

CMP-4008Y Coursework 1

100463737 (adw23utu)
100476374 (rcy24jnu)

Tue, 17 Dec 2024 15:39

PDF prepared using pass-cli-server version 1.3.1 running on Linux 5.4.0-200-generic (amd64).

☑ We agree that by submitting a PDF generated by PASS we are confirming that we have checked the PDF and that it correctly represents our submission.



Contents

AlbumDatabase.java	2
Duration.java	4
Track.java	5
Album.java	6
AlbumCollection.java	7
Application	8
Compiler Invocation	8
Compiler Messages	8
Application Invocation	8
Messages to STDOUT	8
Messages to STDERR	8

AlbumDatabase.java

```

1  import java.io.File;
   import java.io.FileNotFoundException;
3  import java.util.Scanner;

5  public class AlbumDatabase {
   public static void main(String[] args) {
7      AlbumCollection collection = new AlbumCollection(); //Create an empty
        collection fir the album

9      try (Scanner scanner = new Scanner(new File("albums.txt"))) { // Open the
        file
        Album currentAlbum = null;
11     String artistName = "Kraftwerk";

13     while (scanner.hasNextLine()) { //loop through all lines in the file
        String line = scanner.nextLine();

15         // Check if the line contains album info
17         if (line.contains(":") && line.contains("(")) {
            String[] parts = line.split(" : "); // Split the line into
            artist and title+year
19             // https://www.w3resource.com/java-tutorial/string/string_split.
            php
            // Make sure the split operation produces two parts

21             if (parts.length == 2) {
23                 String artist = parts[0];
                String titleAndYear = parts[1];

25                 // Find the position of the parentheses for the year
27                 int startIndex = titleAndYear.lastIndexOf('(');
                int endIndex = titleAndYear.lastIndexOf(')');

29                 // Check if the brackets exist and the title and year
31                 if (startIndex != -1 && endIndex != -1) {
                    String title = titleAndYear.substring(0, startIndex); //
                    Title is before '('
33                     int year = Integer.parseInt(titleAndYear.substring(
                        startIndex + 1, endIndex).trim()); // Year is
                        between parentheses

35                     currentAlbum = new Album(artist, title, year); // Create
                        the album object
                        collection.addAlbum(currentAlbum); // Add the album to
                        the collection
37                 }
39             }

41             // check if the line contains a dash for the track
            if (currentAlbum != null && line.contains(" - ")) {
43                 String[] trackParts = line.split(" - ");
                if (trackParts.length == 2) { // check if the line has both time
                    and title
45                     String trackTitle = trackParts[1];
                    String[] timeParts = trackParts[0].split(":");
47                     //cast time to integers
                    if (timeParts.length == 3) {
49                         try {
                            int hours = Integer.parseInt(timeParts[0]);
51                             int minutes = Integer.parseInt(timeParts[1]);

```

```

53         int seconds = Integer.parseInt(timeParts[2]);
           // add track
           currentAlbum.addTrack(new Track(trackTitle, new
55               Duration(hours, minutes, seconds)));
       } catch (NumberFormatException e) {
           System.out.println("unable to read track/album " +
               line);
57       }
           }
59       }
           }
61     }
    } catch (FileNotFoundException e) {
63         System.out.println("File not found.");
    }

65     // print all albums and tracks
67     for (Album album : collection.getAlbumsSorted()) { // loop through albums
        System.out.println(album);
69     }
    System.out.println("\nTotal playtime of Kraftwerk albums: " + collection.
        getTotalPlaytime("Kraftwerk"));
71    System.out.println("\nAlbum with the shortest title: " + collection.
        getAlbumWithShortestTitle());
    System.out.println("\nLongest track in the collection: " + collection.
        getLongestTrack());
73 }
}

```

Duration.java

```
public class Duration implements Comparable<Duration> {
2    private final int hours;
    private final int minutes;
4    private final int seconds;

6    public Duration(int hours, int minutes, int seconds) {
        this.hours = hours;
8        this.minutes = minutes;
        this.seconds = seconds;
10    }

12    public static Duration fromSeconds(int totalSeconds) {
        int hours = totalSeconds / 3600;
14        int minutes = (totalSeconds % 3600) / 60;
        int seconds = totalSeconds % 60;
16        return new Duration(hours, minutes, seconds);
    }

18    public int totalSeconds() {
20        return hours * 3600 + minutes * 60 + seconds;
    }

22    @Override
24    public int compareTo(Duration other) {
        return Integer.compare(this.totalSeconds(), other.totalSeconds());
26    }

28    @Override
    public String toString() {
30        return String.format("%02d:%02d:%02d", hours, minutes, seconds);
    }
32 }
```

Track.java

```
public class Track implements Comparable<Track> {
2   private final String title;
   private final Duration duration;
4
   public Track(String title, Duration duration) {
6       this.title = title;
       this.duration = duration;
8   }

   public String getTitle() {
10       return title;
12   }

   public Duration getDuration() {
14       return duration;
16   }

   @Override
18   public int compareTo(Track other) {
20       return this.duration.compareTo(other.duration);
   }

22   @Override
24   public String toString() {
       return String.format("Track: %s, Duration: %s", title, duration);
26   }
}
```

