

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-04-2-tableaux-declaration-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Tableau de taille. Moyen de la tournure suivante. Variable de type tableau. Type des valeurs. Valeur de ces éléments. Instruction de déclaration du remplissage. Majorité des cas. Situation suivante. Dehors des indices licites. Cours d'exécution du programme. Deuxième étape. Cases du tableau. Taille de départ. Numéro de la case. Instruction de cette nature.



[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 : déclaration

(Partie 3)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s





Nous venons de voir comment il est possible de déclarer-initialiser en une

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s

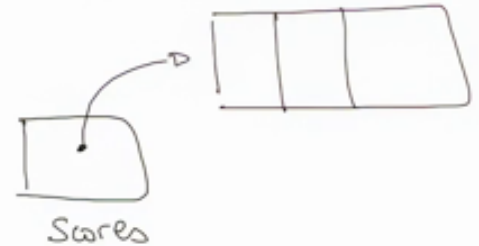


Dans le cas général, on ne connaît pas les valeurs de tous les éléments lors de la déclaration du tableau

On utilise alors plusieurs instructions pour **déclarer et initialiser** :

1. Déclarer le type du tableau
2. Construire le tableau avec :
`new type [taille]`
3. remplir le tableau **élément par élément**

`int [] scores = new int [4];`



seule ligne un tableau de taille fixe en Java. Ceci présuppose que l'on connaît a priori les valeurs à mettre dans le tableau. Dans le cas le plus général, on ne connaît pas ces valeurs et donc il faut séparer l'instruction de déclaration du remplissage effectif du tableau. Voilà comment procéder dans ce cas. Donc on commence par déclarer notre variable de type tableau, ici un tableau de taille fixe d'entiers. Ensuite, il faut allouer les emplacements mémoire, réserver des emplacements mémoire pour notre tableau. Ceci se fait au moyen de la tournure suivante : on utilise le mot réservé « new », on indique ensuite le type des valeurs que l'on veut stocker dans le tableau puis entre crochets, la taille donnée au tableau au moment de sa construction. Alors ce qu'il faut savoir, c'est qu'une fois la taille allouée, cette taille ne peut pas changer en cours d'exécution du programme, c'est pourquoi nous appelons ces tableaux, des tableaux de taille fixe. Donc deuxième étape, nous devons allouer les emplacements mémoire réservés pour le tableau. A ce stade, je suis donc dans la situation suivante : j'ai une variable « score » qui contient une référence, une indirection, vers un

notes

résumé

0m 6s



Valeurs par défaut

Chaque élément d'un tableau reçoit une **valeur par défaut** lors de la construction avec `new`

int	0
double	0.0
boolean	false
char	'\u0000'
(objet quelconque)	(null)

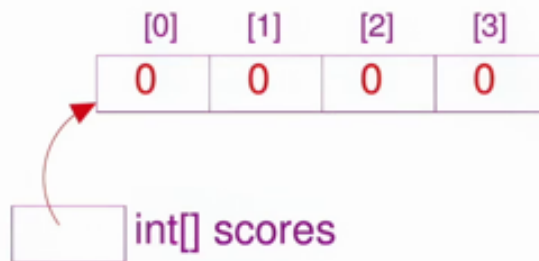


tableau. Ce tableau contient quatre éléments. Quelle va être la valeur de ces éléments à ce stade ? Alors ce qu'il faut savoir, c'est qu'en Java il existe des valeurs par défaut qui sont mises dans les tableaux au moment de sa construction

notes

résumé

1m 25s

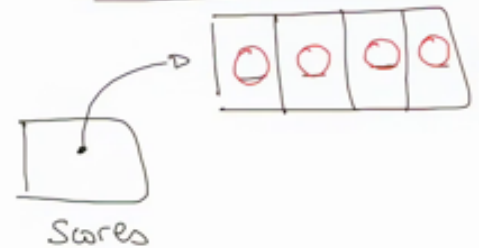


Dans le cas général, on ne connaît pas les valeurs de tous les éléments lors de la déclaration du tableau

On utilise alors plusieurs instructions pour **déclarer et initialiser** :

1. Déclarer le type du tableau
2. Construire le tableau avec :
`new type [taille]`
3. remplir le tableau élément par élément

`int [] scores = new int [4];`



Donc pour la situation qui a précédé nous nous trouvons donc avec un tableau qui contient des zéros. Les valeurs par défaut sont 0 pour les types int et double False pour booléen et il existe d'autres valeurs dédiées pour d'autres types de données. Donc concrètement, lorsqu'une instruction de cette nature est exécutée, nous nous retrouvons dans une situation en mémoire où le tableau a cette allure. Les valeurs par défaut pour chacune des cases sont zéro car je suis en train de travailler avec un tableau d'entiers pour lesquels la valeur par défaut est effectivement zéro. Evidemment, ces valeurs par défaut dans la majorité des cas ne vont pas nous satisfaire, et donc la troisième étape va consister à remplir le tableau élément par élément ; ce qui implique de s'intéresser à comment

notes

résumé

1m 36s



Accès direct aux éléments d'un tableau

Le $i+1^{\text{ème}}$ élément d'un tableau `tab` est accessible au moyen de l'indexation : `tab[i]`

```
int[] tab =
    new int[10];
tab[0]
tab[9]
tab[0] = 4;
```



Attention ! Les indices correspondant aux éléments d'un tableau de taille `T` varient entre 0 et T-1

Le 1^{er} élément d'un tableau `tab` précédemment déclaré est donc `tab[0]` et son 10^e élément est `tab[9]`



Attention ! En cas de débordement une exception est lancée par le programme
 situation d'erreur provoquant l'arrêt du programme si on ne la traite pas
 (la gestion des exceptions n'est pas présentée dans ce cours d'introduction)

Il est impératif que l'élément auquel vous voulez accéder **existe** effectivement !

