

Support de cours

Cours:

## Initiation à la programmation (en Java)

Vidéo:

### Init-JAVA-03-2-IterationsApprofondissementExemple-pt4

Concepts (extraits des sous-titres générés automatiquement) :

**Boucles imbriquées. Boucle for. Instruction d'incrémentation. Variable i. Nouvelle instruction. Exemple de la table de multiplication. Nombre de notes. For i. Instruction d'affichage. Ensemble des tables. Variable j. Nouvelle fois. Solution possible. Branchement conditionnel. Tables de multiplication.**



[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/>



# Itérations : approfondissement et exemples

## (Partie 4)

### Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s





Une **solution**:

```
System.out.println("Entrez le nombre de notes");
int nombre_de_notes = clavier.nextInt();

double somme = 0;

if (nombre_de_notes > 0) {
    for(int i = 1; i <= nombre_de_notes; ++i) {
        System.out.println("Entrez la note numero " + i);
        double note = clavier.nextDouble();
        somme = somme + note;
    }
    System.out.println("Moyenne = " + somme / nombre_de_notes);
}
```

Le bug peut survenir, va survenir quand l'utilisateur entre la valeur 0 pour le nombre de notes. Donc, si je ne fais pas attention, ici, je vais diviser par 0. Alors, pour éviter ça, il y a plusieurs solutions. Une solution possible, c'est de tester, avant de faire le calcul de la moyenne, si le nombre de notes est égal à 0 ou pas. Alors, je peux modifier mon programme, de cette façon, ici, c'est-à-dire que j'ai ajouté un branchement conditionnel, qui va tester si le nombre de notes est positif, strictement, c'est-à-dire, qu'il n'est pas égal à 0. Dans ce cas-là, et bien je vais faire le calcul, comme précédemment.

notes

résumé

0m 1s






Et puis je n'ai plus qu'à fermer l'accolade, ici. Et donc, si le nombre de notes est égal à 0, je ne vais pas exécuter cette division.

[illegible]

résumé

0m 49s







# Boucles imbriquées

Reprenons l'exemple précédent de la table de multiplication par 5:

```
for(int i = 1; i <= 10; ++i) {  
    System.out.println("5 multiplie par " + i + " vaut " + 5 * i);  
}
```

Supposons qu'on veuille maintenant afficher toutes les tables de multiplication, de 2 à 10.

Il suffit de mettre la boucle précédente dans une autre boucle, et de remplacer le 5 par...ce qu'il faut.

Terminons par ce qu'on appelle les boucles imbriquées. Alors, pour cela, je vais revenir sur l'exemple de la table de multiplication par 5, qui a utilisé cette «boucle for»

notes

résumé

1m 5s





# Boucles imbriquées

```
for(int j = 2; j <= 10; ++j) {  
    for(int i = 1; i <= 10; ++i) {  
        System.out.println("5 multiplie par " + i + " vaut " + 5 * i);  
    }  
}
```

affiche 9 fois la table de multiplication par 5

pour afficher la table de 5. Et je vais supposer, maintenant, que je veuille afficher toutes les tables de multiplication de 2 jusqu'à 10. Et bien, il me suffit de mettre ce code à l'intérieur d'une autre «boucle for», et de le modifier légèrement pour obtenir l'ensemble des tables de 2 jusqu'à 10. Alors, je vais commencer par mettre ce code, tout simplement, dans une «boucle for». C'est ce que j'ai fait, ici. Ici, j'ai recopié le code précédent.

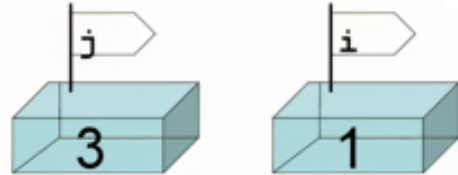
notes

résumé

1m 14s







```
for(int j = 2; j <= 10; ++j) {
    System.out.println("Table de multiplication par " + j + ": ");
    → for(int i = 1; i <= 10; ++i) {
        System.out.println(j + " multiplie par " + i + " vaut " + j * i);
    }
}
```

```
Table de multiplication par 2:
2 multiplie par 1 vaut 2
2 multiplie par 2 vaut 4
...
2 multiplie par 10 vaut 20
Table de multiplication par 3:
|
```

