

## **Projet Java Objet**

### **Gestion des prêts d'une bibliothèque**

\*\*\*\*\*

On désire gérer informatiquement les prêts d'une bibliothèque (sorties, retours avec amende éventuelle). Les abonnés font des emprunts ; ces emprunts concernent des objets empruntables (livres, cds et vidéos).

#### **Les objets empruntables**

La durée d'emprunt est de 3 semaines pour les livres, 3 semaines pour les CDs, 1 semaine pour les vidéos.

Au retour, en cas de dépassement, une amende d'1,5 euro par semaine révolue et par objet est appliquée dans tous les cas.

Un abonné peut emprunter au maximum 6 livres, 8 cds et 3 vidéos.

Les CDs sont éventuellement accompagnés d'un livret et les vidéos ont un marqueur *adulte* (pour les vidéos interdites aux moins de 16 ans). Si le livret manque au retour d'un CD, le retour est tout simplement refusé.

Une interface Empruntable contiendra les spécifications des objets empruntables. Elle sera associée à une classe EmpruntableManager.

#### **Les abonnés**

Une classe Abonné définit les abonnés de la bibliothèque. Un abonné a un numéro, un nom, un prénom, une adresse, une date de naissance, une date limite d'abonnement et un type d'abonnement. Le type d'abonnement indique pour quels objets l'abonné a payé son abonnement :

0 : livres seuls

1 : livres+ CDs

2 : livres+CDs+vidéos

#### **Les emprunts**

Un emprunt est défini par sa date d'emprunt, sa date de retour maximum, sa date de retour effective. Un emprunt fait bien sur le lien entre un abonné et un objet empruntable.

#### **Le lecteur de code barre**

Les abonnés possèdent une carte pourvue d'un code barre. Les objets empruntables ont également un code barre sur la couverture. Un lecteur de code barre, pour les cartes d'abonnés comme pour les objets empruntables, permet d'obtenir le numéro associé. Une classe LecteurCodeBarre du package *mesoutils* doit simuler ce lecteur, sa seule méthode publique est

*public static int litNumero( ) throws LectureImpossibleException*

Lorsqu'un abonné emprunte, il donne sa carte puis successivement tous les objets qu'il souhaite emprunter.

Lorsqu'un abonné rapporte un objet, on passe simplement l'objet au lecteur de code barre.

### **L'application GestionBiblio et son menu**

L'application principale s'articule autour d'un menu dans le main qui permet de gérer toute la bibliothèque, menu contenant les options suivantes :

- l'emprunt d'un objet
- son retour
- l'inscription d'un abonné
- des affichages des abonnés et des objets empruntables (avec critères de sélection)
- toutes autres suggestions

La création initiale des objets empruntables sera le fruit de la lecture d'un fichier préalablement créé (voir le package objetsbiblio plus loin).

Chacune de ces options doit être écrites dans une méthode de classe de la classe GestionBiblio, par exemple *emprunt* ( ) et *retour* ( ) pour les 2 premières. Ces méthodes n'ont pas de paramètres ni de résultat mais peuvent rejeter des erreurs en cas d'incidents (EmpruntImpossibleException, RetourImpossibleException, AmendeException, etc).

### **Les packages**

Les différentes classes et interfaces sont réparties dans 4 packages :

package *empruntable* : l'interface Empruntable, la classe EmpruntableManager, et tout autre type (avec son manager) nécessaire à ce package qui fait le lien entre les 2 suivants ; ce package doit être compilable sans les 2 suivants.

package *objetsbiblio* : les classes des objets empruntables, un main permettant la création d'un fichier pour l'initialisation du main de *GestionBiblio*.

package *appli* : la classe GestionBiblio et les autres classes nécessaires.

package *mesoutils* : les outils comme Keyboard, LecteurCodeBarre (qui sera bien sur une simulation clavier)

Il est évident que la bibliothèque devra pouvoir ajouter de nouvelles catégories d'objets (jeux pour une ludothèque par exemple), la maintenance de votre logiciel devant être la plus simple possible. Idem si le nombre de types d'abonnements augmente. Une classe Constantes regroupant les constantes de l'application (NB\_CATEG\_EMPR, NB\_TYP\_ABTS,...) pourra éventuellement s'avérer utile.

Le rapport du projet (entre 3 et 5 pages) devra présenter les points importants en rapport avec les objectifs de l'enseignement IL1, donc pas de rapport orienté « Notice d'utilisation du logiciel ».

\*\*\*\*\*