

Les classes abstraites et les classes concrètes

Table des matières

Introduction	2
Qu'est-ce qu'une classe abstraite ?	2
Méthodes abstraites	4
Résumé	6

Introduction

L'objectif de ce cours est de vous présenter les classes abstraites et de les identifier dans un diagramme de classes.

Comme l'héritage, les classes abstraites correspondent à une notion importante dans la programmation orientée objet. Les classes abstraites sont apparues dans la programmation orientée objet avant le langage UML. Lors de l'élaboration d'un diagramme de classes, vous devez identifier éventuellement les classes abstraites pour une implémentation future.

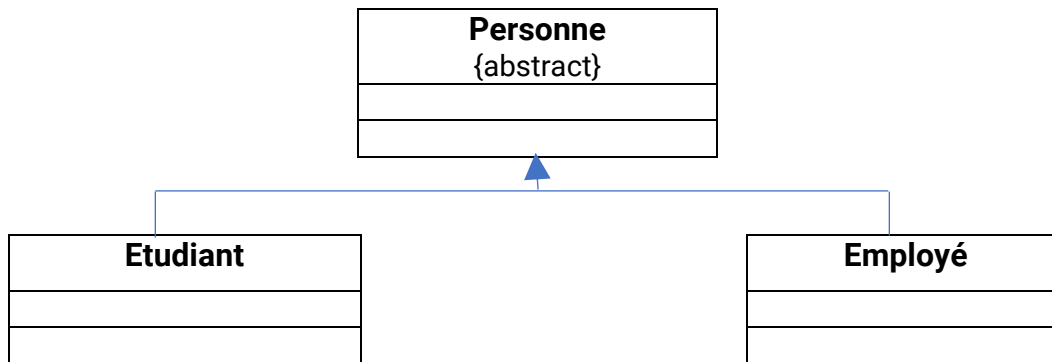
Qu'est-ce qu'une classe abstraite ?

On parle de classe abstraite dans le cas de l'héritage. Dans une relation d'héritage, la super-classe est par définition une classe abstraite. Une classe abstraite a pour **rôle** d'être une superclasse et de factoriser les attributs et les méthodes communs des sous classes.

Une super-classe n'est pas forcément abstraite.

En UML, le mot-clé {abstrait} ou {abstract} est accolé aux classes. Une autre manière souvent usitée de représenter ces méthodes ou classes est d'écrire leur nom en italique.

Soit l'exemple suivant :

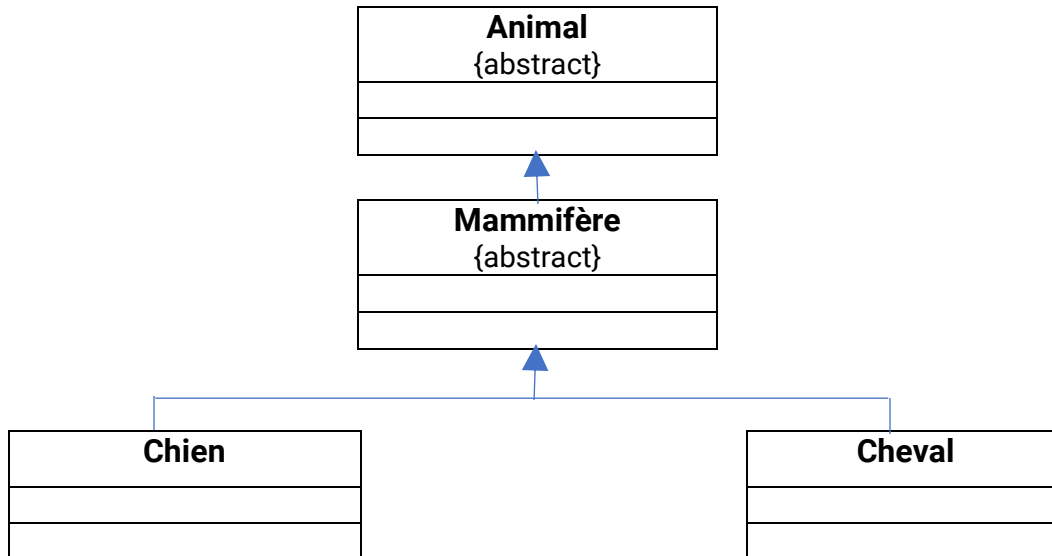


La classe Personne est **une classe abstraite**. Les classes Etudiant et Employé sont deux classes concrètes.