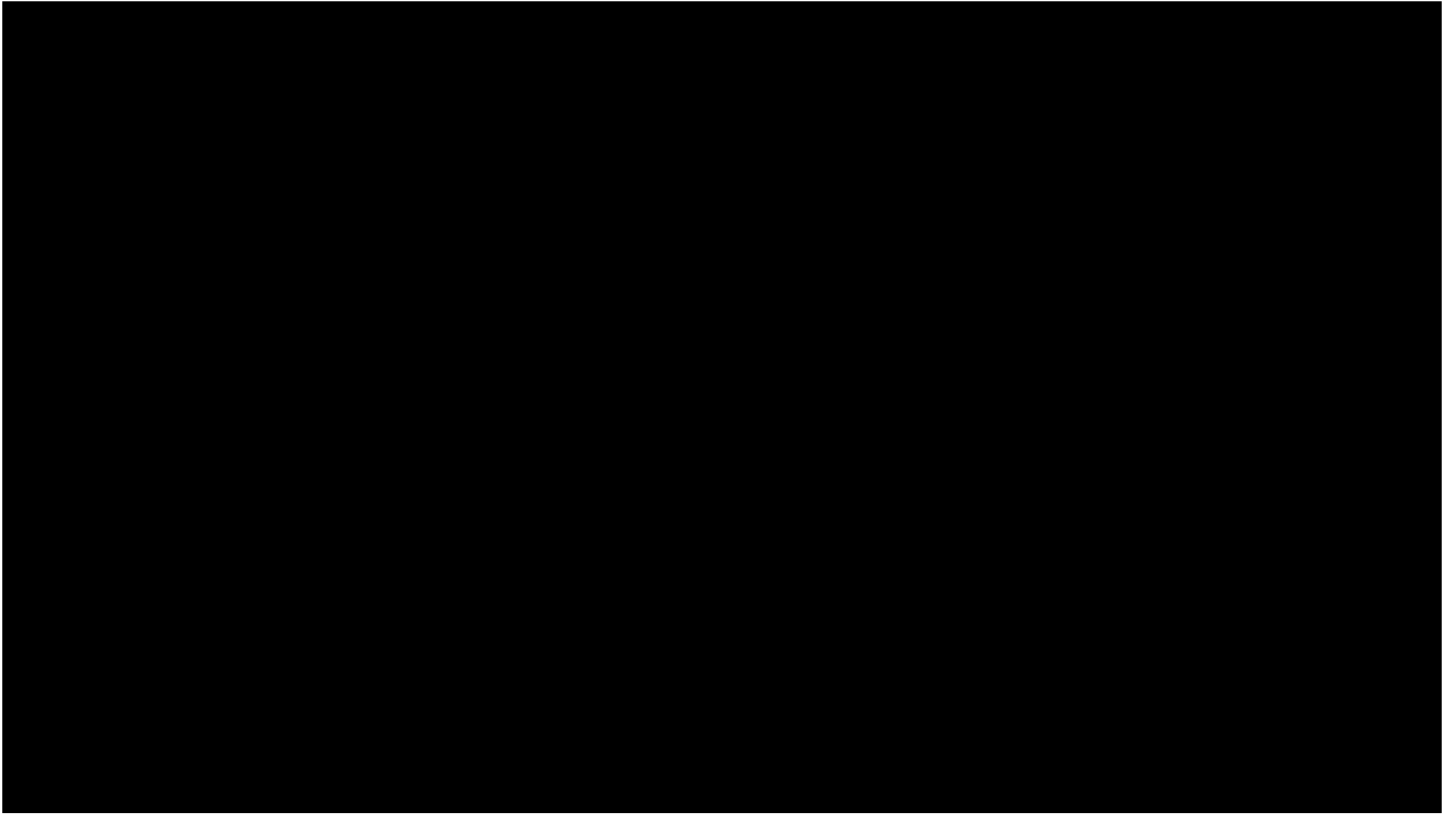
accéder à chacune des cases du tableau pour y mettre une valeur dédiée. Le mécanisme par lequel je peux accéder à la i -ème case d'un tableau est le mécanisme de l'indexation. J'utilise la syntaxe suivante : entre crochets j'indique l'indice, le numéro de la case, que je veux accéder. Ce qu'il faut savoir c'est qu'en Java, l'indice, le numéro de la case, varie entre 0 et $T - 1$, où T est la taille du tableau. Si par exemple, je déclare dans un programme un tableau d'entiers « `tab` » et que je lui alloue une taille de départ qui est de 10, comme ici, alors ce qu'il faut savoir c'est que le premier élément du tableau correspond à l'indice 0 et que le dernier élément du tableau va correspondre à l'indice 9. Je peux par cette notation, altérer, modifier le contenu d'une case donnée, par exemple en écrivant ceci. Alors il est

notes

résumé

2m 25s





important de savoir que si vous débordez au niveau de l'indice du la taille initialement allouée, il y a ce qu'on appelle le « lancement d'une exception », qui à ce stade de votre apprentissage va probablement se traduire dans un programme par l'arrêt abrupte de votre programme. Donc ici par exemple, si j'essaye d'accéder à une case qui est en dehors des indices licites, sachant que le dernier indice licite est 9 pour un tableau de taille 10, on aura une erreur au niveau de l'exécution du programme. programme.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

3m 25s



.....

.....

.....

.....

.....