

Support de cours

Cours:

Introduction à la programmation orientée objet (en Java)

Vidéo:

W13-02-1-protectedmasquage-JAVA-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Fait de l'héritage. Notion de droit d'accès. Membre d'une classe. Séquence vidéo. Attribut petit. Déclaration d'un attribut. Droit d'accès. Lancée de l'héritage. Sous-classe b. Extérieur de la classe. Déclaration du membre. Classes du même paquetage. Cadre des exercices. Modificateur explicite. Droit d'accès privé.



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en Java).)



[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 d'accès `protected`

(Partie 1)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

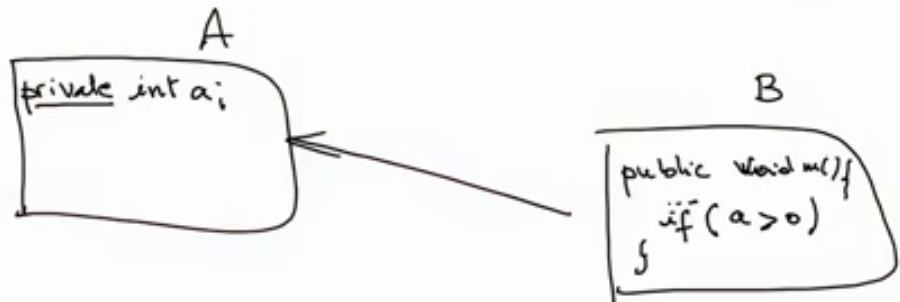
résumé

0m 0s



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`)
- ▶ soit **par défaut** (aucun modificateur) : visibilité depuis toutes les classes du même paquetage (est aussi valable pour le paquetage par défaut que vous utilisez en exercice)



Dans cette séquence vidéo nous continuons sur la lancée de l'héritage et abordons dans ce cadre, la notion de droit d'accès protégé. Vous avez appris jusqu'ici que l'accès à un membre d'une classe, c'est à dire, à un attribut ou à une méthode, peut être soit 'public', ce qui signifie qu'il est utilisable sans restriction aussi bien à l'intérieur qu'à l'extérieur de la classe. Il peut être 'privé', ce qui signifie que le membre en question n'est utilisable qu'à l'intérieur de la classe. En java il existe également, le droit d'accès 'par défaut' qui est en place si on ne met aucun modificateur explicite devant la déclaration du membre en question et dans ce cas, le membre est visible depuis toutes les classes du même paquetage. C'est aussi le cas pour le paquetage par défaut que vous avez souvent l'occasion d'utiliser dans le cadre des exercices, notez qu'il est prévu pour vous, un complément PDF sur le site du cours, concernant la notion de paquetage. La question que nous nous posons maintenant est qu'en est-il des droits d'accès dans une hiérarchie de classes ? Supposons que nous ayons une super-classe grand 'A' dont hérite une sous-classe grand 'B'. Supposons que la super-classe... contient la déclaration d'un attribut : petit 'a' qui serait privé. Sachant que déclarer les attributs en privé est ce que nous avons toujours fait jusqu'ici, la question que nous nous posons est : est-ce que l'attribut en question est accessible dans une méthode de B ? Par exemple, aurions-nous le droit d'écrire quelque chose comme ceci à l'intérieur de B ? Nous avons vu que par le fait de l'héritage, la sous-classe B dispose, bel et bien, de l'attribut petit 'a' hérité de 'A', la question est : a-t-elle le droit d'en disposer librement, de l'utiliser directement, au sein de ses propres méthodes ? En fait, comme nous l'avons vu, le droit d'accès privé n'assure la visibilité

