

TRAVAUX PRATIQUES : série n°2

Implémentation du type abstrait des comptes bancaires

L'objet de ce TP est de fournir une implémentation **correcte** pour le type abstrait des comptes bancaires en répondant au cahier des charges préparé par un responsables du secteur Gestion Clients d'une banque.

Le travail doit **obligatoirement** se dérouler en suivant les **4** étapes du cycle de développement initiées lors de la série n°1 (type abstrait des polynômes), à savoir :

- étape 1: **Spécification** du type abstrait (une spécification est jointe au présent document)
- étape 2: **Validation** de la spécification sous **hets**.(utiliser hets en ligne :**DOLiator**)
- étape 3: **Implémentation** de la spécification.
- étape 4: **Vérification** de l'implémentation.

I-CAHIER DES CHARGES

Le responsable du secteur Gestion Clients d'une banque souhaite lancer un projet pour proposer aux clients certaines fonctionnalités en ligne dont l'exécution est contrôlée en temps réel par un robot superviseur.

Dans une première phase du projet, les fonctionnalités retenues sont les suivantes :

- **créer** un compte bancaire : le solde d'un compte créé est nul; un compte créé n'est pas dans l'état de compte clôturé
- **débiter** un compte,
- **créditer** un compte,
- **clôturer** un compte: un compte ne peut être clôturé que si son solde est nul.
- connaître le nom de son **titulaire**: un même **titulaire** ne peut posséder plusieurs comptes,
- chercher à savoir son **solde**,
- tester si un compte **est clôturé**.

Un compte ne peut être clôturé que si son solde est nul et qu'il n'est pas déjà clôturé. En outre, un compte clôturé ne peut être ni débité ni crédité.

Le titulaire d'un compte ne peut être modifié lorsqu'on exécute les opérations clôturer, débiter ou créditer un compte.

II-SPECIFICATION EN CASL

```
library libraryCompte

from Basic/Numbers get Rat
from Basic/SimpleDatatypes get Boolean
from Basic/CharactersAndStrings get String

spec COMPTE =
  Rat
and Boolean
and String

then sort Compte
pred estCloture : Compte
ops
  creer : Compte;
  creditor : Compte * Rat ->? Compte;
  debiter : Compte * Rat ->? Compte;
  cloturer : Compte ->? Compte;
  titulaire : Compte -> String;
  solde : Compte -> Rat;

forall c1, c2 : Compte; s1 : Rat; p1: String
. def creditor(c1, s1) <=> not estCloture(c1)
. def debiter(c1, s1) <=> not estCloture(c1)
. def cloturer(c1) <=> not estCloture(c1) /\ solde(c1)=0
```

```
. not estCloture(creer)
. solde(creer) = 0

. estCloture(cloturer(c1))
. solde(cloturer(c1)) = 0
. titulaire(cloturer(c1)) = titulaire(c1)

. solde(crediter(c1, s1)) = solde(c1) + s1
. titulaire(crediter(c1, s1)) = titulaire(c1)

. solde(debiter(c1, s1)) = solde(c1) - s1
. titulaire(debiter(c1, s1)) = titulaire(c1)
```

end

III-TRAVAIL FAISANT L'OBJET D'UN COMPTE RENDU PAR CHAQUE ETUDIANT

1-Compréhension de la spécification **casl** ci-dessus du **type abstrait** des comptes bancaires. Noter les mots clés du langage CASL(en bleu) .

Commenter chaque groupe d'axiomes de cette spécification en faisant référence au cahier des charges imposé par le responsable du secteur Gestion Client de la banque. Mettre en évidence la **signature** et la **sémantique** de la spécification en expliquant leurs rôles respectifs.

2-Vérifier la **validation** de la spécification.

3-Proposer une **implémentation** pour cette spécification.

4-Vérifier la **correction** de l'implémentation proposée.

