

E2I4 (option H.F) TP Java n°3

La Banque Funkelfraude installée en Fiscal Paradise Island vous contacte pour développer un programme de gestion de la Banque. Le cahier des charges est le suivant :

La Banque gère un ensemble de comptes. Chaque Compte possède un numéro unique (attribué à sa création), une solde représentant la somme d'argent déposée (ou due à la Banque en cas de découvert). On doit pouvoir bien sûr y déposer et retirer de l'argent (uniquement en euros) et consulter le solde. En cas de retrait, la somme maximum qui est donnée effectivement est égale au solde du compte, majorée d'un découvert maximum. Il existent trois sortes de comptes : les comptes de personnes physiques, les comptes des entreprises , et les comptes numérotés. Les comptes de personnes physiques contiennent l'information du nom et prénom du propriétaire. Elles n'ont pas droit au découvert. Les comptes des entreprises contiennent le nom de l'entreprise (raison sociale) et ont droit à un découvert maximum fixé à la création du Compte (égal au montant du premier dépôt), mais qui peut évoluer selon la politique de la Banque. Finalement les comptes numérotés qui ne dévoilent jamais le nom du propriétaire (mais qui est toujours connu par la Banque) mais affichent un numéro unique d'identification qui est supérieur à 100000. Ces dernières n'ont pas droit à un découvert.

1 Diagramme UML

A l'aide du logiciel umbrello faire un diagramme UML des classes comprenant celles qui vous paraissent pertinentes pour implémenter les différents comptes dont une classe de base nommée Compte. Montrer votre diagramme à l'enseignant pour validation. Après cette validation vous devez :

- Générer le code correspondant avec umbrello.
- Créer avec netbeans un nouveau projet d'application java.
- Copier les sources produites dans le répertoire des sources du projet.
- Rectifier les sources pour les inclure correctement dans le paquet correspondant du projet.

2 Codage

1. Coder la classe Compte qui décrit un compte de base. Coder dans un premier temps seulement les méthodes strictement nécessaires. Une signature commentée sera suffisante pour les autres méthodes imaginées. N'oubliez pas la méthode toString!

2. Écrire la classe qui décrit un compte de personne physique, et codez les méthodes nécessaires. Écrire une classe d'application qui fait les opérations suivantes :
 - Crée un compte pour M. René Lepauvre avec 200 euros et l'affiche
 - M. Lepauvre dépose 50 euros puis on affiche l'état de son compte.
 - M. Lepauvre retire 200 euros et on affiche l'état de son compte.
 - M. Lepauvre retire encore 100 euros et on affiche l'état de son compte.
3. Écrire la classe qui décrit un compte d'entreprise. Ajouter dans la classe d'application les opérations suivantes :
 - On crée un compte pour l'entreprise Bétons Bretons avec 20000 euros. L'état du compte est affiché.
 - Bétons Bretons dépose 2000 euros, puis on affiche l'état du compte.
 - Bétons Bretons retire 25000 euros, et on affiche le (piteux) état du compte.
4. Écrire la classe correspondante aux comptes numérotées, puis ajouter dans la classe d'application les opérations suivantes
 - Créer un compte numérotée pour M. Hosni Benali contenant 1000000 euros.
 - Hosni Benali dépose 2000000 euros et retire 3200000 euros en affichant l'état à chaque transaction.
5. Écrire la classe qui décrit la Banque (qui peut être la classe d'application) avec la gestion des comptes de tous types. On doit pouvoir créer et détruire des comptes. Utilisez un **ArrayList** après avoir étudié la documentation.
 - (a) Créer 5 comptes (dont au moins un de chaque type) et afficher l'état de la Banque (la totalité des comptes).
 - (b) Détruire le troisième compte et afficher l'état de la Banque.
 - (c) Cherchez le compte correspondant à un nom entré au clavier et afficher son contenu.
 - (d) Calculez la quantité d'argent contenu dans les coffres de la banque.