# Projet AppServ Java
# Réservation/emprunt/retour bibliothèque

On désire gérer les réservations/emprunts/retours de livres pour une bibliothèque. Le service ne sera ouvert qu'aux abonnés référencés de la bibliothèque (la création de nouveaux abonnés n'est pas au programme de ce projet). Pour simplifier, on suppose que tout fonctionne 24h/24h

La réservation d'un livre est possible à distance si le livre est disponible ; celui-ci est alors réservé pendant 2h et l'abonné doit passer l'emprunter à l'accueil (sinon la réservation est annulée).
L'emprunt d'un livre sur place est possible si le livre est disponible ou réservé à l'abonné qui vient l'emprunter.
Le retour d'un livre se passe également sur place.

Chaque livre a un numéro. Chaque abonné a également un numéro.

Lors de la réservation ou de l'emprunt, on doit préciser ces 2 numéros. Lors du retour, le numéro de livre suffit.

### *Clients et serveurs*

Une application serveur générale est lancée en permanence sur un ordinateur de la bibliothèque (ou ailleurs) d'adresse IP connue. Cette application attend
- les demandes de **réservation** sur le port 2500,
- les demandes d'**emprunt** sur le port 2600 et
- les **retours** sur le port 2700.
-

Chacune de ces 3 opérations correspondra donc à une application cliente se connectant au port concerné et échangeant les données avec le service concerné.

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).

Le(s) logiciel(s)s client(s) d'**emprunt** et de **retour** est (sont) installé(s) sur les postes d'accueil de la bibliothèque.

### *Coté serveur*

Du coté du serveur, les livres et les abonnés sont tous matérialisés par des objets créés au lancement du programme. L'ajout d'abonnés ou de livres n'est pas à prendre en compte.

La tentative de réservation ou d'emprunt, si elle échoue, doit donner lieu à une exception porteuse du motif de refus détaillé (numéro d'abonné ou de document non reconnu, document déjà réservé ou déjà emprunté, etc). C'est la forme affichable de cette exception qui sera renvoyé au client.

## L'interface Document que devra implémenter votre classe Livre

```
public interface Document {
      int numero();
      void reserver(Abonne ab) throws PasLibreException ;
      void emprunter(Abonne ab) throws PasLibreException;
      void retour(); // document rendu ou annulation réservation
}
```

Cette interface est contractuelle et vous n'avez pas le droit de la modifier (hormis pour les certifications optionnelles).

### *Certifications BretteSoft©*

Les options ci-dessous permettent d'obtenir des niveaux de certifications BretteSoft© ; ne vous lancez dans ces options qu'après avoir terminé et sécurisé votre projet sur une copie.

1. (« Papoose éclairé » BretteSoft©)
*Certains abonnés rendent les livres en retard (parfois avec un gros retard) ; d'autres dégradent les livres qu'ils empruntent ; un abonné, suite à un retard de plus de 2 semaines ou à dégradation de livre constatée au retour, sera interdit d'emprunt pendant 1 mois.*

2. (« Grand Chaman » BretteSoft©)
Le modèle de réservation doit permettre, lorsque le livre n'est pas disponible, de placer une alerte nous avertissant par courriel du retour du livre. L'aspect courriel pourra être simulé par l'écriture d'informations sur un fichier log, voire par un simple message, mais la certification suppose l'exploration des bibliothèques de mail java.

### *A rendre*

*Le projet est à réaliser par binômes ou trinômes et à rendre le vendredi 12 janvier minuit par courriel à jean-francois.brettearisdescartes.fr. Aucun retard ne sera accepté (même avec points de pénalités).*

*Le projet est fait pour s'entrainer et assimiler les principes et techniques vues en cours et TPs. Le coefficient est donc faible (1) en comparaison du dst (3) qui est un contrôle individuel du niveau.*