Cours du Master PISE

Jean-Baptiste.Yunes@u-paris.fr
http://yunes.informatique.univ-paris-diderot.fr/

Note

- ce support a été établi à partir des ouvrages :
    - de Pierre-Alain Muller (Modélisation UML)
    - de Pascal Roques (UML en action, UML par la pratique)
- des supports :
    - de Colette Johnen (Université Bordeaux - LabRI)

Statique

Objet ?

- **Atome** formé
  - d'un **état**
  - d'un **comportement**
- Forte cohésion interne
- Faible couplage externe

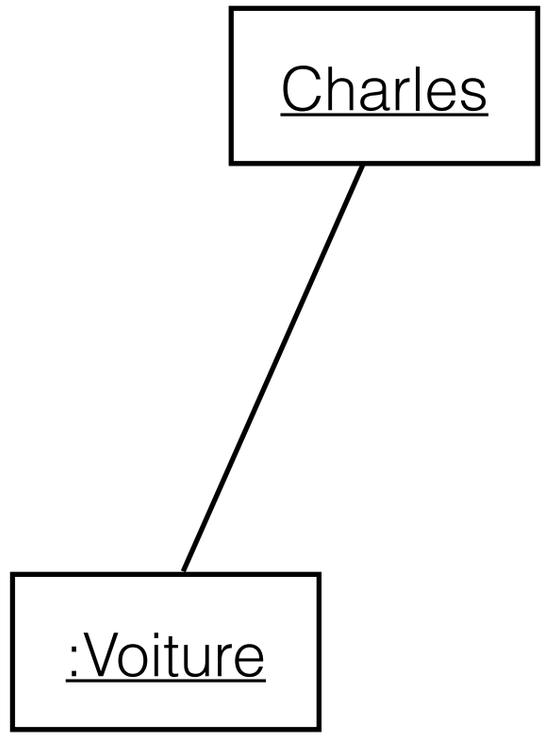
Objet ?

- Un rectangle
  - nom souligné (Charles)
- Lorsque le nom n’est pas signifiant
  - préfixe : suivi par un nom générique (:Voiture, :Billet)

Objet ?

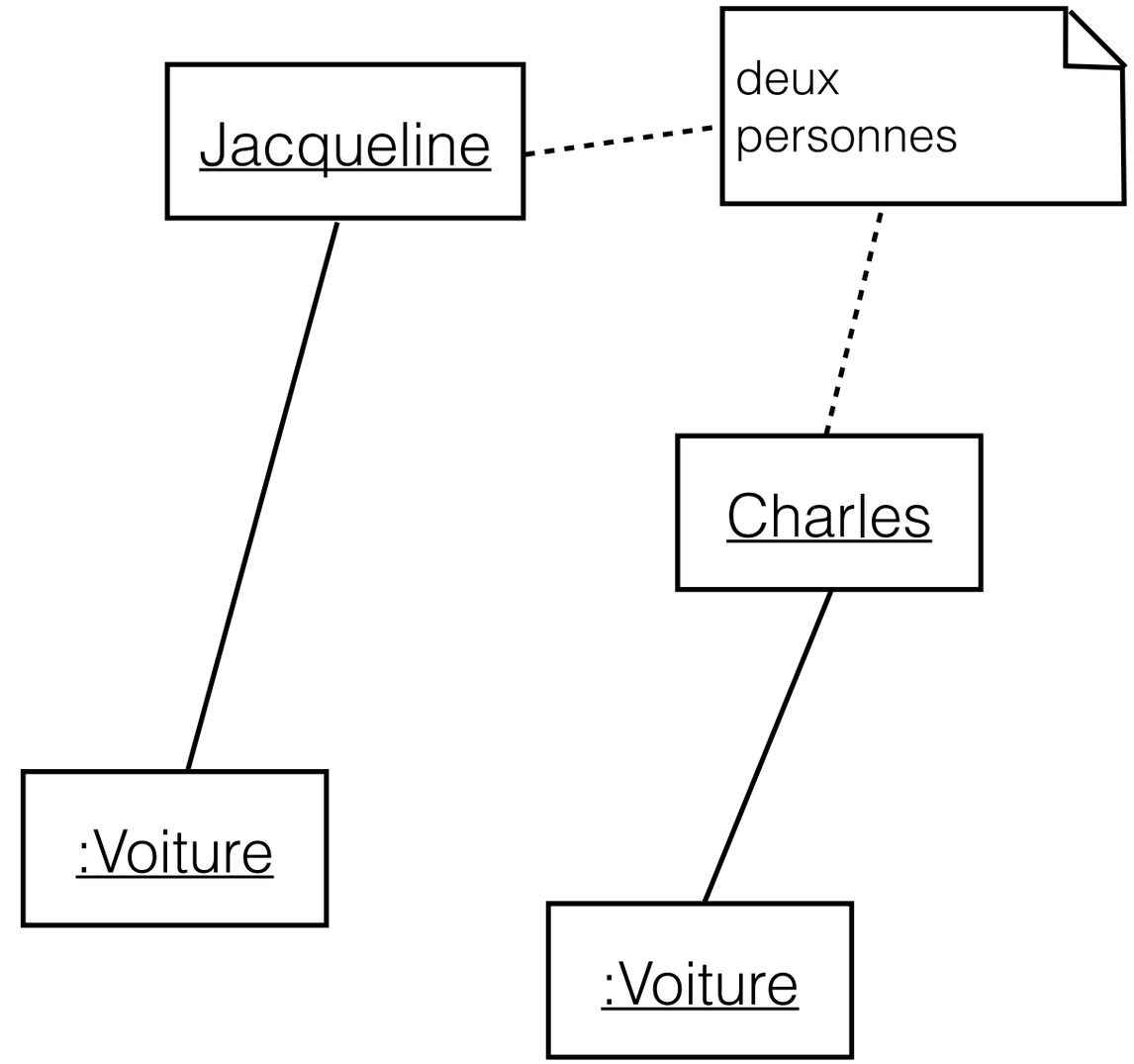
- Un trait entre deux objets symbolise un **lien**

- par exemple que Charles possède une :Voiture (qui n'est pas nommée)



Objet ?

- Une documentation est possible à l'aide d'une **annotation**
  - format libre
  - reliée aux entités par des pointillés
- On obtient un **diagramme d'objets**



Objet ?

