THIBAUD Erwan
GORAND Tancrède

Rédactions Containers

En C++, les containers sont des structures de données qui permettent de stocker et manipuler des éléments.

- Le container vector (vecteur) :
  Un vecteur est un tableau utilisé pour stocker des éléments. Il est très simple et rapide d'ajouter ou de supprimer un élément, mais cela prend plus de temps lorsque ces modifications sont effectuées au centre du vecteur, car il est nécessaire de parcourir tous les éléments pour y accéder.

  Nous pouvons utiliser les vecteurs si nous voulons stocker une suite de chiffre mais que nous ne savons pas exactement combien nous voulons en stocker. Nous pouvons églament par la suite accéder fréquemment aux chiffres stockés pour les manipuler sans problème.

```
std::vector<int> numbers;

numbers.push_back(10);
numbers.push_back(20);
numbers.push_back(30);
//...
```

- Le container list (liste) :
  La liste est une suite d'éléments stockés comme pour un vecteur. Cependant la liste est plus longue à parcourir mais une fois que l'élément recherché est trouvé, l'insertion et la suppression sont rapides. En général, les containers list sont des listes doublement liées. C'est-à-dire qu'il s'agit de nœuds qui contiennent les données, la référence du nœud en question et la référence au nœud précédent dans la liste.

  Nous pouvons prendre l'exemple d'une liste d'élèves d'une classe que nous pouvons stocker dans une liste .
  Le choix d'une liste est pertinent parce qu'il est facilement possible d'ajouter des étudiants au TD partout dans la liste. Cela est pratique si nous voulons ranger les étudiants dans l'ordre alphabétique.

```cpp
std::list<std::string> listeEleves;

listeEleves.push_back("Erwan");
listeEleves.push_back("Tancrède");
//...

listeEleves.sort();
```

- Le container deque (double queue ou file à deux extrémités) :
  Un container deque est une file d'éléments qui permet l'ajout et la suppression d'éléments au début et à la fin de la queue. Cependant, un peu comme pour le vecteur, il est plus long d'accéder aux éléments du centre du container.

  Nous pouvons voir dans l'exemple suivant comment ajouter des entiers au début ou à la fin de la liste.

```cpp
deque<int> doubleque;
doubleque.push_back(10);
doubleque.push_front(20);
doubleque.push_back(30);
doubleque.push_front(15);

cout  << doubleque.size() << endl;
```

- Le container map (mappe) :
  Le container map est une structure de données où chaque élément est associé à une clé. Cela permet une recherche efficace et rapide mais l'insertion ou la suppression des éléments sont moins efficaces que les autres containers.

  Nous pouvons utiliser map pour créer un dictionnaire et ainsi stocker des valeurs associés à des clés.

```cpp
map<string, string> dictionnaire;
dictionnaire["right"] = "droite";
dictionnaire["left"] = "gauche";
dictionnaire["top"] = "haut";

cout << "Dictionnaire anglais-français : " << endl;

map<string, string>::iterator it = dictionnaire.begin();
while (it != dictionnaire.end()) {
    cout << it->first << " => " << it->second << endl;
    it++;
```