Dossier Projet AppServJava

POOREEA Fardeen 201
JIANG Janik 202
TRUONG Hubert 202

SOMMAIRE

~

~

1. Le sujet

Une médiathèque avec 3 services :

Le projet consiste à développer une application permettant de gérer les interactions d'une médiathèque avec ses abonnés. Cette application possède plusieurs services :

- Un service de réservation pour réserver un document pendant 2 heures
- Un service d'emprunt pour emprunter des documents disponibles
- Un service de retour afin de gérer les retours des documents.

Les différents types de client pouvant interagir avec le logiciel sont :

- Les postes de la médiathèque qui permettent d'emprunter ou de retourner un document
- Les ordinateurs personnels des adhérents dont l'unique action possible est la réservation

Les ressources partagées :

Tout d'abord, on repère les ressources partagées :

Les instances de Document : Certaines fonctions de Document comme reserver() ont besoin d'être Thread-Safe. Par exemple, un document ne peut pas être réservé par deux personnes en même temps. Si deux instances d'Abonné font cette requête sur un même document, alors ces 2 requêtes doivent être traité l'une après l'autre afin de soulever une exception lors de la 2ème demande de type ReservationException. C'est pourquoi nous avons utilisé des blocs de code synchronized (this) {…} pour verrouiller l'accès concurrent au livre courant grâce à son moniteur de Hoare.

2. Coté Serveur & Coté Client

Coté Serveur :

Coté Serveur, nous avons une classe appli, qui va lancer les services. On a 3 classes Service qui correspondent à la réservation, l'emprunt et le retour. Ces 3 services héritent d'une classe abstraite Service pour éviter la redondance de code. Enfin on a une classe TaskReservation qui va décrire l'action à réaliser si le délai de réservation est dépassé.
Nous avons également utilisé le pattern de conception Singleton afin de pouvoir accéder à l'unique instance de la médiathèque dans l'application.
Pour les documents, on a préféré créer une classe abstraite DocumentAbstrait qui implémentera la classe Document pour que la médiathèque puisse évoluer sans difficulté vers des livres, CD, etc…
La classe DVD hérite donc de la classe abstraite DocumentAbstrait.
Enfin pour les abonnes, on a créé une interface.

Coté Client :

Nous avons créé une classe client qui permet de dialoguer avec le serveur de façon synchrone. Cette classe est appelée par une classe Appli qui prend deux paramètres (Le nom d'host et le numéro du port) pour pouvoir se connecter Socket correspondant aux service voulu.
On a deux classe appli, une appli médiathèque et une autre maison.

3. Certification BretteSoft

« Géronimo » BretteSoft©

La fonctionnalité de Géronimo est de bannir (Interdiction d'emprunt) pendant 1 mois un abonné qui rend un document abimé ou en retard de plus de 2 semaines.
Nous avons donc ajouté des attributs à l'Abonne pour savoir si l'abonne en question est bannis et si oui la date de fin. Nous avons également ajouté la date d'emprunt au document pour comparer avec la date de rendu et un attribut pour savoir si le document est abime.
Enfin on a crée une exception de type EmpruntException qui renvoie un message personnalisé lorsqu'un abonné essaye d'emprunter.

« Sitting bull » BretteSoft©

Sitting Bull permet l'envoi d'un mail en tant qu'alerte lorsqu'un document devient disponible.
Malheureusement, nous n'avons pas pu compléter cette certification en temps. Cependant nous avons effectué des recherches pour l'envoi de mail.

5. Les difficultés & Améliorations

La maintenabilité du code

Il est possible d'effectuer des améliorations au niveau de la maintenabilité. On pense qu'il est possible d'appliquer certains design pattern

Le programmation Client Serveur

En effet, on a du mal avec la compréhension Client Serveur au début.

Le travail en groupe & la motivation

Comme chaque travail en groupe, il faut bien s'organiser. Trouver des horaires qui conviennent aux membres du groupe. On a pris un peu de temps à bien démarrer également dû à un manque de motivation.

Conclusion

En conclusion, ce projet nous a permis de mieux comprendre la programmation Client Serveur.
On a consolidé nos bases sur JAVA.
C'est un bon projet qui nous a permis d'en apprendre davantage.