

به نام خدا



تمرین سری چهارم

مأده بادان فیروز - ۹۸۲۲۲۰۰۹

مقدمه:

در این تمرین سعی شده تا جایی که ممکن است Netflix شبیه سازی شود. همانند تمرین سری قبل بخش زیادی از ابزار کار برای هندل کردن درخواست‌های کاربر در کلاس‌های دیگر پیاده سازی شده و بخش دیگری هم در main و همان منوی اصلی رسیدگی می‌شود. در این برنامه TVShow به صورت یک کلاس abstract تعریف شده است که فیلم و سریال از آن ارث می‌برند. TVShow هر برنامه‌ی تلویزیونی‌ای در نظر گرفته شده که بتوان دید، مثل فیلم یا سریال. اینجا نمی‌توان instance از TVShow ساخت. بلکه صرفاً یک مفهوم انتزاعی مشترک بین فیلم و سریال است.

همچنین duration به عنوان بازه‌ی زمانی‌ای در نظر گرفته شده که هر قسمت از یک سریال بین چند دقیقه تا چند دقیقه است.

طراحی و پیاده سازی:

User
<ul style="list-style-type: none">- username: String- password: String- favorites: ArrayList<TVShow>- watchHistory: ArrayList<TVShow>- recommendedList: ArrayList<TVShow>
<ul style="list-style-type: none">+ User(username: String, password: String)+ setUsername(username: String): void+ getUsername(): String+ setPassword(password: String): void+ getPassword(): String+ searchByTitle(title: String): ArrayList<TVShow>+ searchByGenre(genre: String): ArrayList<TVShow>+ searchByReleaseYear(year: int): ArrayList<TVShow>+ addToFavoriets(show: TVShow): void+ viewFavoriets(): ArrayList<TVShow>+ watch(show: TVShow): void+ viewWatchHistory(): ArrayList<TVShow>+ recommend(show: TVShow): void+ getRecommendations(): ArrayList<TVShow>+ updateTVShow(mainTVShow: TVShow, updatedTVShow: TVShow): boolean+ toString(): String

Duration
- minValue: int - maxValue: int
+ Duration(minValue: int, maxValue: int) + setMinValue(minValue: int, maxValue: int): void + getMinValue(): int + setMaxValue(maxValue: int): void + getMaxValue(): int + toString(): String

<<Abstract>> TVShow
- title: String - genre: String - rate: double - yearOfPublication: int - generateCountry: String - castList: ArrayList<String> - raters: ArrayList<User> - advisers: ArrayList<User>
+ TVShow(title: String, genre: String, rate: double, yearOfPublication: int, generateCountry: String, castList: ArrayList<String>) + TVShow(show: TVShow) + setTitle(title:String):void + getTitle(): String + setGenre(genre: String): void + getGenre(): String + setRate(rate:double): void + getRate(): double + setYearOfPublication(year: int): void + getYearOfPublication(): int + setGenerteCountry(generateCountry: String): void + getGenerateCountry(): String + setCastList(castList: ArrayList<String>): void + getCastList(): ArrayList<String> + setAdvisers(advisers: ArrayList<User>): void + getAdvisers(): ArrayList<User> + updateRate(newRate: double, rater: User): boolean + recommend(recommender: User): boolean + toString(): String

Movie
<ul style="list-style-type: none"> - lengthInMinutes: int
<ul style="list-style-type: none"> + Movie(title: String, genre: String, rate: double, yearOfPublication: int, lengthInMinutes: int, generateCountry: String, castList: ArrayList<String>) + Movie(movie: Movie) + setLength(length: int): void + getLength(): int + soString(): String

Serial
<ul style="list-style-type: none"> - numOfSeasons: int - episodes: ArrayList<Movie> - duration: Duration
<ul style="list-style-type: none"> + Serial(title: String, genre: String, rate: double, yearOfPublication: int, generateCountry: String, numOfSeasons: int, duration: Duration, castList: ArrayList<String>) + Serial(serial: Serial) + setNumOfSeasons(numOfSeasons: int): void + getNumOfSeasons(): int + setEpisodes(episodes: ArrayList<Movie>): void + getEpisodes(): ArrayList<Movie> + setDuration(duration: Duration): void + getDuration(): Duration + addEpisode(newEpisode: Movie): void + updateDuration(episode: Movie): void + updateCastList(episode: Movie): void + toString(): String

NetflixService
<ul style="list-style-type: none"> - currentUser: User - indexOfCurrentUser: int - users: ArrayList<User> - shows: ArrayList<TVShow>
<ul style="list-style-type: none"> + NetflixService() + getCurrentUser(): User + addShow(show: TVShow): void + createAccount(username: String, password: String): boolean + login(username: String, password: String): boolean + logout(): void + searchByTitle(title: String): ArrayList<TVShow> + searchByGenre(genre: String): ArrayList<TVShow> + searchByReleaseYear(year: int): ArrayList<TVShow> + watch(show: TVShow): void + addToFavoriets(show: TVShow): void + recommend(show: TVShow): void + recommend(serial: Serial, episode: Movie): void + rate(show: TVShow, newRate: double): void + rate(serial: Serial, episode: Movie, newRate: double): void + addEpisode(serial: TVShow, episode: Movie): Serial + updateCurrentUser(updatedUser: User): void + findUser(username: String): User + doesShowExist(show: TVShow): boolean

آنچه نیازمند توضیح است Overload کردن دو تابع rate و recommend در کلاس NetflixService است. همانطور که مشاهده می‌شود در ورژن دوم هر کدام برای کار با یک قسمت از یک سریال ساخته شده در حالی که ورژن اول می‌تواند برای یک فیلم یا یک سریال به خوبی عمل کند. اما برای امتیازدهی به یک قسمت یا پیشنهاد کردن یک قسمت - و اعمال تغییرات لازم و به‌روز کردن لیست‌هایی که شامل سریال مورد نظر هستند - به یک تابع rate جدید و recommend جدید نیاز است. برای همین این دو متد Overload شده‌اند.

سنجش و ارزیابی:

در کلاس Test در قسمتی که recommend امتحان شد با مشکل مواجه شد. برای حل آن با تکه‌تکه کامنت کردن کد و run گرفتن اشکالش پیدا شد. به خاطر این مشکل و حل آن یک commit برای کلاس serial اضافه شد که تابع updateEpisode از آن حذف شد چون دیگر نیازی به آن نبود و حتی باعث ایجاد ارور همان بود. چون در حین اجرای کد، کاری که این تابع انجام می‌داد انجام می‌شد و فراخوانی آن بی‌جا بود.

نتیجه‌گیری:

- از کلاس‌های abstract برای در نظر گرفتن مفاهیم انتزاعی می‌توان بهره برد که از خود این کلاس لازم نیست مثالی ساخته شود. بلکه کلاس‌های دیگر از آن ارث می‌برند و از آن‌ها مثال ساخته می‌شود.
- اگر تمامی متدهای یک کلاس abstract باشند این کلاس دیگر interface می‌شود. که در این تمرین با توجه به رویکردی که من در پیش گرفتم نیاز نشد.