- Tout objet a une **identité**
- Donc trois caractéristiques fondamentales :
  - état
  - comportement
  - identité

Objet ? État

- L'**état** d'un objet est l'ensemble des valeurs instantanées de tous les attributs d'un objet

- Un **attribut** prend sa valeur dans un domaine

MonCompteCourant
position = 153540€

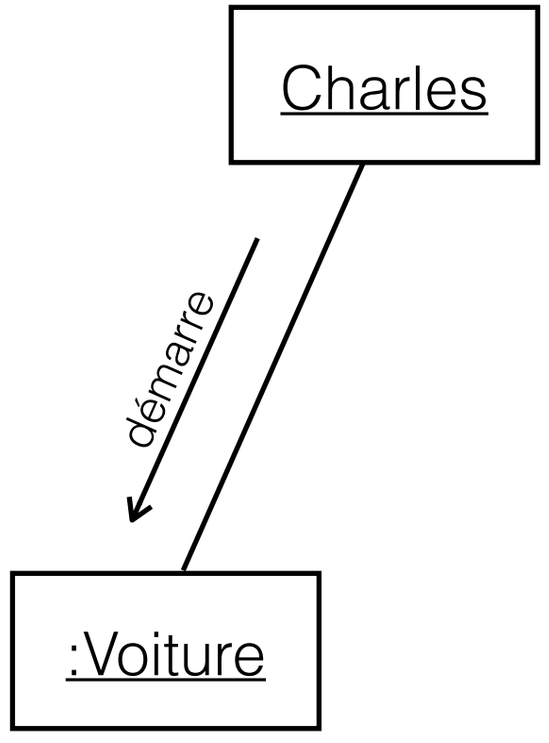
MonVélo
taille = 26” couleur = Rouge vitesses = 21

Objet ? Comportement

- Le **comportement** est l'ensemble des **actions** de l'objet

- Un atome de comportement est une **opération**

- une opération est déclenchée à la réception d'un **message**

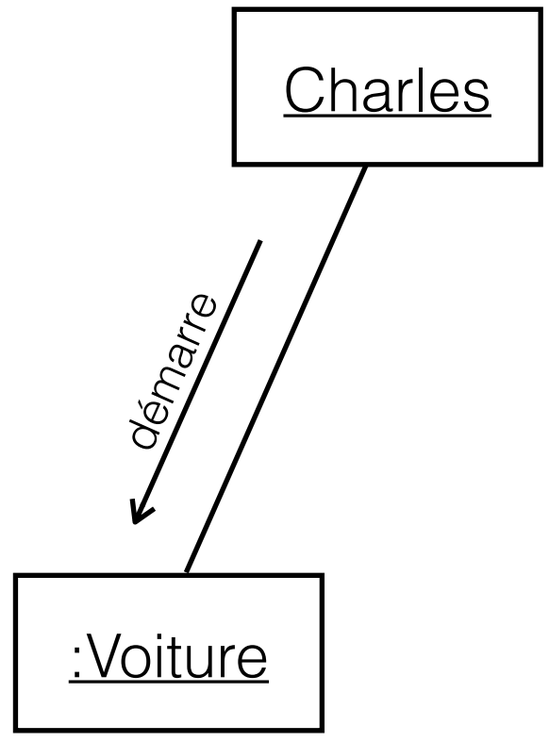


Objet ? Comportement

- Ici Charles démarre la :Voiture

- « démarre » est le **message**

- ce diagramme indique que la :Voiture est dotée de la faculté de démarrer (une **opération** correspondante)



Objet ?

- état et comportement sont liés :
- une opération peut dépendre de l'état courant
- l'état peut-être modifié par une opération

Objet ? Identité

- L'**identité** d'un objet est un attribut implicite (nul besoin de l'expliciter, ni dans la modélisation, ni dans le code) qui caractérise l'objet
- ceci distingue tout objet d'un autre
- ex. : numéro de sécurité sociale, immatriculation numéro de série (ou appelle parfois cet attribut **clé naturelle**)

Objet ?

- Les objets ont parfois des caractéristiques liées au domaine de réalisation
- Dans le domaine informatique on trouve :
    - la **persistance**
        - capacité à transcender le temps (congélation)
    - la **transmission**
        - capacité à transcender l’espace (migration)
    - l’**ubiquité**
        - capacité à être présent en plusieurs lieux

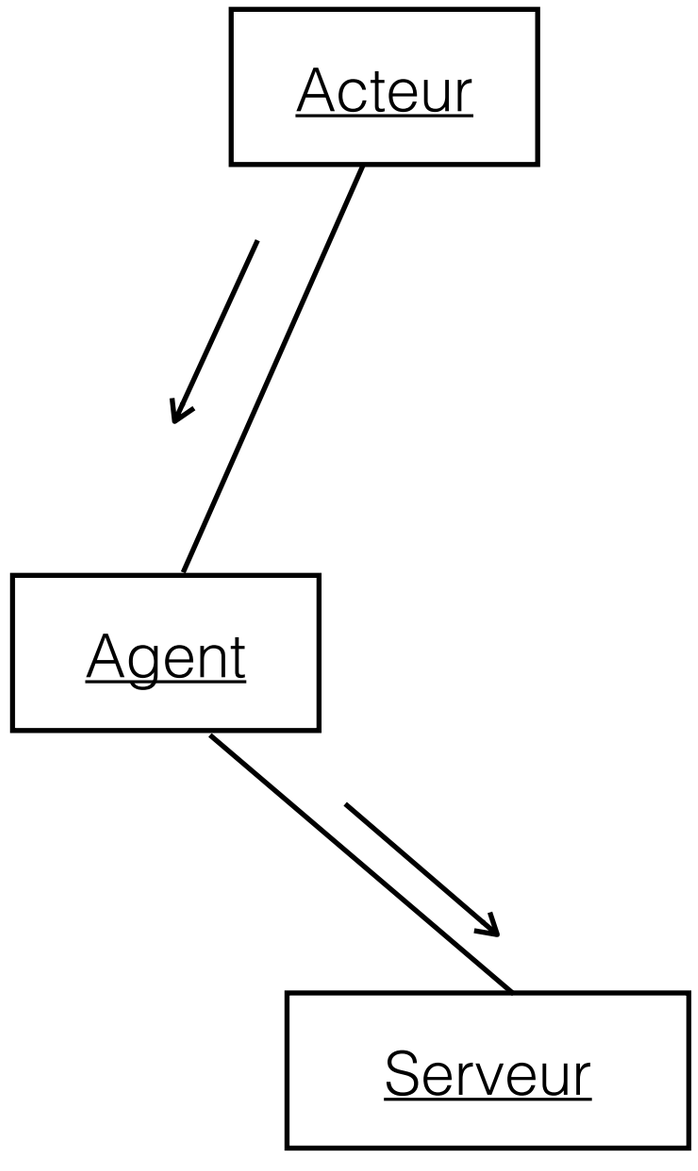
Objet ? Communication

- C’est la **collaboration** effective **entre objets** qui constitue l’exécution d’un programme objet

