

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-02-2-Conditions-pt4

Concepts (extraits des sous-titres générés automatiquement) :

Opérateur logique. Évaluation de ce premier opérande. Exemples d'utilisation de l'opérateur. Petit programme. Petite illustration concrète de l'utilisation de l'opérateur. Évaluation de cet opérande. Non logique. Évaluation de ce second opérande. Ensemble de l'expression. Opérateur unaire. Cas particulier. Évaluation de l'expression globale. Négation de la valeur de cet opérande. Dernier opérateur logique usuel. Évaluation de cet opérande retourne.



[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/>

Conditions

(Partie 4)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



Exemple avec l'opérateur logique ||

```
System.out.println("Entrez deux valeurs:");
int n = scanner.nextInt();
int m = scanner.nextInt();

if ((m >= 0) || (n >= 0)) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}
```



Une petite illustration concrète de l'utilisation de l'opérateur logique OU, maintenant, supposons ici que nous ayons un petit programme qui

notes

résumé

0m 1s



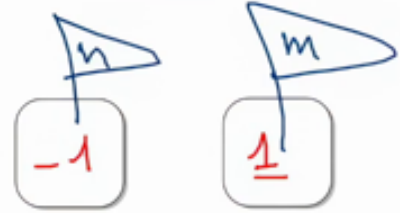
Exemple avec l'opérateur logique ||

```

System.out.println("Entrez deux valeurs:");
int n = scanner.nextInt();
int m = scanner.nextInt();

if ( (m >= 0) || (n >= 0) ) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}

```



demande à l'utilisateur d'introduire deux valeurs, et nous souhaitons que l'une ou l'autre au moins de ces valeurs soit positive. Les deux valeurs sont lues depuis l'entrée standard et stockées dans deux variables n et m. Nous souhaitons donc que l'une ou l'autre des valeurs introduites soit supérieure à zéro, nous allons donc exprimer une condition, il nous suffit que soit m soit supérieur ou égal à zéro, soit n soit supérieur ou égal à zéro. Puisqu'il suffit que l'une ou l'autre des conditions soit vérifiées, nous allons naturellement utiliser l'opérateur logique OU. Supposons que l'utilisateur introduise les valeurs -1 et 1, l'évaluation de ce premier opérande va retourner "true" puisque la valeur 1 est bel et bien supérieur ou égal à zéro, et nous avons vu qu'il

notes

résumé

0m 13s



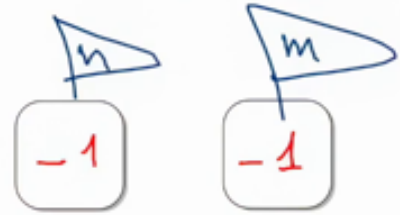
Exemple avec l'opérateur logique ||

```

System.out.println("Entrez deux valeurs:");
int n = scanner.nextInt();
int m = scanner.nextInt();
if ( (m >= 0) || (n >= 0) ) {
    System.out.println("au moins une valeur est positive");
} else {
    System.out.println("les deux valeurs sont negatives");
}

```

Handwritten annotations: *false* above $(m \geq 0)$, *false* above $(n \geq 0)$, and *false* above the $||$ operator.



suffit que l'un ou l'autre des opérandes soit "true" pour que l'ensemble de l'expression soit "true" aussi, par conséquent, nous allons ici brancher sur le bloc positif vrai de l'instruction if, et indiquer à l'utilisateur par un message qu'effectivement au moins une des valeurs introduites est positive. Supposons maintenant que l'utilisateur ait introduit les valeurs -1 et -1, donc deux valeurs négatives. Dans ce cas l'évaluation de cet opérande va retourner évidemment "false", l'évaluation de ce second opérande retourne aussi "false", et dans ce cas, nous sommes dans la situation où l'évaluation de l'expression globale retourne "false" puisque les deux opérandes valent "false". Dans ce cas particulier, nous allons donc

notes

résumé

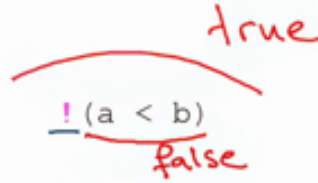
1m 1s



Les opérateurs logiques

L'opérateur logique ! (NON):

par exemple, la condition



 $!(a < b)$

est vraie si $(a < b)$ est fausse, et fausse si $(a < b)$ est vraie.

Nous verrons des exemples d'utilisation de cet opérateur plus loin dans la suite du cours.

brancher sur le bloc else de l'instruction conditionnelle, et signifier à l'utilisateur par un message que aucune des deux valeurs introduites n'est positive. Dernier opérateur logique usuel, le NON logique, la négation, qui en Java s'exprime au moyen du point d'exclamation. Contrairement aux deux opérateurs que nous venons de voir, l'opérateur NON, la négation, est un opérateur unaire, c'est-à-dire qu'il n'attend qu'un seul opérande. Il a pour vocation de retourner la négation de la valeur de cet opérande. Par exemple, supposons que l'évaluation de cet opérande retourne "false", ce qui signifie ici concrètement que a serait supérieur ou égal à b, la négation de la valeur "false" va être "true".

notes

résumé

1m 49s



Supposons maintenant que l'évaluation de cet opérande retourne au contraire "true", dans ce cas, la négation de "true" va être "false". Nous aurons l'occasion d'examiner plusieurs exemples d'utilisation de l'opérateur négation dans la suite du cours. suite du cours.

2m 37s

