CMP-4008Y Coursework 1 - Album Database

ssq16shu (100166648)

Wed, 13 Dec 2017 12:47

Contents

Duration.java

```java
/*
* File              :   Duration.java
* Author            :   100166648 T.Tufnail-Clarke
* Description       :   Stores duration of a Track, Album, Etc
* Final Edit        :   v1.3 11/12/17
* Version History   :   v1.2 10/12/17
                        v1.1 17/11/17
                        v1.0 16/11/17
                        v0.1 11/11/17
 */
package cw_1;

public class Duration {

    //instance variables
    private int hours;
    private int minutes;
    private int seconds;

    //constructors
    public Duration() {
        hours = 0;
        minutes = 0;
        seconds = 0;
    }

    //if passed as int
    public Duration(int hrs, int min, int sec) {
        hours = hrs;
        minutes = min;
        seconds = sec;
    }

    //if passed as string
    public Duration(String enterTime) {
        String[] parts = enterTime.split(":");
        String stHr = parts[0];
        String stMn = parts[1];
        String stSc = parts[2];
        hours = Integer.parseInt(stHr);
        minutes = Integer.parseInt(stMn);
        seconds = Integer.parseInt(stSc);
    }

    //override allows this method to replace toString from default library
    @Override
    public String toString() {
        return hours + ":" + minutes + ":" + seconds;
    }

    //Accessor methods
    public int getHour() {
        return hours;
    }

    public int getMin() {
        return minutes;
    }

    public int getSec() {
        return seconds;
    }

    //setMethods
    public void setHour(int tmp) {
        hours = tmp;
    }

    public void setMinutes(int tmp) {
        minutes = tmp;
    }

    public void setSeconds(int tmp) {
        seconds = tmp;
    }

    //convert total run time of each track to seconds.
    //allows for easy comparison of track duration
    public int toSeconds() {
        int totalSeconds = (hours * 3600) + (minutes * 60) + (seconds);
        return totalSeconds;
    }

    //Compares two durations and returns length as an interger
    public int compareTo(Duration t) {
        int x = toSeconds();
        int y = t.toSeconds();
        int biggest;
        if (y > x) {
            biggest = y;
        } else {
            biggest = x;
        }
        return biggest;
    }

    //test harness
    public static void main(String[] args) {

    }

}
```

Track.java

```
/*
* File                :   Track.java
* Author              :   100166648 T.Tufnail-Clarke
* Description         :   Stores details of an album track as member variables
* Final Edit          :   v1.2 11/12/17
* Version History     :   v1.1 10/12/17
                          v1.0 24/11/17
                          v0.3 17/11/17
                          v0.2 16/11/17
                          v0.1 11/11/17
 */

package cw_1;

public class Track implements Comparable<Track> {

    //instance variables
    public String title;
    public Duration duration;

    //constructors
    public Track() {
        title = "";
        duration = new Duration();
    }

    public Track(String title, Duration duration) {
        this.title = title;
        this.duration = duration;
    }

    //Accessor methods
    public String getTitle() {
        return title;
    }

    public Duration getDuration() {
        return duration;
    }

    //set methods
    public void setTitle(String t) {
        title = t;
    }

    public void setDur(Duration s) {
        duration = s;
    }

    //Convert to string
    @Override
    public String toString() {
        return title + " " + duration;
    }

    //Compares two Track durations
    public int compareTo(Track t) {
        int comparison = this.duration.compareTo(t.getDuration());
        return comparison;
    }
    //Test harness
    public static void main(String[] args) {
    }

}
```

