0;  
 for (Song song : songs) {  
 id++;  
 System.*out*.print("Song " + id + ": ");  
  
 if (song.gettopHits() > count)  
 song.print();  
 }  
 }  
//method by which computer identifies user criteria for play count.  
  
 private void topHits() {  
  
 int topRequest = InputReader.*getInt*("Insert number to find playcount above given criteria.");  
 printSongs(topRequest);  
 }  
  
}

//classes and strings for methods to run to, separated for simple access.  
  
public class Song  
{  
 private int id;  
  
 private String title;  
  
 private String artist;  
  
 private int playCount;  
  
 private String removeSong;  
  
 public Song(String title, String name, int count)  
 {  
 this.title = title;  
 this.artist = name;  
 this.playCount = count;  
 this.removeSong = title;  
 this.id = id;  
  
 }  
  
 public void print()  
 {  
 System.*out*.println(title + " by " + artist + " plays: " + playCount);  
 }  
 public String getTitle() {return title;}  
  
 public int gettopHits() {return playCount;}  
}