Support de cours

Cours:

## Initiation à la programmation (en Java)

Vidéo:

## Init-JAVA-03-1-IntroIteration-pt1

Concepts (extraits des sous-titres générés automatiquement) :

**Boucle for. Instructions d'affichage. Mot-clé for. Seule instruction d'affichage. Tel cas. Instruction d'incrémentation. Nombre de tour de boucle. Initialisation d'une variable. Branchements conditionnels. Mot-clé if. Ensemble de cet affichage. Deuxième structure de contrôle. Opérateur d'incrémentation. Bloc d'instruction. Corps de la boucle.**

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

vers la vidéo

EPFL

# Itérations : introduction
(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s


deuxième structure de contrôle, qui sont les itérations.

notes

résumé

0m 1s

# La boucle `for`

Une boucle `for` permet de répéter un nombre donné de fois la même série d'instructions.

Par exemple, si on fait:

```
for(int i = 0; i < 5; ++i) {
  System.out.println("le carre de " + i + " vaut " + i * i);
}
```

le programme affichera les carrés des 5 premiers entiers:

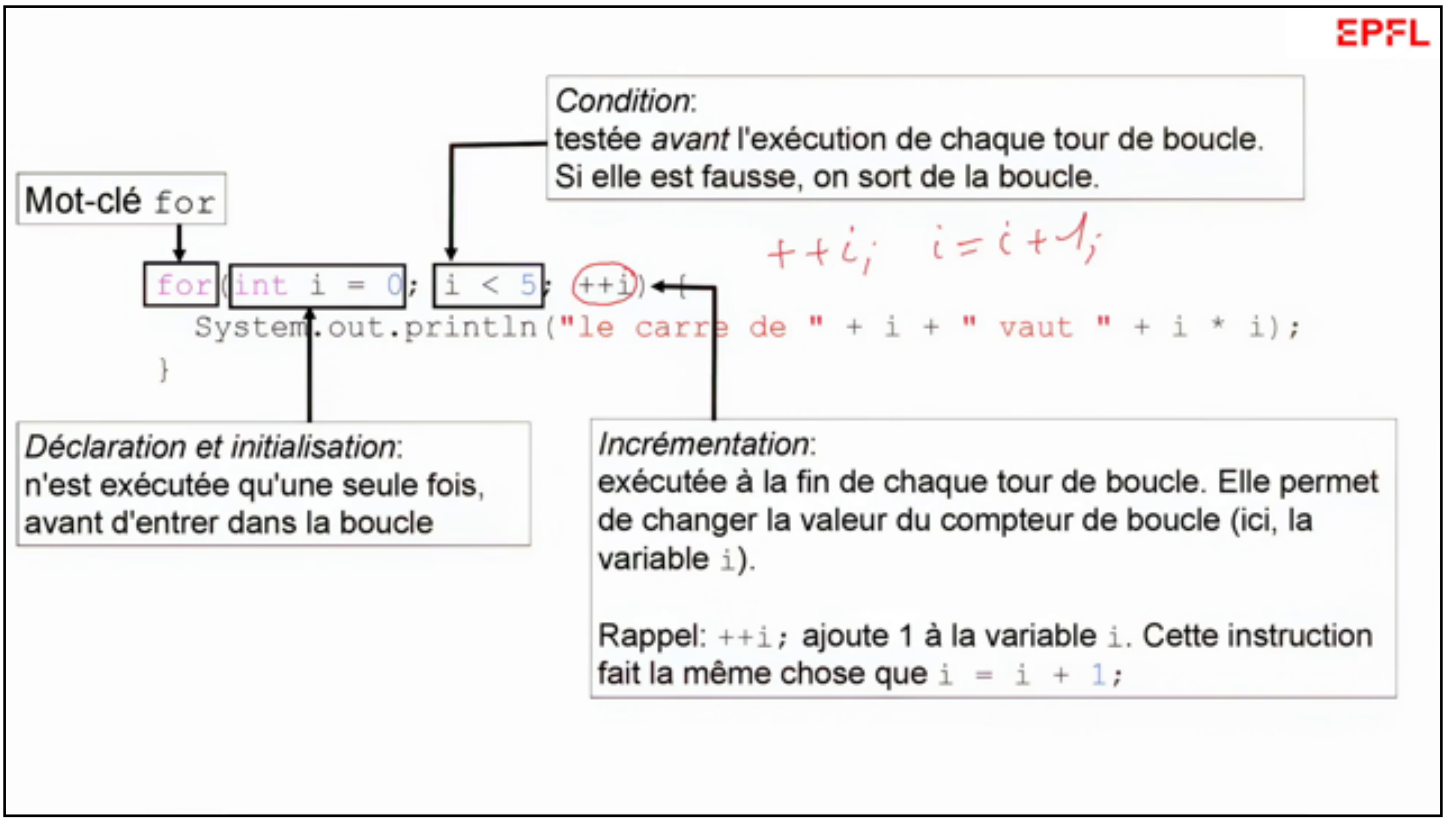
```
le carre de 0 vaut 0
le carre de 1 vaut 1
le carre de 2 vaut 4
le carre de 3 vaut 9
le carre de 4 vaut 16
```