Album.java

```
/*
* File                :   Album.java
* Author              :   100166648 T.Tufnail-Clarke
* Description         :   Stores details of an album with instance variables
                          storing name of artist and title of album as string
                          objects and track objects storing contents of album
* Final Edit          :   v1.5 11/12/17
* Version History     :   v1.4 10/12/17
                          v1.3 04/12/17
                          v1.2 01/12/17
                          v1.1 24/11/17
                          v1.0 16/11/17
                          v0.2 11/11/17
 */
package cw_1;

import java.util.ArrayList;

public class Album implements Comparable<Album> {

    //instance variables
    private String artist;
    private String album;
    private ArrayList<Track> trackList;

    //constructors
    //default
    public Album() {
        artist = null;
        album = null;
        trackList = new ArrayList();
    }

    //passed album details
    public Album(String a, String b, ArrayList<Track> c) {
        artist = a;
        album = b;
        trackList = c;
    }

    //passed album and artist, create empty trackList
    public Album(String a, String b) {
        artist = a;
        album = b;
        trackList = new ArrayList();
    }

    //Accessor methods
    public String getArt() {
        return artist;
    }

    public String getAlb() {
        return album;
    }

    public ArrayList getTrack() {
        return trackList;
    }

    //set metods
```

```java
    public void setArt(String s) {
        artist = s;
    }

    public void setAlb(String a) {
        album = a;
    }

    public void setArray(ArrayList r) {
        trackList = r;
    }

    //Convert to string
    @Override
    public String toString() {
        return artist + " -- " + album + "\n" + trackList + "\n";
    }

    //alphabetically compares two albums
    @Override
    public int compareTo(Album t) {
        //Compares two albums
        int comparison = this.artist.compareTo(t.getArt());

        if (comparison == 0) {
            comparison = this.album.compareTo(t.getAlb());
        }
        return comparison;
    }

    //iterates through tracklist and returns the longest track
    public Track findLong() {
        Track longest = new Track();
        for (int i = 0; i < trackList.size(); i++) {
            if (trackList.get(i).getDuration().toSeconds()
                    > longest.getDuration().toSeconds()) {
                longest = trackList.get(i);
            }
        }
        return longest;
    }

    //tracklist counter
    public int counter() {
        return trackList.size();
    }

    //test harness
    public static void main(String[] args) {
        Duration d = new Duration("13:33:23");
        Duration d1 = new Duration("1:21:30");
        Track t1 = new Track("Track1", d1);
        Track t2 = new Track("Track2", d);

        ArrayList<Track> tst = new ArrayList();
        tst.add(t1);
        tst.add(t2);
        tst.add(t1);
        Album ab = new Album("A0", "AB", tst);
        System.out.println(ab.toString());
    }

}
```

