

TP de Programmation Objet en C++ n°2

Mise à niveau en Master 2 Informatique

Septembre 2010

1 Extension de la classe Vecteur

- Écrire une classe `VecteurRotation` héritant de la classe `Vecteur` et dans laquelle on trouvera en plus une notion d'angle.
- Certaines opérations réalisables sur la classe `Vecteur` n'étant pas forcément souhaitables pour la classe `VecteurRotation`, on pourra utiliser un héritage protégé.
- Il faudra penser à modifier la classe `Vecteur` pour déclarer `virtual` certaines de ses méthodes.
- Il faudra en particulier redéfinir l'opérateur d'affectation et la méthode `afficher` de la classe `VecteurRotation` en évitant toute recopie de code de la classe `Vecteur`.

Difficultés abordées

- ⇒ Héritage protégé.
- ⇒ Méthodes virtuelles.
- ⇒ Factorisation de code.

2 Extension de la classe Point

- Ecrire une classe `Point6ddl` héritant de la classe `Point` pour lui ajouter la notion de rotation (à l'aide de la classe `VecteurRotation` par exemple), qui permette d'écrire :

```
Point3D * p3D = new Point6ddl (1, 2, 3, 4, 5, 6, 7) ;  
std::cout << "*p3D = <" << *p3D << ">" << std::endl ;
```

Et qui produise comme résultat :

```
*p3D = <1 2 3 4 5 6 7>
```
- Il faudra penser à modifier la classe `Point` pour déclarer `virtual` certaines de ses méthodes.
- Il faudra en particulier redéfinir l'opérateur d'affectation et la méthode `afficher` de la classe `Point6ddl` en évitant toute recopie de code de la classe `Point`.
- La notion de rotation pourra être ajoutée soit par composition (agrégation d'un `VecteurRotation`) soit par héritage multiple (en héritant également de la classe `VecteurRotation`).

Difficultés abordées

- ⇒ Héritage public.
- ⇒ Méthodes virtuelles.
- ⇒ Pointeurs.
- ⇒ Polymorphisme et liaison dynamique.
- ⇒ Héritage multiple.

3 Une classe générique regroupant différentes sortes d'objets affichables

- Ecrire une classe générique `Nuage` permettant de stocker des pointeurs vers des objets et de les afficher sur demande.
- Cette classe générique pourra utiliser par exemple la classe `list` de la STL, et utiliser des itérateurs pour parcourir ses éléments et les afficher.
- Il peut être intéressant de redéfinir l'opérateur d'injection d'un élément dans un `Nuage`.
- Instancier cette classe générique pour y ranger des pointeurs vers des `Points`, et y placer effectivement des pointeur vers des instances de la classe `Point` et des pointeurs vers des instances de la classe `Point6ddl`.
- Vérifier le bon fonctionnement de l'affichage de ce nuage de points.

Difficultés abordées

⇒ Classes génériques.