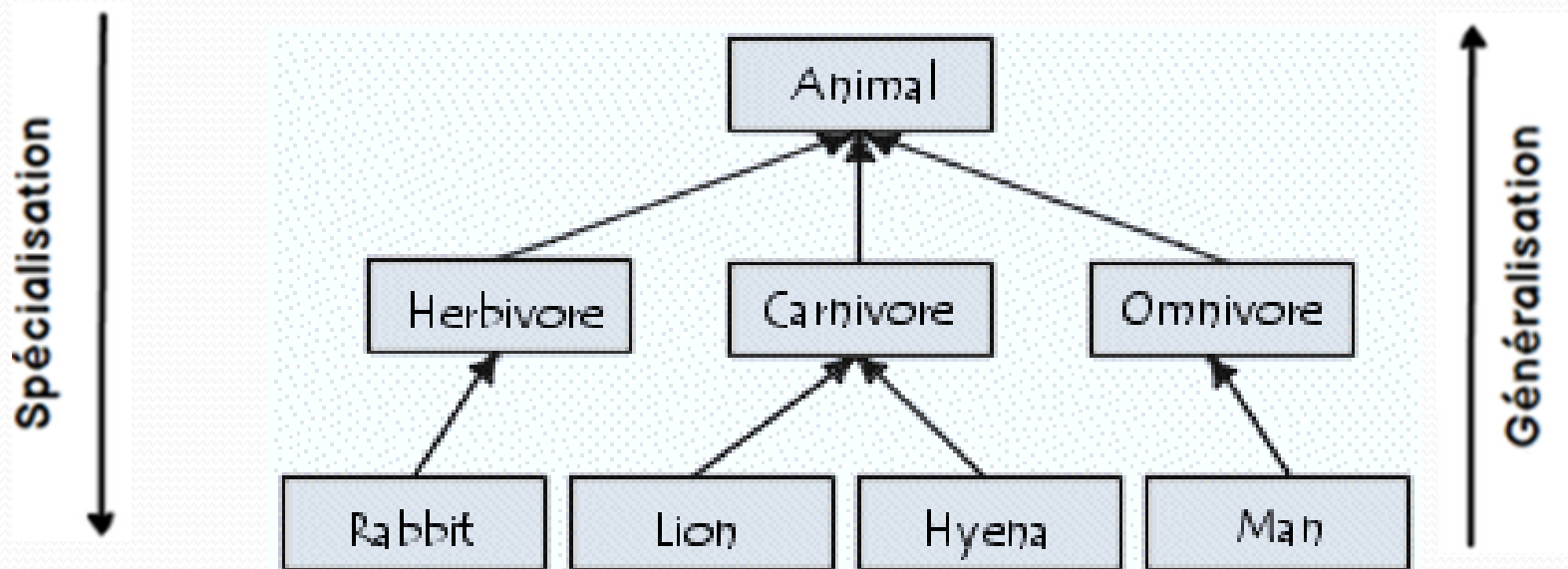


# L'héritage

# Le principe de l'héritage

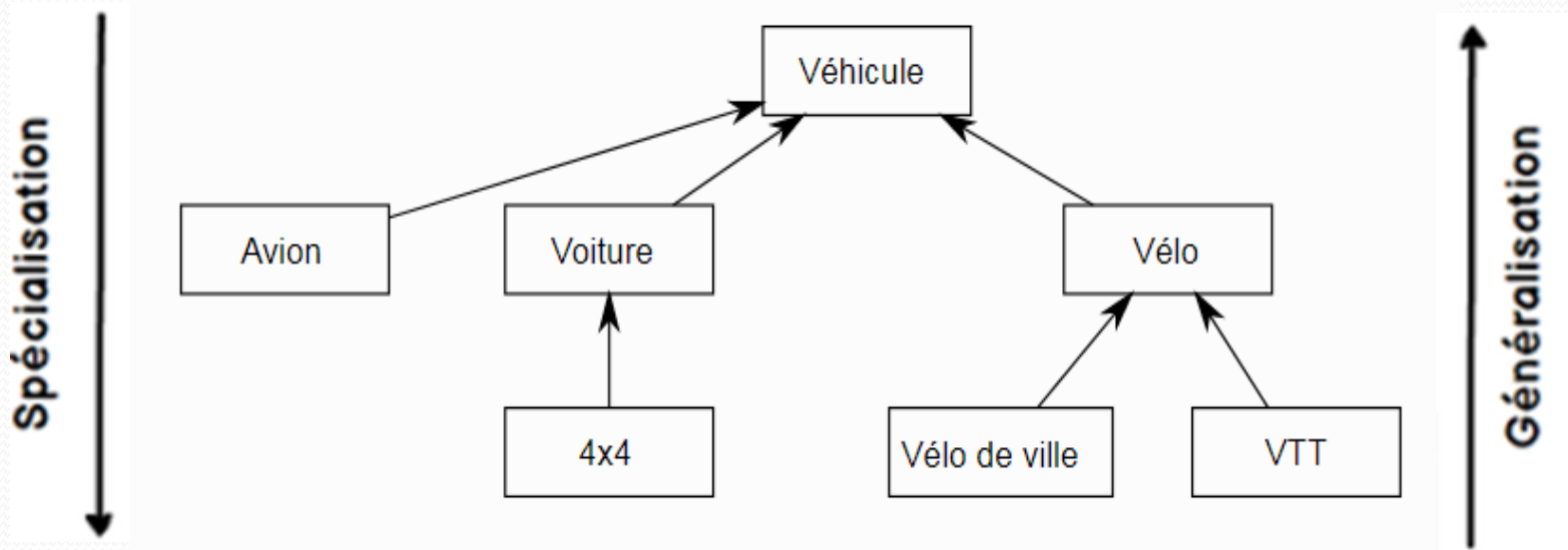
- L'**héritage** est une notion qui définit une relation de **spécialisation** ou de **généralisation** entre deux classes classes.
- Dans la POO, on dit:
  - Une classe **Chat**, **hérite** d'une autre classe **Animal**
  - Une classe **Directeur** et une classe **Formateur**, **héritent** d'une classe **Employé**
  - Une classe **voiture** et une classe **vélo**, **héritent** de la classe **véhicule**
- Dans la modélisation, on dit que la classe chat est une **spécialisation** de la classe Animal, et Animal est une **généralisation** de chat