AlbumCollection.java

```java
/*
* File                :   AlbumCollection.java
* Author              :   100166648 T.Tufnail-Clarke
* Description         :   Reads and stores details of a collection of albums
                          including member variable containing album objects
* Last Edit           :   04/12/17
* Version History     :   v2.1 04/12/17
                          v2.0 01/12/17
                          v1.0 24/11/17
                          v0.1 11/11/17
 */
package cw_1;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;

public class AlbumCollection {

    //instance variables
    private ArrayList<Album> albCol;

    //constructors
    //default
    public AlbumCollection() {
        albCol = new ArrayList();
    }

    //passed an array list
    public AlbumCollection(ArrayList<Album> newAdd) {
        albCol = newAdd;
    }

    //Read file
    public static AlbumCollection readDoc() throws FileNotFoundException,
            IOException {

        BufferedReader in = new BufferedReader(new FileReader("albums.txt"));

        //holding variables (compiler demanded variables to be declaired)
        String line, time = null, title = null, art = null, alb = null;
        ArrayList<Track> collect = new ArrayList();
        ArrayList<Album> albPass = new ArrayList();

        //reading line and splitting into strings depending on first char
        while ((line = in.readLine()) != null) {

            if (Character.isLetter(line.charAt(0))) {

                if (!collect.isEmpty()) {
                    Album current = new Album(art, alb, collect);
                    albPass.add(current);
                }
                //reset collect for new album pass
                collect = new ArrayList();

                String[] splitArr = line.split(" : ");
                art = splitArr[0];
                alb = splitArr[1];
            } else if (Character.isDigit(line.charAt(0))) {
                String[] splitArr = line.split(" - ");
                time = splitArr[0];
                title = splitArr[1];
                Duration durr = new Duration(time);
                Track mus = new Track(title, durr);
                collect.add(mus);
            }
        }
        Album current = new Album(art, alb, collect);
        albPass.add(current);
        AlbumCollection aC = new AlbumCollection(albPass);
        in.close();
        return aC;
    }

    //sorts the album collection alphabetically
    public void albumSort() {
        Collections.sort(albCol);
    }

    //find longest track in album(s)
    public Track trackCheck() {
        Track find = new Track();
        Track compare = new Track();
        for (int i = 0; i < albCol.size(); i++) {
            compare = albCol.get(i).findLong();
            if (compare.getDuration().toSeconds()
                    > find.getDuration().toSeconds()) {
                find = compare;
            }
        }
        return find;
    }

    //find album with most tracks
    public Album getMostTracks() {
        Album album = null;
        int biggest = 0;
        int biggestI = -1;
        for (int i = 0; i < albCol.size(); i++) {
            if (albCol.get(i).counter() > biggest) {
                biggest = albCol.get(i).counter();
                biggestI = i;
                album = albCol.get(i);
            }
        }
        return album;
    }

    //find total run time
    public int albumTime() {
        //this was originally written as a chain of methods but has been broken
        //into multiple varables to make following it for debug easier
        String artists = "Pink Floyd";
        int totalSeconds = 0;
        int albumSeconds = 0;
        ArrayList<Track> tracks = new ArrayList();
        //loop through albums
        for (int i = 0; i < albCol.size(); i++) {
            if (albCol.get(i).getArt().equals(artists)) {
                //take track list
```

```java
                tracks = albCol.get(i).getTrack();
                //loop through track list and add each time to total duration
                for (int j = 0; j < tracks.size(); j++) {
                    albumSeconds += tracks.get(j).getDuration().toSeconds();
                }
                totalSeconds += albumSeconds;
                albumSeconds = 0;
            }
        }
        return totalSeconds;
    }

    //Convert to string
    @Override
    public String toString() {
        return albCol.toString();
    }

    //Test harness
    public static void main(String[] args) {

    }

}
```

AlbumDatabase.java

```java
/*
* File                :   AlbumDatabase.java
* Author              :   100166648 T.Tufnail-Clarke
* Description         :   Program stores and manipulates details of
                          music album colection
* Final Edit          :   v2.3 13/12/17
* Version History     :   v2.2 11/12/17
                          v2.1 04/12/17
                          v2.0 01/12/17
                          v1.0 24/11/17
                          v0.4 24/11/17
                          v0.3 17/11/17
                          v0.2 16/11/17
                          v0.1 11/11/17
 */
package cw_1;

import java.io.FileNotFoundException;
import java.io.IOException;
//Scanner disabled for PASS submission as only used for user input switch
//import java.util.Scanner;

public class AlbumDatabase {

    public static void main(String[] args) throws IOException,
            FileNotFoundException {

        AlbumCollection aC = AlbumCollection.readDoc();

        //User selection -allows for segmented use/testing of tasks
        //Disabled and replaced with all functions in line for PASS submission

        /*int userChoice;
        System.out.println("File successfully read to Album Collection\n"
                + "please select a numeric option:\n\n"
                + "1: Display entire collection as read in\n"
                + "2: Display entire collection in aphabetic order\n"
                + "3: Display total run time of Pink Floyd albums\n"
                + "4: Display album with largest number of tracks\n"
                + "5: Display full details of longest track in collection\n");

        Scanner keyboard = new Scanner(System.in);
        userChoice = keyboard.nextInt();
        switch (userChoice) {
            case 1: {
                System.out.println("\nList as read in from file:\n");
                System.out.println(aC);
                break;
            }
            case 2: {
                System.out.println("Alphabetically sorted list:\n");
                aC.albumSort();
                System.out.println(aC);
                break;
            }
            case 3: {
                System.out.println("Total play time of Pink Floyd albums:\n");
                System.out.println(aC.albumTime()
                + " seconds");
                break;
            }
            case 4: {
                Album album = aC.getMostTracks();
                System.out.println("Album with largest collection of tracks:\n"
                        + album + "\nNumber of tracks: " + album.counter());
                break;
            }
            case 5: {
                System.out.println("Longest track in collection:\n"
                        + aC.trackCheck());
                break;
            }
        }*/

        //Linear list of functions for PASS submission
        System.out.println("\nList as read in from file:\n");
        System.out.println(aC + "\n\n");
        System.out.println("Alphabetically sorted list:\n");
        aC.albumSort();
        System.out.println(aC + "\n\n");
        System.out.println("Total play time of Pink Floyd albums:");
        System.out.println(aC.albumTime()
                + " seconds\n\n");
        Album album = aC.getMostTracks();
        System.out.println("Album with largest collection of tracks:"
                + "\n" + album + "Number of tracks: " + album.counter()
                + "\n\n");
        System.out.println("Longest track in collection:\n"
                + aC.trackCheck());
    }

}
```

