COURS PROGRAMMATION ORIENTÉ OBJET

CHARLES 'ARYS' YAICHE

MÉTHODE PARTICULIÈRE : ACCESSEURS (GET & SET)

- En programmation orienté objet, la modification et la lecture des **attributs** directement dans **l'objet** n'est pas considéré comme une bonne pratique pour des raison de sécurité.
- On créer en général des **méthodes** qui vont modifier et lire ces **attributs**

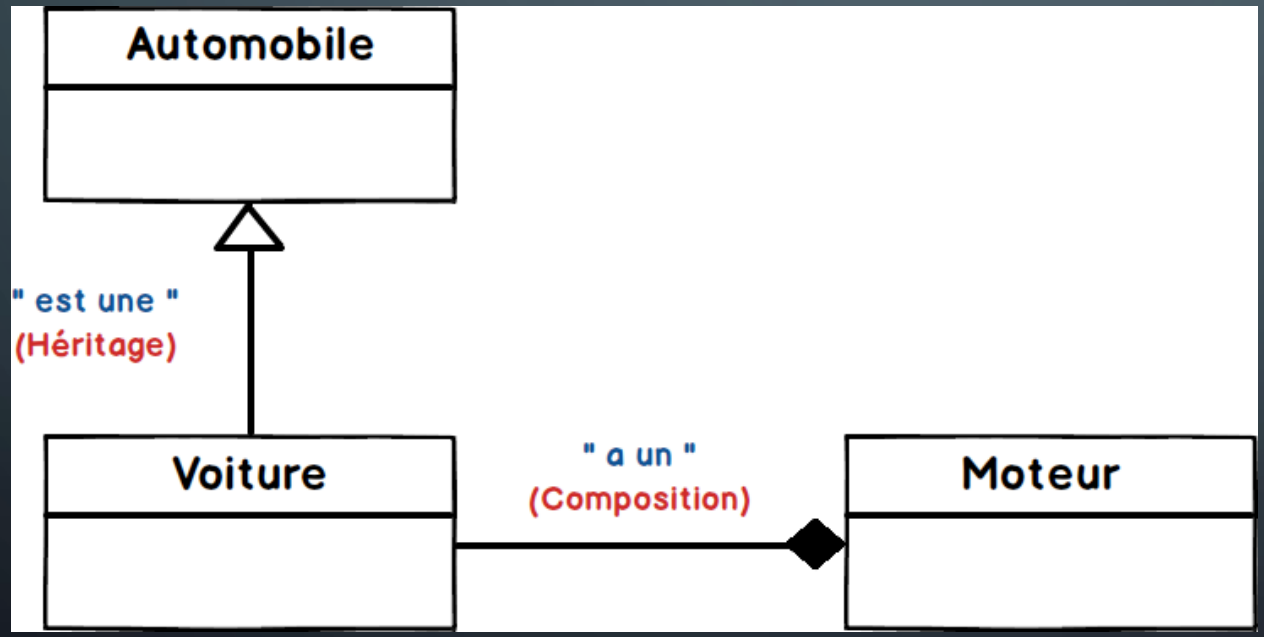
ENCAPSULATION

L'encapsulation est un mécanisme permettant de protéger le champs d'action des attributs et méthode d'une classe

- Public
  - Toutes les **classe** peuvent avoir accès au **Attributs** et **Méthode** publiques
- Private
  - L'accès au données est limité au **méthode** de la **classe** elle-même.
- Protected
  - L'accès au données est limitée au **méthode** de la **classe** d'une même famille (nous verrons le concept d'héritage plus tard)

