

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-02-3-ErreurstypeBool-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Branchements conditionnels. Premier piège. Compilateur java. Intérieur du branchement conditionnel. Deuxième exemple d'erreur. Instruction vide. Seul symbole égal. Symboles égal. Bonne idée. Dehors du branchement conditionnel. Première instruction. Troisième erreur. Version correcte. Branchement conditionnel. Programmeurs débutants.



[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/>

Erreurs de débutant

Le type `boolean` (Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s





Voyons maintenant plusieurs erreurs commises fréquemment par

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



.....

.....

.....

.....

.....

Erreurs classiques

Le test d'égalité s'écrit \equiv , et pas $=$ affect

```
if (a = 1) // !!!
```

n'est pas accepté par le compilateur.

des programmeurs débutants quand il s'agit d'écrire des branchements conditionnels. Le premier piège qui vous guette est que le test d'égalité s'écrit avec deux symboles égal comme ici et non pas un seul symbole égal qui est pour l'affectation.

notes

résumé

0m 6s



Erreurs classiques

```
if (a == 1);  
    System.out.println("a vaut 1");
```



Ce code-ci est donc incorrect puisqu'il n'y a qu'un seul symbole égal, la version correcte est évidemment: `if (a = = 1)`, et ce code-ci qui est incorrect n'est tout simplement pas accepté par le compilateur Java. Voici un deuxième exemple d'erreur, peut-être voyez-vous le problème? En fait il ne devrait pas y avoir de

notes

résumé

0m 25s



Erreurs classiques

```
[ if (a == 1) ; // !!!
  System.out.println("a vaut 1");
```

a vaut 1 est toujours affiché quelle que soit la valeur de a!

Le point-virgule est considéré comme une instruction, qui ne fait rien.

Le code précédent est compris par le compilateur comme:

```
if (a == 1)
;
System.out.println("a vaut 1");
```

l'instruction `System.out.println("a vaut 1");` est donc située après le `if`.

point-virgule ici et si on exécute ce code, il sera accepté par le compilateur et le message "a vaut 1" sera affiché quelque soit la valeur de a, donc c'est un petit peu mystérieux, mais pour comprendre pourquoi il faut savoir que le point-virgule est considéré comme une instruction, une instruction qui ne fait rien. Ce code ci, j'aurais pu donc l'écrire de cette façon-là, qui est plus facile à interpréter, c'est-à-dire que le point-virgule, cette instruction vide, est considérée à l'intérieur du branchement conditionnel, mais cette instruction est considérée en dehors du branchement conditionnel. Si a vaut 1, on va rentrer dans le branchement conditionnel, exécuter l'instruction vide, c'est-à-dire ne

notes

résumé

0m 49s



Si on utilise des accolades même quand il n'y a qu'une instruction dans le bloc, et qu'on écrit le test de la façon suivante:

```
if (a == 1) {  
→ System.out.println("a vaut 1");  
}
```

l'erreur précédente a beaucoup moins de chance d'arriver.

rien faire et quelque soit la valeur de a sauter à cette instruction-ci et afficher le message "a vaut 1". Notez au passage que si nous avons utilisé des accolades alors qu'il n'y a qu'une seule instruction dans le branchement conditionnel, l'erreur que nous venons de voir aurait eu beaucoup moins de chance d'arriver.

notes

résumé

1m 37s



Erreurs classiques

Ne pas oublier les accolades, l'indentation ne suffit pas:

```
if (n < p)
→ System.out.println("n est plus petit que p");
  max = p;
else
  System.out.println("n est plus grand ou egal a p");
```

génère à la compilation l'erreur:

error: 'else' without 'if'

Voici une meilleure présentation du code précédent:

```
if (n < p)
- System.out.println("n est plus petit que p");

max = p;
else
  System.out.println("n est plus grand ou egal a p");
```