Donc, celui qui m'affiche la table de multiplication par 5, que j'ai mis dans une autre «boucle for», ici, qui va faire varier une variable  $j$  de 2 jusqu'à 10. Donc,  $j$  va prendre neuf valeurs, de 2 à 10. Donc, l'ensemble de ce code va m'afficher neuf fois la table de multiplication par 5. Comme maintenant, je veux plutôt toutes les tables de 2 jusqu'à 10, et bien, je n'ai qu'à modifier ce 5 ci, pour afficher plutôt la valeur de la variable  $j$ , puisque  $j$  varie de 2 à 10. Et je vais aussi modifier ce 5, surtout, pour utiliser la variable  $j$ , dans cette expression. Je vais, donc, obtenir ce code-ci, où j'ai maintenant  $j$ , ici, ainsi que  $j$  dans cette expression. Et je dis maintenant que l'ensemble de ce code m'affiche les tables de 2 à 10. C'est ce que nous allons vérifier, maintenant, en détaillant ce code pas à pas. On commence, donc, avec une première «boucle for» qui déclare une variable  $j$ , qui l'initialise à 2. La condition est « $j \leq 10$ », et l'instruction d'incrément est « $++j$ ». Donc,  $j$  va prendre les valeurs 2, 3 jusqu'à 10. Et pour l'instant,  $j$  va être initialisé à 2. On entre, maintenant, dans cette «boucle for», qui commence par cette nouvelle instruction que j'ai ajoutée pour afficher le nombre de la table, qu'on va afficher, c'est-à-dire que je vais afficher : «table de multiplication par», ensuite, la valeur de  $j$ , ensuite «:». Comme  $j$  vaut 2, je vais simplement afficher 2, pour l'instant. On entre, maintenant, dans cette «boucle for» qui utilise une variable  $i$ , et qui l'initialise à 1. La condition est « $i \leq 10$ ». L'instruction d'incrément est « $++i$ ».  $i$  va donc, prendre les valeurs 1, 2, jusqu'à 10. Et pour l'instant,  $i$  contient la valeur 1. On entre donc dans cette «boucle for», c'est-à-dire qu'on va exécuter cette instruction d'affichage, qui affiche la valeur

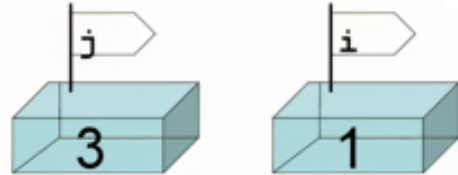
## notes

## résumé

1m 49s







```
for(int j = 2; j <= 10; ++j) {
    System.out.println("Table de multiplication par " + j + ": ");
    → for(int i = 1; i <= 10; ++i) {
        System.out.println(j + " multiplie par " + i + " vaut " + j * i);
    }
}
```

```
Table de multiplication par 2:
2 multiplie par 1 vaut 2
2 multiplie par 2 vaut 4
...
2 multiplie par 10 vaut 20
Table de multiplication par 3:
|
```

de j, c'est-à-dire 2. Ensuite «multiplié par», la valeur de i, c'est-à-dire, pour l'instant, 1, ensuite «vaut», ensuite, la valeur de cette expression qui est 2 fois 1, C'est-à-dire 2. C'est la seule instruction qui est dans la boucle «for i». On va, donc, revenir, ici, pour exécuter l'instruction d'incrémentement ; i va prendre maintenant la valeur 2. La condition est toujours vraie. Donc, on entre une nouvelle fois dans la boucle «for i», c'est-à-dire qu'on va exécuter cette instruction, cette fois-ci, avec i qui vaut 2 ; j vaut toujours 2. On obtient l'affichage «multiplié par» ; on affiche la valeur de i, qui vaut, donc, 2 maintenant. On affiche «vaut» et la valeur de cette expression qui vaut maintenant, 2 fois 2, c'est-à-dire 4. Et ainsi de suite, jusqu'à ce que i prenne la valeur 10. On va exécuter l'instruction d'incrémentement «++i», qui va faire que i prenne la valeur 11. La condition va devenir fausse. Donc, on va sortir de cette boucle, c'est-à-dire arriver ici. On arrive maintenant à la fin de la boucle «for j». On revient donc maintenant ici, pour exécuter l'instruction d'incrémentement «++j», qui va donner la valeur 3 à j, puisque j vaut 2. La condition est toujours vraie. Et on entre, une nouvelle fois, dans la boucle «for j», c'est-à-dire qu'on recommence par exécuter, le corps de la boucle «for j». On affiche ce message-ci, avec cette fois-ci, j qui vaut 3. Et on rentre une nouvelle fois dans cette «boucle for». Et on recommence avec i, qui vaut 1. L'instruction à

#### notes

#### résumé



l'intérieur du corps de la boucle «for i» va être exécutée. Cette fois-ci avec j qui vaut 3. On va obtenir l'affichage «multiplié par», i vaut 1 de nouveau, et on va afficher la valeur de l'expression «j \* i», c'est-à-dire 3 fois 1, donc, simplement 3. Et ainsi de suite, jusqu'à ce que i prenne la valeur 10. On va donc, pour chaque nouvelle valeur de j, dérouler complètement la boucle la plus interne, c'est-à-dire, la boucle «for i». i va varier de 1 à 10, pour afficher l'ensemble de la table de multiplication, pour la valeur courante de j, jusqu'à ce que j prenne la valeur 10. ce que j prenne la valeur 10.

6m 37s

