Projet ArchiLog
Réservation/emprunt/retour médiathèque

On désire gérer les réservations/emprunts/retours de documents pour une médiathèque. Le service ne sera ouvert qu'aux abonnés référencés de la médiathèque (la création de nouveaux abonnés n'est pas au programme de ce projet). Pour simplifier, on suppose que tout fonctionne 24h/24h. Les documents concernés ici sont des DVDs mais cela doit pouvoir évoluer sans difficulté vers des livres, CDs, etc.

La réservation d'un DVD par un abonné est possible à distance (depuis chez l'abonné) si le DVD est disponible en médiathèque ; celui-ci lui est alors réservé (aucun abonné ne peut l'emprunter ni le réserver) et l'abonné doit passer à la médiathèque dans les 2h l'emprunter (sinon la réservation est annulée).
L'emprunt d'un DVD sur place est possible si le DVD est disponible ou réservé à l'abonné qui vient l'emprunter.
Le retour d'un DVD se passe également sur place.

Chaque DVD a un numéro, un titre et un indicateur booléen *adulte*, signifiant, s'il est à *true*, réservé aux plus de 16 ans. Cette caractéristique est spécifique aux DVDs ; elle n'existe ni pour les livres, ni pour les CDs.
Chaque abonné a un numéro, un nom et une date de naissance. Les autres informations sont laissées à votre discrétion.
Lors de la réservation ou de l'emprunt, on doit préciser ces 2 numéros. Lors du retour, le numéro de DVD suffit (vous pouvez ramener un DVD trouvé dans la rue par exemple ou pour le compte de quelqu'un d'autre). Pour simplifier, on suppose que ces numéros sont connus de l'abonné qui réserve/emprunte/rend un document.

Clients et serveurs

Une application serveur générale est lancée en permanence sur un ordinateur de la médiathèque (ou ailleurs) d'adresse IP connue. Cette application attend
- les demandes de **réservation** sur le port 3000,
- les demandes d'**emprunt** sur le port 4000 et
- les **retours** sur le port 5000.

Chacune de ces 3 opérations correspondra donc à une application cliente se connectant au port concerné et échangeant les données avec le service concerné, suivant le modèle logiciel vu en tps. Un client bttp2.0 avec le numéro de port récupéré via les arguments du main ou par lecture d'un fichier de configuration sera bien sur apprécié.

Ces 3 serveurs d'écoute sur 3 ports différents sont contractuels : vous ne pouvez pas choisir une autre solution technique.

Un logiciel client de **réservation** est installé sur l'ordinateur des adhérents (à terme, une version mobile sera envisagée). Lorsqu'un abonné lance ce client chez lui, le service de réservation lui envoie le catalogue et lui demande son numéro d'abonné et le numéro de document qu'il souhaite réserver.
Les logiciels clients d'**emprunt** et de **retour** sont installés sur des bornes dédiées à la médiathèque.

Coté serveur

Les DVDs et les abonnés sont tous matérialisés par des objets créés au lancement du programme à partir d'une base de données SQL. A priori, seules 2 tables sont nécessaires avec un schéma relationnel permettant de savoir qui a emprunté quoi. L'aspect ponctuel de la réservation n'a pas à apparaitre dans cette persistance de données.
L'ajout d'abonnés ou de DVDs n'est pas à faire pour ce projet.
Le reste de l'application se déroule intégralement en mémoire centrale.
En option : si vous envisagez un arrêt/redémarrage du serveur, vous sauvegarderez les données dans leur état actuel à l'arrêt du serveur.
La tentative de réservation ou d'emprunt, si elle échoue, doit donner lieu à un message de refus précis (« ce DVD est déjà emprunté », « vous n'avez pas l'âge pour emprunter ce DVD », « ce livre est réservé jusqu'à 12h25 », etc).

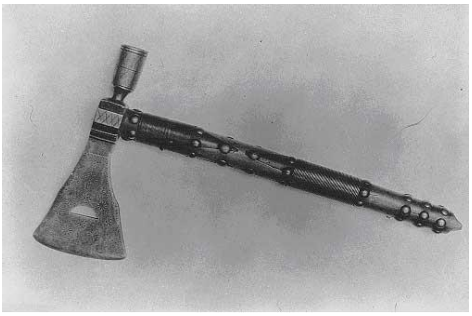
L'interface Document que devra implémenter votre classe DVD

```
public interface Document {
    int numero();

    @return null si pas emprunté ou pas réservé
    Abonne emprunteur() ; // Abonné qui a emprunté ce document
    Abonne reserveur() ; // Abonné qui a réservé ce document

    @pre ni réservé ni emprunté
    void reservationPour(Abonne ab) ;

    @pre libre ou réservé par l'abonné qui vient emprunter
    void empruntPar(Abonne ab);

    @brief retour d'un document ou annulation d'une réservation
    void retour();
}
```

Cnsidérez cette interface comme la clé de voute de votre projet : elle est contractuelle et vous n'avez pas le droit de la modifier

Certifications BretteSoft©

Les certifications **BretteSoft**© (décrites à part) permettent d'intégrer la tribu comme guerrier des plaines; ne faites ça qu'après avoir montré sur le sujet vos capacités à chasser le bison proprement.



A rendre

Le projet est à réaliser par binômes. Aucun trinome ne sera accepté et les monomes seront mal vus du Wakan Tanka correcteur. Les binomes peuvent être inter-parcours.
3 séances de tp de 3h lui sont consacrées en C-7, D-1 et D-2. Un emploi du temps spécifique sera élaboré pour D-1 et D-2 vous permettant de travailler avec votre binôme quel que soit son groupe. C'est malheureusement impossible pour C-7.
La remise est fixée au plus tard le ***dimanche 2 avril 23h59*** *sur le cours moodle : dépôt du dossier sous forme d'UN fichier .zip nommé des noms des membres du groupe (mettez juste NOM_Prenom de chaque membre, ni le groupe, ni ProjetJavaAppServeur, etc…) et contenant le code source (Java source files) et un rapport de présentation pdf de quelques pages dont le contenu sera précisé en tp.*
Aucun retard ne sera accepté (même avec points de pénalités).