Hacettepe University

BBM104 - 2021 Spring

ASSIGNMENT 2

INHERITANCE, ACCESS MODIFIERS

April 14, 2021

MELİKE NUR DULKADİR

21992919

1. Problem

In this assignment, we are supposed to develop a simple Movie Database System similar to IMDB with using inheritance mechanism and access modifiers. Programming language of this assignment is Java.

2. Implementation

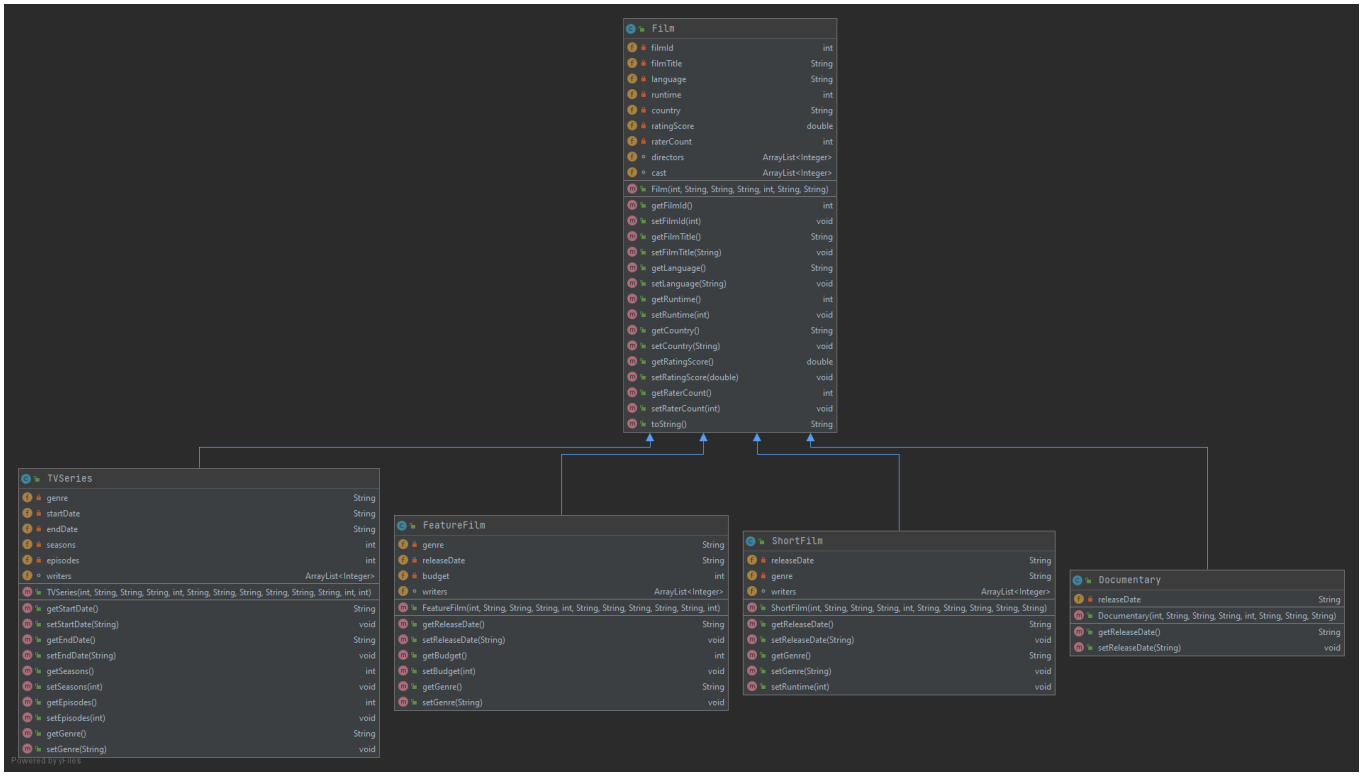
First, I set up an inheritance mechanism with the help of the information given in pdf. Person and Film became main classes accordingly. User and Artist classes extended the Person class, Performer, Writer and Director classes extended the Artist class. There were also 3 classes that extended the Performer class: the Actor Childactor and the StuntPerformer class. FeatureFilm, ShortFilm, Documentary and TVSeries extended the Film class. I have defined attributes that are common to all persons such as id, name, surname, country as private in the Person class, so that these attributes can be accessed in a controlled manner using only the getter and setter. I also defined a constructor for all Person's subclasses. I set private all the fields of all classes except the arraylists because they are controlled. I created the same structure in film classes.

