University of Newcastle
School of Electrical Engineering and Computer Science
SENG1110/6110 Programming Assignment 1 – Semester 1, 2017
Due: By electronic submission (Blackboard) by 11:59pm on **Wed 19/Apr/2017.**

Movie database

Introduction

The objective of this assignment is the implementation of an object oriented program using Java. The assignment 2 will be an extension of this assignment.

Before you start

Carefully read the problem description below. Make sure that you have all the information necessary to create the program. Do not assume what is necessary. There is a discussion board forum: assignment 1. Post your questions there and check regularly. Start the assignment as soon as possible.

Problem Description

Your task in this assignment is to create a simple movie database with the following functionality:

1. Will allow the user to **enter a new movie** into the **movie database**.
2. Will allow the user to create **playlists** from their **movies**.
3. Will allow the user to request the **list of movies** in the database
4. Will allow the user to request the **list of playlists** and the respective movies.
5. Will allow the user to request the **list of movies in a specific playlist.**
6. Will prevent the user from adding a movie when the database is full.
7. Will prevent the user from adding a movie to a playlist that is full.
8. Will allow the user to request a **list of all movies** with the **same director**.
9. Will allow the user to delete a **movie** from a **playlist** (but not from the movie **database**).
10. Will allow the user to delete a **playlist** (but not those movies from the **database**).
11. Will allow the user to delete a **movie** from the **database** (and any associated **playlist(s)**).

Your program must give appropriate messages to the user on an attempt to:

- add a movie to the database when full,
- add a movie to a playlist when full,
- delete a playlist that does not exist.
- delete a movie that does not exist.
- add a movie that already exist (note that 2 movies are identical if name, director and duration are the same).
- Note that Strings (name of the movie and director) must not be case sensitivity. For example, “the godfather”, “The godfather”, “The Godfather” or even “the godFather” are the same movie.

Program Requirements:

Your program should implement four classes, which store the following data:

- `Movie.java` – storing the following details about a movie.
    - `name` – the name of the movie.
    - `director` – the director of the movie.

- `fileSize` – the size of the file in terms of memory (MB).
- `duration` – the length of the movie in minutes

- `MovieDatabase.java` – stores all current movies in the system.
  - `movie1, movie2, movie3, movie4` – All current Movie objects stored in the system.
- `Playlist.java` – stores up to 3 movies at a time.
  - `movie1, movie2, movie3` – Movies objects
  - `totalDuration` – the total playing time of all movies stored
  - `totalSize` – the current sum of all movie file sizes stored in this playlist.
- `Interface.java` – provides the user interface.
  - `playlist1, playlist2` – Can store up to two Playlist objects.
  - `database` – stores a MovieDatabase object, with all movies in the system.

All the data components of your classes need to be **private** (this means that you are applying the principles of ***encapsulation***).

Additionally, your classes need to have methods that provide the functionality outlined in the problem description. The only class which should have a **main method** is `Interface.java`, which should create an instance of the class Interface and call the run() method which will have code to provide the user with a menu to allow them to perform any of the tasks outlined in the problem description. You can use TIO or GUI (it is your choice). The template for this is below. The class Interface also will be the only one that will receive inputs and show outputs.

```
public class Interface {
	private void run() {
		//This method should control the flow of the program
		//and have all code for accessing the playlists
		//and movie database
	}
	public static void main(String[] args){
		Interface intFace = new Interface();
		intFace.run();
	}

}
```

You **cannot use arrays** in this assignment. Your solution must be your own work. **Marks** will be awarded for: layout, both visual (variable names, indentation) and structural (scope of variables, use of methods); documentation (comments); and ability of the submission to perform as specified. A more detailed marking schema will be available in Blackboard.

What to submit.

You should submit the Java program (`Movie.java, Playlist.java, MovieDatabase.java` and `Interface.java`) and the assignment cover sheet electronically under Assignment 1 link in Blackboard.

Extra Work for SENG6110 students

You need to provide a UML class diagram of your program.

In the Blackboard you will find a new forum in the discussion board: "assignment1". Any question about the assignment 1 you can post there. Check this forum regularly.

Prof Regina Berretta.
Mar - 2017