

Middle East Technical University
Department of Computer Engineering

CENG 443 Intro. to Object-Oriented Prog. Lang. and Systems
Spring 2021-2022

Homework#3 – Movies & Directors
Version 1.0

Due date: June 12, 2022, 23:59

1. Introduction

In this assignment, you will write a Java class with Stream processing commands and Lambda expressions to query a csv file containing information about the movies. A sample file is attached, but your program can be tested with some additional data files of the same form.

The input file has been adapted from the IMDB movies file downloaded from *Kaggle.com* only for educational use in this course. It has a header line naming the fields at the top, and its gross (revenue) field contains the value -1 if the revenue for that movie is unknown. Please, examine the sample file for the details.

You are asked to write the following queries:

Query-1: Titles of all the movies.

Query-2: Directors who directed a movie with a rating of at least 8.5.

Query-3: Director who made an Adventure movie with the least known gross revenue.

Query-4: Directors who have directed both a successful movie with a rating of at least 8.5 and also a moderate movie with a rating of at most 8.

Query-5: Sum of the duration of the movies directed by the director of the earliest movie in the list.

The MovieQuery class you are going to develop will contain the method

```
public static void queryAll(String inFileName, String outFileName)
```

and any other helper methods.

This method is expected to read the movies data from the input csv file `<inFileName>.csv` and write only the output of each query to a separate output file named `<outFileName>-<queryNumber>.out`.

Hence, if this method is called as

```
MovieQuery.queryAll("movies", "result");
```

then it will read the data records from the file `"movies.csv"` and write the outputs of the queries from 1 to 5 into the files `"result-1.out"`, `"result-2.out"`, ..., `"result-5.out"`, respectively.

2. Important Details

- All the text outputs must be in uppercase and sorted order with no duplicates.
- You must NOT use any traditional looping or conditional structures in your solutions. Instead, you can use Stream processing statements and Lambda expressions.
- In the query outputs, each item should be printed on a separate line.

3. Grading

- Clean implementation with JavaDoc/Java comments explaining your code..... 20%
- Tests to check whether your implementation matches the specifications listed..... 80%

4. Submission

Submission will be made via ODTUClass. You will submit your class file called MovieQuery.java (or a zip file named hw3.zip containing this class and your helper classes, if any). A penalty of 5 x LateDay² will be applied for submissions that are late, at most three days.

Your submission must be your original solution. The regulations will be applied in case of high similarity scores.