

Support de cours

Cours:

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

Vidéo:

W14-02-1-protected-CPP-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Droit d'accès. Hiérarchie de classe. Attribut privé. Séquence vidéo. Membres d'une classe. Lancée de l'héritage. Intérieur de la classe. Droits d'accès. Fait de l'héritage. Méthode de la sous-classe. Visibilité totale. Droit d'accès privé. Objet de type. Super-classe. Cadre d'une relation d'héritage.



[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/>

Héritage : droit protégé

(Partie 1)

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

Jean-Cédric Chappelier, Jamila Sam et Vincent Lepetit

...

notes

résumé

0m 0s





Dans cette séquence vidéo, nous continuons sur la lancée de l'héritage et abordons deux notions qui sont liées : le droit d'accès protégé

notes

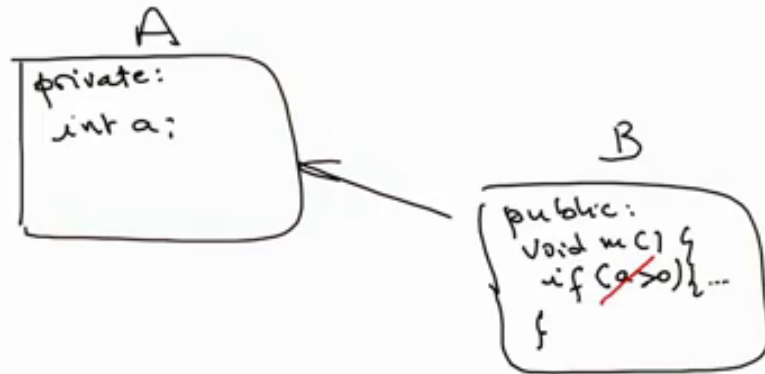
résumé

0m 1s



Jusqu'à maintenant, l'accès aux membres (attributs et méthodes) d'une classe pouvait être :

- ▶ soit **public** : visibilité totale à l'intérieur et à l'extérieur de la classe (mot-clé **public**)
- ▶ soit **privé** : visibilité uniquement à l'intérieur de la classe (mot-clé **private**)



et la notion de masquage dans une hiérarchie de classe. Vous avez appris jusqu'ici que l'accès aux membres d'une classe, c'est-à-dire soit à un attribut ou à une méthode, peut-être soit public, ce qui veut dire que le membre en question a une visibilité totale, il est accessible aussi bien à l'intérieur de la classe dans lequel il est défini qu'à l'extérieur. S'il est privé, il a une visibilité restreinte uniquement à l'intérieur de la classe dans lequel il est défini. La question que nous nous posons maintenant est : qu'en est-il des droits d'accès lorsque l'on se trouve dans le cadre d'une relation d'héritage ? Supposons que l'on est une super-classe « A » dont hérite une sous-classe « B », supposons qu'à l'intérieur de la super-classe « A », nous avons déclaré un attribut privé, ce que nous avons fait jusqu'ici pour tous les attributs et nous nous posons maintenant la question de savoir si dans une méthode de la sous-classe « B » par exemple, une méthode comme ceci nous avons le droit d'utiliser cet attribut directement. Nous avons vu que par le fait de l'héritage, la classe « B » dispose bel et bien d'un attribut « a » hérité de A. Mais, est-ce qu'elle a le droit d'y accéder directement, est-ce qu'elle a le droit par exemple de faire ceci avec un accès direct à l'attribut hérité ? En fait, comme le droit d'accès privé assure la visibilité uniquement à l'intérieur de la classe, l'attribut « a » défini dans cette classe n'est accessible que dans la classe « A ». On pourrait accéder à cet attribut directement uniquement dans les méthodes de la classe « A » et par ailleurs. La réponse à la question que nous nous posons est donc « non », nous n'avons pas le droit d'accéder à «

notes

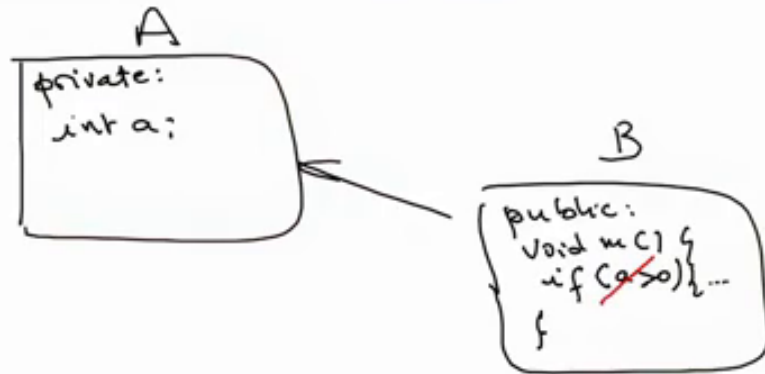
résumé

0m 13s



Jusqu'à maintenant, l'accès aux membres (attributs et méthodes) d'une classe pouvait être :

- ▶ soit **public** : visibilité totale à l'intérieur et à l'extérieur de la classe (mot-clé **public**)
- ▶ soit **privé** : visibilité uniquement à l'intérieur de la classe (mot-clé **private**)



a » directement dans une méthode de la classe « B » si cet attribut est défini comme privé dans la super-classe. Et ceci nous place dans une situation un petit peu particulière, un objet de type « B » dispose bel et bien de l'attribut « a » par héritage, mais il ne peut pas y accéder directement. Comme il peut sembler naturel

notes

résumé