Application

Compiler Invocation

```
javac.exe -Xlint:unchecked -Xlint:deprecation -d classes Duration.java Track.java
    Album.java AlbumCollection.java AlbumDatabase.java
```

Compiler Messages

```
Album.java:31: warning: [unchecked] unchecked conversion
        trackList = new ArrayList();
                    ^
  required: ArrayList<Track>
  found:    ArrayList
Album.java:45: warning: [unchecked] unchecked conversion
        trackList = new ArrayList();
                    ^
  required: ArrayList<Track>
  found:    ArrayList
Album.java:71: warning: [unchecked] unchecked conversion
        trackList = r;
                    ^
  required: ArrayList<Track>
  found:    ArrayList
Album.java:116: warning: [unchecked] unchecked conversion
        ArrayList<Track> tst = new ArrayList();
                               ^
  required: ArrayList<Track>
  found:    ArrayList
AlbumCollection.java:29: warning: [unchecked] unchecked conversion
        albCol = new ArrayList();
                 ^
  required: ArrayList<Album>
  found:    ArrayList
AlbumCollection.java:45: warning: [unchecked] unchecked conversion
        ArrayList<Track> collect = new ArrayList();
                                   ^
  required: ArrayList<Track>
  found:    ArrayList
AlbumCollection.java:46: warning: [unchecked] unchecked conversion
        ArrayList<Album> albPass = new ArrayList();
                                   ^
  required: ArrayList<Album>
  found:    ArrayList
AlbumCollection.java:58: warning: [unchecked] unchecked conversion
                collect = new ArrayList();
                          ^
  required: ArrayList<Track>
  found:    ArrayList
AlbumCollection.java:120: warning: [unchecked] unchecked conversion
        ArrayList<Track> tracks = new ArrayList();
                                  ^
  required: ArrayList<Track>
  found:    ArrayList
AlbumCollection.java:125: warning: [unchecked] unchecked conversion
                tracks = albCol.get(i).getTrack();
                                               ^
  required: ArrayList<Track>
  found:    ArrayList
10 warnings
```