# Schéma explicatif



- La classe **Lion** hérite de la classe **carnivore**
- La classe **carnivore** hérite de la classe **Animal**

# Schéma explicatif



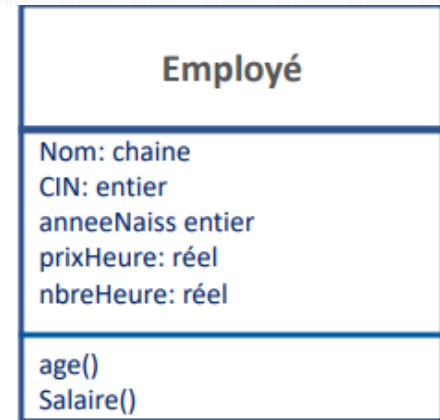
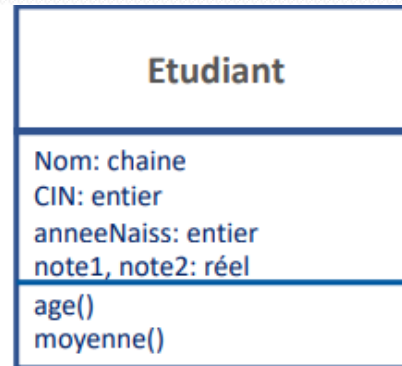
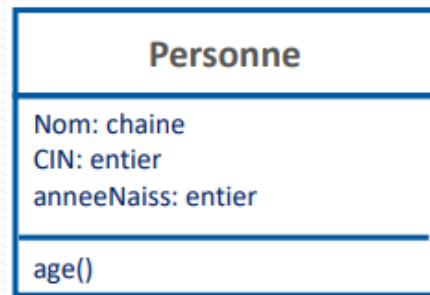
**Avion, Voiture et Vélo** sont des véhicules

# Mots techniques relatifs à l'héritage

- Lorsque on travaille avec le principe de l'héritage:
  - On dit que la classe Animal est une **classe mère** ou **classe parente** ou **classe de base** ou **super-classe**
  - La classe Chat qui hérite de la classe Animal, on l'appel **classe fille** ou **classe dérivée** ou **sous classe** .

