NFA032 : TP n°8 (Références)

10 avril 2018

Objectifs et rappels

L'objectif de ce Tp est de travailler sur une application qui exploite la posibilité d'avoir accès à un même objet via différentes variables ou *partage de données*. Prenons l'exemple d'un objet Compte de Java : il est représenté par son adresse mémoire, dite *référence* (ou pointeur). Si cette référence est contenue dans plusieurs variables (par exemple, plusieurs cases de tableau), on aura accès à l'objet compte (pouvant le voir/modifier) en passant par l'une ou l'autre de ces variables. La figure 1 montre un code où des objets sont partagés par les cases d'un tableau et par plusieurs variables, ainsi qu'un dessin où les flèches (représentant les pointeurs) illustrent ce partage. Etudiez ce code et son dessin en vous assurant de bien comprendre le pourquoi des flèches représentées.

Exercice 1 : Banque, Comptes et Titulaires

Pour réaliser cet exercice recopiez dans votre projet le paquetage `sourcesExo1`.

On veut travailler sur la modélisation d'une banque et de titulaires de compte détenus par cette banque. Une banque possède une liste de comptes. Un titulaire possède un nom et tous les comptes dont il est titulaire. On voit donc qu'un même objet compte peut être à la fois dans la liste de la banque, et dans celle d'un titulaire.

Un même compte peut avoir plusieurs titulaires (c'est le cas par exemple des comptes joints pour un couple). Pour un compte joint, le même objet sera partagé entre tous les objets titulaires de ce compte. Par exemple, l'objet compte correspondant au compte joint de titulaires Paul et Fatima se trouvera dans la liste de comptes du titulaire Paul, dans les comptes du titulaire Fatima et aussi dans les comptes de la banque. Nous vous donnons du code à compléter pour les classes Banque, Titulaire ainsi qu'une classe Compte finalisée. Quelques brèves explications sur ces classes suivent. **Attention : il pourra être utile ou nécessaire d'ajouter d'autres méthodes dans ces classes**.

Titulaire

Un titulaire possède un nom et un arraylist permettant de stocker ses comptes ainsi qu'une méthode permettant d'y ajouter un nouveau compte. Il peut réaliser sur ses comptes toutes les opérations usuelles sur un compte.