Alors à quoi va servir cette nouvelle structure de contrôle? Supposons par exemple que je veuille afficher le carré des cinq premiers entiers, comme ici, c'est-à-dire que je veuille afficher le carré de zéro vaut zéro, le carré de un vaut un, le carré de deux vaut quatre, jusque le carré de quatre vaut seize. On pourrait, pour obtenir cet affichage, utiliser cinq instructions d'affichage, mais en fait, dans un tel cas, on peut, je dirais même on doit, utiliser une itération ou ce qu'on appelle aussi aussi une boucle for, c'est-à-dire

notes

résumé

0m 6s




qu'en fait ce code va permettre d'obtenir l'ensemble de cet affichage en bouclant ou en itérant sur cette seule instruction d'affichage. Et je vais maintenant détailler ce code pour vous expliquer pourquoi. Une itération, ou ce qu'on appelle donc aussi, une boucle for, commence tout simplement avec le mot-clé for, vient ensuite la déclaration et l'initialisation d'une variable qui va servir à contrôler le nombre de tour de boucle. Cette déclaration et initialisation n'est exécutée qu'une seule fois avant d'entrer dans la boucle. Vient ensuite quelque chose que vous pouvez reconnaître, puisqu'il s'agit d'une condition, que vous avez rencontrée dans la vidéo sur les branchements conditionnels. Cette condition sera testée avant d'entrer dans la boucle. Si elle est vraie, on va continuer à exécuter la boucle, si elle est fausse, on va sortir de la boucle. Vient ensuite une instruction d'incrémentation, qui dans cet exemple s'écrit ++i. Je vous rappelle quand on utilise l'opérateur d'incrémentation ++ sur la variable i, c'est équivalent à écrire i = i +1, c'est-à-dire que cette instruction va ajouter un à la variable i.

notes

résumé

0m 49s

```
for(int i = 0; i < 5; ++i){
  System.out.println("le carre de " + i + " vaut " + i * i);
}
```

*Corps de la boucle:*
Bloc d'instructions qui seront exécutées à chaque tour de boucle.

Cette instruction d'incrémentation va permettre de faire évoluer la variable qui permet de contrôler le nombre de tour de boucle, et elle n'est exécutée qu'à la fin de chaque tour de boucle. La déclaration et l'initialisation de la variable qui sert à contrôler le nombre de tour de boucle, la condition et l'instruction d'incrémentation sont entourées par des parenthèses et sont séparées par des points-virgule. Vient ensuite un bloc d'instruction, qui

notes

résumé

2m 13s

Comme pour le `if`, les accolades ne sont obligatoires que si plusieurs instructions doivent être répétées.

Si il n'y a qu'une seule instruction, on peut ne pas utiliser d'accolades:

```
for(int i = 0; i < 5; ++i)
  System.out.println("i = " + i);
```

Mais, toujours comme pour le `if`, il est conseillé de garder les accolades:

```
for(int i = 0; i < 5; ++i) {
  System.out.println("i = " + i);
}
```

constitue le corps de la boucle, et qui est, qui contient les instructions qui vont être exécutées à chaque tour de boucle. Comme pour le branchement conditionnel, qui utilise le mot-clé if, les accolades ne sont obligatoires que quand plusieurs instructions doivent être répétées, c'est-à-dire quand le corps de la boucle contient plusieurs instructions.

notes

résumé

2m 49s

C'est-à-dire que si je n'ai qu'une seule instruction à répéter, comme ici, je ne suis pas obligé de mettre des accolades ici et ici. En revanche, même dans ce cas, il est conseillé d'utiliser des accolades, c'est-à-dire mettre une accolade ouvrante avant l'instruction et une accolade fermante après l'instruction. l'instruction.

notes

résumé

3m 13s