



Analyse et Programmation Orientées Objets / C++

Standard Template Library

Services et notions de base (1)

➔ Encapsulation de la librairie C

Modification de la désignation des fichiers d'en tête (directive **include**)

Introduction de nouveaux entête (fichiers de déclaration)

Réservation de l'espace de nommage **std::** (directive **using**)

Définition des exceptions standards (entete **stdexcept** et classe **exception**)

Normalisation d'une méthode **what** propre aux classes d'exceptions

Services et notions de base (2)

➡ Abstraction des types de données (1)

Abstraction des types de base (concept de **traits**)

Abstraction des tableaux (classe **valarray**)

Abstraction des flux et introduction des tampons

Abstraction des pointeurs (concept d'**itérateur**)

Abstraction des fonctions (concept de **foncteur**)

Abstraction des collections (concept de **conteneur**)

Services et notions de base (3)

➡ Abstraction des types de données (2)

Gestion personnalisée de la mémoire (concept d'**allocateur**)

Introduction de types complémentaires (**string**, **pair**, **complex**, **auto_ptr**, ...)

Introduction des locales et des facettes

Services et notions de base (4)

➡ Normalisation

Norme ISO 14882:1998 et suivantes

Usage massif des classes génériques (**template**)

➡ Difficultés pour le débutant

Les abstractions interagissent entre elles !

Les conteneurs (1)

➡ Définition

Ensemble de classes génériques aptes à :

- conserver et à restituer des structures de données (indépendamment de leurs types !)
- fournir les services de manipulation associés (ajout, suppression, remplacement, filtres, recherche, ...)

Les conteneurs (2)

➡ Classification

- les séquences
- les conteneurs associatifs

Les séquences

➔ Classification

- les listes (classe **list**)
- les vecteurs (classe **vector**)
- les dequeues (classe **deque**)
- les piles (classe **stack**)
- les files (classe **queue**)
- les files de priorité

Les conteneurs associatifs

➔ Premier niveau de classification

- les associations (classe **map**)
- les ensembles (classe **set**)

➔ Second niveau de classification

- clé unique
- clés multiples (préfixe *multi*)

Bibliographie

- ➔ **Cours C/C++ Christian Casteyde**
- ➔ **Cours C/C++ Bruce Eckel**
- ➔ <http://www.developpez.biz/download/stl.pdf>