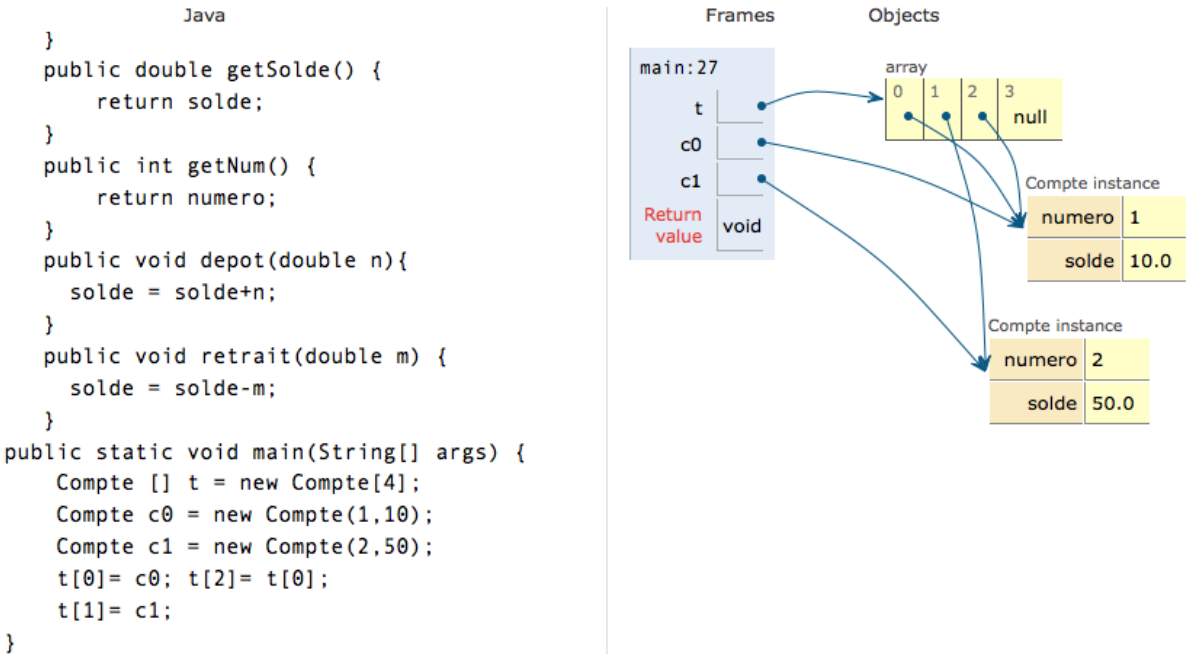

FIGURE 1 – un dessin avec références

Banque

Les comptes ont tous un numéro unique et seule la banque peut créer des nouveaux comptes : en particulier, le constructeur Compte *est non public*. Quand la banque crée un nouveau compte, elle l'ajoute à sa liste de comptes, mais aussi dans la liste de comptes de chacun des titulaires de ce compte. De même, chaque nouveau compte se voit attribuer par la banque un nouveau numéro, exclusif pour ce compte. Pour réaliser la création de comptes, il faut une méthode dans Banque qui crée un compte avec en paramètre la liste des titulaires (par exemple sous forme de tableau), et qui lui attribue un numéro de compte (unique). La méthode retournera le numéro attribué au nouveau compte.

Les méthodes de la banque devant agir sur un compte particulier prennent en paramètre le numéro de ce compte. La Banque devra implanter une méthode pour effectuer un dépôt sur un numéro de compte, une méthode pour obtenir le bilan de soldes de tous ses comptes, et une méthode pour tester si un numéro correspond à un compte de la banque. Il n'y aura pas de méthode permettant de faire de retrait : donner le numéro de compte ne suffit pas pour autoriser un retrait : il faudrait montrer que c'est le titulaire qui fait le retrait. Or, la banque ne possédant pas de registre de titulaires, n'a pas moyen de le vérifier. Enfin, *aucune méthode publique ne doit retourner en résultat un objet Compte*. Cela permet d'assurer que seuls les titulaires et la banque pourront les voir/modifier.

Question 1 : étude du code fourni + dessin

Le code fourni est incomplet et donc ne fonctionne pas correctement. Le but de cette première question est de comprendre ce code avant de le modifier ou de l'exécuter.

Ecrivez une classe `TesterOpsBanque` avec une méthode `main` qui réalise une situation où trois titulaires, Paul, Pierre et Fatima ont des comptes dans une banque *BNP*. Paul et Fatima ont un compte joint. Fatima a en plus un compte personnel, de même que Pierre. Dessinez la situation en faisant apparaître sur un schéma tous les objets et tableaux en vous inspirant du dessin de la figure 1.

Question 2 : compléter le code fourni

Complétez le code des classes données en respectant les commentaires mis dans le code. Il peut être utile d'ajouter de nouvelles méthodes dans ces classes. *Conseil : commencez pas modifier les classes avec le moins de dépendances vis-à-vis d'autres classes non encore terminées*. Ici, cela revient à commencer par la classe Titulaire.

Question 3 : opérations sur les comptes d'un titulaire

Si ce n'est pas encore fait ajoutez les méthodes nécessaires pour permettre à un titulaire d'effectuer des opérations de dépôt, de retrait et d'obtention du solde sur un de ses comptes à partir de son numéro.

Question 4 : virements

On désire écrire deux méthodes de virement. La première effectue un virement depuis un compte dont on est titulaire vers un compte dont on ne connaît que le numéro (et dont on n'est pas forcément titulaire), et dont on connaît la banque qui possède ce compte. La deuxième méthode est une version surchargée de la première, qui effectue un virement entre deux comptes d'un même titulaire. Dans quelle classe faut-il placer ces méthodes ?

Question 5

Ici, il s'agit de comprendre les conséquences de l'organisation du code donnée. Les situations examinées sont hypothétiques : rien est à modifier dans le code développé. Répondez aux questions suivantes sous forme de commentaires dans la classe Banque :

1. Créez un nouveau paquetage `essaiBanque`. Ajoutez dedans une classe `TesterOpsBanque2` avec une méthode main. Peut-on créer un objet compte dans cette classe ? Expliquez pourquoi.
2. Supposons (1) le constructeur de Compte est délcaré public ; (2) dans la classe Banque une méthode permet d'obtenir le compte associé à un numéro de compte. Supposons maintenant qu'on a réussi à obtenir le numéro d'un compte dont on n'est pas titulaire. Peut-on écrire un code qui syphone ce compte ? Comment ?

A rendre

La totalité des exercices de ce Tp (sauf les dessins) sont à rendre en séance, au format zip.