Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-03-5-Blocs-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Variables déclarées. Variables locales. Instruction if. Variable j. Variables globales. Première ligne. Variables d'instance. Exemple précédent. Variables de classe. Bloc. Variables. Moment. Utilisation. Besoin. Conseil.

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

vers la vidéo

EPFL

Fonctions : blocs
(Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s

Notion de portée

- Les variables déclarées à l'intérieur d'un bloc sont appelées **variables locales** (au bloc). Elles ne sont accessibles qu'à l'intérieur du bloc.

```
if (i != 0) {
  int j = 0;

  ...
  j = 2 * i;
  ...
}
// A partir d'ici, on ne peut plus utiliser j
```

Des variables déclarées comme ceci dans un bloc et qui ne sont accessibles donc que dans ce bloc, c'est ce qu'on appelle des

notes

résumé

0m 1s

Notion de portée

- Les variables déclarées à l'intérieur d'un bloc sont appelées **variables locales** (au bloc). Elles ne sont accessibles qu'à l'intérieur du bloc.

- Les variables déclarées en dehors de `main` sont de portée **globales** (à la classe).
  Elles sont accessibles dans toute la classe.

Bonne pratique: Déclarer les variables au plus près de leur utilisation.

variables locales, elles sont locales à ce bloc. On peut aussi déclarer des variables hors de tout bloc, dans la classe dans laquelle on est en train de travailler, c'est ce qu'on appelle les variables globales à la classe. On distinguera dans le cours orienté objet, les variables d'instance, des variables de classe, mais ça nous emmènerait ici, un petit peu trop loin. Le conseil qu'on peut par contre vous donner ici c'est

notes

résumé

0m 13s

Notion de portée

Déclarer les variables au plus près de leur utilisation

Par exemple, si la variable `j` n'est pas utilisée après la condition,

écrivez:

```
if (i != 0) {
    int j = 0;
    ...
    j = 2 * i;
    ...
}
```

plutôt que:

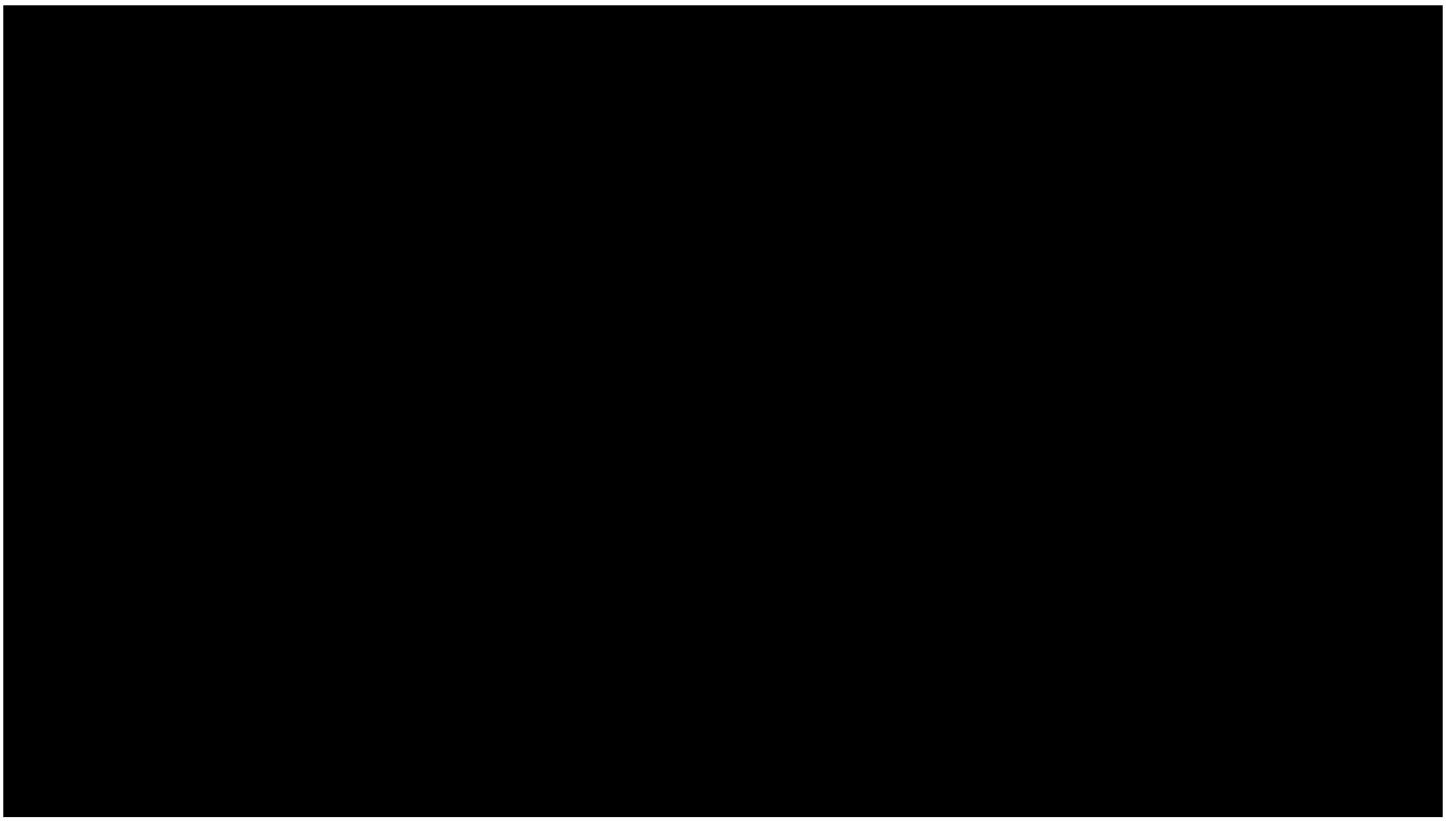
```
int j = 0;
if (i != 0) {
    ...
    j = 2 * i;
    ...
}
```

de déclarer vos variables au plus proche de leur utilisation. Qu'est-ce-que cela veut dire? Prenons un exemple. Si je reprends l'exemple précédent avec le bloc ici contrôlé par une instruction if, dans lequel j'utilise la variable j, mais on suppose que je n'utilise donc plus la variable j après ce bloc. A ce moment là, je mets donc la déclaration ici au plus proche de son utilisation, donc dès que j'en ai besoin, avant la première ligne où je vais utiliser l'instruction j. Par opposition, donc dans la même situation où j'ai besoin ici d'une variable j qui n'est plus utilisée après le bloc contrôlé par l'instruction if, si je déclarais ici savoir cette variable, je peux tout à fait le faire, le déclarer avant le bloc, ça ne serait pas déclarer la variable au plus proche de son utilisation, puisque cette variable j

notes

résumé

0m 37s



serait déclarée hors du bloc et pourrait donc être utilisée ensuite, par la suite. L'idée c'est de la déclarer le plus près de l'endroit où on l'utilise. de l'endroit où on l'utilise.

notes

résumé

1m 25s