

Support de cours

Cours:

Introduction à la programmation orientée objet (en Java)

Vidéo:

W12-01-constrintro-JAVA-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Deuxième différence. Nom de la classe. Classe rectangle. Éventuels paramètres nécessaires. Définition de la méthode. Constructeur. Corps de la méthode. Nom. Nouvelle instance. Différences. Type de retour. Méthode. Constructeurs. Exemple. Classe.



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (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/>

Constructeurs (introduction)

(Partie 2)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

résumé

0m 0s



Les constructeurs (2)

Les constructeurs sont des méthodes presque comme les autres. Les différences sont :

- ▶ *pas* de type de retour (pas même `void`)
- ▶ *même nom* que la classe
- ▶ invoqués *systématiquement* à chaque fois qu'une instance est créée.

```
Rectangle(double h, double l)
{
    hauteur = h;
    largeur = l;
}
```

Pour déclarer un constructeur, il suffit simplement d'ajouter une méthode qui a le même nom que le nom de la classe, suivi d'éventuels paramètres nécessaires aux initialisations, et suivi comme d'habitude de la définition de la méthode en question. Par exemple, pour la classe `Rectangle`, on définirait donc une méthode « `Rectangle` » qui a le même nom que la classe, qui est donc un constructeur, qui reçoit deux paramètres « `h` » et « `l` » pour pouvoir initialiser dans le corps de la méthode en question, respectivement la hauteur, l'attribut « `hauteur` » et l'attribut « `largeur` ». Les constructeurs sont donc des méthodes presque comme les autres. Les trois différences à noter sont d'abord qu'un constructeur n'a pas de type de retour. Nous n'avons rien mis ici, même pas `void`, donc ça veut dire que le constructeur n'a même pas de type de retour. La deuxième différence c'est que le constructeur est obligé d'avoir un nom qui est exactement le même que le nom de la classe. Enfin, un constructeur doit être systématiquement appelé à chaque fois qu'une nouvelle instance est créée. Appelé avec une syntaxe que l'on va détailler dans un instant. que l'on va détailler dans un instant.

notes

résumé

0m 1s

