

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Déclaration et initialisation des vector (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Tableau dynamique. Dernière façon. Copie d'un tableau. Façons suivantes. Accolade ouvrante. Séquence initiale de valeurs. Éléments tous nuls. Exemple. Accolade fermante. Tab2. Parenthèses d'initialisation. Nom de la variable. Éléments. Valeur différente. Copie élément.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Tableaux : déclaration et initialisation des `vector` (Partie 2)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Initialisation d'un tableau dynamique

Avec taille initiale et initialisation des éléments à la même valeur :

```
vector<type> identificateur(taille, valeur);
```



On peut aussi initialiser un tableau dynamique à l'aide d'une copie d'un autre tableau dynamique :

```
vector<type> identificateur(reference);
```

où *reference* est une référence à un tableau de même type de base *type*.

Exemples :

```
vector<int> tab1(5, 1);  
vector<int> tab2(tab1);
```



correspondent toutes deux à la déclaration d'un tableau d'entiers dont les 5 éléments de départ sont initialisés à la valeur 1.

Enfin, dernière façon d'initialiser un tableau dynamique. On peut aussi initialiser un tableau dynamique en faisant une copie d'un tableau qui existait au préalable. Donc par exemple ici je peux déclarer un tableau qui s'appelle "tab2", qui va être une copie. Donc ici on va copier élément par élément le tableau "tab1" dans "tab2" mais on aura bien deux tableaux qui, au départ, sont dans la même situation. "tab1" et "tab2" ont au départ, tous les deux cinq éléments qui valent tous 1. Mais ensuite chacun de ces deux tableaux "tab1" et "tab2" évoluera séparément. Et par exemple on pourra éventuellement modifier des éléments dans "tab2", et faire varier "tab2" comme on veut, indépendamment donc de "tab1". Ce sont donc bien deux tableaux complètement séparés.

notes

résumé

0m 1s



En C++11, il y a cinq façons d'initialiser un tableau dynamique :

- ▶ vide

```
vector<int> tab;
```

- ▶ avec un ensemble de valeurs initiales

```
vector<int> tab(20, 35, 26, 38, 22);
```

- ▶ avec une taille initiale donnée et tous les éléments « nuls »

```
vector<int> tab(5);
```

- ▶ avec une taille initiale donnée et tous les éléments à une même valeur donnée

```
vector<int> tab(5, 1);
```

- ▶ avec une copie d'un autre tableau

```
vector<int> tab(tab2);
```

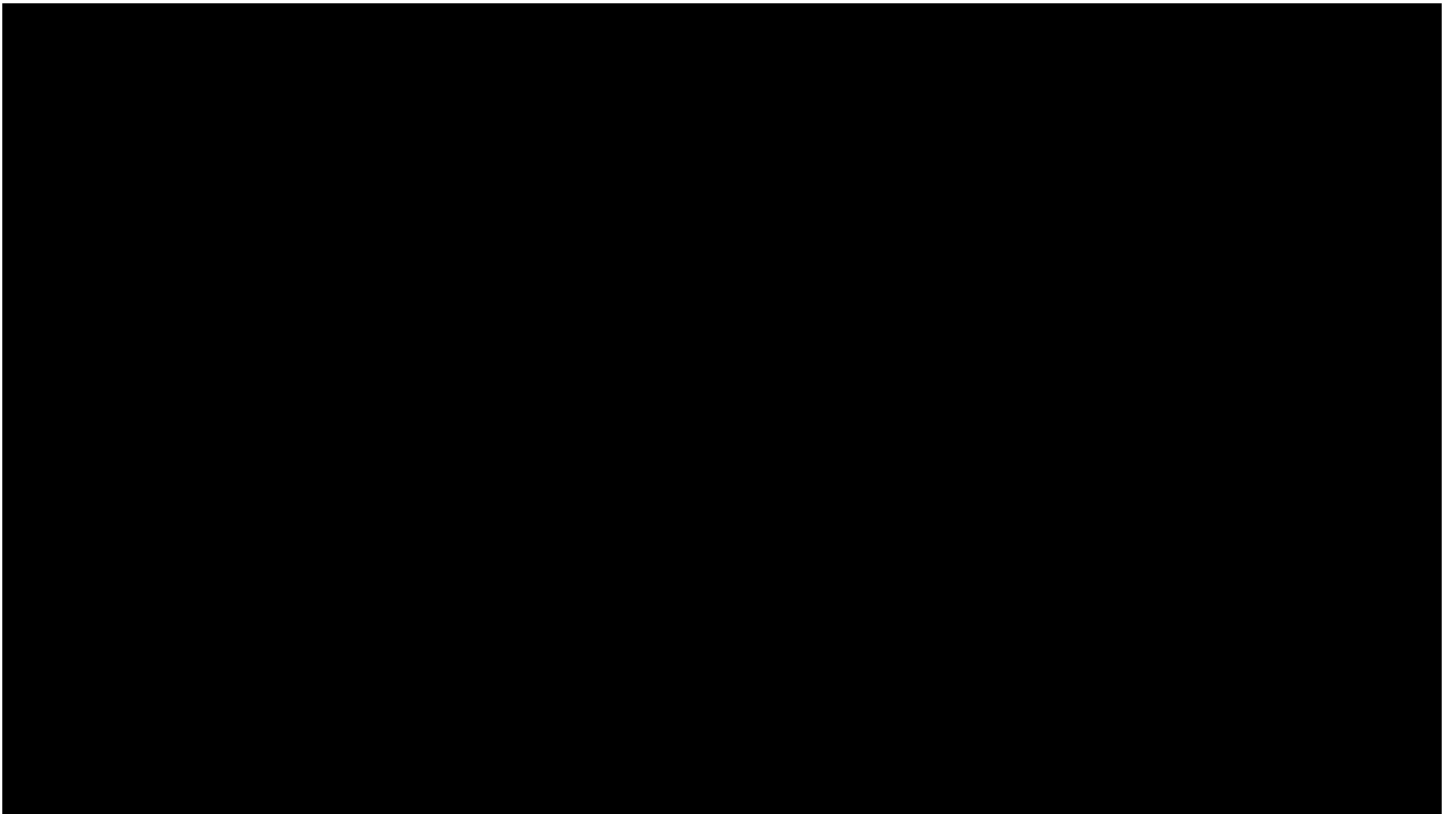
On a bien ici une copie de "tab1" dans "tab2". Donc pour résumer, on a les cinq façons suivantes pour initialiser un tableau dynamique : Soit vide, on ne met rien du tout, j'insiste là-dessus, derrière le nom de la variable dans la déclaration. Le tableau sera alors vide. Soit l'initialiser avec une séquence initiale de valeurs. Pour cela on va mettre une accolade ouvrante, une accolade fermante dans les parenthèses d'initialisation. Soit l'initialiser avec un nombre fixé d'éléments tous nuls. A ce moment-là, entre parenthèses, on mettra juste un entier. Soit l'initialiser avec un nombre fixé d'éléments, donc on donnera la taille, mais avec une valeur différente de 0. Donc ici on va fournir une valeur qui est de même type que le type annoncé pour chacun des éléments.

notes

résumé

1m 1s





Soit enfin, faire une copie d'un tableau qui existait déjà, et on va faire une copie élément par élément. Voilà donc les cinq façons d'initialiser un tableau dynamique.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 49s