- Cette collaboration repose sur la **communication par message** entre objets

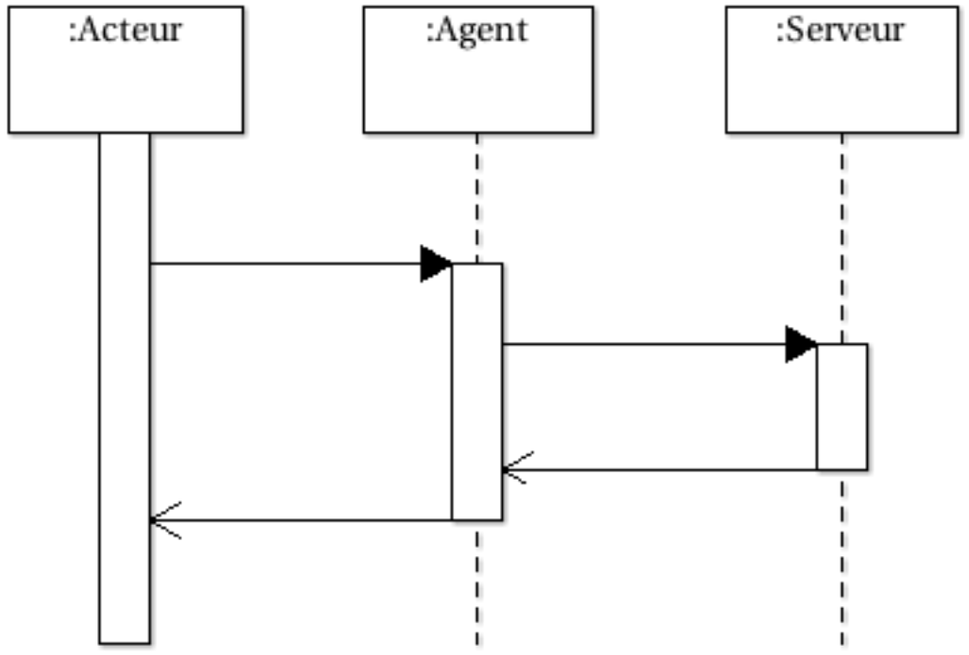
Objet ? Communication

- Il existe trois catégories de comportements :
  - l'acteur
  - l'agent
  - le serveur
- Illustré dans un **diagramme de collaboration**



Objet ? Communication

- La séquence d'interaction peut être représentée par un **diagramme de séquence**



Objet ? Communication

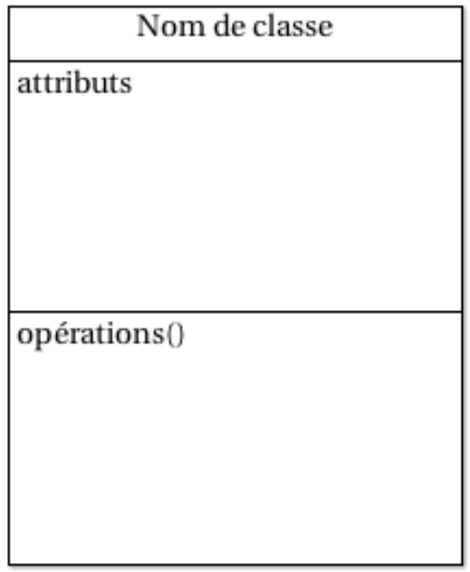
- Il existe 5 catégories de messages :
  - les **constructeurs** (*constructors*) qui créent des objets
  - les **destructeurs** (*destructors*) qui détruisent des objets
  - des **sélecteurs** (*selectors*) qui consultent l'état d'un objet
  - des **modificateurs** (*modifiers*) qui modifient l'état d'un objet
  - des **itérateurs** (*iterators*) qui visitent une structure (objet ou collection d'objets)

Classe ?

- **Représentation conceptuelle** d'un ensemble d'objets similaires permettant de parler de tous en oubliant les détails qui les distinguent
  - classification
    - la **classe** : généralité
    - l'**objet** : particularité
- Note : beaucoup de langage de programmation orientés objets utilisent la notion de classe (mais pas tous).

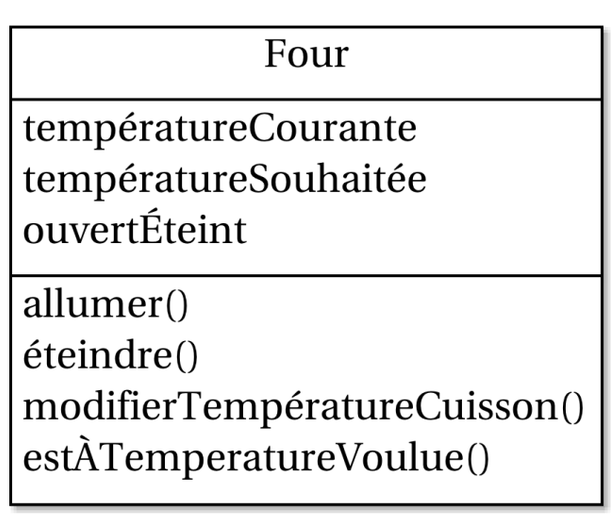
Classe ?

- La représentation d'une **classe**, trois compartiments :
  - **nom**
  - liste des **attributs**
  - liste des **opérations**



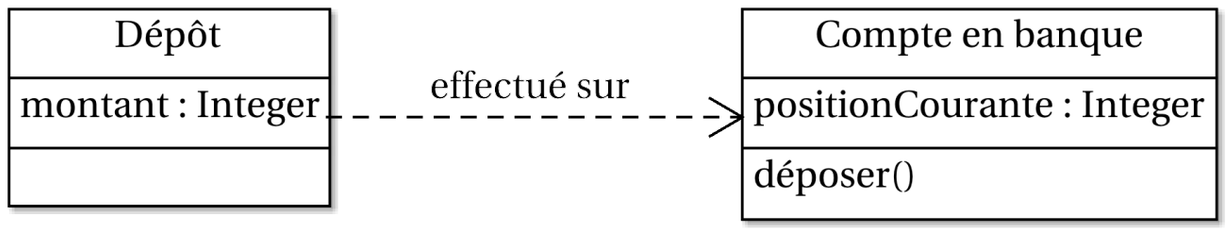
Classe ?

- Exemple :
  - un four de cuisine



Classe ?

- On peut typer les attributs ou les opérations
- Les abstractions peuvent représenter de l'immatériel :
  - un dépôt sur un compte en banque



Classe ?

- Les classes sont généralement décrites à (au moins) deux niveaux :
  - le niveau **abstrait** qui décrit le type (au sens des types des langages de programmation)
  - le niveau **concret** qui décrit la réalisation effective de la classe (son mode opératoire effectif)
- C’est pourquoi on parle de **classe abstraite** ou **classe concrète**

Classe ?

