

Support de cours

Cours:

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

Vidéo:

W12-01-constrintro-CPP-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Spécificités des constructeurs. Nom de la classe. Déclaration de la classe. Partie publique. Exemple de départ. Constructeurs de la classe. Construction de chaque instance de la classe. Seul constructeur. Attributs hauteur. Programmeurs utilisateurs de la classe. Déclaration de chaque instance. Paramètres différents. Valeurs initiales. Définition du constructeur. Choix du programmeur de la classe.



[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 3)

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;
}
```

Comme les autres méthodes :

- ▶ les constructeurs peuvent être surchargés
- ▶ on peut donner des valeurs par défaut à leurs paramètres

(exemples dans la suite)

Une classe peut donc avoir **plusieurs constructeurs**, pour peu que leur liste de paramètres soit différente.

Voilà pour les spécificités des constructeurs. Cependant, comme les autres méthodes, les constructeurs peuvent être surchargés

notes

résumé

0m 1s



A man with short, dark hair and glasses is looking down. He is wearing a black t-shirt. The background is plain white.

notes

0m 13s





de garder maintenant des accesseurs, des modificateurs à chacun des attributs. C'est un choix du programmeur de la classe « Rectangle » de décider si oui ou non, on peut ensuite encore modifier ces attributs hauteur et largeur ou si simplement la valeur initiale suffit. Voyons maintenant comment les constructeurs sont appelés. On a dit qu'ils étaient appelés automatiquement lors de la déclaration de chaque instance, mais comment passer les paramètres ?

notes

résumé

1m 37s



La *déclaration avec initialisation* d'un objet se fait comme pour une variable ordinaire.

Syntaxe :

NomClasse *instance*(*valarg1*, ..., *valargN*);

où *valarg1*, ..., *valargN* sont les valeurs des arguments passés au constructeur.

Exemple :

```
Rectangle r1(18.0, 5.3); // invocation du constructeur à 2 paramètres
```

Pour passer les paramètres, il suffit simplement, lors de la déclaration d'une instance d'une certaine classe, de mettre entre parenthèses ici les différentes valeurs que l'on va passer comme arguments aux paramètres du constructeur correspondant. Par exemple, dans la classe « Rectangle », où l'on avait un constructeur avec deux paramètres,

notes

résumé

2m 1s



```
class Rectangle {
public:
    Rectangle(double h, double L)
    {
        hauteur = h;
        largeur = L;
    }
    double surface() const
    { return hauteur * largeur; }
    // accesseurs/modificateurs si nécessaire
    // ...
private:
    double hauteur;
    double largeur;
};
```

on va passer ici lors de la déclaration de l'instance « r1 » de la classe « Rectangle » entre parenthèses deux valeurs pour les paramètres donc respectivement hauteur et Largeur du constructeur que l'on avait déclaré précédemment. que l'on avait déclaré précédemment.

notes

résumé

2m 25s

