

EXAMEN

Matière: PROGRAMMATION OBJET

Documents: Non autorisés

Enseignants: Mejdoub S., Refai S., Mhafidhi H.

Durée: 1h30'

Classes : DSI2

Date: 13/12/2024

Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.

PROBLEME

PTS

On souhaite créer une application Java qui simule une plateforme de streaming comme Netflix. Cette plateforme gère différents types de contenus (films et séries) et des utilisateurs (avec ou sans abonnement). L'objectif est de permettre aux utilisateurs de lire des contenus en respectant leurs droits d'accès.

Partie 1 : Gestion du contenu

Pour gérer le contenu sur la plateforme, nous aurons besoin des éléments suivants :

- **Interface Streamable** : Cette interface définit les méthodes suivantes :

- **play()** : Démarre la lecture du contenu.
- **getDuration()** : Retourne la durée du contenu en minutes.

```
public interface Streamable {  
  
    void play(); // Méthode pour lire un contenu  
  
    int getDuration(); // Méthode pour récupérer la durée totale d'un contenu  
  
}
```

- **Classe abstraite Content** : Cette classe représente un contenu général sur la plateforme et implémente l'interface Streamable.

- Attributs :

- **title (String)** : Le titre du contenu.
- **genre (String)** : Le genre du contenu.

- **isFree** (boolean) : Indique si le contenu est payant ou gratuit.
 - Constructeur paramétré pour initialiser les attributs.
 - Méthode abstraite **afficheDetails()** pour afficher les détails spécifiques du contenu et sa durée.
 - Implémentation de la méthode **play()** : La méthode play() de la classe Content doit afficher :
- "Lecture de: [type contenu]" afficher les détails du contenu.
- **Classes dérivées Movie et Serie :**
 - Movie : possède un attribut **duration** (durée en minutes).
 - Serie : possède deux attributs : **nbEpisodes** (nombre d'épisodes) et **dureeEpisode** (durée de chaque épisode en minutes).
 - Les deux classes possèdent un constructeur et doivent implémenter les méthodes **afficheDetails()** et **getDuration()** .

1. Écrire le code des classes **Content**, **Movie** et **Serie**

Partie 2 : Gestion des Utilisateurs (Classe User)

- Attributs :
 - **name** (String) : Le nom de l'utilisateur
 - **isSubscribed** (boolean) : true si l'utilisateur a un abonnement, false sinon.
 - **watchedContent** (tableau de content) : l'historique des contenus regardés
 - **nbWatched**(int) : Nombre de contenus regardés
 - **MAX_WATCHED** : la capacité maximale de watchedContent constante fixée à 100.
- Les utilisateurs abonnés peuvent accéder à tous les contenus, y compris les contenus payants.
- Les utilisateurs non abonnés n'ont accès qu'aux contenus gratuits.

La classe **User** inclut :

- Un constructeur pour initialiser ses attributs.

- Une méthode **boolean canPlay(Content content)** : qui prend un objet Content en paramètre et détermine si l'utilisateur peut lire ce contenu.
- Une méthode **addWatchedContent(Content content)** : Cette méthode permet d'ajouter un contenu à l'historique de visionnage du user.
- Une méthode **String[] getWatchedGenres()** : qui retourne un tableau des genres uniques des contenus regardés par un utilisateur.

2. Ecrire le code de la classe **User**

Partie 3 : Plateforme de streaming

- **Gestion des exceptions :**

- Une exception personnalisée **ContentNotFoundException** si un utilisateur tente de lire un contenu inexistant.
- Une exception **AccessDeniedException** si un utilisateur sans abonnement tente de lire un contenu payant.

3. Ecrire le code des classes exceptions

- **Classe StreamingPlatform :**

Cette classe contient les éléments suivants :

- Deux tableaux : un pour les contenus et un pour les utilisateurs.
- Nombre d'utilisateurs effectif et nombre de contenus effectif
- Un constructeur permettant d'initialiser les attributs de la classe en spécifiant la capacité des tableaux des contenus et des utilisateurs.
- Une méthode **addContent(Content content)** permettant d'ajouter un nouveau contenu à la plateforme.
- Une méthode **addUser(User user)** permettant d'ajouter un nouveau utilisateur à la plateforme.
- Une méthode **Content findContent(String title)** qui permet de rechercher un contenu par son titre . Si le contenu n'existe pas, une exception de type **ContentNotFoundException** est levée.

- Une méthode `void playContent(User user, String title)` qui permet à un utilisateur de lire un contenu, avec une vérification des droits d'accès. Une exception de type `AccessDeniedException` est levée si l'accès est refusé.

NB :Ne pas oublier de mettre à jour la liste des contenus regardés par l'utilisateur

- Une méthode `Content[] getRecommendations(User user)` qui renvoie un tableau de recommandations pour l'utilisateur user, en s'appuyant sur les genres des contenus qu'il a visionnés, avec un maximum de 10 recommandations.

Nb .il faut vérifier que le contenu n'a pas été regardé avant de le recommander.

4. Ecrire le code de la classe **StreamingPlatform**

5. Ecrire une méthode **main** qui permet de :

- Créer une plateforme
- Créer, ajouter et afficher 4 contenus
 - `Movie("Inception", "Sci-Fi", false, 148)`
 - `Serie("Stranger Things", "Thriller", false, 25, 50)`
 - `Movie("Charlie Chaplin", "Comedy", true, 75)`
 - `Serie("Friends", "Comedy", true, 236, 22)`
- Créer deux utilisateurs :
 - user1 : nom "Imene", sans abonnement
 - user2 : nom : "jihene", avec abonnement
- Tenter de lire des contenus pour vérifier les droits d'accès des utilisateurs :
 - Imene regarde "Inception" et "Charlie Chaplin"
 - Jihene regarde "Stranger Things" et "Friends"
- Afficher les recommandations pour l'utilisateur "Jihene"
- Gérer les exceptions de manière appropriée.