- Les caractéristiques **essentielles** sont décrites dans l'**abstraction**

- Les **détails** dans la **réalisation**
  - Le masquage des détails s'appelle l'**encapsulation**

- **boîte noire** (selon wikipédia) : représentation d'un système sans considérer son fonctionnement interne

Classe ?

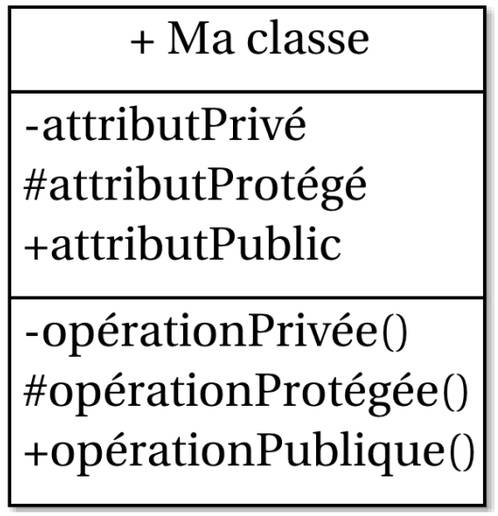
- L'**encapsulation** est essentielle :
  - elle interdit les accès non autorisés aux données et opérations contenues ⇝ garantie d'**intégrité**
  - elle interdit la dépendance vis-à-vis de la réalisation ⇝ **diminution du couplage** (code anti-spaghetti)

Classe ?

- L'**encapsulation** comporte trois niveaux de « protection » :
- **privé** : niveau opaque pour l'extérieur
- **protégé** : niveau opaque pour quiconque n'est pas explicitement ou implicitement autorisé
- **publique** : aucune restriction (plus d'encapsulation)

Classe ?

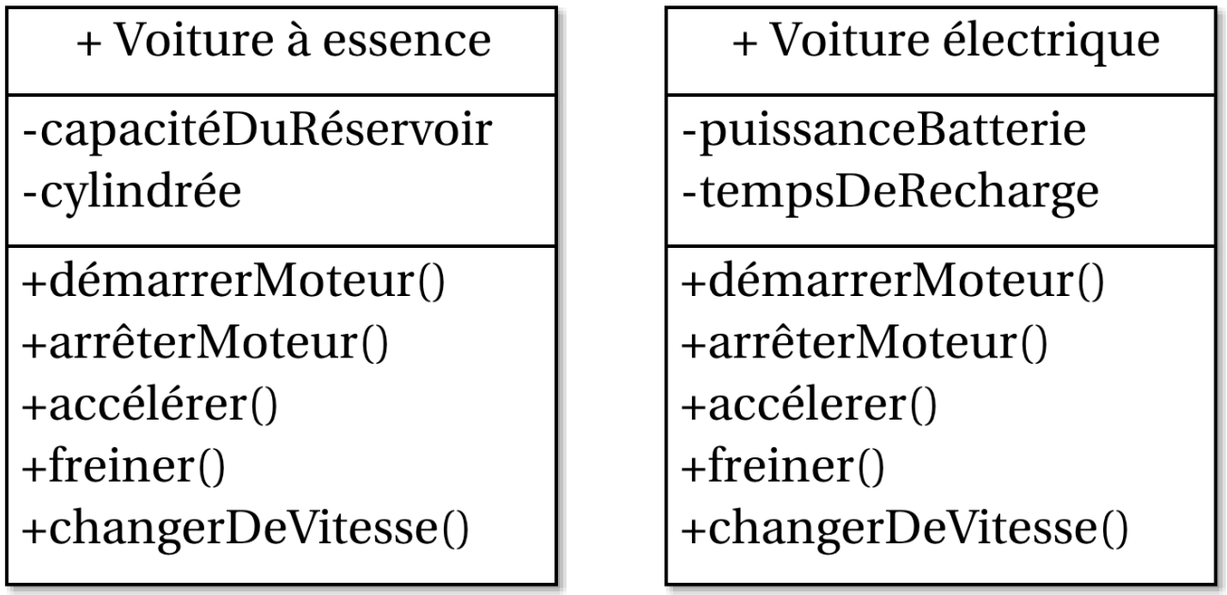
- Représentation des niveaux d'encapsulation



Classe ?

