44-563 Developing Web Applications and Services

Project 05

**Create a Java application project, NOT a web application.**

**This is a Java console application.**

**NOTE: You must follow the generic guideline posted on the course website in the unit “Projects”**

1) The following DTD describes an XML language for storing information about songs.

<?xml version="1.0" encoding="UTF-8"?>

<!--

Document: songs.dtd

Description: Song information.

Song lengths are in minutes (should it be an integer number NOT a floating point).

Song ratings are integers between 0 and 10 (inclusive).

-->

<!ELEMENT songList (song+)>

<!ELEMENT song (title, singer, length, ratingList)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT singer (#PCDATA)>

<!ELEMENT length (#PCDATA)>

<!ELEMENT ratingList (rating+)>

<!ELEMENT rating (#PCDATA)>

2) In this project, you will create Java classes that can be used to parse data stored in an XML file that conforms to this DTD.

a) For testing purposes, you should begin by creating such an XML file named songs.xml that contains information for several songs. You **MUST** use this name for your XML data file.

b) **We will test your program with our own data file, so be sure your program works for any data having the specified format.**

c) Here is an example of information for one song:

<song>

<title>What Do You Mean</title>

<singer>Justin Bieber</singer>

<length>5</length>

<ratingList>

<rating>8</rating>

<rating>9</rating>

</ratingList>

</song>

3) Create a Java class named Song with the following elements:

a) Four instance variables:

private String title;

private String singer;

private int length;

List<Integer> ratings;

b) A constructor with four parameters, one for each of four instance variables, in the order listed above.

c) Four getter methods, one for each of the four instance variables.

d) A method

public double getAverageRating()

that returns the average of the ratings.

e) A toString() method that returns a string consisting of the title, singer, length, and average rating using the format "%-25s %-25s %3d %5.1f". Hint: use the String.format() method.

4) Create a Java class SongParser with the following elements:

a) Two instance variables:

private DocumentBuilder builder;

private XPath path;

b) A no-arg constructor that initializes the two instance variables.

c) A method

public List<Song> getSongsFromXMLFile(String fileName)

that parses an XML file containing song information that conforms to song.dtd. This method returns a list of Song objects representing the information in the XML file.

5) Create a Java class named SongInformation that has only a main() method. This method first creates a new SongParser object and then calls the getSongsFromXMLFile() method with argument "songs.xml" to obtain the list of songs. It then produces the output listed below. Be sure to include appropriate labels for the information.

a) A list showing the title, singer, length, and average rating of every song.

b) A list giving the title of each song that is more than (include) six minutes long.

c) The title of the song that has the lowest average rating.

d) The title of the longest song.

e) The average length of the songs.

SAMPLE OUTPUT

(Your program will be tested using a file different from the one used to produce this output.)

Song Information

Title Singer Length Average Rating

What Do You Mean Justin Bieber 5 8.5

The Hills The Weeknd 7 9.0

Cool For The Summer Demi Lovato 6 6.0

Photograph Ed Sheeran 4 7.5

All Songs More Than Six Minutes Long

The Hills

Cool For The Summer

The song with the lowest average rating is Cool For The Summer.

The longest song is The Hills.

The average length of the songs is 5.5 minutes.