

Support de cours

Cours:

Introduction à la programmation orientée objet (en C++)

Vidéo:

W12-01-constrintro-CPP-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Nom de la classe. Cas de la classe. Déclaration d'une instance de la classe. Type de retour. Prototype du constructeur. Nom du constructeur. Constructeur. Déclaration d'une instance. Nom. Exemple. Méthode. Constructeurs. Différences. Appel. Rectangle.



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en C++).)



[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 C++)

Jean-Cédric Chappelier, Jamila Sam 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 *automatiquement* à chaque fois qu'une instance est créée.

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

Rectangle rect;

Pour déclarer un constructeur, il suffit simplement de déclarer comme ceci une méthode qui a le même nom que le nom de la classe. Ensuite, cette méthode peut recevoir si nécessaire des paramètres puis, on écrit comme ceci sa définition comme toute méthode. Par exemple, dans le cas de la classe « Rectangle », on pourrait déclarer le constructeur, ici « Rectangle » -- c'est le même nom que le nom de la classe -- qui reçoit deux paramètres ici pour pouvoir initialiser la hauteur et la largeur. Les constructeurs sont des méthodes presque comme les autres. Les différences, c'est qu'il n'y a pas de type de retour, on n'a pas noté « void », on n'a rien noté devant le prototype du constructeur, ça veut dire qu'il n'y a pas de type de retour. Le nom du constructeur doit être absolument le même nom que le nom de la classe et enfin, les constructeurs sont invoqués automatiquement lors de la déclaration d'une instance de la classe. Par exemple, si j'écris, même simplement, comme ceci, la déclaration d'une instance « rect » de la classe « Rectangle »

notes

résumé

0m 1s





et bien, ici il y a automatiquement un appel, à ce que l'on verra plus tard, que l'on appelle le constructeur par défaut. Il y a automatiquement appel à un constructeur chaque fois qu'une instance est créée que cet appel soit explicite ou non. que cet appel soit explicite ou non.

notes

résumé

1m 13s