- Le degré privé est réservé à la réalisation

- Le degré public à la spécification abstraite

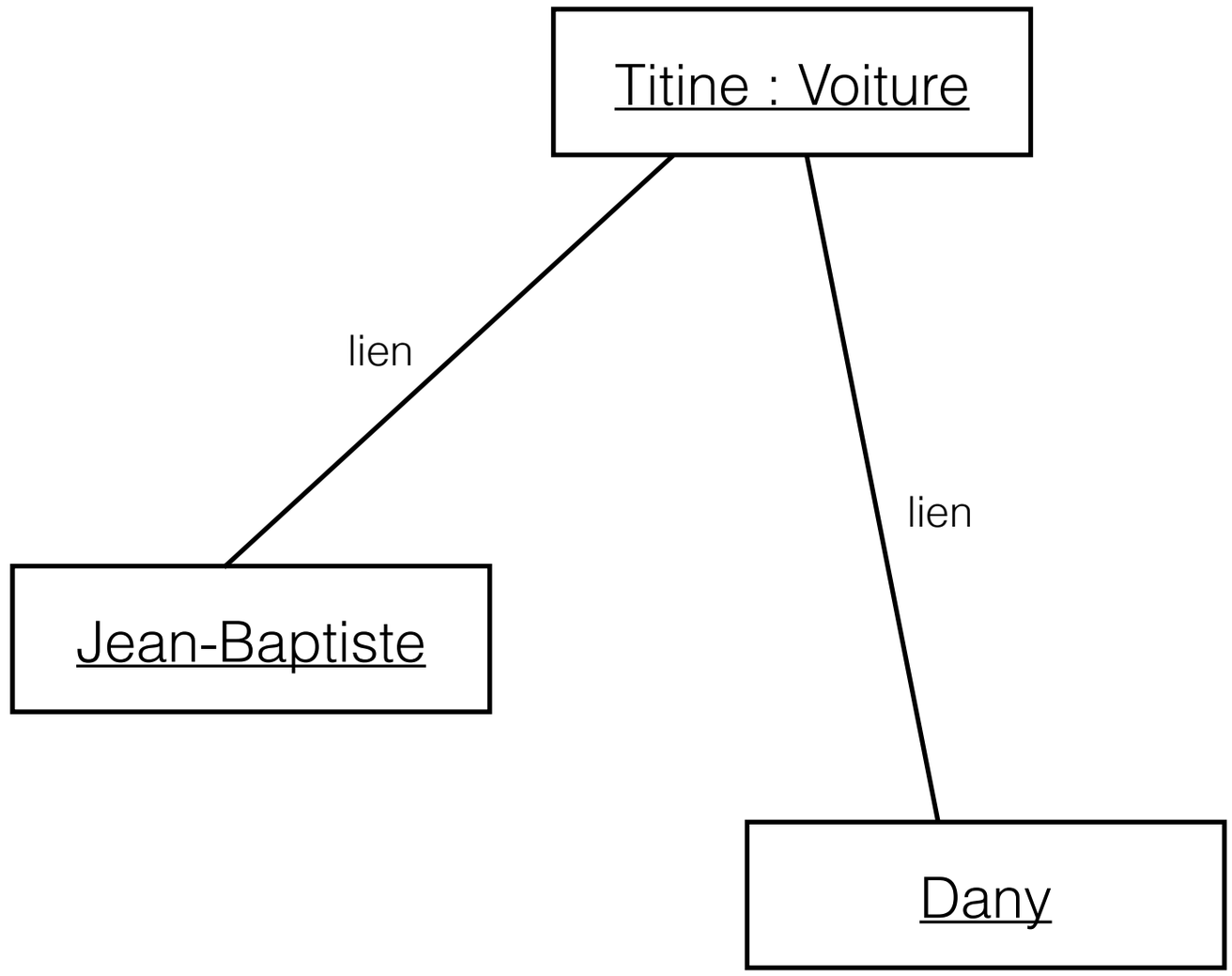


Relations ?

- Les liens entre objets peuvent aussi être abstraits
- les **relations entre classes** sont l'abstraction des **liens entre objets**, comme les classes sont l'abstraction des objets

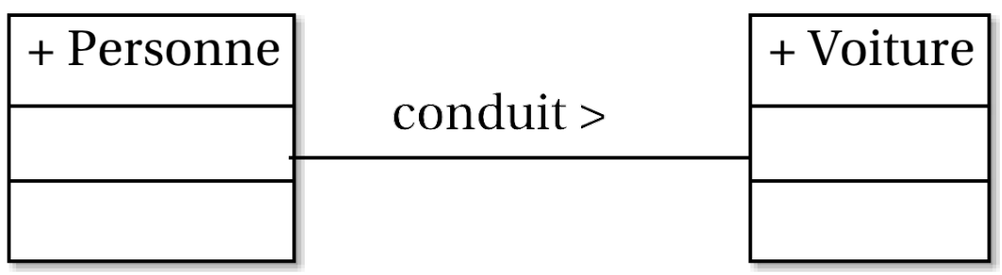
Relations ?

- **Liens** entre objets



Relations ?

- **Relations** entre classes
  - qualification verbale de la relation

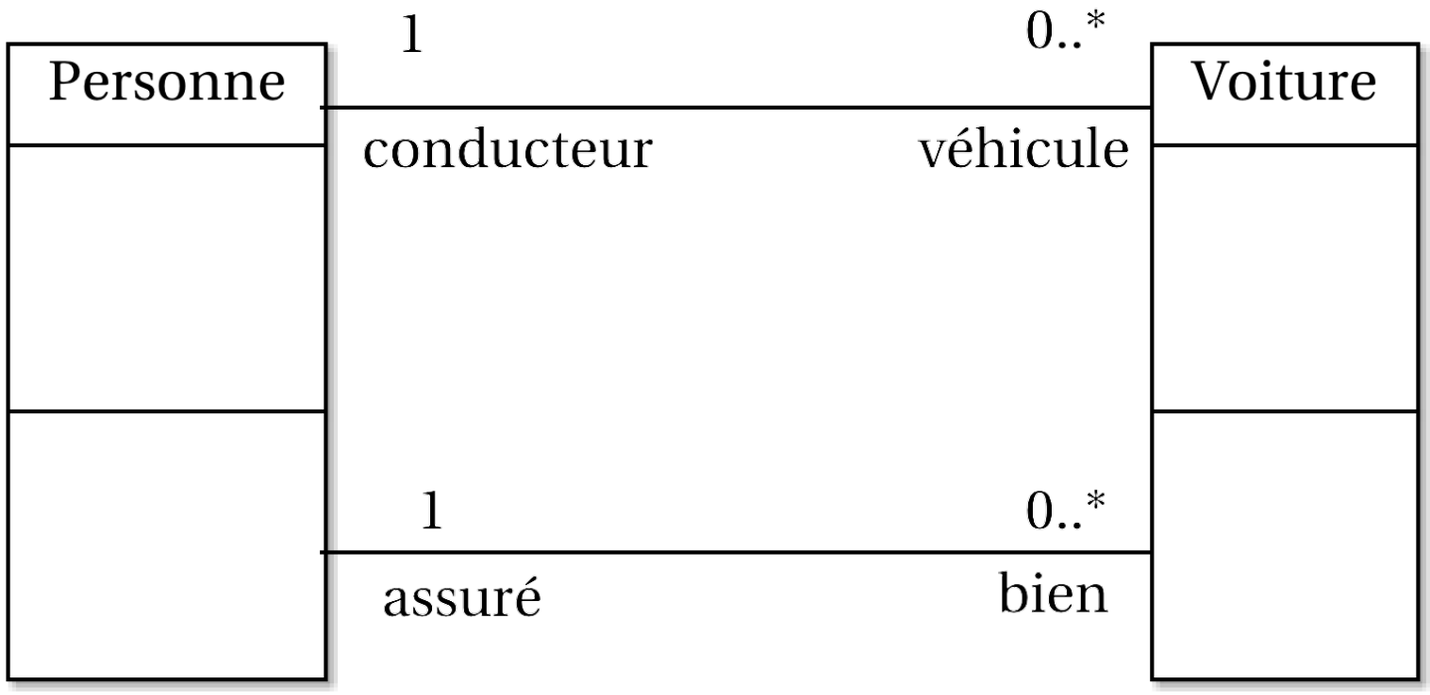


Relations ?

- Il peut y avoir plusieurs relations entre deux mêmes classes
  - on peut **qualifier** les rôles

Relations ?