# Avantages de l'héritage (problématique)

- Considérons la définition des 3 classes Personne, Etudiant et Employé suivantes



**Sans héritage on aura des problème :**

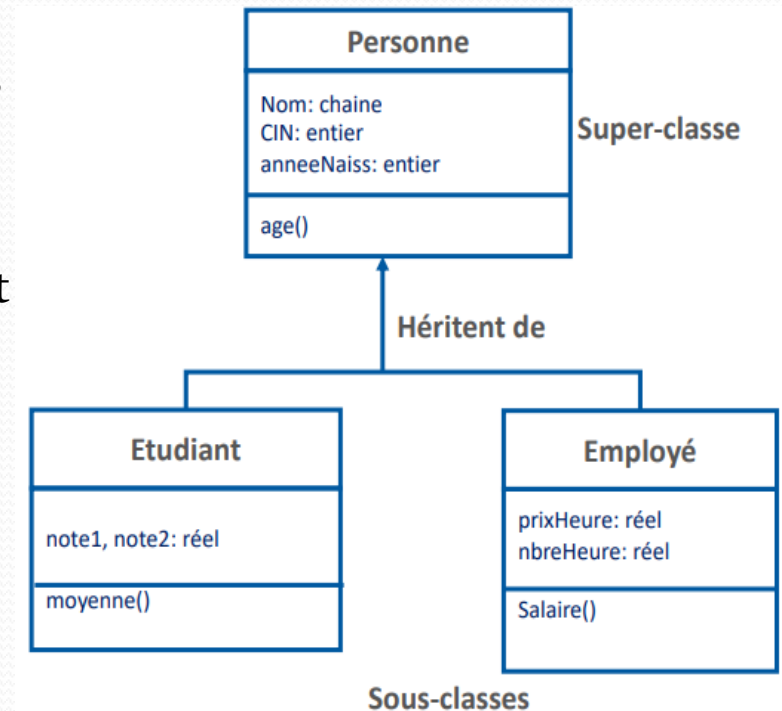
- Duplication du code.
- Toute modification d'un attribut ou d'une méthode doit être répétée dans chaque classe où ils ont été déclarés.

(Exp : si on souhaite modifier le type de l'attribut anneeNaiss, il faut le faire dans chacune des 3 classes)

# Avantage de l'héritage (Solution)

- **Solution :**

- Placer dans la classe mère les propriétés en **commun** à toutes les autres classes.
- On ne garde dans les classes filles que les attributs ou méthodes qui leur sont spécifiques.
- Les classes dérivées **héritent** automatiquement des attributs (et des méthodes) qui n'ont pas besoin d'être réécrits.



# Que peut on faire avec l'héritage?

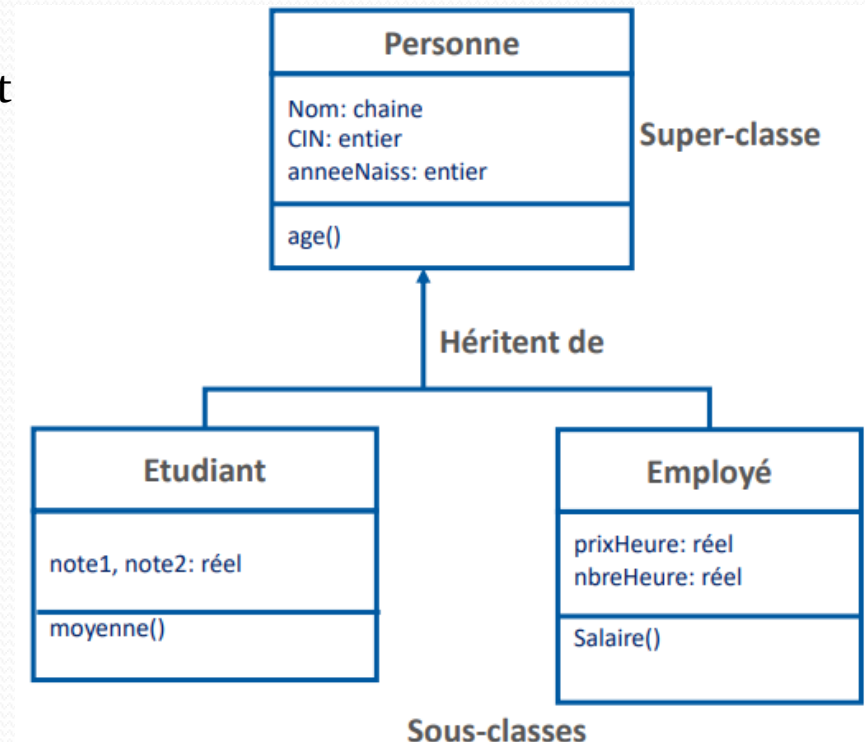
- On utilise l'héritage pour faciliter le codage et éviter la répétition de certaines parties de code.
- L'héritage va nous permettre de :
  1. **Récupérer** le comportement standard de la classe mère à partir des propriétés et méthodes définies dans celle-ci.
  2. **Ajouter** des fonctionnalités supplémentaires en créant de nouvelles propriétés et méthodes dans la classe dérivée.
  3. **Modifier** le comportement standard de la classe mère en substituant ou **redéfinissant** ses comportements dans les classes dérivées.



