

Support de cours

Cours:

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

Vidéo:

W11-02-progclasses-CPP-pt4

Concepts (extraits des sous-titres générés automatiquement) :

Attributs d'une classe. Instances de la classe figurecoloree. Paramètres de l'extérieur de la classe. Extérieur de la classe figurecoloree. Figure de la classe. Couleur rouge. Partie de la classe figurecoloree. Attribut de figurecoloree. Couleur. Exemple. Instance rouge de la classe couleur. Méthodes. Besoin. Classe. Figure.



[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/>

Classes, objets, attributs et méthodes en C++ (Partie 4)

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

Jean-Cédric Chappelier, Jamila Sam et Vincent Lepetit

...

notes

résumé

0m 0s





Ceci dit, ce n'est pas parce qu'on n'a pas besoin de passer les attributs d'une classe comme arguments à ces méthodes

notes

résumé

0m 1s



Paramètres des méthodes

Mais ce n'est pas parce qu'on n'a pas besoin de passer les attributs de la classe comme arguments aux méthodes de cette classe, que les méthodes n'ont *jamais* de paramètres.

Les méthodes **peuvent avoir des paramètres** : ceux qui sont nécessaires (et donc *extérieurs à l'instance*) pour exécuter la méthode en question !

Exemple :

```
class FigureColoree {
    // ...
    void colorie(Couleur c) { /* ... */ }
    // ...
};

FigureColoree une_figure;
Couleur rouge;
// ...
une_figure.colorie(rouge);
// ...
```

que les méthodes n'ont jamais de paramètres. Bien sûr, si une méthode a besoin de recevoir des paramètres de l'extérieur de la classe pour pouvoir fonctionner, il faudra déclarer les paramètres à la méthode. Par exemple, ici, supposons qu'on imagine qu'on ait une classe de « figurecolorée » et dans cette classe, on aurait différents attributs et en particulier aussi, une méthode qui permet de colorier la figure laquelle méthode pourrait donc recevoir typiquement une couleur et elle modifierait les instances de la classe FigureColoree pour les colorier dans cette couleur, « Couleur » étant lui-même un autre type qu'on aurait défini par ailleurs une classe ou un typedef On pourrait donc par exemple déclarer une instance, une figure de la classe figurecolorée une instance rouge de la classe couleur, et puis un tout petit peu plus tard, en anticipant sur une syntaxe qui va vous être présentée, colorier la figure avec la couleur rouge. Ce qui m'intéresse ici, c'est de vous montrer que la couleur rouge vient de l'extérieur de la classe figurecolorée. Elle ne fait pas partie de la classe figurecolorée. C'est pour ceci que l'on a besoin de la passer, ce n'est pas un attribut de figurecolorée, on a besoin de la passer comme paramètre à la méthode colorie. comme paramètre à la méthode colorie.

notes

résumé

0m 13s

