Les ranges et vues sur les collections

Les ranges

Les algorithmes utilisent des itérateurs pour manipuler les collections de données. Une contrainte forte est que les paires d'itérateurs doivent provenir de la même collection.

```
vector<int> v1(10), v2(10);
find(begin(v1), end(v1), 0); // ok
find(begin(v1), end(v2), 0); // erreur
```

Pour éviter ce type d'erreur, un range encapsule une paire d'itérateurs dans une classe. Il est plus facile dans ces conditions de vérifier la cohérence des itérateurs.

1. Ecrire une classe Range contenant une paire d'itérateurs

- sémantique de valeur
- en mode débug, conserver une référence sur la collection et vérifier que les itérateur sont cohérents ?

2. Réécrire les algorithmes de la bibliothèque standard pour prendre en compte les ranges

http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n4128.html

DP proxy ?

Problématique : on a une collection spécifique et l'on veut la manipuler comme si c'était une collection d'un autre type. Ou une sous collection

2 solutions :

- écrire une collection intermédiaire, qui du point vue utilisateur se comporte comme une collection de type attendu et utilise en interne la collection donnée = vue
- écrire un itérateur spécial, qui fait le conversion. prochain chapitre

string_view

Cf le TS

array_view

tableau multi dimensionnel

autre ?

collection pour “voir” que les nombres pairs ?

Chapitre précédent	Sommaire principal	Chapitre suivant

Cours, C++