

Support de cours

Cours:

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

Vidéo:

W15-03-polymasq-CPP-pt4

Concepts (extraits des sous-titres générés automatiquement) :

Développeur de la classe b. Mot-clé override. Nouvelle méthode virtuelle. Programmeur de la classe b. Méthode de a. donc. Faute de frappe. Classe b. Méthode fl. Substitution d'une méthode. Telle petite différence. Protection du programmeur de la sous-classe. Cause de cette faute de frappe. Lieu d'un f1. Méthode. L minuscule.



[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/>

Masquage, substitution et surcharge

(Partie 4)

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

Jean-Cédric Chappelier, Jamila Sam et Vincent Lepetit

...

notes

résumé

0m 0s



```

class A {
    // ...
    virtual void f1();
    virtual void f2() const;
        void f3();          // non virtuelle (oubli?)
    virtual void f4() final; // pas de redéfinition
};

class B : public A {
    // ...
    virtual void f1() override; // OK
    virtual void f1() override; // Erreur faute de frappe : 1 <-> l
    virtual void f2() override; // Erreur
        void f3() override; // Erreur: non virtuelle
    virtual void f4();          // Erreur : f4 était final
};

```

Ici, nous voyons donc que nous avons une faute de frappe. Le développeur de la classe B a écrit `fl` minuscule au lieu d'un `f1`. C'est, je peux vous le garantir, une faute assez difficile à trouver à la relecture. Une telle petite différence entre un `l` minuscule et un `1` échappe souvent à la relecture. Donc ici, si on avait oublié le mot-clé `override` ce que pensait peut-être faire, évidemment sans avoir cette ligne là, la personne qui aurait écrit ça dans la classe B, ce serait de substituer la méthode `f1`. Mais au lieu de substituer la méthode `f1`, à cause de cette faute de frappe, il aurait en fait réintroduit une nouvelle méthode virtuelle au niveau de B. Ce que le mot-clé `override` permet de faire, c'est que le compilateur va justement vérifier si ceci, comme le spécifie ici le programmeur de la classe B est bien une substitution d'une méthode de A. Dans ce cas, comme il n'existe pas de méthode `fl` dans A le compilateur va indiquer que c'est une erreur. Cette méthode `fl` de `override` ne substitue absolument aucune méthode de A. Donc c'est bien une protection du programmeur de la sous-classe B contre des éventuelles erreurs. Un autre type d'erreur, par exemple, c'est sur la méthode `f2`. Le programmeur de la classe B prétend ici redéfinir la méthode `f2` avec un `override`. Pourtant on a ici une erreur de compilation. Est-ce que vous voyez pourquoi ? Est-ce que vous voyez pourquoi ?

notes

résumé

0m 1s

