



HACETTEPE UNIVERSITY

Author: Muhammed Görkem KOLA

ID: b2200765032

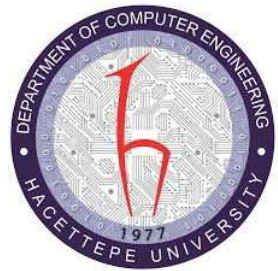
Lecturers: Sevil ŞEN, Cemil ZALLUHOĞLU

Teaching Assistants: Bahar GEZİCİ, Merve ÖZDEŞ

REPORT OF ASSIGNMENT2

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING



2020-2021 Spring Semester

2021.16.04

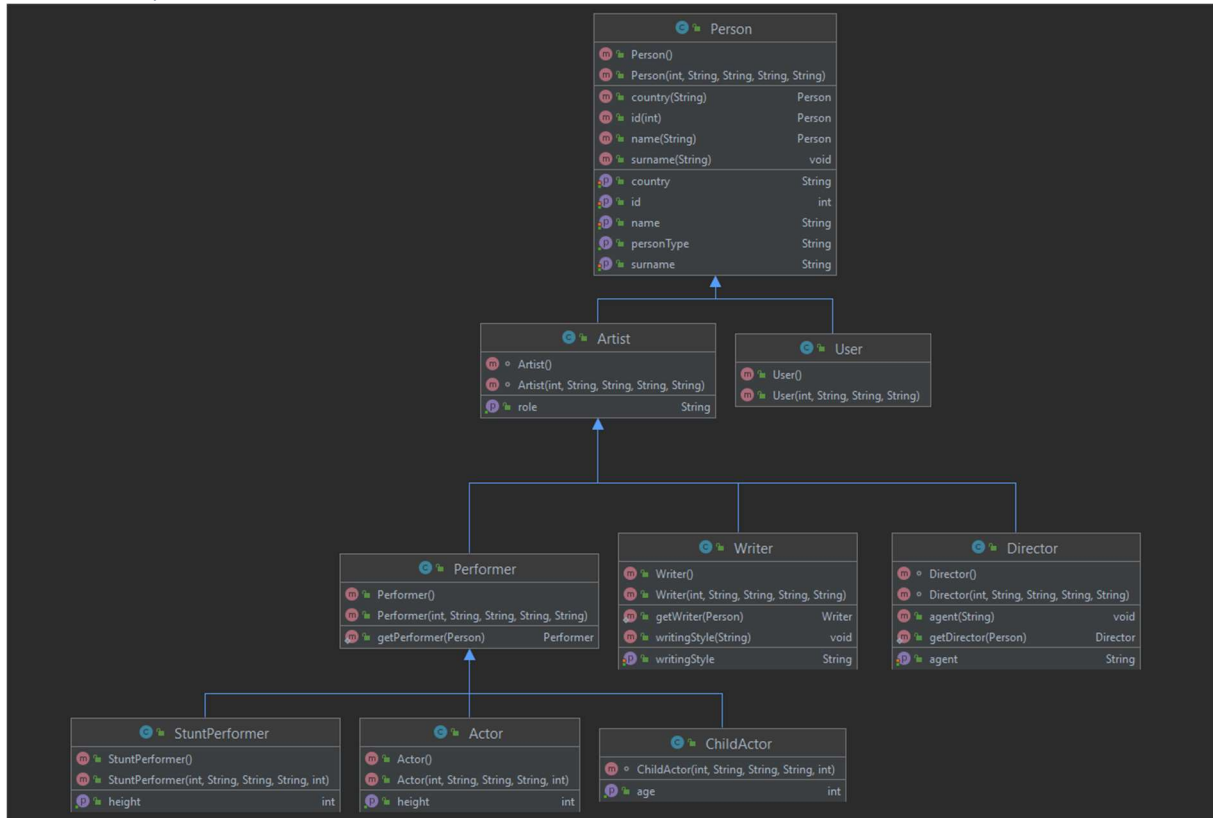
1.1 CLASS DIAGRAMS



1.2 SOLUTION

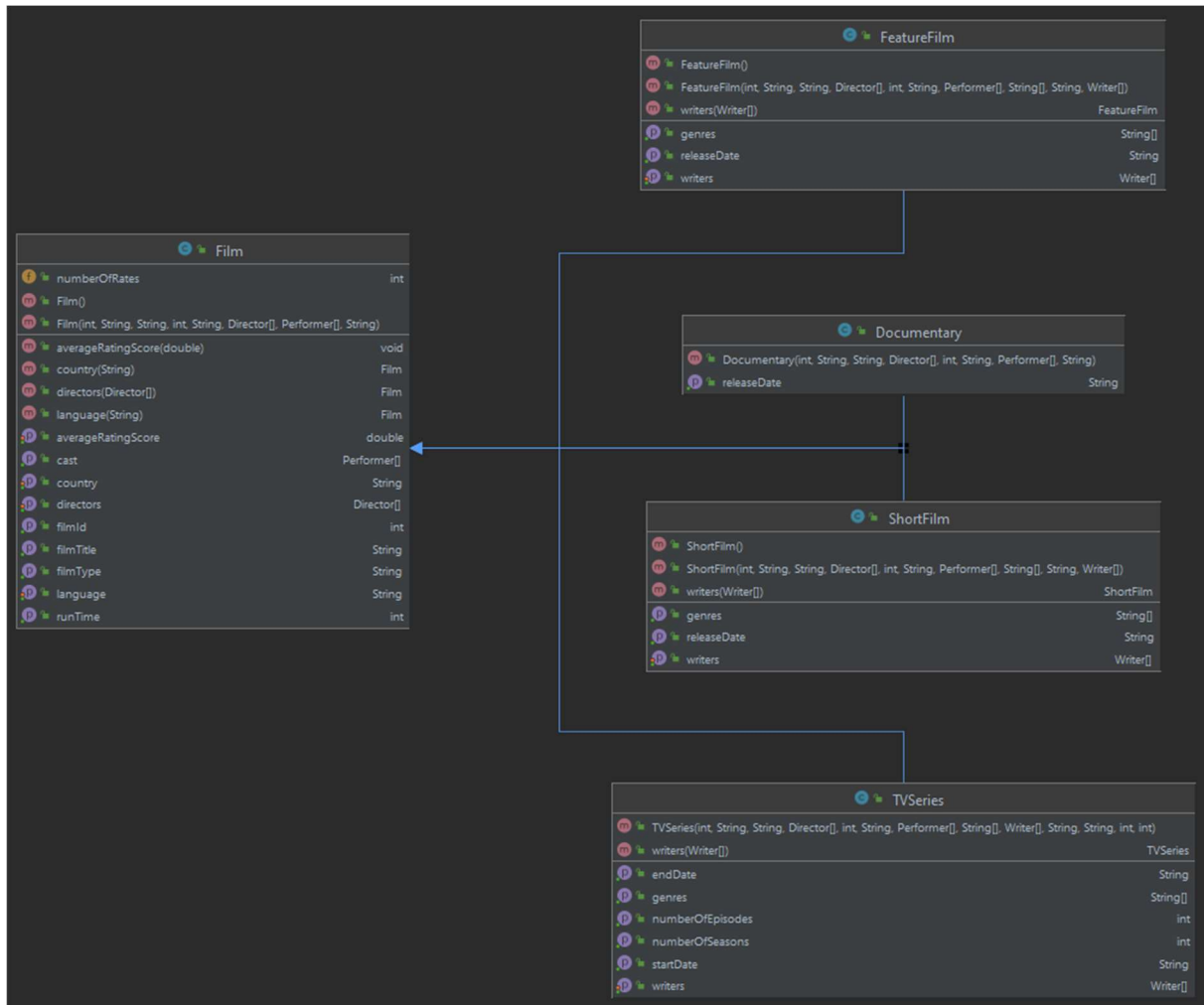
- In this Assignment, we are expected to create a Movie Database System and we are given some important informations such as Film types.
- I created many classes to fit in polymorphism because it is easier to see if there whether a problem and to update the program.
- There are 2 classes "Artist, User" inherite Person class, 3 classes "Performer, Director Writer" inherites Artist Class, 3 classes "Actor, StuntPerformer, ChildActor" inherite Performer class.
- There are 4 classes "ShortFilm, FeatureFilm, Documentary, TVSeries" inherite Film class.
- There are 3 classes "ReadingCommandsFile, ReadingFilmsFile, ReadingPeopleFile" inherite FileInput class.
- The other classes does not inherite another class because it would be unnecessary for other classes.