Application Invocation

```
javaw.exe cw_1.AlbumDatabase
```

Messages to STDOUT

```
List as read in from file:

[The Jimi Hendrix Experience -- Are you Experienced?
[Foxy Lady 0:3:22, Manic Depression 0:3:46, Red House 0:3:53, Can You See Me 0:2:35, Love or Confusion 0:3
, Pink Floyd -- Dark Side of the Moon
[Speak to Me 0:1:30, Breathe 0:2:43, On the Run 0:3:36, Time 0:7:1, The Great Gig in the Sky 0:4:36, Money
, Jordi Savall & Hesperion XX -- Folias and Canarios
[Folias Pavana Con Su Glosa (Antonio De Cabezon) 0:2:6, Fantasia (Alonso Mudarra) 0:2:49, Tiento De Falsas
, The Beatles -- Rubber Soul
[Drive My Car 0:2:25, Norwegian Wood (This Bird Has Flown) 0:2:1, You Won't See Me 0:3:18, Nowhere Man 0:2
, Blondie -- Parallel Lines
[Hanging on the Telephone 0:2:17, One Way or Another 0:3:31, Picture This 0:2:53, Fade Away and Radiate 0:
, The Dave Brubeck Quartet -- Take Five
[Blue Rondo a la Turk 0:6:44, Strange Meadow Lark 0:7:22, Take Five 0:5:24, Three to Get Ready 0:5:24, Kat
, Goldfrapp -- Supernature
[Ooh La La 0:3:24, Lovely 2 C U 0:3:25, Ride a White Horse 0:4:41, You Never Know 0:3:27, Let It Take You
, Kraftwerk -- The Man Machine
[The Robots 0:6:11, Spacelab 0:5:51, Metropolis 0:5:59, The Model 0:3:38, Neon Lights 0:9:3, The Man-Machi
, Led Zeppelin -- Led Zeppelin IV
[Black Dog 0:4:54, Rock and Roll 0:3:40, The Battle of Evermore 0:5:51, Stairway to Heaven 0:8:2, Misty Mo
, Marillion -- Script for a Jester's Tear
[Script for a Jester's Tear 0:8:44, He Knows You Know 0:5:23, The Web 0:8:52, Garden Party 0:7:19, Chelsea
, Miles Davis -- Kind of Blue
[So What 0:9:22, Freddie Freeloader 0:9:46, Blue in Green 0:5:37, All Blues 0:11:33, Flamenco Sketches 0:9
, Neil Pye -- Neil's Heavy Concept Album
[Hello Vegetables 0:0:26, Hole In My Shoe 0:3:40, Heavy Potato Encounter 0:0:42, My White Bicycle 0:3:31,
, Pulp -- Different Class
[Mis-Shapes 0:3:46, Pencil Skirt 0:3:11, Common People 0:5:50, I Spy 0:5:55, Disco 2000 0:4:33, Live Bed S
, Nimoy & Shatner -- Spaced Out
[King Henry The Fifth 0:3:0, Elegy For The Brave 0:3:18, Highly Illogical 0:2:20, If I Had A Hammer [The H
, Pink Floyd -- Wish You Were Here
[Shine On You Crazy Diamond (Part One) 0:13:33, Welcome to The Machine 0:7:26, Have A Cigar 0:5:7, Wish Yo
, Pink Floyd -- Animals
[Pigs on the Wing (Part I) 0:1:25, Dogs 0:17:3, Pigs (Three Different Ones) 0:11:25, Sheep 0:10:25, Pigs o
, Pink Floyd -- Meddle
[One of These Days 0:5:57, A Pillow of Winds 0:5:7, Fearless 0:6:5, San Tropez 0:3:40, Seamus 0:2:13, Echo
, Kraftwerk -- Trans Europe Express
[Europe Endless ("Europa Endlos") 0:9:42, The Hall of Mirrors ("Spiegelsaal") 0:7:57, Showroom Dummies ("S
, Pink Floyd -- Momentary Lapse of Reason
[Signs of Life 0:4:24, Learning to Fly 0:4:53, The Dogs of War 0:6:5, One Slip 0:5:10, On The Turning Away
]


Alphabetically sorted list:

[Blondie -- Parallel Lines
[Hanging on the Telephone 0:2:17, One Way or Another 0:3:31, Picture This 0:2:53, Fade Away and Radiate 0:
, Goldfrapp -- Supernature
[Ooh La La 0:3:24, Lovely 2 C U 0:3:25, Ride a White Horse 0:4:41, You Never Know 0:3:27, Let It Take You
, Jordi Savall & Hesperion XX -- Folias and Canarios
[Folias Pavana Con Su Glosa (Antonio De Cabezon) 0:2:6, Fantasia (Alonso Mudarra) 0:2:49, Tiento De Falsas
, Kraftwerk -- The Man Machine
[The Robots 0:6:11, Spacelab 0:5:51, Metropolis 0:5:59, The Model 0:3:38, Neon Lights 0:9:3, The Man-Machi
, Kraftwerk -- Trans Europe Express
[Europe Endless ("Europa Endlos") 0:9:42, The Hall of Mirrors ("Spiegelsaal") 0:7:57, Showroom Dummies ("S
, Led Zeppelin -- Led Zeppelin IV
[Black Dog 0:4:54, Rock and Roll 0:3:40, The Battle of Evermore 0:5:51, Stairway to Heaven 0:8:2, Misty Mo
, Marillion -- Script for a Jester's Tear
[Script for a Jester's Tear 0:8:44, He Knows You Know 0:5:23, The Web 0:8:52, Garden Party 0:7:19, Chelsea
, Miles Davis -- Kind of Blue
```

```
[So What 0:9:22, Freddie Freeloader 0:9:46, Blue in Green 0:5:37, All Blues 0:11:33, Flamenco Sketches 0:9
, Neil Pye -- Neil's Heavy Concept Album
[Hello Vegetables 0:0:26, Hole In My Shoe 0:3:40, Heavy Potato Encounter 0:0:42, My White Bicycle 0:3:31, 
, Nimoy & Shatner -- Spaced Out
[King Henry The Fifth 0:3:0, Elegy For The Brave 0:3:18, Highly Illogical 0:2:20, If I Had A Hammer [The H
, Pink Floyd -- Animals
[Pigs on the Wing (Part I) 0:1:25, Dogs 0:17:3, Pigs (Three Different Ones) 0:11:25, Sheep 0:10:25, Pigs o
, Pink Floyd -- Dark Side of the Moon
[Speak to Me 0:1:30, Breathe 0:2:43, On the Run 0:3:36, Time 0:7:1, The Great Gig in the Sky 0:4:36, Money
, Pink Floyd -- Meddle
[One of These Days 0:5:57, A Pillow of Winds 0:5:7, Fearless 0:6:5, San Tropez 0:3:40, Seamus 0:2:13, Echo
, Pink Floyd -- Momentary Lapse of Reason
[Signs of Life 0:4:24, Learning to Fly 0:4:53, The Dogs of War 0:6:5, One Slip 0:5:10, On The Turning Away
, Pink Floyd -- Wish You Were Here
[Shine On You Crazy Diamond (Part One) 0:13:33, Welcome to The Machine 0:7:26, Have A Cigar 0:5:7, Wish Yo
, Pulp -- Different Class
[Mis-Shapes 0:3:46, Pencil Skirt 0:3:11, Common People 0:5:50, I Spy 0:5:55, Disco 2000 0:4:33, Live Bed S
, The Beatles -- Rubber Soul
[Drive My Car 0:2:25, Norwegian Wood (This Bird Has Flown) 0:2:1, You Won't See Me 0:3:18, Nowhere Man 0:2
, The Dave Brubeck Quartet -- Take Five
[Blue Rondo a la Turk 0:6:44, Strange Meadow Lark 0:7:22, Take Five 0:5:24, Three to Get Ready 0:5:24, Kat
, The Jimi Hendrix Experience -- Are you Experienced?
[Foxy Lady 0:3:22, Manic Depression 0:3:46, Red House 0:3:53, Can You See Me 0:2:35, Love or Confusion 0:3
]


Total play time of Pink Floyd albums:
13580 seconds


Album with largest collection of tracks:
Nimoy & Shatner -- Spaced Out
[King Henry The Fifth 0:3:0, Elegy For The Brave 0:3:18, Highly Illogical 0:2:20, If I Had A Hammer [The H
Number of tracks: 24


Longest track in collection:
Echoes 0:23:31
```

Messages to STDERR

None.