

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-04-3-tableaux-usage-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Ensemble des valeurs d'un tableau. Chapitre des itérations. Nombre de limitations. Boucle for classique. Exemple. Première limitation. Type d'itération. Travers de la variable val. Deuxième limitation. Ensemble de valeurs. Troisième limitation. Itération classique. Dernière limitation. Boucles for. Taille du tableau.



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



[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 : traitements courants

(Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



Attention, les itérations sur ensemble de valeurs :

```
for(Type element : tableau)
```

est très simple et élégant mais :

- ① ► ne permet pas modifier le contenu du tableau
- ② ► ne permet d'itérer que sur un seul tableau à la fois : il n'est pas possible de traverser en une passe deux tableaux pour les comparer par exemple
- ③ ► ne permet l'accès qu'à un seul élément : on ne peut pas par exemple comparer un élément du tableau et son suivant
- ④ ► itère d'un pas en avant seulement.

```
int[] tab = {1, 2, 3};  
for (int val : tab) {  
  
}
```

Nous nous sommes intéressés à comment itérer sur l'ensemble des valeurs d'un tableau. Nous avons vu qu'il était possible de le faire avec une boucle for classique ce qui nous a amené à discuter le problème du calcul de la taille du tableau. Mais avant de clore sur le chapitre des itérations revenons un petit peu sur l'autre type d'itération, les boucles for permettant d'itérer sur un ensemble de valeurs. Nous avons vu que ce genre de boucle for s'écrit ainsi. Nous pouvons constater aisément qu'il s'agit d'une façon très simple et très élégante d'itérer. Nous n'avons même pas ici à nous préoccuper de la taille du tableau. Cependant, il existe un certain nombre de limitations à cette boucle for que nous ne rencontrons pas avec l'itération classique et qui sont les suivantes, que nous allons détailler maintenant. Première limitation : une itération sur ensemble de valeurs ne va pas nous permettre de modifier le contenu du tableau. Donc ici, si j'ai déclaré/initialisé un tableau de cette façon et que je souhaite itérer sur ce tableau au moyen d'une itération sur ensemble de valeurs. Le fait d'écrire quelque chose comme ceci, ne va pas permettre de modifier le contenu du tableau. Je ne peux pas modifier le tableau en ayant recours à une notation de ce genre là. Deuxième limitation : je ne peux parcourir qu'un seul tableau à la fois. Donc je ne peux itérer que sur un seul tableau. Ici, une notation de ce type n'existe pas. Si on imagine qu'on aimerait, par exemple, itérer sur deux tableaux en même temps, par une notation de ce genre. Pour comparer les valeurs du premier tableau avec les valeurs du deuxième tableau, ceci n'existe pas. Nous ne pourrions pas, avec une itération sur ensemble de

notes

résumé

0m 1s



Attention, les itérations sur ensemble de valeurs :

`for(Type element : tableau)`

est très simple et élégant mais :

