

Java

Cours 4

Benoit Simard



Les conditions

if / else

La condition doit renvoyer un **boolean true / false**

```
1 ---  
2 if( condition ) {  
3     // Exécution du code si condition remplie  
4 }  
5 else {  
6     // Exécution du code si la condition n'est pas remplie  
7 }  
8 ---
```



Les opérateurs logiques

- `==` : permet de tester l'égalité.
- `!=` : permet de tester l'inégalité.
- `<` : strictement inférieur.
- `≤` : inférieur ou égal.
- `>` : strictement supérieur.
- `≥` : supérieur ou égal.
- `&&` : l'opérateur ET.
- `||` : le OU. Même combat que le précédent.



L'algèbre de Boole

La conjonction

- $1 \wedge 1 = 1$
- $0 \wedge 0 = 0$
- $1 \wedge 0 = 0$
- $0 \wedge 1 = 0$



L'algèbre de Boole

La disjonction

- $1 \vee 1 = 1$
- $0 \vee 0 = 0$
- $1 \vee 0 = 1$
- $0 \vee 1 = 1$



L'algèbre de Boole

La négation

- $\sim(A \wedge B) = \sim A \vee \sim B$
- $\sim(A \vee B) = \sim A \wedge \sim B$



Exemple



For



Switch



Example



while



Example



Do while



Exemple



Java - programmation orienté objet

Tout en java commence par une **classe**, qui représente la définition d'un objet.



L'instance - l'objet

Instancier une classe, permet de créer un objet correspond à la définition de la classe.



Exemple



Le constructeur

C'est la méthode qui sert à créer une instance (ie un objet). Sa déclaration est particulière puisqu'un constructeur ne retourne rien.

Une classe peut avoir plusieurs constructeurs



Le mot `this`

Le `this` n'est pas des plus simples à comprendre.

Ce terme représente la future l'objet. Ainsi `this` pointe déjà vers un'objet. La première utilisation que l'on peut remarquer, est celle qui en est faite dans le constructeur avec argument, où les arguments portent le même nom que les variables d'instance mais ces dernières sont différencierées par l'apparition de `this` devant.



Private, Public & protected

- **Public:*** accessible par toutes les classes ;
 - **Protected:** accessibles par toutes les classes héritées et les classes du même paquetage, inaccessibles par les autres ;
 - **Private :** inaccessible par toute autre classes.
 - **static :** permet de spécifier des variables ou des méthodes de classe, directement accessible sans construction d'objet.



Exemple de programme



A voir

- Héritage
- Création de méthode

