**Main.Java:**

package javaapplication1;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.regex.\*;

/\*\*

\*

\* @author Kaleem Shah @version 1.18

\*/

public class Main {

/\*this method creates 10 song objects and adds them to an array

which is then looped through to add them to the playlist arraylist\*/

public void createDefaultSongs(Playlist myPlaylist){

Song[] songs = new Song[10];

songs[0]= new Song("Electric Dreams at Midnight","Luna Aurora",72);

songs[1]= new Song("Whispers in the Rain","Silver Harmony",45);

songs[2]= new Song("Neon Sunshine","Dreamcatcher",89);

songs[3]= new Song("Crystal Embrace","Midnight Mirage",12);

songs[4]= new Song("Starry Serenade","Celestial Serenity",61);

songs[5]= new Song("Velvet Skies","Phoenix Rising",34);

songs[6]= new Song("Echoes of Tomorrow","Aurora Borealis",98);

songs[7]= new Song("Lost in the Labyrinth of Love","Echo Enigma",27);

songs[8]= new Song("Invisible Wings","Velvet Shadows",55);

songs[9]= new Song("Enchanted Moonlight","Solar Flare Orchestra",76);

for (Song song:songs){

myPlaylist.addSong(song);

}

}

public static void main(String[] args) {

Playlist myPlaylist=new Playlist();//creates a new object of the playlist class called myPlaylist

Main oop = new Main(); // Create an object of the oop class

oop.createDefaultSongs(myPlaylist); // Calls the method createDefaultSongs

String REGEX = "^.\*,[A-Za-z ]+,\\d+$";//regular expression for syntax of adding a song

Scanner myObj = new Scanner(System.in);//creates a scanner object to read inputs

int choice = 11; //sets a variable called choice to 11 which is just used as a place holder until the user inputs their choice

//creates a variable called menu which displays a menu on screen when printed

String menu = "\n 1.)Add a new song \n 2.)Remove a song \n 3.)Print a list of all the songs stored \n 4.)print a list of songs over a certain number of plays \n 5.)update a songs number of plays \n 6.)print all songs by a given artist \n 7.)print a random song \n 0.)Exit";

/\*while loop to continously loop through the menu until

a 0 is inputted as the choice. The user is asked to enter

a choice and if there is an error it will catch it

the myObj.nextLine() consumes the new line character\*/

while (choice!=0){

System.out.println(menu + "\n" + "enter your choice (1 2 3, etc): ");

try{

choice = myObj.nextInt();

}

catch(Exception e){

System.out.println("error, invalid entry");

}

myObj.nextLine();

/\*if the user enters 1 as a choice, then the user

will be prompted to enter the song details, this input

is then error checked using regular expressions

if correct, it will split the input and add the song

if not it will output an error message\*/

if (choice == 1){

System.out.println("enter the song in this format (song name,artist,plays)");

String songDetails = myObj.nextLine();

Pattern pattern = Pattern.compile(REGEX);

Matcher matcher = pattern.matcher(songDetails);

if (!matcher.matches()) {

System.out.println("invalid entry");

}

else{

String[] songDetailsList = songDetails.split(",");

try{

String songName = songDetailsList[0].trim();

String songArtist = songDetailsList[1].trim();

int songPlays = Integer.parseInt(songDetailsList[2]);

Song song1 = new Song(songName,songArtist,songPlays);

myPlaylist.addSong(song1);

System.out.println("successfully added");

}

catch(Exception e){

System.out.println("Invalid Entry");

}

}

}

/\*if the user enters 2 as the choice, allows user

to enter the title of song to be removed, this input

is then used to search the playlist using the removeSong method

if the song is present, it will remove it, if not it will display

a message saying so.\*/

else if (choice == 2){

System.out.println("Enter the title of the song you want to remove: ");

String songTORemove = myObj.nextLine();

boolean Found = myPlaylist.removeSong(songTORemove);

if (!Found){

System.out.println("song not found");

}

else{

System.out.println("song has been removed successfully");

}

}

/\*if the user enters 3 as the choice, the printAll method is

called which loops through the playlist and prints all the songs present\*/

else if (choice == 3){

myPlaylist.printAll();

}

/\*if the user enters 4 as the choice, the user is asked to enter

the minimum number of plays, if the input isnt an integer, the

error is caught and an error message is displayed, if the input

is valid the printAllAbovePlays method is called which searches

through the playlist and outputs all songs above the inputted

number of plays\*/

else if (choice == 4){

System.out.println("enter the minimum number of plays you want to see: ");

try{

int minNumberOfPlays = myObj.nextInt();

boolean songsFound = myPlaylist.printAllAbovePlays(minNumberOfPlays);

if (!songsFound){

System.out.println("no songs with plays above the value");

}

}

catch(Exception e){

System.out.println("Invalid Entry");

}

}

/\*if the user enters 5 as the choice, the user is asked to enter the

song title they would like to update the plays for and the new number

of plays, if not an integer, an error message is displayed. the program

then searches through the playlist and if the song is present, success

is true and the plays has been updated, if not an error message is displayed\*/

else if (choice == 5){

System.out.println("Enter the title of the song you want to update: ");