1.2.1 People Classes



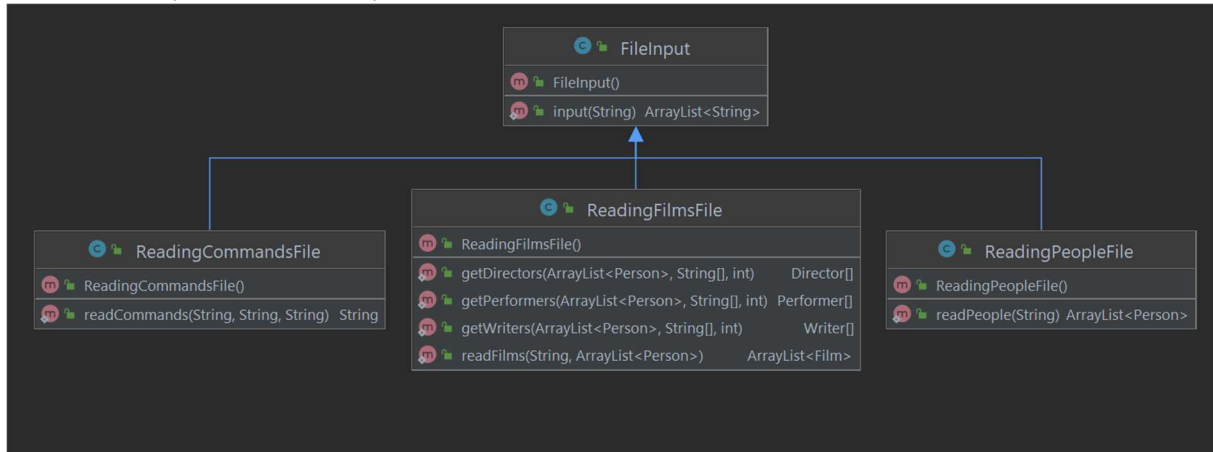
- Person class has an empty constructor, a constructor using fields; country, id, name, personType, surname attributes; getters, setters and fluent setters for attributes.
- Artist class has role variable.
- Performer, Writer, Director classes have getters because I needed to use some of them in my code.
- Writer has writingStyle, Director has agent, StuntPerformer and Actor have height, ChildActor has age attributes.
- I created personType and role variables because when I created that classes I did not know how to use InstanceOf operator.

