

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-04-4-tableaux-affectation-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Contenus des tableaux. Travers de références. Raison de la première condition. Référence particulière. Égalité des contenus des tableaux. Valeur spéciale. Tableaux. Sémantique de l'opérateur de comparaison. Intégralité du tableau. Taille différente. Nombre d'instructions. Affectation de cette nature. Moyen d'une itération. Null. 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 : affectation et comparaison

(Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

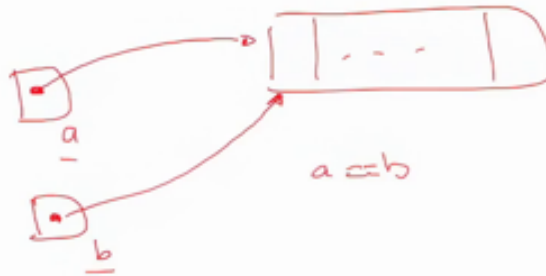
0m 0s



L'opérateur `a == b` teste si les variables `a` et `b` référencent le même emplacement mémoire.

⇒ ce qui est donc le cas lors de l'affectation `b = a;`

L'opérateur `a == b` **ne teste pas** l'égalité des valeurs contenues dans les tableaux pointés par `a` et `b` !



Il est maintenant aisé de comprendre que le fait que les tableaux en Java soient manipulés au travers de références a également un impact sur la sémantique de l'opérateur de comparaison `==`. Si « `a` » et « `b` » sont deux tableaux, le fait de les comparer au moyen de `==` va, bien évidemment, comparer des références, et non pas les contenus des tableaux. Donc, ici je suis dans la situation où j'ai `a`, qui est un tableau, donc qui contient une référence vers le tableau, pareil pour « `b` », donc `a == b` ne retournera `true` que si effectivement la même référence est contenue dans les objets « `a` » et « `b` ». Donc ils pointent vers le même emplacement en mémoire,

notes

résumé

0m 1s



Pour vérifier l'égalité de contenu des tableaux,
il faut écrire explicitement les tests :

```
→ if (a == null || b == null || a.length != b.length) {  
    System.out.println("contenus différents ou nuls");  
}  
else {  
    int i = 0;  
    → while(i < a.length && (a[i] == b[i])) {  
        ++i;  
    }  
    → if (i >= a.length) {  
        System.out.println("contenus identiques");  
    }  
    else {  
        System.out.println("contenus différents");  
    }  
}
```

