

TP n° 5 de C++

Liste chaînée dynamique en C++ - notion élémentaire d'itérateur Généricité sur une structure de donnée dynamique

- 1) Ecrire dans les mêmes fichiers (hpp et cpp) le code C++ pour 3 classes : Cellule (*Cell*), Liste (*List*) et un itérateur de liste (*ItList*). Ces classes ne seront pas implémentées sous forme de classe imbriquée mais on considère que la classe Cell a pour amie la classe List et la classe ItList (à la différence de la STL qui imbrique les classes – concept de « nested classes » qui n'est pas présent dans les concepts objets).

La cellule « Cell » possède un attribut de type entier, et un pointeur sur la cellule suivante (*_next*) (on considère une liste simplement chaînée).

La classe « List » possède comme attribut une tête (*_first*) et une fin de liste (*_last*). La fin de liste pointe sur le dernier élément de la liste (*_first* = *_last* = NULL quand la liste est vide).

Vous implémenterez des méthodes telles que celles proposées dans la STL (Standard Template Library) pour l'ajout/suppression d'un élément, la recherche d'un élément, l'affichage de la liste,... Cherchez les prototypes proposés par la STL pour *push_back*, *push_front*, *pop_front*, *empty*, *clear*, *remove*, ... faites votre choix, implémentez les et testez les.

La classe itérateur de liste (*ItList*) comporte un pointeur sur une liste (*_myList*) ainsi qu'un pointeur sur une cellule courante (*_cellPtr*). Vous proposerez des surcharges pour l'opérateur d'affectation =, pour l'opérateur de déréférencement * (qui donne l'objet pointé par l'itérateur), les opérateurs ++ (pré et postfixés), et les opérateurs de comparaison == et != .

La classe itérateur vous permet d'ajouter une méthodes à la classe liste – *begin()* pour retourner un itérateur sur le début de la liste.

Coder et tester dans un programme principal les classes une à une en proposant les constructeurs que vous jugerez utiles.

- 2) Tester dans le programme principal deux instances d'itérateurs référant la même liste pour faire, par exemple, la comparaison d'éléments 2 à 2 au sein de cette liste et le parcours d'une liste dans le but d'afficher ses éléments un à un.
- 3) Coder ce patron de Liste de manière générique avec la notion de template.