1.2.2 Films Classes

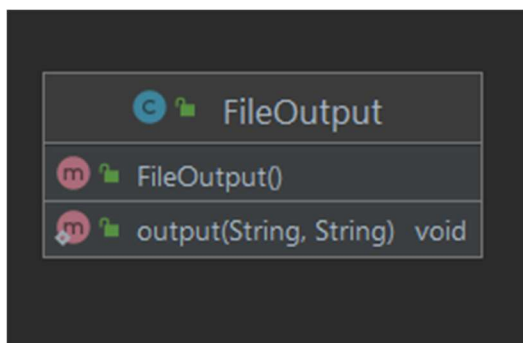


- Film Class has an empty constructor, constructor using fields; an empty constructor, a constructor using fields; numberOfRates, averageRatingScore, cast country, directors, filmID, filmTitle, filmType, language, runTime attributes; getters, setters and fluent setters.
- FeatureFilm class has an empty constructor, a constructor using fields; genres, releaseDate, writers attributes; getters, a setters, a fluent setter.
- Documentary has a constructor using fields; releaseDate attribute and getter for it.
- ShortFilm class has an empty constructor, a constructor using fields; genres, releaseDate, writers attributes; getters, a setter, a fluent setter.
- TVSeries has a constructor using fields; end date, genres, numberOfEpisodes, numberOfSeasons, startDate, writers attributes; getters, a setter, a fluent setter.

1.2.3 FileInput and FileOutputClasses



- FileInput class has an empty constructor; input function to read a file and returns a list.
- ReadingCommandsFile has an empty constructor, a constructor using fields; readCommands function to read commands file and returns a string that will be written on a file
- ReadingFilmsFile has an empty constructor; getDirectors, getPerformers, getWriters, readFilms function. readFilms function returns an Film ArrayList to use later.
- ReadingPeopleFile has an empty constructor; a function returns a Person ArrayList to use later.



- FileOutput class has an empty constructor and an output function that writes a file on a given path.

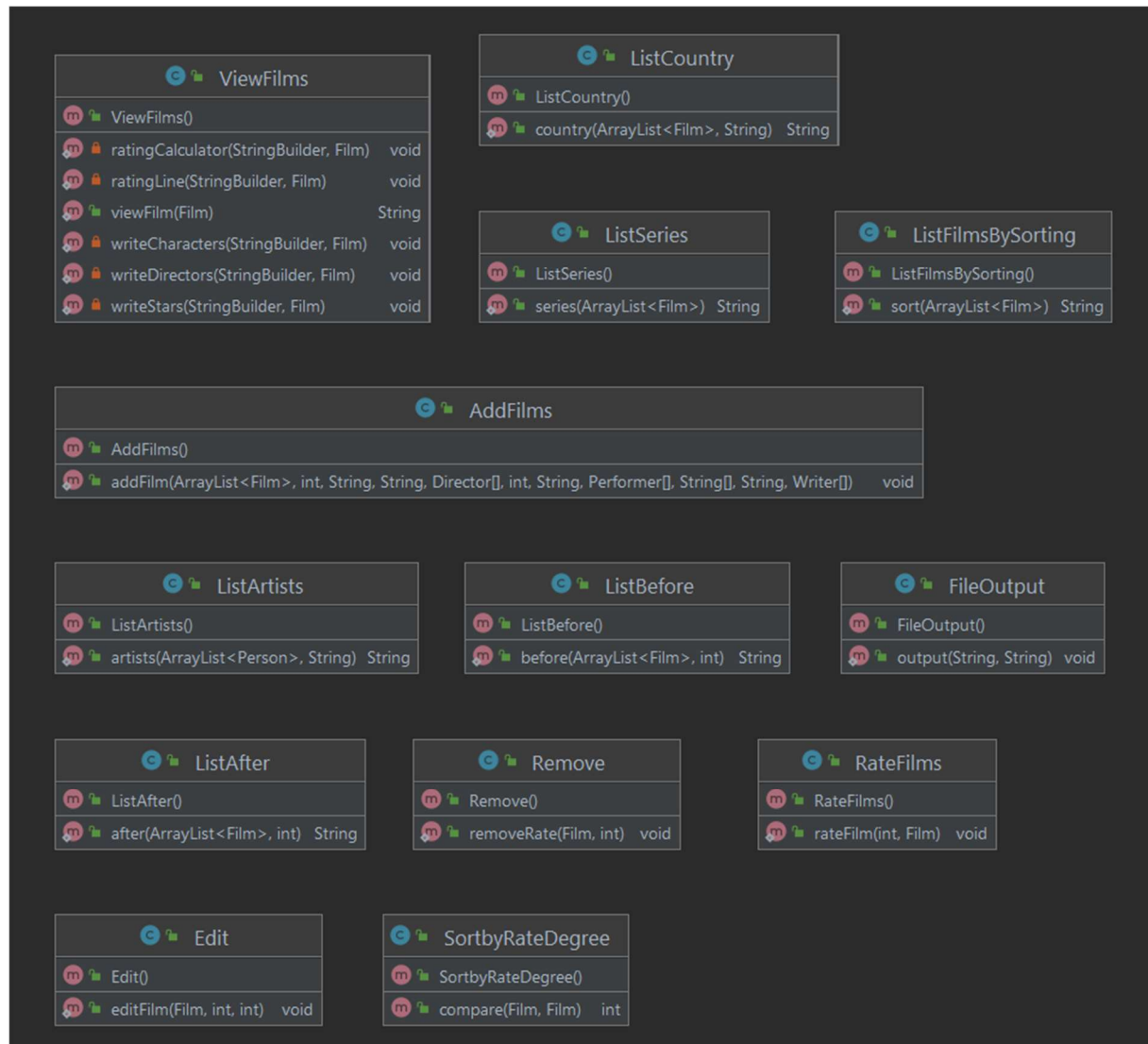
1.2.4 ReturnFilm and ReturnPerson Classes

