

TP2

Novembre 2023

1 Société Géniale!!!

L'objectif de ce TP est de réaliser en Java une application décrivant de façon rudimentaire le fonctionnement d'une banque.

La banque gère des comptes. Chaque compte possède un numéro unique décerné lors de la création du compte. Chaque compte dispose également d'un solde pour représenter la somme d'argent disponible sur ce compte (lorsque le solde sera négatif on parlera de découvert). On doit pouvoir effectuer des opérations bancaires de base comme créditer ou débiter un compte (dans ce cas, on tiendra compte alors de la valeur du découvert autorisé) ou encore opérer un virement entre deux comptes.

On va considérer que la banque gère trois types de compte :

- les comptes de personnes physiques (*i.e.* vous et moi). Un compte de personne physique contient le nom et le prénom du détenteur. La valeur du découvert autorisé est définie lors de la création du compte comme étant égale à l'apport initial de la personne. Cette valeur de découvert pourra être modifiée selon la politique commerciale de la banque.
- les comptes d'entreprises. Un compte d'entreprise est défini par le nom de l'entreprise. Par ailleurs, le découvert autorisé par la banque pour ce type de compte est nul.
- les comptes anonymes. Ce sont des comptes comparables aux comptes de personnes physiques mais ils ne dévoilent jamais le nom du détenteur du compte. Seul le numéro de compte peut être affiché.

Pour représenter une banque vous allez définir une classe ayant comme attribut une liste de comptes (vous utiliserez la classe `ArrayList`, cf. documentation *Java*) et une chaîne de caractères pour stocker le nom de la banque.

Il vous est demandé d'utiliser systématiquement les commentaires à la *javadoc* pour documenter vos classes (cf. TP1).

Question 1. Définir une hiérarchie au niveau des classes de comptes. Proposer un schéma UML pour cette hiérarchie en faisant figurer les principales méthodes et les attributs de ces classes.

Question 2. Programmer chacune des quatre classes de type `Compte` (la méthode de virement bancaire sera faite après la réalisation de la classe `Banque`). Vous prendrez soin de redéfinir pour chaque classe la méthode `toString()`. Etablissez un chaînage des constructeurs entre chacune des classes. N'oubliez pas de faire les méthodes d'accès si nécessaire pour les attributs privés. Testez chacune de vos méthodes au fur et à mesure dans une méthode `main` au sein de la classe en cours de développement par exemple.

Question 3 Une fois que vous avez réalisé vos classes, programmez dans la méthode `main` une boucle permettant d'initialiser aléatoirement un tableau de comptes de chaque type. Pour les noms fournis au constructeur lors de l'initialisation, on utilisera par exemple la concaténation du prénom *Louis* avec un entier (1,2,...).

Question 4. Définir la classe `Banque` permettant de réaliser et de gérer les opérations de gestion des comptes. Vous devez implémenter les méthodes qui réalisent les tâches suivantes :

- Ouverture d'un compte auprès de la banque (*i.e.* une méthode qui ajoute un compte à la liste des comptes de la banque).
- Suppression d'un compte en fonction de son numéro de compte (il faudra donc réaliser au préalable une méthode qui recherche un tel compte dans liste des comptes gérés par la banque).
- Affichage de tous les comptes de la banque. Pour cela vous redéfinirez la méthode `toString()`.

Question 4. Réaliser au sein de la classe `Banque` un algorithme de tri par sélection qui triera les comptes en fonction de leur solde (tri décroissant). Le tri par sélection consiste à rechercher le plus grand élément d'une liste non triée et à le mettre en queue de la liste triée en construction. Il vous est fortement conseillé de décomposer l'algorithme du tri par sélection en plusieurs méthodes implémentant des tâches plus élémentaires.

Question 5. Faire l'affichage des comptes d'un certain type (*i.e.* compte de personne physique, compte d'entreprise ou compte anonyme).

Question 6. Ajouter à la classe `Compte` une méthode qui réalise un virement bancaire. Cette méthode prendra en argument le numéro du compte bénéficiaire, la banque et le montant. On prendra soin de traiter les cas d'erreur (*i.e.* compte recherché inexistant).