

EXERCICE 1

1- Ecrire une structure `Point` formée de deux double (abscisse et ordonnée) et d'un nom, qui est un caractère. Ecrire les fonctions :

- `init` qui permet d'initialiser un point avec une abscisse, une ordonnée et un nom;
- `print` qui affiche les caractéristiques d'un point ;
- `translate` qui pratique la translation d'un point selon deux valeurs de type double.

2- Reprendre la définition du type `point` en transformant la structure en une classe.

3- Discuter la notion d'encapsulation au niveau de la classe.

4- Utiliser cette notion pour écrire une fonction nommée `distant` calculant la distance entre deux points.

5- Répartissez le contenu du programme de l'exercice précédent entre les fichiers `Point.h` (la classe) et `Point.cpp` (les méthodes).

EXERCICE 2

Le but de cet exercice est d'écrire une classe `Compte` pour gérer les comptes bancaires d'un certain nombre de clients. Les informations suivantes sont disponibles sur ces comptes :

- un compte est défini par trois informations : le numéro du compte, le nom du client et le solde ;
- un même client peut avoir plusieurs comptes.

Écrire la classe `Compte` en donnant des méthodes d'initialisation, de consultation du compte, de retrait et de dépôt.

EXERCICE 3

On veut à présent écrire une classe `Banque` pour gérer les comptes des différents clients.

Cette nouvelle classe définira par conséquent un tableau `comptes` contenant tous les comptes que l'on désire gérer. Cette classe devra définir les méthodes suivantes :

```
void add(Compte c);
```

```
// ajoute un compte à la liste des comptes à gérer
```

```
void delete(int numero);
```

```
// supprime le compte dont le numéro est donné par numero  
Compte ;
```

```
getCompte(int numero);
```

```
// retourne le compte dont le numéro est donné par numero
```

Écrire cette classe.