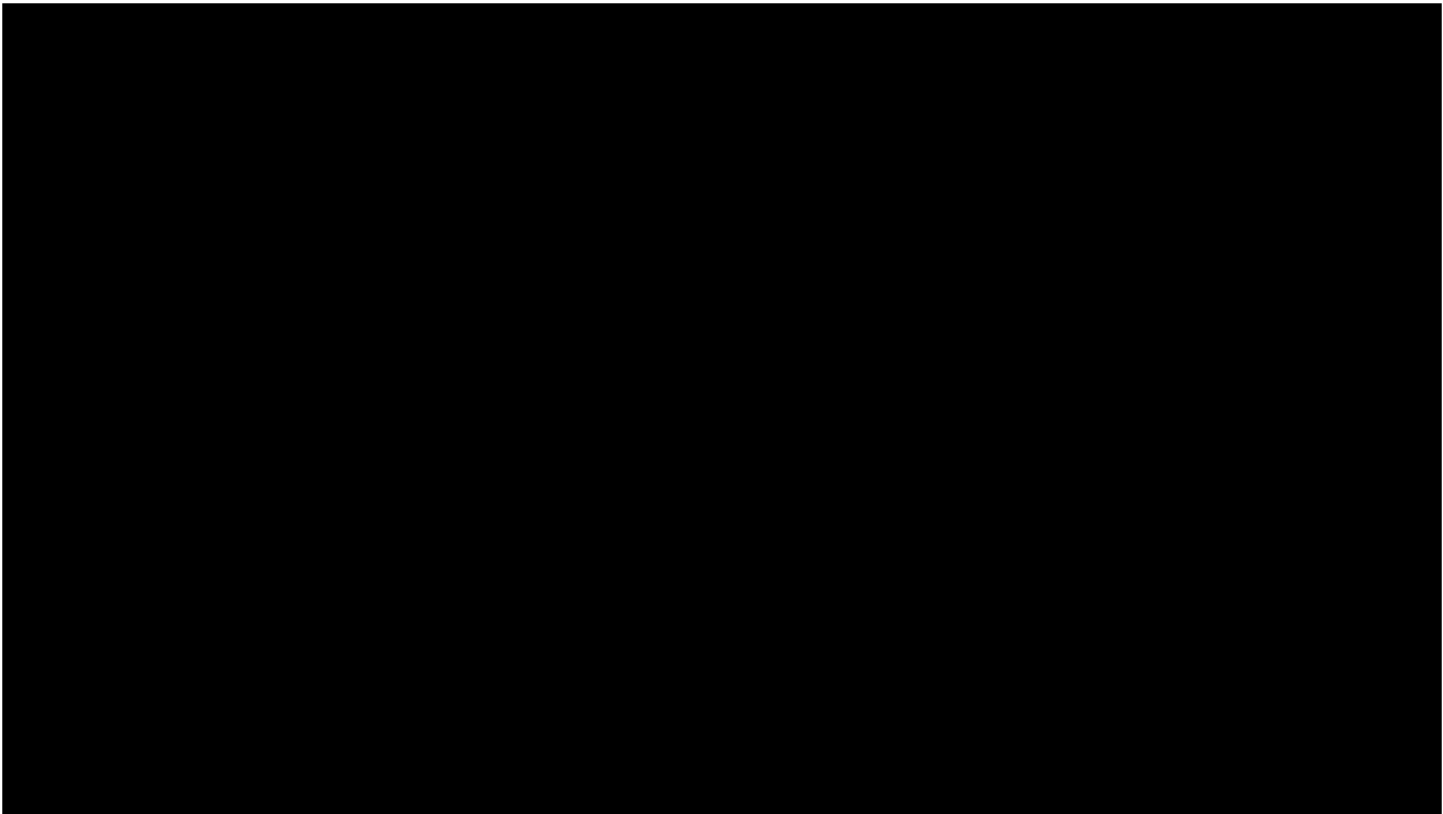
- ① ► ne permet pas modifier le contenu du tableau
- ② ► ne permet d'itérer que sur un seul tableau à la fois : il n'est pas possible de traverser en une passe deux tableaux pour les comparer par exemple
- ③ ► ne permet l'accès qu'à un seul élément : on ne peut pas par exemple comparer un élément du tableau et son suivant
- ④ ► itère d'un pas en avant seulement.

```
int[] tab = {1, 2, 3};  
for (int val : tab) {  
  
}
```

valeurs procéder au parcours simultané de deux tableaux. Troisième limitation : Au travers de la variable val, je ne peux, avec une itération sur ensemble de valeurs, accéder qu'à un seul élément à la fois. Par exemple il ne va pas être possible de comparer un élément du tableau avec son suivant. Je ne peux accéder au travers de val qu'à un seul élément à la fois. Enfin, dernière limitation : je ne peux itérer que d'un pas en avant seulement.

notes

résumé



Si par exemple, j' imagine de pouvoir sauter de deux en deux dans le tableau, ceci n'est pas possible avec une itération sur ensemble de valeurs. A noter que les itérations classiques ne présentent aucune de ces limitations. Il est possible de tout faire avec une itération for classique. une itération for classique.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

2m 13s

