

## TP 5

Le but de ce TP est de manipuler la généricité en C++. On va définir une classe générique *ListeOrdonnee* permettant de stocker une liste d'objets stockés de manière ordonnée.

### Exercice 1

Définir la classe *ListeOrdonnee* paramétrée par le type d'objets *T* qu'il contient. On stockera ces objets sous forme d'un tableau, d'une manière séquentielle dans un premier temps (pas ordonnée). On définira une variable statique *ListeOrdonnee::TAILLE\_MAX* qui sera initialisée à 100, et on supposera qu'on insèrera pas plus de 100 objets dans notre liste. La classe possède les méthodes suivantes:

- un constructeur par défaut uniquement (on connaît la taille du tableau : *ListeOrdonnee::TAILLE\_MAX*)
- un destructeur
- une surcharge de l'opérateur «
- *add* qui prend en entrée un objet de type *T*, et l'insère à la suite des éléments présents dans le tableau (il faudra donc un attribut qui mémorise le nombre d'objets dans la liste).

Ecrire la classe *ListeOrdonnee* et la tester avec des entiers et des objets de type *Personne*

### Exercice 2

On va maintenant rendre notre classe *ListeOrdonnee* ordonnée. Cependant, afin d'ordonner des objets, il faut pouvoir les comparer. La classe *ListeOrdonnee* doit donc également être paramétrée par une fonction de comparaison *int compare(T t1, T t2)* qui retourne -1, 0 ou 1 si respectivement *t1* est strictement inférieur, égal ou strictement supérieur à *t2*.

Modifier la classe *ListeOrdonnee* (les templates et la méthode *add*) afin d'avoir une liste ordonnée. Définir également une méthode *compareInt* et *comparePersonne* qui comparent respectivement des entiers et des personnes. Tester à nouveau.

### Question bonus

Ajouter la méthode *remove* qui prend en entrée un objet de type *T* et le supprime de la liste s'il existe.