# Récupération du comportement de la classe mère

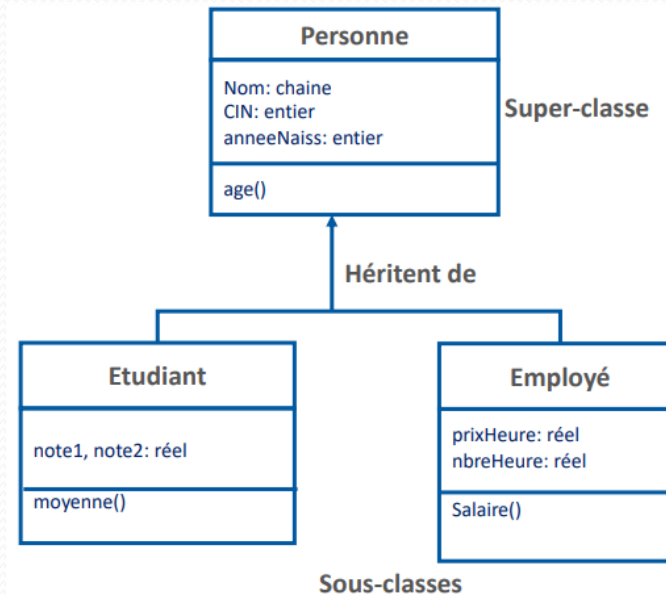
Remarquez que la classe personne contient

- les attributs: nom, CIN, Anneenaiss
- la méthode age()



Les classes Etudiant et Employé **ne contiennent pas** ces attributs et la méthode age(): cela signifie que les classes Etudiant et Employé **récupère ces attributs et la méthode age() grâce à l'héritage**.

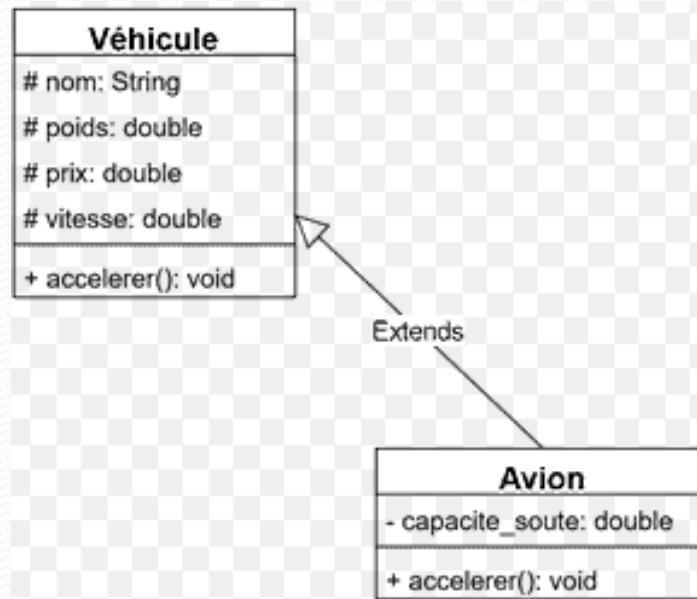
# Ajouter des fonctionnalités dans la classe dérivée



La classe Etudiant, **contient les attributs note1, note2 et la méthode moyenne()** qui ne sont pas dans la classe mère.

