Streaming Service
Individual Assignment Part B

Outline

For this assignment, you will develop the *Streaming Service* application in Java which allows a user to query digital content (films or music) available on a streaming service.

Your Task

To complete the assignment, implement all methods for the Streaming Service in Java as specified below.

Getting Started

The streaming.zip archive contains five Java files. Create a new Java Project in Eclipse and add the source files. The AssignmentDemo.java file contains a main method. You should be able to run this method with the following message appearing in the console: *Welcome to the Streaming Service.*

Methodology

Use correct access modifiers for all instance variables with get and set methods. Do not add new methods aside from those described below. Make sure you include your name, student number and course code (COMP503, ENSE502 or ENSE602) in a comment section at the top of each class. Follow the order below, uncommenting the appropriate code in AssignmentDemo.java to test your code. AssignmentDemo also has an example of the console output.

This assessment has been prepared by Dr. Kenneth Johnson at the Auckland University of Technology. Questions? Ask the team at programming2@aut.ac.nz

DigitalContent: This is an abstract class with three instance variables for the digital content’s title, publisher and release dates (as strings). Each instance variable has get and set methods. DigitalContent objects can only be instantiated by supplying input parameters for all three instance variables. Implement a toString method to return a string describing the digital content. Implement the match method to return true if the input query is contained in either the title or publisher or release date. The query is not case sensitive. Implement the Comparable interface to compare titles.

Film: This class extends DigitalContent with a string instance variable to store the cast members on the film. Create get and set methods. Write a constructor that initializes all instance variables for Film objects. Write a toString method which uses super to invoke DigitalContent toString, but also adding the cast members to the output string. Override the match method to also include a check if the query match the cast. It must invoke the superclass match in DigitalContent.

Music: This class extends DigitalContent with a string instance variable to store the artist’s name. Create get and set methods. Write a constructor that initializes all instance variables for Music objects. Write a toString method which uses super to invoke DigitalContent’s toString, but also adding the artist’s name to the output string. Override the match method to also include a check if the query matches the artist. It must invoke the superclass match in DigitalContent.

StreamingService: This class maintains a private ArrayList of digital content. Do not include get or set methods for this instance variable. Write a default constructor to initialise the ArrayList. Write an add method which takes a DigitalContent object to add to the list. Write the match method to return an arraylist of matching digital contents.

This assessment has been prepared by Dr. Kenneth Johnson at the Auckland University of Technology. Questions? Ask the team at programming2@aut.ac.nz

Marking Scheme

Your total mark for this assignment is out of 50

Class	Marks	Grade A 80% - 100%	Grade B 65% - 79%	Grade C 50% - 64%	Grade D Fail
DigitalContent	20 marks	OOP paradigm consistently used for implementation of all functionality	Inconsistent use of OOP paradigm but functionality is correct	Poor use of OOP paradigm and incorrect functionality	Absent classes or code does not compile
Music and *Film*	15 marks				
StreamingService	10 marks				
Code Quality: -Whitespace -Naming - Commenting	5 marks	Whitespace is comprehensively consistent. All naming is sensible and meaningful. Code reuse is present. Student Name/Number and Course Code included	Whitespace is comprehensively consistent. Majority of naming is sensible	Whitespace is comprehensively consistent. Inconsistencies in naming	Whitespace is inconsistent and hence code is difficult to read. No student identification

Submission

Submission is online via Blackboard.

Please submit only .java files in an archive file

Name the archive file as: lastname-firstname-studentid-coursecode.zip

Do not rename any of the assignment source code files. Do not submit the .class files. Do not submit any eclipse project files etc.

This assessment has been prepared by Dr. Kenneth Johnson at the Auckland University of Technology. Questions? Ask the team at programming2@aut.ac.nz