Adriana Abella Kuri

A01329591

Programación orientada a objetos

Situación problema: modelado de servicio de streaming

Dr. Ariel Lucien García Gamboa

Índice de contenido

Introducción

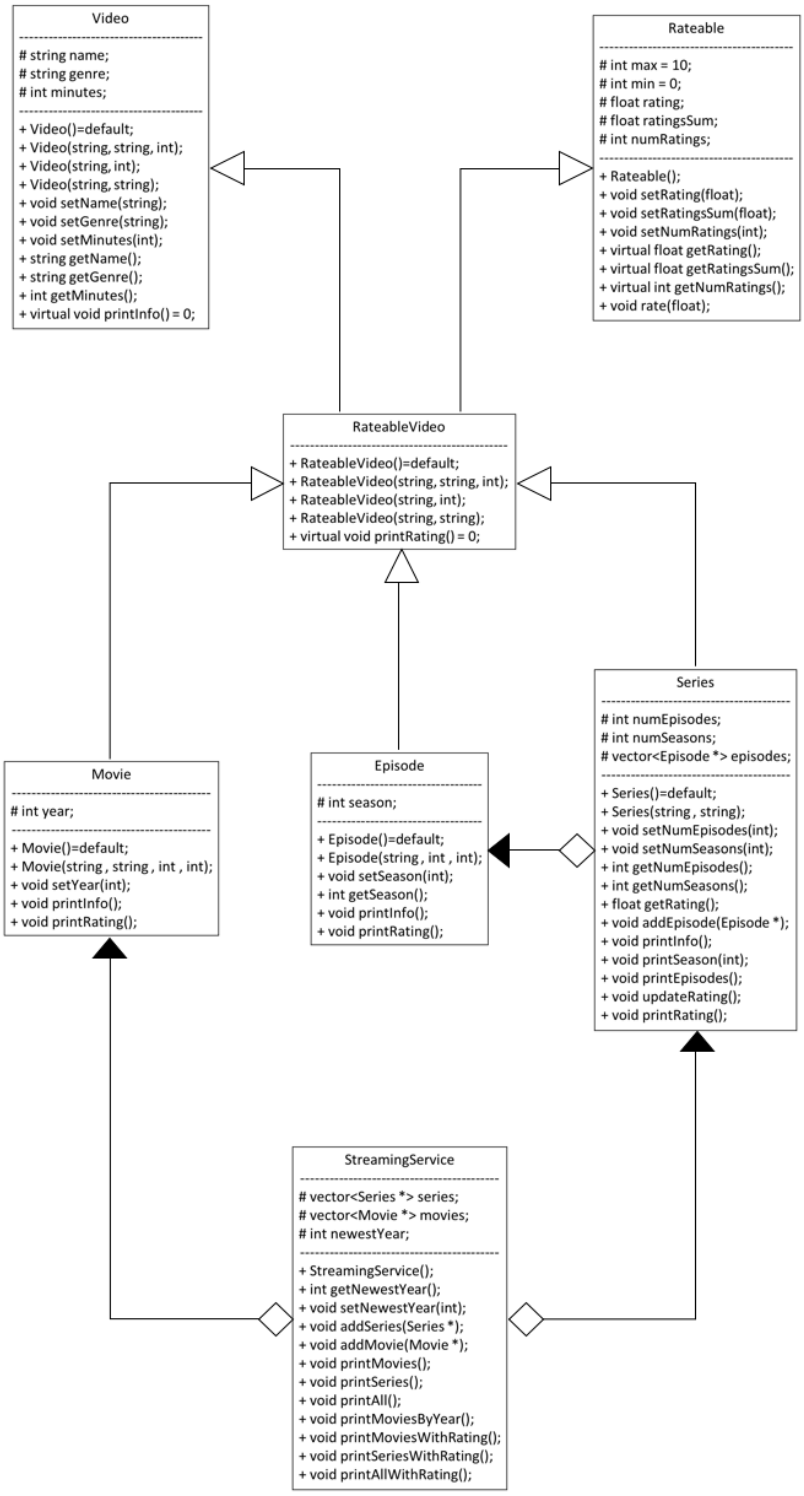
La situación problema consistía en modelar un servicio de streaming en donde se puedan calificar y visualizar las calificaciones tanto de películas como de series y sus capítulos. El rango de las calificaciones para todo tipo de videos va del 0 a 10. El programa es capaz de crear instancias de películas, series y capítulos con nombre, duración y genero. También es capaz de asignarles una calificación y agregarlos a una instancia de servicio de streaming, la cual es capaz de imprimir los nombres de las películas y las series con sus respectivas calificaciones. El programa también puede imprimir la información de una serie, la información de una temporada de la serie y la información de todos sus capítulos. De igual manera puede imprimir una lista de las películas en orden del año en el que fueron estrenadas.

