

## TP N°4

## Gestion d'un compte bancaire

**Objectif :**

Le but de ce TP est d'écrire les classes nécessaires à la gestion des comptes des clients d'une banque.

Nous allons simplifier considérablement le problème, notamment en supposant que chaque client a exactement deux comptes : un compte courant et un compte épargne.

**Exercice 1 : Classe CompteCourant**

Ecrire une classe *CompteCourant* pour la représentation des comptes courants. Cette classe possèdera deux données correspondant au numéro de compte bancaire et à son solde, et sera munie des méthodes suivantes :

- un *constructeur*,
- une méthode *String toString* ( ) qui renverra une chaîne contenant les données du compte courant,  
**Exemple** : "Compte courant numéro 123456, solde = 3500.0 euros"
- une méthode *crediter* qui ne renvoie pas de résultat et crédite le compte d'un montant donné (passé en paramètre). La fonction affichera à l'écran un message indiquant l'opération effectuée sur le compte,
- une méthode *debiter* qui ne renvoie pas de résultat et débite le compte d'un montant donné (passé en paramètre), sous les conditions suivantes : si le solde est suffisant alors le compte est débité de la somme demandée, si le solde n'est pas suffisant mais positif alors le compte est débité du solde restant, sinon on refuse le débit. La fonction affichera à l'écran un message indiquant l'opération effectuée sur le compte.

**Exercice 2 : Classe CompteEpargne**

Ecrire une classe *CompteEpargne* pour la représentation des comptes épargne. Cette classe possèdera quatre données correspondant au numéro de compte, à son solde, au taux d'intérêt, ainsi qu'un booléen indiquant si le compte est bloqué (vrai si le compte est bloqué, faux sinon).

Elle sera munie des méthodes suivantes :

- un *constructeur*,
- une méthode *String toString* ( ) qui renverra une chaîne contenant les données du compte épargne,  
**Exemple** : "Compte épargne numéro 987654, a 2.5 % de taux d'intérêts, de solde = 400.0 euros, compte non bloqué."
- une méthode *crediter* qui ne renvoie pas de résultat et crédite le compte d'un montant donné (passé en paramètre). La fonction affichera à l'écran un message indiquant l'opération effectuée sur le compte,
- une méthode *debiter* qui ne renvoie pas de résultat et débite le compte d'un montant donné (passé en paramètre), sous les conditions suivantes : si le solde est suffisant et le compte n'est pas bloqué alors le compte est débité de la somme demandée, sinon on refuse le débit. La fonction affichera à l'écran un message indiquant l'opération effectuée sur le compte,
- une méthode *ajouterInterets* qui ajoute les intérêts au solde courant. Les intérêts seront calculés par la méthode *calculInterets* de la classe Client puis passés en

paramètre à cette méthode.

- une méthode *float getSolde()* qui renvoie le solde du compte Epargne,

### Exercice 3 : Classe Client

Ecrire une classe *Client* pour la représentation des clients de la banque. Cette classe possèdera trois données correspondant au nom du client et à ses deux comptes (compte courant et compte épargne).

Elle sera munie des méthodes suivantes :

- un *constructeur*,
- une méthode *consulterComptes* pour réaliser l'affichage à l'écran de toutes les informations relatives au client,
- une méthode *getNom*, accesseur pour le nom du client,
- une méthode *crediter* qui ne renvoie pas de résultat et crédite l'un des deux comptes d'un montant donné : elle reçoit en paramètre un booléen qui indique quel compte créditer (le compte courant si ce booléen est à vrai, et le compte épargne sinon),
- une méthode *debiter* qui ne renvoie pas de résultat et débite l'un des deux comptes d'un montant donné : elle reçoit en paramètre un booléen qui indique quel compte débiter (le compte courant si ce booléen est à vrai, et le compte épargne sinon),
- une méthode *calculInterets* qui calcule et ajoute les intérêts au compte épargne, elle renvoie en résultat les intérêts calculés. Elle utilisera pour cela, la méthode *ajouterInterets* de la classe *CompteEpargne*.

### Exercice 4 : Classe ApplicationBancaire

Ecrire une classe *ApplicationBancaire* dont la fonction main crée un client de nom Dupont ayant un compte courant de solde 3500 € et un compte épargne non bloqué, à 2,5% de taux d'intérêt, de solde 400 €. On affichera les informations relatives à ce client, puis on demandera de créditer 200 € au compte courant et de débiter 50 € du compte épargne. Enfin on calculera les intérêts de ce client, puis on affichera à nouveau les informations relatives à ce client.

On créera un deuxième client de nom Martin ayant un compte courant de solde 200 € et un compte épargne bloqué, à 3,5% de taux d'intérêt, de solde 1000 €. On affichera les informations relatives à ce client, puis on demandera de débiter 500 € du compte courant et de débiter 500 € du compte épargne. Enfin on calculera les intérêts de ce client, puis on affichera à nouveau les informations relatives à ce client.

1. Remplacer les méthodes infos des classes *CompteCourant* et *CompteEpargne* par des définitions de la méthode *toString*. Adapter le corps de la méthode *consulterComptes* de la classe *Client* en conséquence. Tester au moyen de votre classe *AppliBancaire*.
2. Modifier la classe *ApplicationBancaire* pour que toutes les informations nécessaires à la création des clients et de leurs comptes soient saisies au clavier. Une session d'utilisation de cette application se déroulera d'une façon similaire à ci-dessous :

**// Saisie**

Nom du premier client ?  
Dupont  
Numéro de compte courant ?  
123456  
Solde du compte courant ?  
3500  
Numéro de compte épargne ?  
987654  
Solde du compte épargne ?  
400  
Taux d'intérêts ?  
2.5  
Est-ce un compte bloqué (y/n) ?  
N

## // Affichage

### // appel de la méthode consulterCompte() de la classe Client

Comptes de M. Dupont :

Compte courant numéro 123456, solde = 3500.0 euros

Compte épargne numéro 987654, a 2.5 % de taux d'intérêts, de solde = 400.0 euros, compte non bloqué.

### // appel de la méthode crediter() de la classe Client

On crédite 200.0 euros au compte 123456

### // appel de la méthode debiter() de la classe Client

On retire 50.0 euros du compte 987654

### // appel de la méthode calculerIntérêts() de la classe Client

Les intérêts du compte épargne de M. Dupont sont de 8.75 euros

### // appel de la méthode consulterCompte() de la classe Client

Comptes de M. Dupont :

Compte courant numéro 123456, solde = 3700.0 euros

Compte épargne numéro 987654, a 2.5 % de taux d'intérêts, de solde = 358.75 euros, compte non bloqué.

Nom du  
... deuxième client ?  
Martin

**Diagramme de classes de l'application :**