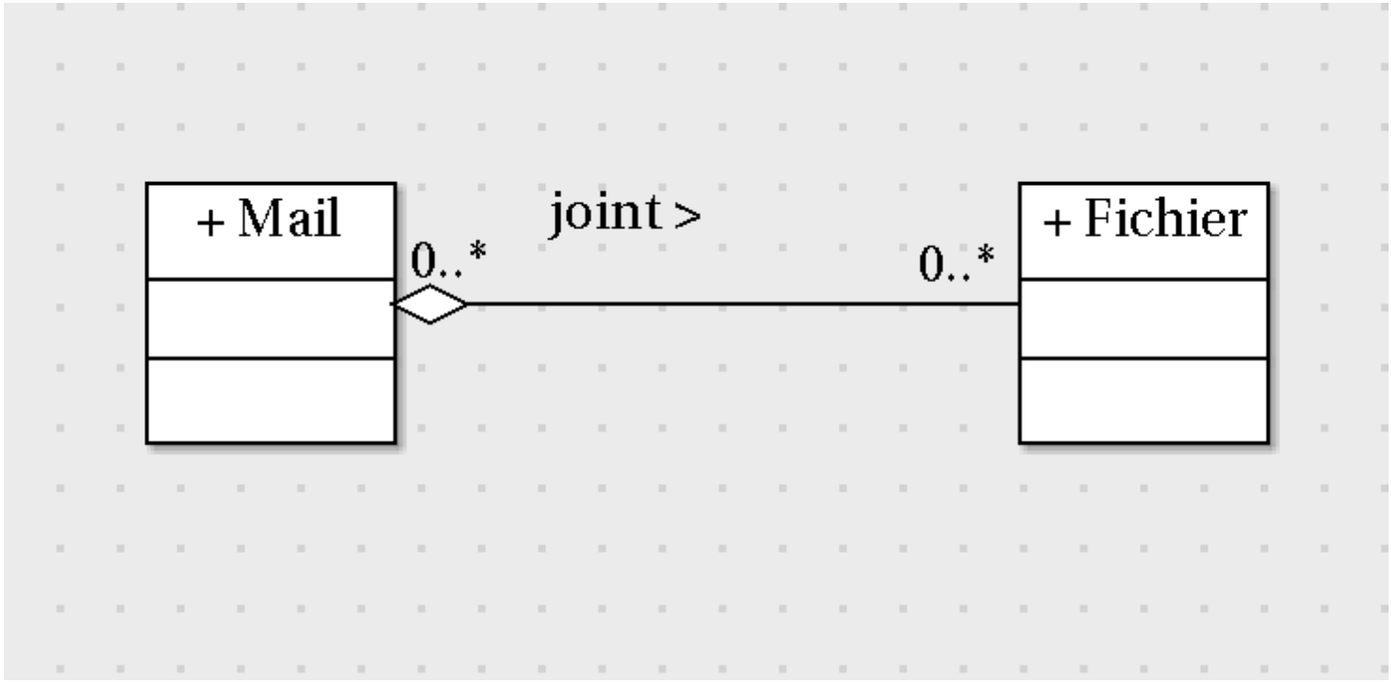
- Les relations peuvent avoir une multiplicité
  - 1
  - 0..1
  - 0..*
  - 1..*
  - M..N
  - *



Relations ?

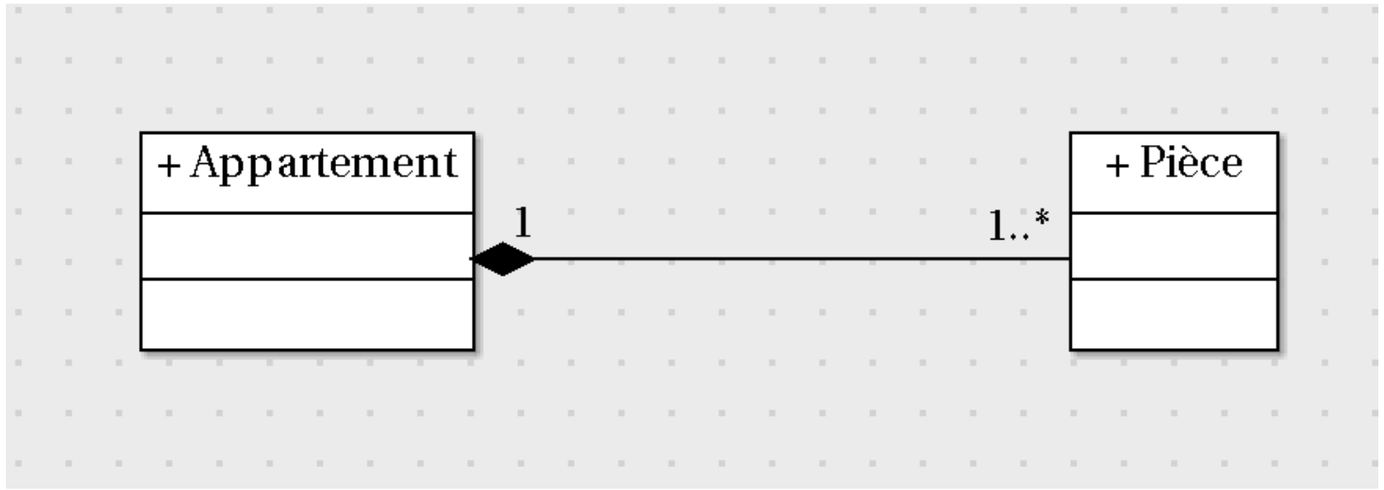
- Certaines relations sont plus fortes que d'autres
  - la plus faible étant la simple **association**
  - plus forte est l'**agrégation**
  - encore plus est la **généralisation-spécialisation**

Relations ?



