

TP2

1) Soit **point** une classe comportant :

- deux composantes x et y (privées) de type entier ,
 - trois fonctions membres:
 - une fonction **Affiche** permettant d'afficher les coordonnées x et y,
 - une fonction **Initialise** qui permet d'initialiser ces coordonnées,
 - et une fonction **Deplace** permettant de déplacer le point de coordonnées x et y par dx, dy.
- a) Ecrire le programme qui permet l'instanciation d'un objet de la classe point et l'appel de ces fonctions membres.
- b) Remplacer la fonction Initialise par un constructeur jouant le même rôle.

2) Ecrire une classe **vecteur** comportant:

- trois composantes (privées) représentant respectivement l'abscisse, l'ordonnée et la hauteur,
- une fonction **affiche**,
- deux constructeurs qui affichent l'adresse de l'objet créé:
 - l'un, sans arguments, initialisant chaque composante à 0,
 - l'autre, avec 3 arguments, représentant les composantes, où par défaut, la hauteur est nulle.
- un destructeur où on affiche l'adresse de l'objet détruit.

3) Ajouter à la classe **vecteur** précédente deux fonctions membres nommées:

- **prod_scal** fournissant en résultat le produit scalaire de deux vecteurs,
- **somme** permettant de calculer la somme de deux vecteurs.

4) Modifier la classe **vecteur** précédente (exercice 3) de manière que toutes les transmissions de valeurs de type vecteur aient lieu:

- a) par adresse,
- b) par référence.

5) Soit **pile_entier** une classe permettant de gérer une pile d'entiers. Ces derniers sont conservés dans un tableau d'entiers alloués dynamiquement. La classe comporte les fonctions membres suivantes:

- **pile_entier(int)** : constructeur allouant dynamiquement un emplacement de n entiers,
- **pile_entier()** : constructeur sans arguments allouant par défaut un emplacement de 20 entiers,
- **~pile_entier()** : destructeur,
- **void empile(int p)** : ajoute l'entier p sur la pile,
- **int depile()** : fournit la valeur de l'entier situé en haut de la pile, en le supprimant de la pile,
- **int pleine()** : fournit 1 si la pile est pleine, 0 sinon.
- **int vide()** : fournit 1 si la pile est vide, 0 sinon.

- a) Ecrire la classe **pile_entier** et toutes ces fonctions membres.
- b) Ecrire une fonction main utilisant des objets automatiques et dynamiques du type **pile_entier** défini précédemment.
- c) Mettre en évidence les problèmes posés par des déclarations de la forme:

```
pile_entier a(10) ;  
pile_entier b = a ;
```

- d) Ajouter à la classe **pile_entier** le constructeur par copie permettant de régler les problèmes précédents.