TD3 : UML, patterns création

1 Représentation UML

Le but de cet exercise est de modéliser un système d'achat en ligne de type Amazon réalisé avec J2EE.

Tout d'abord, télécharger le fichier td1amazon.zip. et intégrer le à votre projet courant

Nous utiliserons l'éditeur UML Violet (très simple et autonome) ou celui intégré à Eclipse, ou argoUML....

1. Faire le diagramme de classe avec simplement les classes, leurs attributs et méthodes.
2. Maintenant, vous allez modifier le diagramme. Faire apparaître les relations d'héritage sur le diagramme: CD, DVD & Book héritent de Item.
3. La prochaine étape va être de créer des associations. Modéliser le fait qu'un Kart ne puisse avoir qu'un client, qui ne doit pas forcément être renseigné au départ.
   Regarder maintenant les apports à la classe Kart. Faites apparaître le champs Customer sur le diagramme Kart.
4. Ajouter les associations suivantes avec les cardinalités appropriées en fonction des indications ci dessous :
   - Un Kart peut avoir plusieurs Item. Ajouter une association en utilisant une ArrayList.
   - Ajouter une classe abstraite Product commune à CD, DVD et Book.
   - Un Product doit avoir un Editor. Un Editor peut avoir plusieurs Product.
   - Un Book doit avoir un ou plusieurs Author. Un Author peut avoir écrit plusieurs Book.
   - Un Book peut avoir une BookCover
   - Un Dvd possède plusieurs Languages.

Exercice 2 – Un objet ... un seul !

On veut avoir un pool de connexions à une DB.

```
Class DBConnexionPool {
	private List<DBConnexion> connexions = new ArrayList<DBConnexion>();
	public void register( DBConnexion connexion) {
		connexions.add(connexion);
	}
	public synchronized DBConnexion getConnexion() {
		for ( DBConnexion c : connexions) {
			if ( ! c.isBusy) {
				c.setBusy();
				return c;
			}
		}
		DBConnexion c = new DBConnexion();
		c.setBusy();
		connexions.add(c);
	}
	public synchronized void releaseConnexion(DBConnexion c) {
		c.setFree();
	}
}
```

Quel est le problème si chaque utilisateur de cette classe crée une instance ?
Comment faire de cette classe un *Singleton* ?

La solution intuitive est-elle thread-safe ?
donnez 3 solutions différentes pour la rendre thread-safe ?
Laquelle ou lesquelles préconisez-vous ?

Pourquoi ne pas juste rendre la classe totalement statique ?

Exercice 3 – (Ex 8 / TD précédent)

La version de base du jeu est livrée, le CD est gravé.

Le jeu a été prévu pour pouvoir supporter des extensions, vendues séparément par des éditeurs indépendants :

- des nouveaux canards
- des nouvelles manières de voler, de cancaner, ...

Comment ?