- L'**agrégation** est
  - **dissymétrique**
  - **lâche**

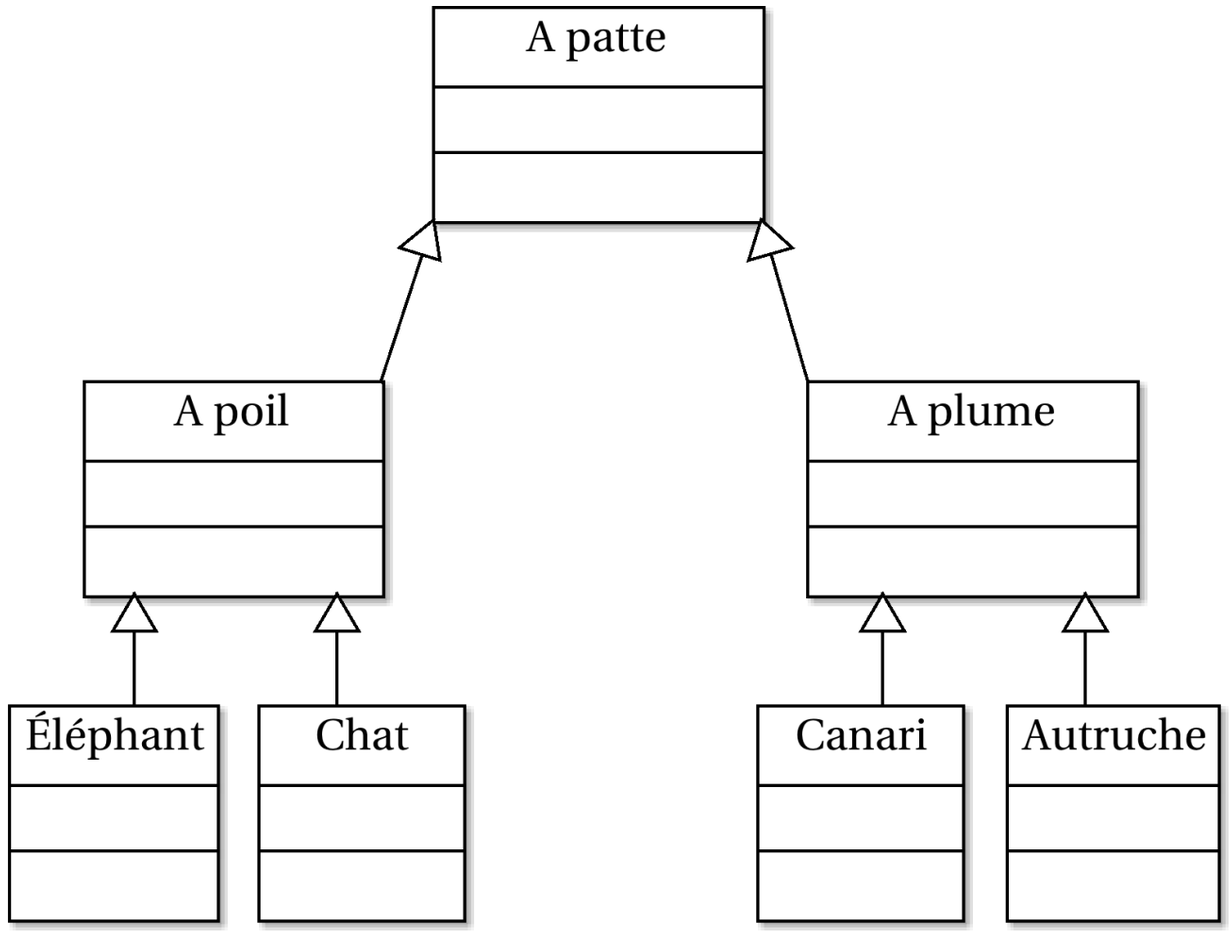
Relations ?



- La **composition** est
  - **dissymétrique**
  - **très forte**

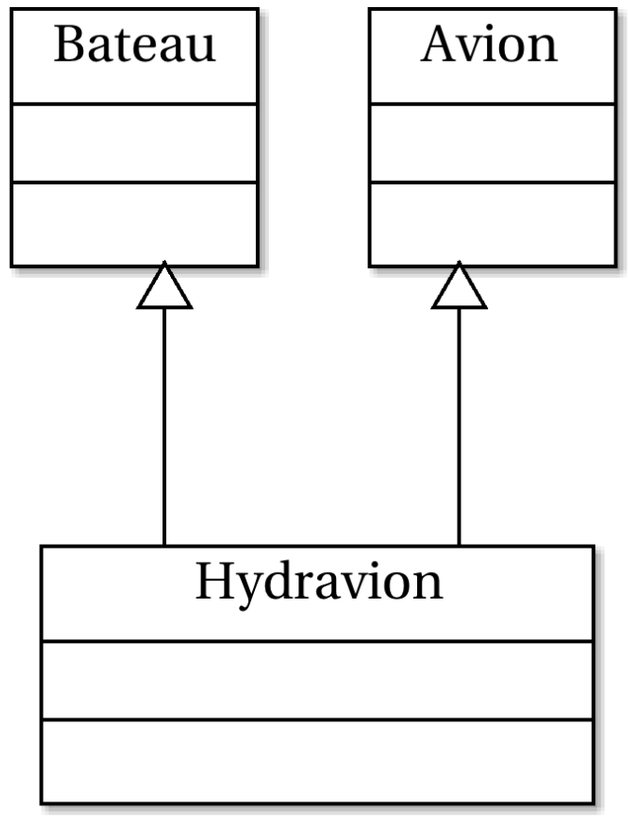
Relations ?

- La **généralisation-spécialisation** est
  - **non symétrique**
  - **non réflexive**
- elle n'est jamais nommée car elle signifie **toujours** *est un* ou *est une sorte de*



Relations ?

- La **généralisation** peut être **multiple**



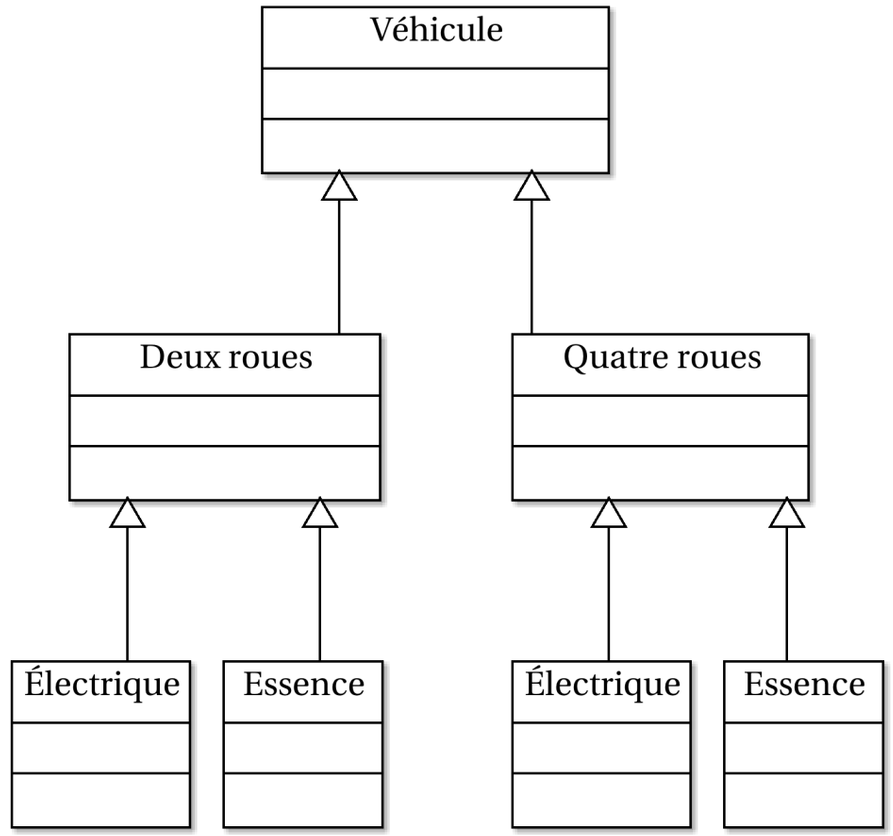
Relations ?