En fait la classe Personne a été créée pour factoriser les attributs et les méthodes communs aux classes Etudiant et Employé. Dans le monde réel, il y a soit un étudiant soit un employé. Une personne est abstraite, on sait pas de qui s'agit-il. On parle ainsi de classe abstraite. Dans la programmation objet, lorsque vous définissez une classe abstraite, vous ne pouvez pas l'instancier, c'est-à-dire, vous ne pouvez pas créer des objets de cette classe : Elle est abstraite, il n'y aura pas d'objets réellement de cette classe.

De façon générale, une classe concrète possède une existence physique c'est le cas d'un client, d'un équipement, d'un produit par exemple alors qu'une classe abstraite n'a pas d'existence propre, c'est-à-dire nous ne trouvons pas des objets d'une classe abstraite. C'est une classe qui n'a pas d'instance directe mais dont les classes filles ont des instances.

Dans l'exemple suivant, les deux classes Animal et Mammifère sont des classes abstraites. Les classes Chien et classe Cheval sont deux classes concrètes.



Méthodes abstraites

Une classe abstraite encapsule **des méthodes dites abstraites**, c'est-à-dire dont le traitement n'est pas défini dans la classe. Par contre, les classes héritant d'une classe abstraite doivent **obligatoirement** redéfinir ses méthodes abstraites.

Une méthode est abstraite lorsqu'on la déclare, sans donner son implémentation, pas d'accolades mais un simple « ; » à la suite de la signature de la méthode.

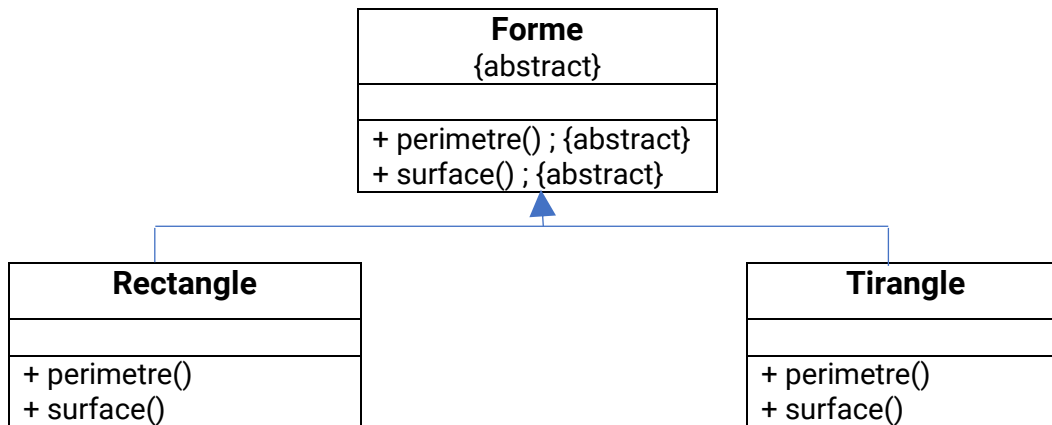
Une classe abstraite contient au moins une méthode abstraite.

En UML, le mot-clé {abstrait} ou {abstract} est accolé aux méthodes abstraites

Dès qu'une classe comporte **une méthode abstraite**, elle est **abstraite**.

L'intérêt d'une méthode abstraite est qu'elle nous offre la certitude qu'elle existera dans toutes les classes dérivées.

Soit l'exemple suivant modélisant une hiérarchie de forme géométrique. La classe mère **Forme** est une classe abstraite et elle comporte les deux méthodes abstraites : `perimetre()` et `surface()`. Au niveau de la classe `Forme`, les deux méthodes `perimetre` et `surface` ne sont pas définies. Nous trouvons que leurs noms et leurs signatures. Ce sont des méthodes abstraites. Dans les classes concrètes : `Rectangle` et `Triangle`, nous devons définir obligatoirement l'implémentation des deux méthodes `perimetre` et `surface`.



Résumé

Les classes abstraites permettent de créer des classes génériques **définissant certaines méthodes sans les implémenter** et fournissant une implémentation commune de certaines autres opérations pour l'héritage de classes.

Ce qu'il faut retenir :

- ✓ Une classe abstraite est une classe qui ne peut pas être instanciée.
- ✓ Une classe abstraite peut contenir des méthodes déjà implémentées.
- ✓ Une classe abstraite peut contenir des méthodes non implémentées.
- ✓ Une classe abstraite est héritable.
- ✓ On peut construire une hiérarchie de classes abstraites.
- ✓ Pour pouvoir construire un objet à partir d'une classe abstraite, il faut dériver une classe non abstraite en une classe implémentant toutes les méthodes non implémentées.
- ✓ Une méthode déclarée dans une classe, non implémentée dans cette classe, mais juste définie par la déclaration de sa signature, est dénommée méthode abstraite.