ReturnPerson		
ReturnPerson()		
returnPeopleByType(ArrayList<Person>, String)	ArrayList<Person>	
returnPersonById(ArrayList<Person>, int)		Person?

ReturnFilm		
ReturnFilm()		
returnFilmById(ArrayList<Film>, int)		Film?
returnFilmsByType(ArrayList<Film>, String)	ArrayList<Film>	

- ReturnPerson has an empty constructor and returnPeopleByType function returns a People ArrayList by a Type and returnPersonById function that returns a Person by using person ID.
- ReturnFilm has an empty constructor and returnFilmById function that returns a Film by using film ID and returnFilmsByType function returns a Film ArrayList.

1.2.5 Command Classes



- ViewFilms has an empty constructor; public viewFilm function; private ratingCalculator, ratingLine, writeCharacters, writeDirectors, writeStars functions. viewFilm function returns a string that will be added on a finalText, ratingCalculator function transforms average rating to a nice form and ratingLine function adds the ratingLine to the finalText, writeCharacters function adds characters on the finalText and writeDirectors and writeStars functions write characters to finalText.
- ListCountry has an empty constructor and country function that adds all the films shot in a specified country to finalText.
- ListSeries has an empty constructor and series function that adds all the series to finalText.
- ListFilmsBySorting function has an empty constructor and sort function shows films in a descending order by using SortByRateDegree class's compare function.
- SortByRateDegree has an empty constructor and compare function sorts films in a descending order by using averageRatingScore.
- AddFilms has an empty constructor and addFilm function that adds a Film to Film ArrayList
- ListArtists has an empty constructor and artists function that adds all artists by categorizing to finalText.
- ListBefore has an empty constructor and before function that adds films which shot before a specified year to finalText.

- ListAfter has an empty constructor and after function that adds films which shot after a specified year to finalText.
- Remove has an empty constructor and removeRate function that deletes a user's rate for a film from an ArrayList.
- RateFilms has an empty constructor and rateFilm function that adds a user's rate to an ArrayList.
- Edit has an empty constructor and editFilm function that edits a user's rate for a film in an ArrayList.

1.3 ADVENTEGES AND DISADVENTEGES

1.3.1 Adventeges:

- Program has many classes and it is important for polymorphism.
- It is readable and has many command Lines.
- When a person wants to update it is very easy to update.

1.3.2 Disadventeges:

- Some methods are too long.
- Some methods are unnecessary.

REFERENCES:

- stackoverflow.com
- btkakademi.com
- BBM102 Lecture Notes