

# Cours Objet – TP2 – [www.laFac.com](http://www.laFac.com)

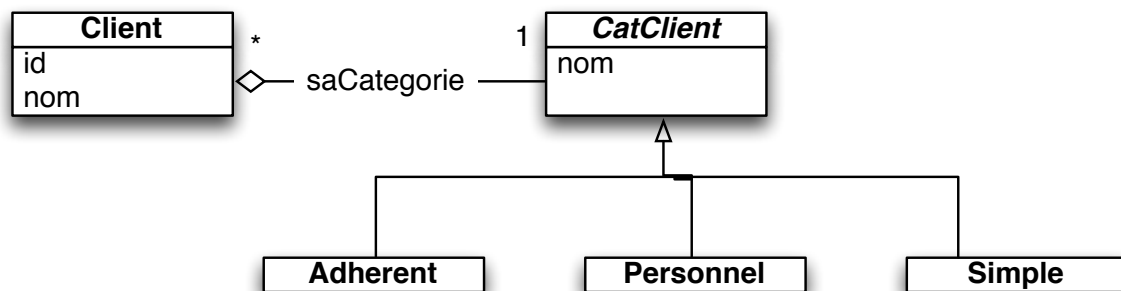
## Explication de la hiérarchie de classes

Le diagramme de classes complet se trouve en page [suivante](#).

Dans le texte nous avons tout d'abord reconnu le design pattern « State », en effet un client peut changer dynamiquement de catégorie, il fallait donc une solution permettant cette demande. Comme vu en TD ce design pattern le permet.

Nous avons donc la forme suivante pour modéliser la catégorie d'un client :

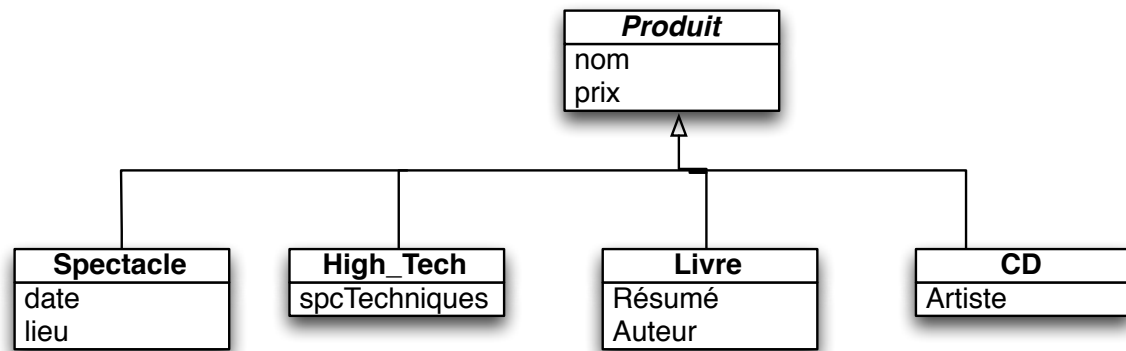
"State"



Nous avons ensuite constaté qu'il était possible de faire des classes **Personnel** et **Simple** des singletons. Ces dernières n'ont besoin d'êtreinstanciées qu'une seule fois. Par contre nous avons fait le choix de ne pas appliquer cela à la classe **Adherent**. Ceci permettant de gérer les points de fidélité facilement.

Concernant la gestion du **Panier**, nous avons choisi de créer une classe **Panier**, le contenu de celui-ci est enregistré dans une **HashMap** ayant pour clé le produit et pour valeur la quantité du produit associé. La classe **Panier** contient des méthodes permettant de calculer le prix du panier. Des que l'on ajoute ou retire un produit on calcule à nouveau l'ensemble des réductions applicables au panier.

Nous avons créé une classe abstraite pour produit, toutes les catégories héritent de celle-ci. Ce choix nous oblige à supprimer un produit puis le recréer si nous souhaitons le changer de catégorie, ce qui n'est pas trop important puisque mise à part une erreur de saisie un produit de devrait pas changer de catégorie. Cette option permet en outre d'ajouter facilement de nouvelles catégories.



Pour les promotions nous avons opté pour la même solution.

## Diagrammes de séquence

## Diagramme des classes

