

COURS DE C++

12 - Standard Template Library

TEMPLATES ET UTILISATION

- Il peut être utile de paramétrer une classe par une autre
- Typiquement : une classe liste pouvant être utilisée pour faire des listes d'entiers, de flottants, d'objets d'une classe donnée ou même de listes
- Une classe paramétrable est appelée patron (template)
- On utilise la syntaxe `A` pour utiliser la classe patron `A` avec la classe `B`
- Si `B != C` alors `A != A<C>`, il faut voir une classe patron comme une famille de classes (foncteur)

STANDARD TEMPLATE LIBRARY

- C'est une bibliothèque incluse avec tous les compilateurs
- Elle fournit des structures de données et des algorithmes génériques
- La généricité est implementée grâce aux templates

VECTOR TABLEAUX INTELLIGENTS

- `#include <vector>`
- définis un template `vector<>`
- Construction :
 - `vector<int> v;`
 - `vector<Point> p(12,Point(1.0,3.0)); // un tableau de 12 points initialisés à (1.0,3.0)`

ITERATOR

- Chaque type de donnée de la STL fourni une classe permettant de le parcourir : itérateurs
- Par exemple `vector<int>::iterator`
- Une instance de cette classe est une sorte de pointeur vers les objets contenus dans le vector
- `v.begin()` renvoie l'itérateur pointant sur le premier élément
- `v.end()` renvoie l'itérateur pointant "dans le vide" après le dernier élément (sentinelle)
- `++iter` fait pointer l'itérateur vers l'élément suivant
- `*iter` renvoie la valeur pointé par l'itérateur
- `for (vector<int>::iterator iter = v.begin(); iter != v.end(); ++iter) { } // parcours`

VECTOR MANIPULATION

- Récupération du i-ème élément :
 - `v[i]` ou `v.at(i)` ce dernier vérifie que i est dans les bornes
- Ajout de donnée :
 - `v.push_back(x);` // rajoute x à la fin de v
 - `v.insert(iter, y);` // rajoute y avant l'élément pointé par iter
 - pour ajouter un élément en tête il suffit de faire : `v.insert(v.begin(), x);`
- Taille : `v.size()`
- Suppression de donnée : `v.pop_back();` ou `v.erase(iter);`