

## TP n°5

### Banque

Dans ce TP on va implémenter du code pour la banque modélisée au TP5.

**Exercice 1** Écrivez une classe `Banque`. Écrivez une classe `Salarie` avec des champs `nom`, `numeroID`, `salaire` avec leurs accesseurs et avec les constructeurs et les destructeurs. Dans la classe `Banque`, créez un champ `listeDesSalaries`. Modifiez constructeurs et destructeurs pour mettre à jour ce champ. Écrivez les sousclasses `Dirigeant`, `Comptable`, `Conseiller`, `AgGuichet`, `AgSecurite`, `ConsAdm`, en ajoutant éventuellement des champs et des accesseurs. Pour chacune de ces classes, écrivez un constructeur de copie d'un Salarié.

**Exercice 2** Écrivez une classe `Equipe`. Réalisez la relation d'agrégation avec un dirigeant et une liste de salariés. Chaque salarié ne peut appartenir qu'à une seule équipe : modifiez la classe `Salarie` pour tenir compte de cela. Écrivez les constructeurs, destructeurs et accesseurs. Écrivez une méthode pour ajouter un salarié à l'équipe. Écrivez une méthode pour éliminer tous les dirigeants de la liste de salariés.

Écrivez une sousclasse `EquipeDeSecurite`. Réfléchissez à sa réalisation.

**Exercice 3** Écrivez une classe `Branche` qui agrège plusieurs équipes. Ajouter à la classe `Banque` un champ `listeDesBranche`. Dans la classe `Branche` écrivez une méthode `void deplace(Equipe & e, Branche & b)`

**Exercice 4** Écrivez les classe `Devise`, `Compte`, `Client`, `MoyenDePaiement`. Réalisez à l'aide d'une classe d'association `CompteClient` la relation entre clients et compte. Chaque client a un conseiller.

Écrivez une méthode pour le dépôt d'argent, le retrait. Écrivez une méthode pour le transfert d'argent entre deux compte d'un client.

**Exercice 5** La banque a décidé que chaque conseiller peut suivre au plus 4 clients. Quand un nouveau client est créé, il faut le confier au premier conseiller libre. S'il n'y a pas de conseiller, on devra en embaucher un. Réaliser cela en modifiant les classes de votre banque.

**Exercice 6** De temps à autre la banque augmente le salaire à tous ses employés. Pour faire cela on utilise une méthode `void augmentation()` qui parcourt la `listeDesSalaries` Chaque salarié sera augmenté de façon différente :

- un Conseiller d'administration reçoit une augmentation de 15%
- un Dirigeant qui n'est pas conseiller d'administration reçoit 10%
- Les Comptables et le Conseiller reçoivent 60 euros
- Les Agents de Sécurité et de Guichet reçoivent 30 euros

Ajoutez à la classe `Salarie` une méthode virtuelle pure `void augmente()`. Pour chaque sousclasse, redéfinissez cette méthode selon les critères décrits.