String songTitleToFind = myObj.nextLine();

try{

System.out.println("enter the new number of plays");

int updatedPlays = myObj.nextInt();

boolean success = myPlaylist.setSongPlays(songTitleToFind,updatedPlays);

if (!success){

System.out.println("song not found");

}

else{

System.out.println("song successfully updated");

}

}

catch(Exception e){

System.out.println("not a valid number");

}

}

/\*if the user enters 6 as the choice, the user is asked to enter

the name of the artist, it is then used to search through the playlist

for all songs that are present by said artist, each song is then outputted\*/

else if (choice == 6){

System.out.println("enter the name of the artist whos songs you would like to see: ");

String artistToFind = myObj.nextLine();

myPlaylist.findByArtist(artistToFind);

}

/\*if the user enters 7 as the choice, a random integer between 0 and the number

of songs in the playlist is chosen, the index of the playlist at that number is

then outputted (with the getRandomSong() method\*/

else if (choice == 7){

myPlaylist.getRandomSong();

}

//if the user enters 0 as the choice, program outputs a goodbye message and ends

else if (choice == 0){

System.out.println("GoodBye");

}

//if the user enters anything other than 0,1,2,3,4,5,6,7, the loop will just continue

else{

System.out.println("Not a valid choice");

}

}

}

}

**Song.java:**

package javaapplication1;

/\*\*

\*

\* @author Kaleem Shah @version 1.18 \*/

//creates a Song class with three attributes of title, artist and plays

public class Song {

private String title;

private String artist;

private int plays;

//creates a constructor which is empty so a blank object can be made

public Song(){

}

//creates a constructor which takes 3 inputs to set the title, artist and plays

public Song(String Title,String Artist,int Plays){

this.title=Title;

this.artist=Artist;

this.plays=Plays;

}

//getter method to return the title when called

public String getTitle(){

return this.title;

}

//getter method to return the artist when called

public String getArtist(){

return this.artist;

}

//getter method to return the plays when called

public int getPlays(){

return this.plays;

}

//setter method to set the title to the inputed string

public void setTitle(String newTitle){

this.title = newTitle;

}

//setter method to set the artist to the inputed string

public void setArtist(String newArtist){

this.artist = newArtist;

}

//setter method to set the number of plays to the inputted integer

public void setPlays(int newPlays){

this.plays = newPlays;

}

//prints the current objects title, artist and plays

public void print(){

System.out.println(this.title + "," + this.artist + "," + this.plays);

}

}

**Playlist.java:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package javaapplication1;

import java.util.ArrayList;

import java.util.Random;

/\*\*

\*

\* @author Kaleem Shah @version 1.18

\*/

public class Playlist {//creates a Playlist class with an arraylist called playlist as an attribute

private ArrayList<Song> playlist;

public Playlist(){//constructor, creates a new arraylist

this.playlist=new ArrayList<>();

}

public void addSong(Song song){//takes the song object passed through and adds it to the playlist array list

this.playlist.add(song);

}

public int getLength(){//gets the number of items/size of the arraylist

return this.playlist.size();

}

public void clearPlaylist(){//clears the playlist

this.playlist.clear();

}

public void printAll(){//searches through each Song object in the playlist and prints them

for (int index=0;index < playlist.size();index++){

Song current=playlist.get(index);

current.print();

}

}

/\*takes the song title entered by the user and if the song is in the playlist, it

removes it and returns true, if not in the playlist it returns false\*/

public boolean removeSong(String songTitle){

boolean songFound = false;

for (int index=0;index < playlist.size();index++){

Song current=playlist.get(index);

if (current.getTitle().equalsIgnoreCase(songTitle)){

playlist.remove(index);

songFound = true;

}

}

return songFound;

}

/\*searches for the song in the playlist, if found,

it updates the plays and returns true, if not found, returns false\*/

public boolean setSongPlays(String songTitle,int songPlays ){

boolean songFound = false;

for (int index=0;index < playlist.size();index++){

Song current=playlist.get(index);

if (current.getTitle().equalsIgnoreCase(songTitle)){

current.setPlays(songPlays);

songFound = true;

}

}

return songFound;

}

/\*searches for all songs in the playlist that have the plays value higher than

the value the user entered, if there are songs that match this, then it calls

the method to print it and returns true, if none match, returns false.\*/

public boolean printAllAbovePlays(int songPlays){

boolean songsFound = false;

for (int index=0;index < playlist.size();index++){

Song current=playlist.get(index);

if (current.getPlays()>songPlays){

current.print();

songsFound = true;

}

}

return songsFound;

}

/\*searches the playlist for all songs by an entered artist and prints them, if none found, it outputs a message.\*/

public void findByArtist(String songArtist){

boolean artistFound = false;

for (int index=0;index < playlist.size();index++){

Song current=playlist.get(index);

if (current.getArtist().equalsIgnoreCase(songArtist)){

current.print();

artistFound = true;

}

}

if (!artistFound){

System.out.println("no songs by this artist");

}

}

public void getRandomSong(){//sets a random integer from 0 to the size of the playlist, then prints the song which has that index

Random rand = new Random();

int n = rand.nextInt(this.playlist.size());

Song randomSong = this.playlist.get(n);

randomSong.print();

}

}