

Programmation objet

Objectif : Comprendre au travers d'un exemple simplifié de comptes bancaires, les notions de propriétés d'instances et de classes, d'héritage, de redéfinition de méthodes, de polymorphisme et de liaison dynamique.

Exercice 1 : Compte bancaire simple

Nous nous intéressons à un compte simple caractérisé par un solde exprimé en euros, positif ou négatif. Il est possible de créditer ce compte ou de le débiter d'un certain montant.

- 1.1 Spécifier en UML la classe `CompteSimple`.
- 1.2 Proposer des tests de la classe `CompteSimple`.
- 1.3 Écrire la classe `CompteSimple`.

Exercice 2 : Compte courant

En fait, une banque conserve, pour chaque compte, l'historique des opérations qui le concernent (on se limite ici aux opérations de crédit et de débit). On souhaite modéliser un tel compte qu'on appelle compte courant. En plus des méthodes d'un compte simple, un compte courant offre des méthodes pour afficher l'ensemble des opérations effectuées (`afficherRelevé`) ou seulement les opérations de crédit (`afficherRelevéCrédits`) ou de débit (`afficherRelevéDébits`).

Pour représenter l'historique, on utilisera une liste. Pour enregistrer une opération, on conservera simplement le montant de l'opération précédé d'un signe indiquant s'il s'agit d'un crédit (montant positif) ou d'un débit (montant négatif).

- 2.1 Indiquer quelle est la relation entre un compte simple et un compte courant et compléter le diagramme de classes UML.
- 2.2 Écrire la classe `CompteCourant` correspondant à un compte courant.
- 2.3 Écrire un sous-programme (un test par exemple), qui montre que les opérations sont bien enregistrées dans l'historique.

Exercice 3 : La banque

Une banque est bien entendu un organisme qui gère un grand nombre de comptes, qu'ils soient simples ou courants.

- 3.1 *État de la classe Banque.* On choisit de stocker tous les comptes dans une unique liste. Indiquer quels sont les attributs de la classe `Banque`. Écrire le constructeur de la classe `Banque`.
- 3.2 *Ouverture des comptes.* Définir sur la classe `Banque` une méthode pour ouvrir un compte simple et une méthode pour ouvrir un compte courant.
- 3.3 *Cumul des soldes des comptes.* Définir une méthode sur la banque qui donne le cumul des soldes disponibles sur chacun des comptes. D'autres informations pourraient être fournies telles que le nombre de comptes débiteurs, la somme des débits, etc.

3.4 Frais de tenue de compte. Nous considérons que la banque prélève périodiquement des frais de tenue de compte. Écrire une méthode qui débite sur tous les comptes la somme de 2 euros.

3.5 Édition des relevés. Écrire une opération qui édite (à l'écran) les relevés de tous les comptes courants (et seulement des comptes courants, le relevé n'a pas de sens pour un compte simple).

3.6 Autre organisation des comptes. Proposer une autre organisation des comptes qui faciliterait l'édition des relevés. Discuter cette solution.

Exercice 4 : Numéro de compte

Pour les différencier, chaque compte possède un numéro unique. On suppose que les numéros sont des entiers et qu'ils sont attribués par ordre croissant en commençant à 10001. Les numéros de compte sont donc 10001, 10002, 10003, etc. Dans la suite, nous envisageons deux solutions pour attribuer les numéros de compte.

4.1 On souhaite que l'attribution du numéro de compte soit de la responsabilité des classes `CompteSimple` et `CompteCourant`. Indiquer les modifications à apporter à ces classes.

4.2 On suppose maintenant que c'est la banque qui gère et attribue les numéros de compte. Indiquer les modifications à apporter aux classes `CompteSimple` et `CompteCourant`.