Diagrama de clases UML y argumentación

Propuse el siguiente diseño para el programa porque me pareció necesario que las clases Movie, Episode y Series tuvieran todas un nombre, género, y duración en minutos, por eso todas ellas heredan de la clase RateableVideo, que a su vez hereda de la clase Video, que tiene todos los atributos que mencioné (name, genre, minutes). También necesitaban todas una función que imprimiera toda su información (printInfo()), pero en todas ellas debía ser diferente y es por eso que la declaré como una función virtual = 0. Esto hace que la clase Video sea abstracta y el programa no permite que se hagan instancias de esta clase.

Creé la clase Rateable para que fuera posible calificar todas las instancias de las clases que heredaran de ella. Hice una clase abstracta llamada RateableVideo que hereda tanto de Video como de Rateable. Las clases Movie, Episode y Series son todas clases hijas de RateableVideo, por lo que todas tienen un nombre, género, minutos y calificación.

Adicionalmente la clase Movie tiene el atributo year, y la clase Episode el atributo season para indicar a qué temporada de su serie pertenece.


Video
string name;
string genre;
int minutes;
+ Video()=default;
+ Video(string, string, int);
+ Video(string, int);
+ Video(string, string);
+ void setName(string);
+ void setGenre(string);
+ void setMinutes(int);
+ string getName();
+ string getGenre();
+ int getMinutes();
+ virtual void printInfo() = 0;
Rateable
int max = 10;
int min = 0;
float rating;
float ratingsSum;
int numRatings;
+ Rateable();
+ void setRating(float);
+ void setRatingsSum(float);
+ void setNumRatings(int);
+ virtual float getRating();
+ virtual float getRatingsSum();
+ virtual int getNumRatings();
+ void rate(float);
RateableVideo
+ RateableVideo()=default;
+ RateableVideo(string, string, int);
+ RateableVideo(string, int);
+ RateableVideo(string, string);
+ virtual void printRating() = 0;
Movie
int year;
+ Movie()=default;
+ Movie(string , string , int , int);
+ void setYear(int);
+ void printInfo();
+ void printRating();
Episode
int season;
+ Episode()=default;
+ Episode(string , int , int);
+ void setSeason(int);
+ int getSeason();
+ void printInfo();
+ void printRating();
Series
int numEpisodes;
int numSeasons;
vector<Episode *> episodes;
+ Series()=default;
+ Series(string , string);
+ void setNumEpisodes(int);
+ void setNumSeasons(int);
+ int getNumEpisodes();
+ int getNumSeasons();
+ float getRating();
+ void addEpisode(Episode *);
+ void printInfo();
+ void printSeason(int);
+ void printEpisodes();
+ void updateRating();
+ void printRating();
StreamingService
vector<Series *> series;
vector<Movie *> movies;
int newestYear;
+ StreamingService();
+ int getNewestYear();
+ void setNewestYear(int);
+ void addSeries(Series *);
+ void addMovie(Movie *);
+ void printMovies();
+ void printSeries();
+ void printAll();
+ void printMoviesByYear();
+ void printMoviesWithRating();
+ void printSeriesWithRating();
+ void printAllWithRating();

La clase Series contiene un vector de Episode para poder administrar y acceder a sus capítulos, por lo cual las dos clases tienen una relación de agregación. La clase Streaming Service tiene dos vectores como atributos, uno de tipo Movie y otro de tipo Series, lo que hace que tenga una relación de agregación con ambas clases.

Ejemplo de ejecución

Creación de instancia de servicio de streaming

```
StreamingService nf;
```

Creación de instancias de películas

```
Movie others("The Others", "Thriller", 104, 2001);
others.rate(9.6);
others.rate(7.8);
nf.addMovie(&others);

Movie fightclub("Fight Club", "Thriller", 151, 1999);
fightclub.rate(8);
fightclub.rate(9.5);
fightclub.rate(9.2);
nf.addMovie(&fightclub);

Movie notebook("The Notebook", "Drama", 123, 2004);
notebook.rate(8.2);
notebook.rate(9.4);
notebook.rate(9.1);
nf.addMovie(&notebook);

Movie chihiro("Spirited Away", "Fantasy", 125, 2001);
chihiro.rate(8.5);
chihiro.rate(9);
chihiro.rate(9.3);
nf.addMovie(&chihiro);

Movie basterds("Inglourious Basterds", "Drama", 153, 2009);
basterds.rate(8.9);
basterds.rate(10);
basterds.rate(9);
nf.addMovie(&basterds);

Movie psycho("Psycho", "Horror", 109, 1960);
psycho.rate(8.7);
psycho.rate(9.6);
psycho.rate(9.5);
nf.addMovie(&psycho);

Movie jojo("Jojo Rabbit", "Drama", 108, 2019);
jojo.rate(9.7);
jojo.rate(9);
nf.addMovie(&jojo);

Movie coraline("Coraline", "Fantasy", 100, 2009);
coraline.rate(8);
coraline.rate(8.6);
nf.addMovie(&coraline);

Movie orfanato("El Orfanato", "Horror", 100, 2007);
orfanato.rate(8.8);
orfanato.rate(8.7);
nf.addMovie(&orfanato);

Movie warrior("Warrior", "Drama", 140, 2011);
warrior.rate(8.8);
warrior.rate(8.7);
nf.addMovie(&warrior);

Movie alice("Alice in Wonderland", "Fantasy", 75, 1951);
alice.rate(8.8);
alice.rate(8.7);
nf.addMovie(&alice);
```

Se crea una película con su nombre, género, duración en minutos y año de estreno. Después se califica dos o tres veces y por último se agrega al servicio de streaming.