COMPOSITION

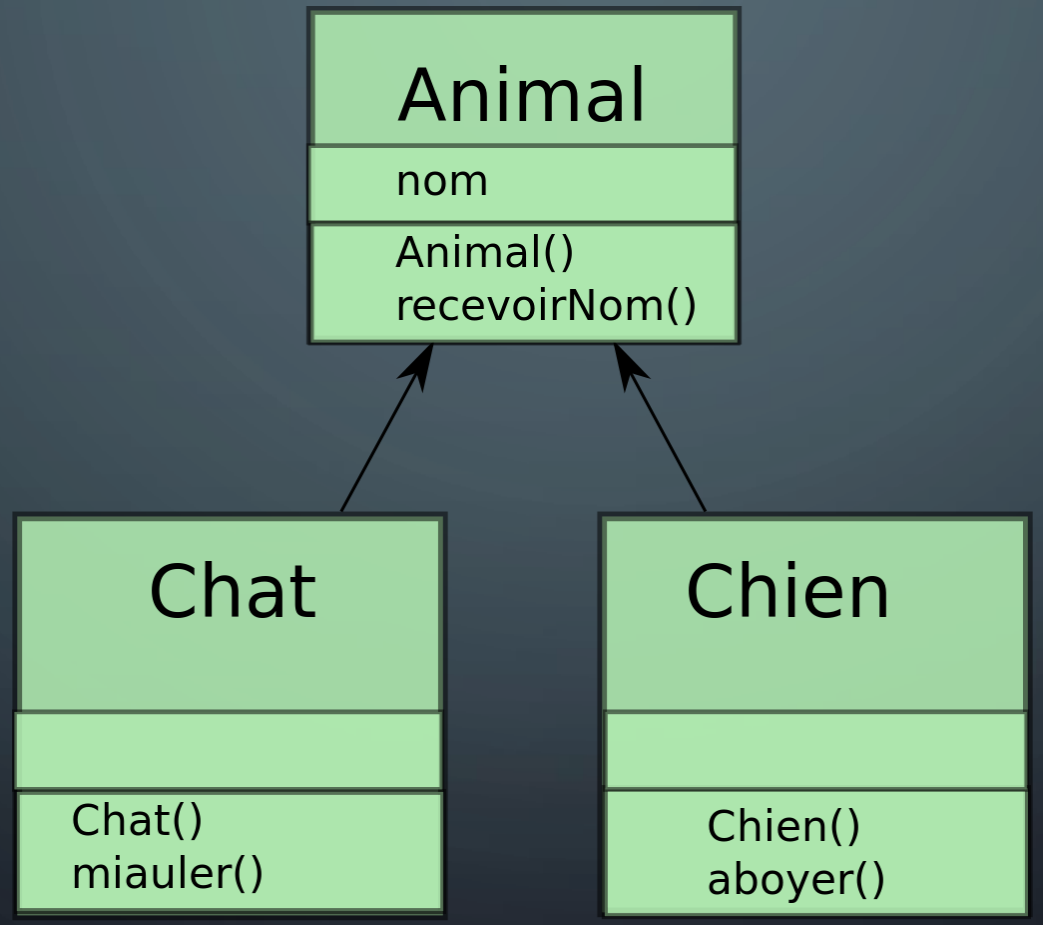
La composition est la capacité d'une classe a intégrer d'autres objets en attributs



HÉRITAGE

- L'héritage 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).

HÉRITAGE


Animal
nom
Animal()
recevoirNom()
Chat
Chat()
miauler()
Chien
Chien()
aboyer()

CLASSE ABSTRAITE

- Une classe abstraite et une classe qui n'a pas vocation à être complète ni instancié, elle sert surtout de classe pour des classe dérivée

- Exemple : Être vivant est une classe abstraite, humain dérive d'être vivant.

INTERFACE

- Uniquement prototype (ou signature) de méthode
- Re-implémentation des méthode d'une interface (java)

DIFFÉRENCE CLASSE ABSTRAITE & INTERFACE

Classe abstraite	Interface
Non Instanciable	Non Instanciable
Toutes les méthodes ne sont pas forcément implémenté	Uniquement prototype de méthode
Servent à factoriser du code	
Dans beaucoup de langage (java, C#, php) il n'y a qu'un seul héritage possible	Plusieurs implémentations sont en général possible

POLYMORPHISME

- le **polymorphisme** permet à une fonction d'avoir différente forme

```
1.  Vehicule v = new Voiture();
2.  v.demarrer();
```

SOURCES

- Schémas Poo vs procédural :
  - https://waytolearnx.com/2018/09/difference-entre-programmation-procedurale-et-orientee-objet.html
- Définition héritage
  - https://fr.wikipedia.org/wiki/H%C3%A9ritage_(informatique)
- Héritage vs polymorphisme
  - https://waytolearnx.com/2018/09/difference-entre-heritage-et-polymorphisme.html
- Composition
  - https://waytolearnx.com/2018/08/difference-entre-heritage-et-composition.html