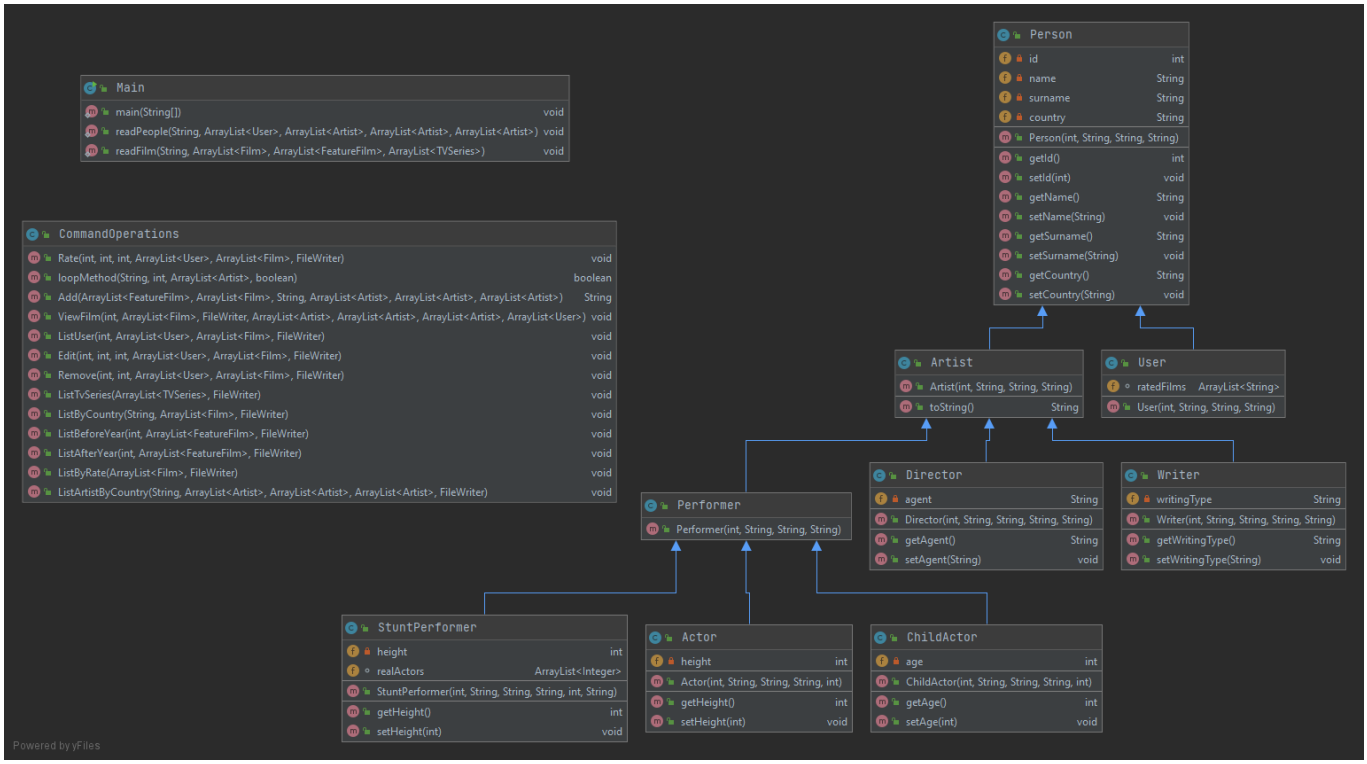
In the file reading section, I first read the *person.txt* file and added each person to the required arrays. Then I read *films.txt* file and added the films to the relevant arraylists.I've written the necessary functions to perform each command in *commands.txt*. Accordingly, the algorithm performs relative operations according to the requested command and prints the results to *output.txt*. According to these commands, users can rate, then change his/her rate or delete, and list all their votes. New feature films can be added to the system, feature films that were published before or after a certain year can be listed, the details of every film in the system can be viewed or the films can be listed according to the their countrys or their rate degrees.

3. Comments

This problem demonstrated the use of OOP logic in the real world, taught us to use inheritance mechanics better. It enabled me to develop securely in the solution by using the OOP concept and necessary access modifiers, avoiding code duplication and encapsulating data.

4. UML Diagrams





5. References

- **https://www.javatpoint.com/inheritance-in-java**
- **https://www.w3schools.com/java/java_inheritance.asp**
- **https://www.tutorialspoint.com/java/java_access_modifiers.htm**