En la próxima captura de pantalla se está llamando la función de StreamingService llamada printMoviesWithRating(), la cual imprime el título de las películas seguida del promedio de sus calificaciones.

```
nf.printMoviesWithRating();
```

```
Movies (rating)
-------------
The Others: 8.7
Fight Club: 8.9
The Notebook: 8.9
Spirited Away: 8.93333
Inglourious Basterds: 9.3
Psycho: 9.26667
Jojo Rabbit: 9.35
Coraline: 8.3
El Orfanato: 8.75
Warrior: 8.75
Alice in Wonderland: 8.75
Tarzan: 8.75
Shawshank Redemption: 8.75
The Nightmare Before Christmas: 8.45
-------------
```

También es posible imprimir la información de una película en específico.

```
Movie chihiro("Spirited Away", "Fantasy", 125, 2001);
chihiro.rate(8.5);
chihiro.rate(9);
chihiro.rate(9.3);
nf.addMovie(&chihiro);
chihiro.printInfo();
```

```
Movie: Spirited Away
Genre: Fantasy
Minutes: 125
Year: 2001
Rating: 8.93333
```

Creación de instancias de series y episodios

```
Series goodplace("The Good Place", "Comedy");
nf.addSeries(&goodplace);

Episode fine("Everything Is Fine", 42, 1);
fine.rate(8.4);
fine.rate(9.2);
goodplace.addEpisode(&fine);

Episode flying("Flying", 45, 1);
flying.rate(8.7);
flying.rate(9.1);
goodplace.addEpisode(&flying);

Episode cockroach("Team Cockroach", 41, 2);
cockroach.rate(8.9);
cockroach.rate(9.6);
goodplace.addEpisode(&cockroach);

Episode answer("The Answer", 48, 4);
answer.rate(7.5);
answer.rate(8.4);
goodplace.addEpisode(&answer);
```

Se crea la serie con su nombre y género, y se añade al servicio de streaming. Se crea cada capítulo con su nombre, duración y temporada, se califican y se agregan a la serie a la que pertenecen. También se pueden calificar los episodios una vez que han sido agregados a la serie o calificar la serie directamente.

```
goodplace.printInfo();
goodplace.printEpisodes();
```

Con las líneas de código anteriores se manda a imprimir la información de la serie

```
Series: The Good Place
Genre: Comedy
Seasons: 4
Episodes: 4
Minutes: 176
Rating: 8.80357
```

y todos sus capítulos por temporada.

```
The Good Place, Season 1

Episode: Everything Is Fine
Season: 1
Minutes: 42
Rating: 8.8

Episode: Flying
Season: 1
Minutes: 45
Rating: 8.9

The Good Place, Season 2

Episode: Team Cockroach
Season: 2
Minutes: 41
Rating: 9.25

The Good Place, Season 3
This season doesn't have any episodes yet.

The Good Place, Season 4

Episode: The Answer
Season: 4
Minutes: 48
Rating: 7.95
```

Como ninguno de los capítulos agregados pertenecía a la tercera temporada, aparece un mensaje que dice que esa temporada aún no tiene ningún episodio.

Con la siguiente línea de código podemos mandar a imprimir la lista completa de las series que fueron añadidas al StreamingService nf.

```
nf.printSeries();
```

```
All series:
-------------
The Good Place
Fullmetal Alchemist Brotherhood
Black Mirror
La Casa de Papel
How I Met Your Mother
Dark
El Gran Hotel
Rick and Morty
-------------
```

Cuando una serie, película o episodio no ha recibido ninguna calificación y se imprime su información, aparece el siguiente mensaje.

```
Series: Black Mirror
Genre: Thriller
Seasons: 5
Episodes: 10
Minutes: 560
Rating: This series hasn't been rated yet.
```

Conclusión Personal

El programa que creé cumple muy bien su función aunque creo que podría llevarse mucho más lejos con algunas modificaciones como clases amigas que administraran el servicio de streaming, ordenar las películas o series por calificación o algún menú intuitivo que permitiera al usuario interactuar con naturalidad con el programa. Creo que es un buen comienzo y estoy satisfecha con la implementación de herencia y relaciones de clases, clases abstractas, sobreescritura de métodos y los demás conceptos aprendidos durante el semestre.