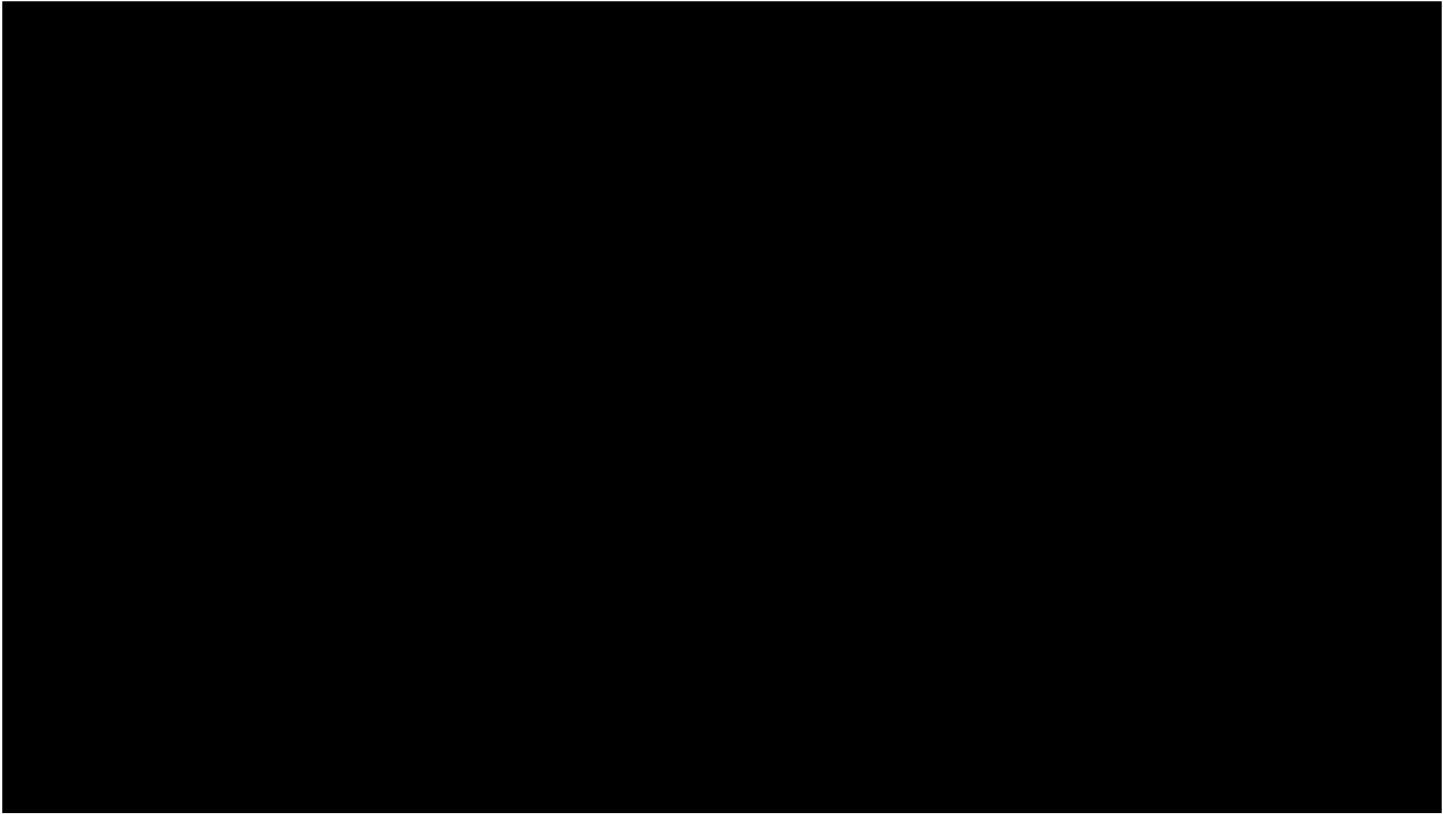
ce qui est évidemment le cas lorsque, au préalable, j'ai réalisé une affectation de cette nature. Donc l'opérateur == appliqué au tableau ne teste en aucun cas l'égalité des contenus des tableaux pointés par « a » et « b ». Si l'on veut comparer des contenus, il faut comme tout à l'heure, pour l'affectation, procéder au moyen d'une itération. Donc si l'on veut plutôt comparer le contenu de deux tableaux, il va falloir écrire un certain nombre d'instructions pour tester si c'est le cas ou pas. On peut commencer, par exemple, par tester si les tableaux sont de même longueur ; si ce n'est pas le cas, on est certain que les contenus ne sont pas identiques. On prend habituellement également la précaution de comparer les tableaux avec la référence particulière, le mot réservé particulier qui est « null ». « null » est une valeur spéciale qui peut être affectée à un tableau et qui indique que le tableau ne référence rien du tout. On prend habituellement cette précaution aussi. Donc si les tableaux ne référencent rien, ou s'ils sont de taille différente, alors nous pouvons d'emblée dire que les contenus sont différents. Sinon, on peut supposer qu'éventuellement ils ont un même contenu, mais il va falloir le tester, et il va falloir parcourir l'intégralité du tableau et tester case à case si les deux contenus sont identiques. Si l'on sort de la boucle en raison de la première condition,

notes

résumé

0m 49s





ceci veut dire qu'on a parcouru l'intégralité du tableau et que la condition était toujours vérifiée case à case, et donc on peut affirmer que les contenus sont identiques. Sinon, eh bien, nous sommes dans la situation où les contenus sont différents. situation où les contenus sont différents.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 13s



.....

.....

.....

.....

.....