Donc l'héritage permet à la classe fille d'avoir ses propres attributs et fonctionnalités.

# Redéfinition de certaines méthodes



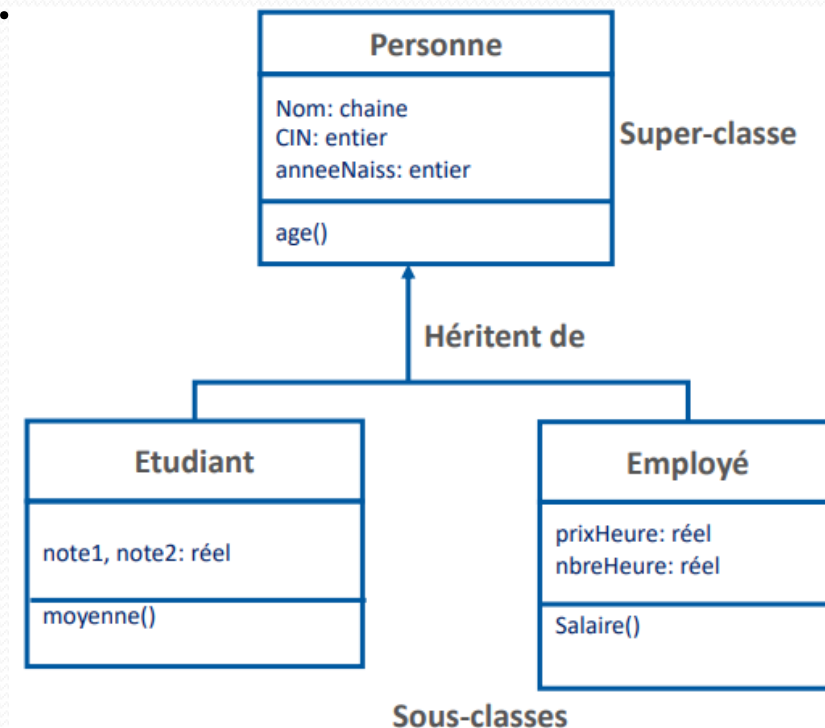
- La classe Avion hérite de la classe véhicule, mais on remarque que les deux classes possèdent la méthode **accélérer()**, ce qui signifie que la classe fille redéfinit la méthode **accélérer()**

# Types d'héritage

- En python on parle de ces types d'héritage :
  1. Héritage simple
  2. Héritage multiple
  3. Héritage en cascade

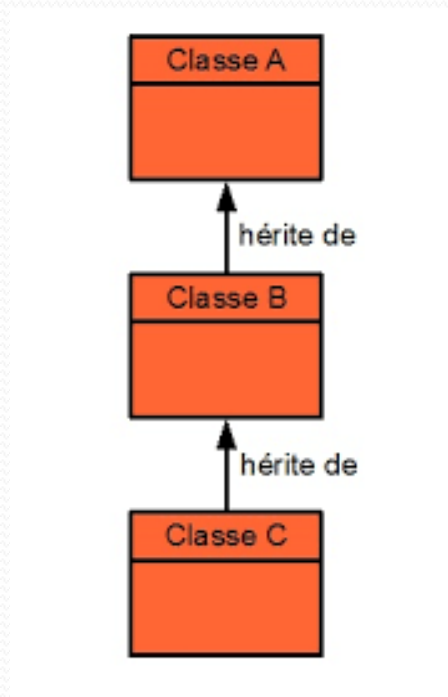
# Héritage simple

- C'est lorsque une classe **hérite d'une seule classe** comme suit:



# Héritage en cascade

- C'est lorsque une classe C hérite d'une classe B qui, à son tour hérite d'une autre classe A



# Héritage multiple

- Lorsque une classe hérite de plusieurs classes, on parle d'un héritage multiple. Exemple : la classe professeur hérite au même temps de la classe personne et la classe employé

