

Associations bi-directionnelles et dérivées

On s'intéresse ici à quelques éléments d'un logiciel destinés à gérer les prêts d'ordinateurs portables aux étudiants¹.

Exercice I Association bi-directionnelle, gestion de la cohérence

Le centre de gestion des prêts maintient la liste des ordinateurs portables pouvant être prêtés. Chaque ordinateur est caractérisé par son numéro d'inventaire et sa marque pour le retrouver plus facilement dans l'armoire. Chaque ordinateur peut être prêté à un étudiant, et depuis une instance d'ordinateur, on peut obtenir directement l'instance d'étudiant correspondant à l'emprunteur. Un étudiant est décrit par son nom (pour simplifier). Un étudiant peut emprunter au plus un ordinateur à la fois. On souhaite avoir un accès direct à l'ordinateur emprunté depuis une instance d'étudiant.

Question I.1. Modélisez la situation décrite ci-dessus, en ajoutant le nécessaire pour y intégrer les comportements suivants :

- Depuis le gestionnaire de prêt, on peut gérer la liste des ordinateurs (en ajouter en refusant les doublons, ou en supprimer).
- Depuis le gestionnaire de prêt, on peut rechercher la liste des ordinateurs portables satisfaisant les deux critères paramétrables suivants : la marque de l'ordinateur (recherche des ordinateurs d'une marque donnée, marque "all" si la marque est indifférente) et le fait d'être disponible à l'emprunt ou pas (recherche des ordinateurs empruntés ou au contraire disponibles).
- Depuis le gestionnaire de prêt, on peut déclencher l'emprunt d'un ordinateur donné par un étudiant donné. Cet emprunt échoue si l'ordinateur est déjà emprunté, ou si l'étudiant a déjà un prêt d'ordinateur en cours (les échecs se matérialiseront par un booléen à faux comme retour de méthode, alors que les succès se matérialisent par un booléen à vrai).
- Depuis le gestionnaire de prêt, on peut clore un emprunt (quand l'étudiant rend l'ordinateur). L'ordinateur est de nouveau empruntable et l'étudiant peut de nouveau emprunter.

Question I.2. Implémentez la modélisation de la question précédente, en prenant bien soin de faire en sorte qu'il ne soit pas possible d'arriver à des situations aberrantes comme un étudiant e1 qui aurait emprunté un ordinateur o1 alors qu'o1 pense être emprunté par un autre étudiant e2.

Question I.3. Il est fort utile de garder un historique des étudiants ayant emprunté un ordinateur (par exemple pour le contacter s'il y a laissé des données).

a- Modélisez cet historique.

b- S'agit-il d'une association dérivée ?

c- Ajoutez à votre code précédent le nécessaire pour la mise en place de l'historique.

Question I.4. On souhaite maintenir depuis le gestionnaire de prêt un accès à tous les étudiants du département informatique. Peut-on modéliser cela avec une association dérivée ? (pas d'implémentation ici)

Exercice II Associations dérivées : implémentation par stockage ou par calcul

Question II.1. On souhaite que le gestionnaire de prêt puisse facilement accéder aux étudiants empruntant actuellement un ordinateur portable. Modélisez cela avec une association dérivée.

Question II.2. Nous allons implémenter cette association de deux façons différentes : l'une en stockant avec redondance l'information, et l'autre en la calculant à la demande. Dupliquez votre code Java actuel : placez chacune des versions dans un package distinct. Implémentez la version "stockage" dans un des packages et la version "calcul" dans l'autre. Pour la version avec redondance, vous veillerez à ce que l'information soit toujours cohérente.

1. Ces prêts existent vraiment au département informatique mais concernent prioritairement les étudiants en master

Exercice III *Vers les classes d'association*

On souhaite finalement conserver pour chaque prêt la date d'emprunt, ainsi qu'une url vers le fichier de contrat de prêt (une chaîne de caractère simple pour simplifier).

Question III.1. Comment modifier la modélisation ?

Question III.2. Même question si on permet finalement qu'un étudiant emprunte plusieurs ordinateurs en même temps.

Question III.3. Même question si on permet le prêt partagé : on prête un même ordinateur à plusieurs étudiants (pour un projet commun).