Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Fonctions spécifiques des vector (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

Travers d'une fonction spécifique. Taille du tableau. Fonctions spécifiques. Tableaux dynamiques. Utilisation d'une fonction spécifique. Nom du tableau. Nom de la fonction spécifique. Type de retour de la fonction. Valeur de retour de cette fonction. Fonction spécifique. Exemple nombre. Fonction size. Premier élément. Fonction pop. Tableaux.

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

vers la vidéo

EPFL

Tableaux : fonctions spécifiques `vector`

(Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

Dans les boucles sur les tableaux,

notes

résumé

0m 1s

Fonctions spécifiques

EPFL

Un certain nombre d'opérations sont **directement attachées** au type `vector`.

L'utilisation de ces opérations spécifiques se fait avec la syntaxe suivante :

```
nom_de_tableau.nom_de_fonction(arg1, arg2, ...);
```

Exemple :

```
vector<double> mesures;
size_t nombre_de_mesures(0);
...
nombre_de_mesures = mesures.size();
...
```

nous avons donc vu comment récupérer la taille du tableau au travers d'une fonction spécifique qui s'appelle size. Il existe en fait beaucoup de fonctions spécifiques associées aux tableaux dynamiques et j'aimerais vous en présenter maintenant quelques-unes. L'utilisation d'une fonction spécifique se fait toujours de la même façon. Ce sera le nom du tableau suivi d'un point collé directement au nom du tableau, suivi du nom de la fonction spécifique. Cette fonction spécifique peut par exemple prendre des arguments, si elle en a besoin. Dans la fonction size il n'y avait pas d'arguments à fournir, mais nous allons voir des exemples où l'on aura besoin de fournir des arguments. Donc par exemple, la fonction size est une fonction spécifique que l'on va appliquer à un tableau. Donc ici le tableau mesure, que l'on aura déclaré comme un tableau dynamique pouvant prendre des double. Si l'on veut récupérer cette taille pour pouvoir l'utiliser dans une variable, je vous rappelle que le type de retour de la fonction size c'est size_t,

notes

résumé

0m 5s

Fonctions spécifiques

EPFL

Quelques fonctions disponibles pour un tableau dynamique nommé `tableau`, de type `vector<type>` :

`tableau.size()` : renvoie la taille de `tableau` (type de retour : `size_t`)

`tableau.front()` : renvoie une référence au 1er élément
`tableau.front()` est donc équivalent à `tableau[0]`

`tableau.back()` : renvoie une référence au dernier élément.
`tableau.back()` est donc équivalent à `tableau[tableau.size()-1]`

`tableau.empty()` : détermine si `tableau` est vide ou non (`bool`).

`tableau.clear()` : supprime tous les éléments de `tableau` (et le transforme donc en un tableau vide). Pas de (type de) retour.

`tableau.pop_back()` : supprime le dernier élément de `tableau`. Pas de retour.

`tableau.push_back(valeur)` : ajoute un nouvel élément de valeur `valeur` à la fin de `tableau`. Pas de retour.

et donc on déclarerait une variable, par exemple nombre_de_mesures de type size_t. On l'initialiserait à 0 par exemple et on pourrait donc l'utiliser pour récupérer la valeur de retour de cette fonction spécifique size. Il existe plusieurs autres fonctions spécifiques propres aux tableaux dynamiques. Par exemple, la fonction front va renvoyer une référence sur le premier élément. Si j'ai donc un tableau qui s'appelle "tableau" tableau.front() c'est exactement la même chose qu'avoir écrit tableau[0]. Il existe la symétrie bien sûr, back, qui renvoie le dernier élément. Alors quel est le dernier élément d'un tableau ? C'est "tableau[la taille du tableau()-1]". Donc le dernier élément, c'est "tableau[tableau.size()-1]. Il existe une fonction qui permet de tester si le tableau est vide ou non. Donc la fonction spécifique empty, va renvoyer un type booléen qui vous dira vrai si le tableau est vide et faux si le tableau contient quelque chose. Vous avez la fonction spécifique clear qui permet de supprimer tous les éléments d'un tableau auquel elle s'applique. Donc toujours la même syntaxe : le_nom_du_tableau.clear(). Après l'appel à cette fonction spécifique clear, le tableau est donc vide. Vous avez ensuite des fonctions qui permettent d'ajouter et de supprimer des éléments derrière un tableau. Donc par exemple, la fonction pop_back, va permettre de supprimer le dernier élément d'un tableau. Et la fonction push_back

notes

résumé

1m 1s

push_back et pop_back

```
vector<double> v(3, 4.5);

v.pop_back();
v.push_back(5.6);
v.push_back(6.7);
v.pop_back();
```

va rajouter un élément à la fin d'un tableau. Je vous rappelle, on parle ici de tableaux dynamiques, dont on peut faire varier la taille, dont on peut rajouter ou supprimer des éléments. Pour rajouter un élément à un tableau, il faut bien sûr en indiquer la valeur. Et donc la fonction push_back, elle va recevoir ici la valeur que l'on veut rajouter. Par exemple, si je considère un tableau dynamique "v", tableau dynamique de double,

notes

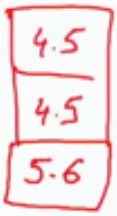
résumé

2m 37s

push_back et pop_back

```
vector<double> v(3, 4.5);

v.pop_back();
v.push_back(5.6);
v.push_back(6.7);
v.pop_back();
```



initialisé au départ avec trois valeurs qui valent toutes 4.5. Par exemple, si je fais un pop_back sur ce tableau "v", pop_back supprime le dernier élément et je me retrouve donc avec un tableau qui n'a plus que deux éléments. Si ensuite je fais un push_back, toujours sur ce même tableau, push_back d'une certaine valeur, pushback* de 5.6, push_back rajoute cet élément derrière le tableau et donc mon tableau "v" contient maintenant 4.5, 4.5 et 5.6.

notes

résumé

3m 1s

Je refais un deuxième push_back ici avec la valeur 6.7, ça veut dire que je rajoute, derrière mon tableau, la valeur 6.7 et le tableau contient donc maintenant les quatre valeurs 4.5, 4.5, 5.6 et 6.7. Enfin, si par exemple je refais un pop_back, pop_back supprime le dernier élément que je venais juste d'ajouter et je me retrouve donc avec ce tableau-ci à trois éléments. avec ce tableau-ci à trois éléments.

notes

résumé

3m 37s