Album.java

```
1  import java.util.ArrayList;
import java.util.List;
3
public class Album implements Comparable<Album> {
5     private final String artist;
private final String title;
7     private final int year;
private final List<Track> tracks;
9
    //copied from the date task, reused for this task
11    public Album(String artist, String title, int year) {
        this.artist = artist;
13        this.title = title;
        this.year = year;
15        this.tracks = new ArrayList<>();
    }
17    //adds track to tracklist
    public void addTrack(Track track) {
19        tracks.add(track);
    }
21    //accessor method
    public String getArtist() {
23        return artist;
    }
25
    public String getTitle() {
27        return title;
    }
29
    public int getYear() {
31        return year;
    }
33
    public List<Track> getTracks() {
35        return tracks;
    }
37
    @Override
39    public int compareTo(Album other) {
        return this.artist.compareToIgnoreCase(other.artist);
41        //refer back to compareTo link
    }
43
    @Override
45    public String toString() {
        return String.format("Artist: %s, Title: %s, Year: %d, Tracks: %d", artist,
            title, year, tracks.size());
47    }
    //format the output of the program
49 }
```

AlbumCollection.java

```

1  import java.util.ArrayList;
   import java.util.Collections;
3  import java.util.List;

5  public class AlbumCollection {
   private final List<Album> albums;

7

   public AlbumCollection() {
8       albums = new ArrayList<>();
   }

11

   public void addAlbum(Album album) {
13       if (album != null) {
           albums.add(album);
15       }
   }

17

   public List<Album> getAlbumsSorted() {
19       Collections.sort(albums); //sort albums
       return albums; //return the albums in order
21   }

23   public Duration getTotalPlaytime(String artist) {
       int totalSeconds = 0;
25       for (Album album : albums) {
           if (album.getArtist().equalsIgnoreCase(artist)) {
27               totalSeconds += album.getTracks().stream()
                                   .mapToInt(track -> track.getDuration().totalSeconds())
                                   .sum();
29           }
31       }
       return Duration.fromSeconds(totalSeconds);
33   }

35   public Album getAlbumWithShortestTitle() {
       return albums.stream()
37           .min((a1, a2) -> Integer.compare(a1.getTitle().length(), a2.getTitle()
                                   .length()))
           .orElse(null);
39       // use compare to compare and get the shortest track
       // https://www.w3schools.com/java/ref_string_compareto.asp#:~:text=The%20
       compareTo()%20method%20compares,equal%20to%20the%20other%20string.
41   }

43   public Track getLongestTrack() {
       return albums.stream()
45           .flatMap(album -> album.getTracks().stream())
           .max(Track::compareTo)
           .orElse(null);
47   }
49 }

```

Application

Compiler Invocation

```
javac -Xlint:unchecked -Xlint:deprecation -encoding UTF-8 -d
  prepasg9265860340921682383classes AlbumDatabase.java Duration.java Track.java
  Album.java AlbumCollection.java
```

Compiler Messages

None.

Application Invocation

```
java AlbumDatabase
```

Messages to STDOUT

```
Artist: Blondie, Title: Parallel Lines , Year: 1978, Tracks: 12
Artist: Goldfrapp, Title: Supernature , Year: 2005, Tracks: 11
Artist: Jordi Savall & Hesperion XX, Title: Folias and Canarios , Year: 1994, Tracks: 20
Artist: Kraftwerk, Title: The Man Machine , Year: 1978, Tracks: 6
Artist: Kraftwerk, Title: Trans Europe Express , Year: 1977, Tracks: 8
Artist: Led Zeppelin, Title: Led Zeppelin IV , Year: 1971, Tracks: 8
Artist: Marillion, Title: Script for a Jester's Tear , Year: 1983, Tracks: 6
Artist: Miles Davis, Title: Kind of Blue , Year: 1959, Tracks: 5
Artist: Neil Pye, Title: Neil's Heavy Concept Album , Year: 1984, Tracks: 20
Artist: Nimoy & Shatner, Title: Spaced Out , Year: 1997, Tracks: 24
Artist: Pink Floyd, Title: Dark Side of the Moon , Year: 1973, Tracks: 10
Artist: Pink Floyd, Title: Wish You Were Here , Year: 1975, Tracks: 5
Artist: Pink Floyd, Title: Animals , Year: 1977, Tracks: 5
Artist: Pink Floyd, Title: Meddle , Year: 1971, Tracks: 6
Artist: Pink Floyd, Title: Momentary Lapse of Reason , Year: 1987, Tracks: 10
Artist: Pulp, Title: Different Class , Year: 1995, Tracks: 12
Artist: The Beatles, Title: Rubber Soul , Year: 1965, Tracks: 14
Artist: The Dave Brubeck Quartet, Title: Time Out , Year: 1959, Tracks: 7
Artist: The Jimi Hendrix Experience, Title: Are you Experienced? , Year: 1967, Tracks: 17
```

Total playtime of Kraftwerk albums: 01:19:11

Album with the shortest title: Artist: Pink Floyd, Title: Meddle , Year: 1971, Tracks: 6

Longest track in the collection: Track: Echoes, Duration: 00:23:31

Messages to STDERR

None.