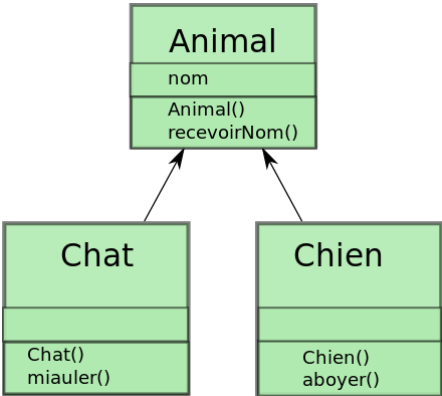


# Héritage (informatique)

En programmation orientée objet, l'**héritage** est un mécanisme qui permet, lors de la déclaration d'une nouvelle classe, d'y inclure les caractéristiques d'une autre classe.

## Sommaire

- Principe
- Particularités des classes
  - Classe abstraite
  - Visibilité des membres
- Voir aussi



Exemple basique de diagramme UML représentant une relation d'héritage

## Principe

L'héritage établit une relation de généralisation-spécialisation qui permet d'hériter dans la déclaration d'une nouvelle classe (appelée classe dérivée, classe fille, classe enfant ou sous-classe) des caractéristiques (propriétés et méthodes) de la déclaration d'une autre classe (appelée classe de base, classe mère, classe parent ou super-classe).

En déclarant une nouvelle classe B par héritage de la classe A, ajoutant de nouveaux membres, on peut alors dire que :

- A est une *généralisation* de B et B est une *spécialisation* de A ;
- A est une *super-classe* de B et B est une *sous-classe* de A ;
- A est la *classe mère* de B et B est une *classe fille* de A.

Lorsqu'une classe fille hérite d'une classe mère, elle peut alors utiliser les caractéristiques de la classe mère.

Par exemple, on déclare d'abord la classe de base *Vehicule* puis, à partir de celle-ci, on déclare les classes dérivées *Automobile* et *Avion* qui héritent alors des caractéristiques de la classe de base *Vehicule*.

On parle d'héritage, car le principe est en quelque sorte le même que celui d'un arbre généalogique; tout comme l'enfant hérite des caractéristiques du parent, une classe fille » héritent des caractéristiques de leur(s) « mère(s) ». Ce principe est basé sur une classe dont les « filles » héritent des caractéristiques de leur(s) « mère(s) ».

En programmation orientée objet, l'héritage permet donc de réutiliser (décomposer un système en composants) et d'adapter les objets grâce au polymorphisme.

## Particularités des classes

### Classe abstraite

- Une **classe abstraite** ne peut être instanciée.
- Si la classe mère est une **classe abstraite**, il y a de forte chance qu'elle présente des caractéristiques abstraites (retardées). Pour être instanciable, la classe fille doit alors les définir, sinon elle sera elle-même abstraite.

- Si la classe mère est instanciable (c'est-à-dire que toutes ses caractéristiques sont définies) alors la classe fille est également instanciable. Il est possible d'y ajouter des caractéristiques, d'utiliser les caractéristiques héritées (appartenant aux parents) et de redéfinir les méthodes héritées. Généralement cette redéfinition se fait par surcharge sémantique (on déclare de nouveau la méthode avec le même nom et la même signature).

## Visibilité des membres

Pour qu'une classe fille puisse hériter des propriétés de la classe mère, il faut que les propriétés de la classe mère possèdent des attributs de visibilité compatibles. Il existe dans la plupart des langages trois niveaux de visibilité :

- `Public` est la visibilité la plus large (accessible directement en dehors de la classe)
- `Private` est la plus restrictive (accessible uniquement depuis la classe courante)
- `Protected` est intermédiaire et est visible par la classe courante, toutes les classes filles, mais inaccessible en dehors des classes mère/filles.

## Voir aussi

- Programmation orientée objet

Ce document provient de « [https://fr.wikipedia.org/w/index.php?title=Héritage\\_\(informatique\)&oldid=153553582](https://fr.wikipedia.org/w/index.php?title=Héritage_(informatique)&oldid=153553582) ».

La dernière modification de cette page a été faite le 31 octobre 2018 à 16:52.

Droit d'auteur : les textes sont disponibles sous licence Creative Commons attribution, partage dans les mêmes conditions ; d'autres conditions peuvent s'appliquer. Voyez les conditions d'utilisation pour plus de détails, ainsi que les crédits graphiques. En cas de réutilisation des textes de cette page, voyez comment citer les auteurs et mentionner la licence.

Wikipedia® est une marque déposée de la Wikimedia Foundation, Inc., organisation de bienfaisance régie par le paragraphe 501(c)(3) du code fiscal des États-Unis.