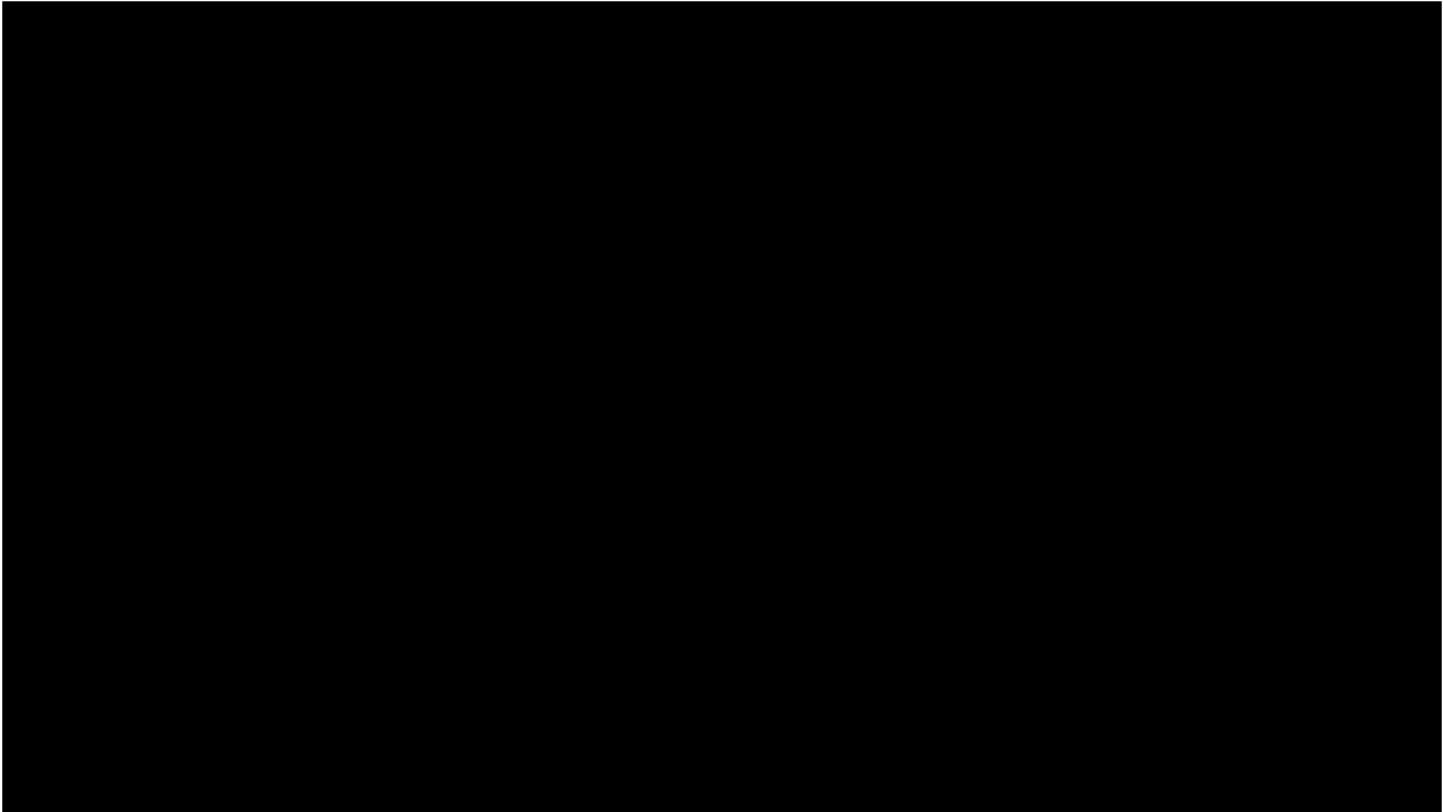
C'est donc une bonne idée d'utiliser systématiquement des accolades, même quand il n'y a qu'une seule instruction dans le bloc. Une troisième erreur est, justement, d'oublier les accolades quand elles sont nécessaires comme dans ce code-ci. Le fait d'avoir indenté ces lignes, c'est-à-dire de les avoir décalées un peu vers la droite, ne suffit pas à ce qu'elles soient à l'intérieur du branchement conditionnel contrairement à d'autres langages. Alors à votre avis que se passe-t-il quand j'essaie de compiler ce code? Et bien, le compilateur va m'afficher ce message d'erreur qui me dit qu'il y a un else sans le mot-clé if. Alors pourquoi? Parce qu'en fait ce code, j'aurais pu l'écrire de cette façon-ci, c'est-à-dire que la première instruction qui est l'instruction d'affichage est bien à l'intérieur

notes

résumé

2m 1s





du branchement conditionnel mais cette deuxième instruction n'y est pas, elle est après le branchement conditionnel et quand le compilateur rencontre ce else, il est déjà sorti du branchement conditionnel qui est ici et ce else n'est donc, pour lui, pas rattaché à une instruction if. C'est pour ça qu'il génère ce message d'erreur.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

2m 49s

