

TP n°3

Classes

Note : ces exercices nécessitent la lecture des supports des six premiers cours que l'on peut retrouver à l'adresse <http://www.liafa.univ-paris-diderot.fr/~yunes/cours/cpp/>

Exercice 1 (Contrôle d'accès)

Soient les classes suivantes :

```
class X
{
    int priv;
public:
    int pub;
    void m();
};

class Y : public X
{
    void my();
};
```

Est-ce que le code suivant est correct ?

```
1. void X::m()
{
    priv = 1;
    pub = 3;
}
```

```
2. void Y::my()
{
    priv=1;
    pub=3;
}
```

```
3. void f(Y * p)
{
    p->priv=1;
    p->pub=3;
}
```

Exercice 2 (Tableaux Multidimensionnels)

Définissez une matrice d'entiers de dimension 2 et écrivez les fonctions d'affichage suivantes :

1. `void Affichej(. .)` qui prend en argument une matrice dont les dimensions sont connues au moment de la compilation.
2. `void Affichej(. .)` qui affiche une matrice en fixant uniquement sa seconde dimension.
3. `void Affiche(. .)` qui affiche une matrice sans connaître ses dimensions au moment de la compilation.

Testez ces trois fonctions en appelant chacune avec des arguments adéquats.

Exercice 3 (Une classe d'association) Avant de commencer relisez la partie *La classe d'association* du cours 5.

Dans cet exercice on modélisera l'association de la classe `Etudiant` à la classe `Cours` par une classe d'association qu'on appelle `Association`. La classe `Cours` possède la méthode privée



FIGURE 1 – Classe d'association.

`void Cours::nouvelEtudiant()` qui incremente `nbtudiants`. Elle est appelée par la classe `Association` lorsqu'une nouvelle association est créée (similaire pour la classe `Etudiant`).

`Association` a les membres `Etudiant* petudiant` et `Cours* pcours` qui représente cette association. Créez le constructeur `Association::Association(Etudiant* petudiant, Cours* pcours)`. Le constructeur va *notifier* `petudiant` et `pcours` qu'une nouvelle association a été créée en appelant leurs méthodes `nouvelEtudiant()` et `nouveauCours()`. Pensez au concept `friend` lorsque vous appelez ces fonctions.

Pour tester votre programme, créez un tableau d'étudiants et un tableau de cours. Créez quelques associations et vérifiez le bon fonctionnement des compteurs `nbcours` et `nbtudiants`.

Exercice 4 (Modélisation d'une Banque) Implémentez le diagramme de classe de la Figure 2.

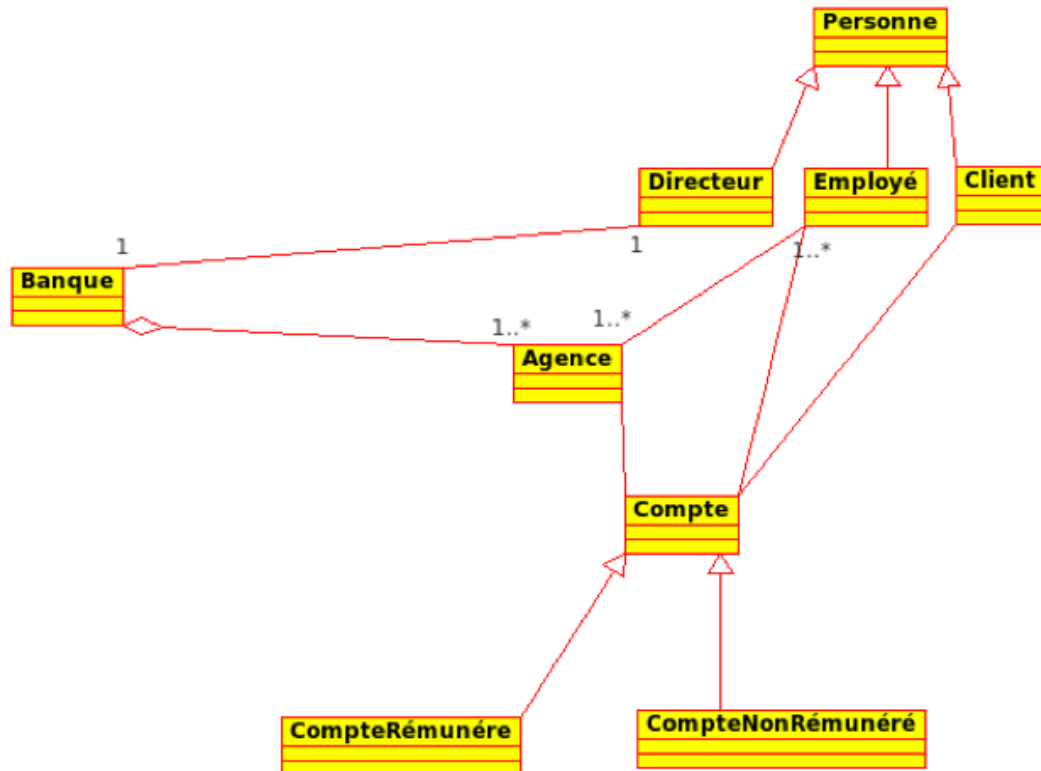


FIGURE 2 – Diagramme de classe d'une banque.