

TEMPLATE

1 Objectifs du TD

- 1) Ecriture d'une classe template.
- 2) Réviser la surcharge des d'opérateurs.
- 3) Utilisation des classes templates dans un programme.
- 4) Héritage entre classe template.

EXERCICE 1 – La classe Array.

- 1) Ecrivez le modèle de classe Array qui devra permettre de créer des vecteurs de n'importe quelle taille, avec des éléments de n'importe quel type. Seuls les opérateurs [] et « devront exister pour cette classe.
- 2) Ecrivez un programme principal : instancie un Array de "double" de taille 10, initialise les valeurs du tableau avec des nombres aléatoires et qui affiche le tableau sur la sortie standard.

Remarque – *Le type des élément et la taille du tableau sont génériques. La classe Array a donc la structure suivante :*

```
template <typename TYPE, unsigned int SIZE>
class .... {
};
```

EXERCICE 2 – La classe Vector3D

- 1) Ecrivez le modèle de classe Vector3D héritant de Vector et possédant les opérateurs + et += ainsi qu'un constructeur prenant trois paramètres pour initialiser les trois coordonnées de type générique du vecteur.
- 2) Ecrivez un programme principal qui : instancie 2 Vector3D de "float", initialise les valeurs des vecteurs avec les valeurs (2,3,4) et (5,5,5) et qui affiche le resultat de l'addition des ces deux Vector3D.

EXERCICE 3 – La classe Array2D

- 1) Ecrivez le modèle de classe Array2D qui devra permettre de créer des tableaux à deux dimensions de n'importe quelle taille, avec des éléments de n'importe quel type. Seuls les opérateurs [] et « devront exister pour cette classe.
- 2) Ecrivez un programme principal qui : instancie un Array2D de "string" de taille 3x3, initialise les valeurs du tableau et qui affiche le tableau sur la sortie standard.

Remarque – *La taille du code de la classe Array2D est très très courte, (4-5 lignes).*