notes

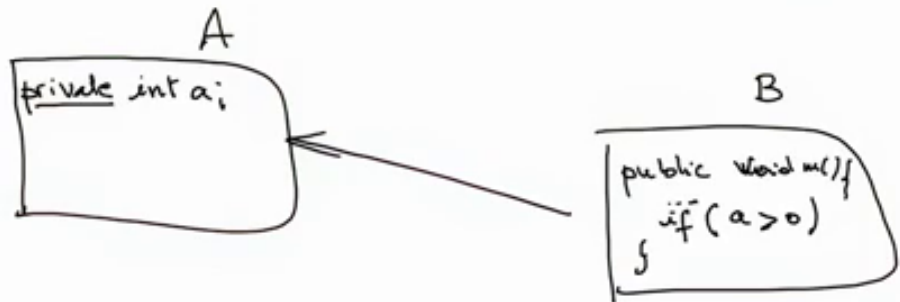
résumé

0m 1s



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

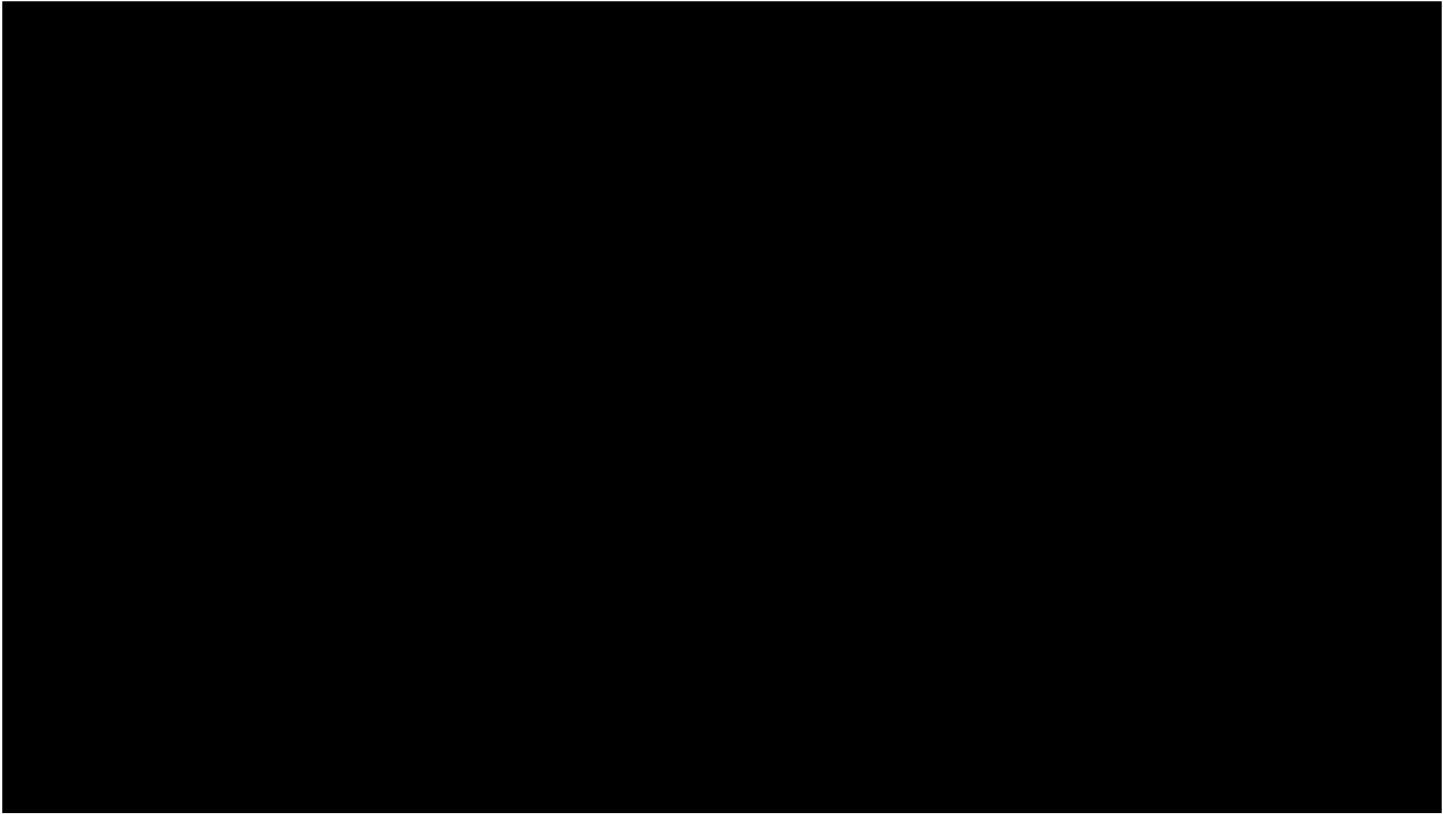
- ▶ soit **public** : visibilité totale à l'intérieur et à l'extérieur de la classe (mot-clé `public`)
- ▶ soit **privé** : visibilité uniquement à l'intérieur de la classe (mot-clé `private`)
- ▶ soit **par défaut** (aucun modificateur) : visibilité depuis toutes les classes du même paquetage (est aussi valable pour le paquetage par défaut que vous utilisez en exercice)



qu'à l'intérieur de la classe dans laquelle le membre est défini. Ceci signifie que l'attribut petit 'a' n'est utilisable ici directement, que dans les méthodes de la classe grand 'A'.

notes

résumé



Ce qui signifie également que ceci n'est pas permis. Nous nous trouvons donc ici dans une situation un peu paradoxale qui est que la classe B, héritant de A, dispose bel et bien d'un attribut petit 'a' mais ne peut pas en faire usage librement comme ici, elle ne peut pas y accéder directement comme dans le cas de cet exemple. comme dans le cas de cet exemple.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

2m 13s