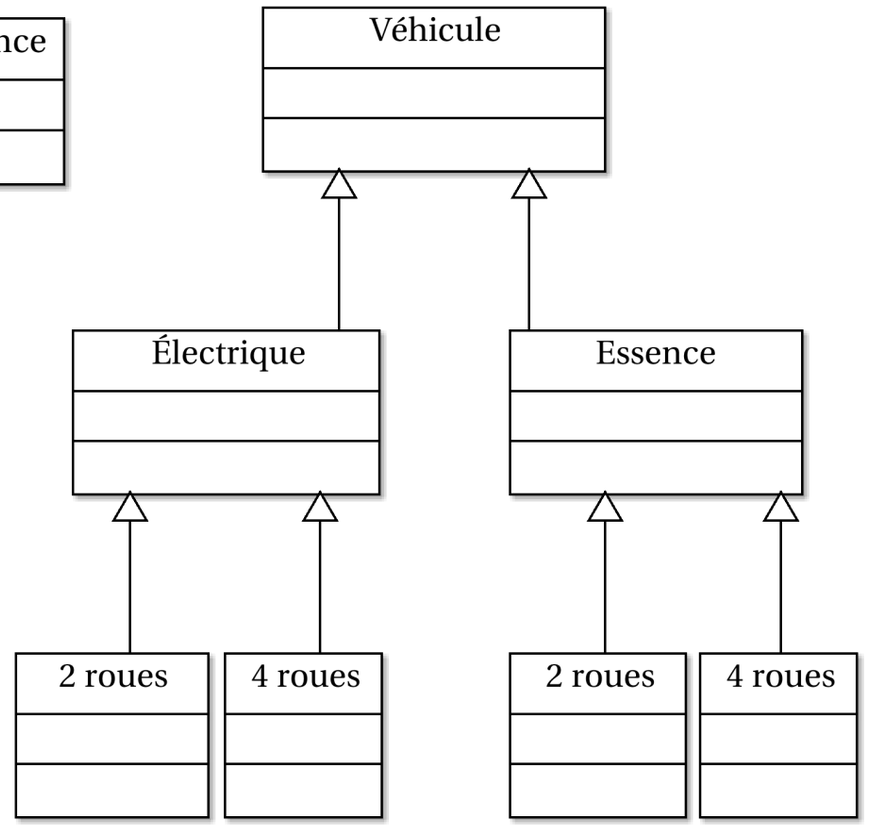
- Attention : obtenir une bonne généralisation par classification (ou pas) est difficile

  - il y a des pièges, des difficultés…

Relations ?

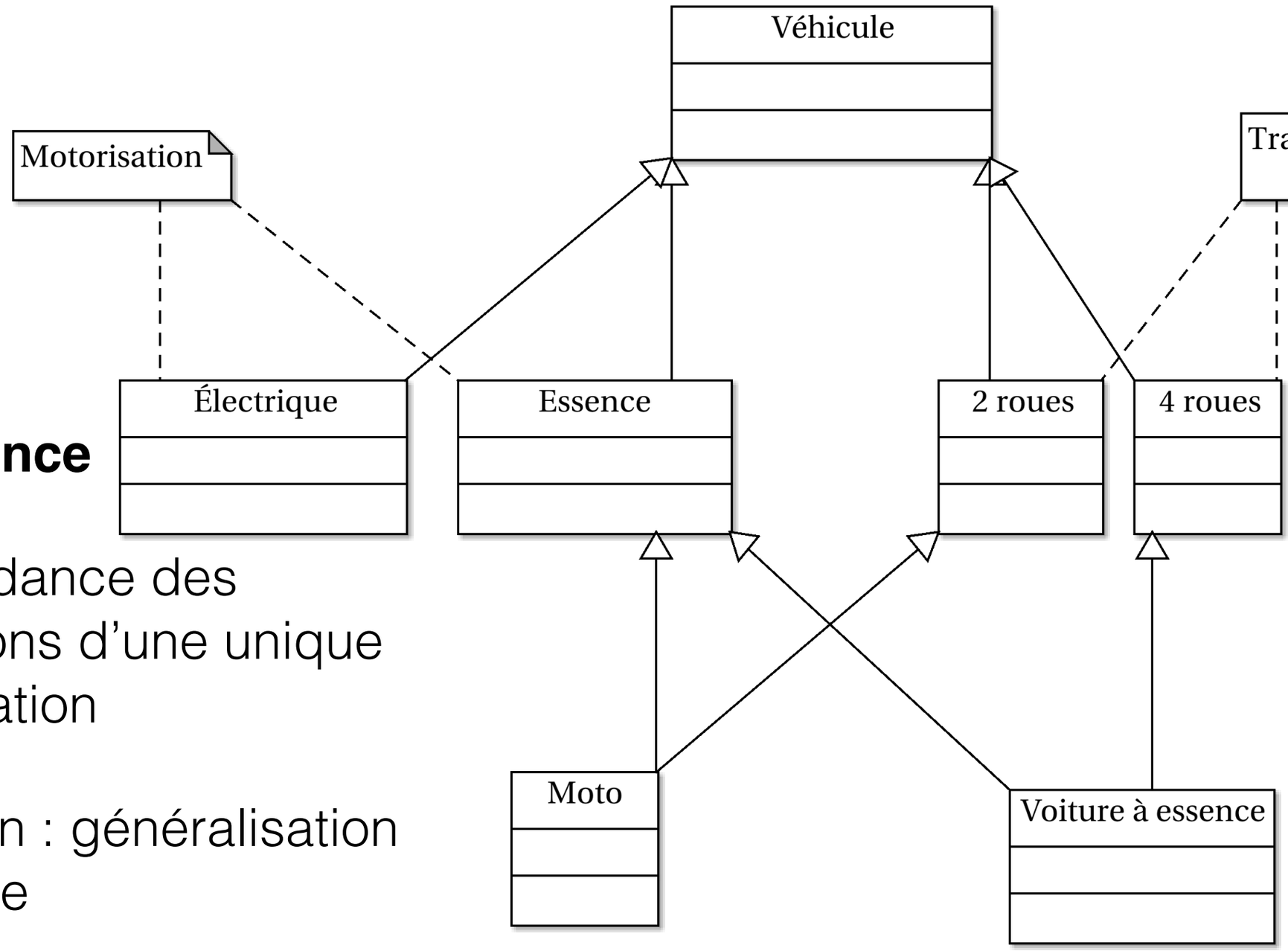
- La **covariance**





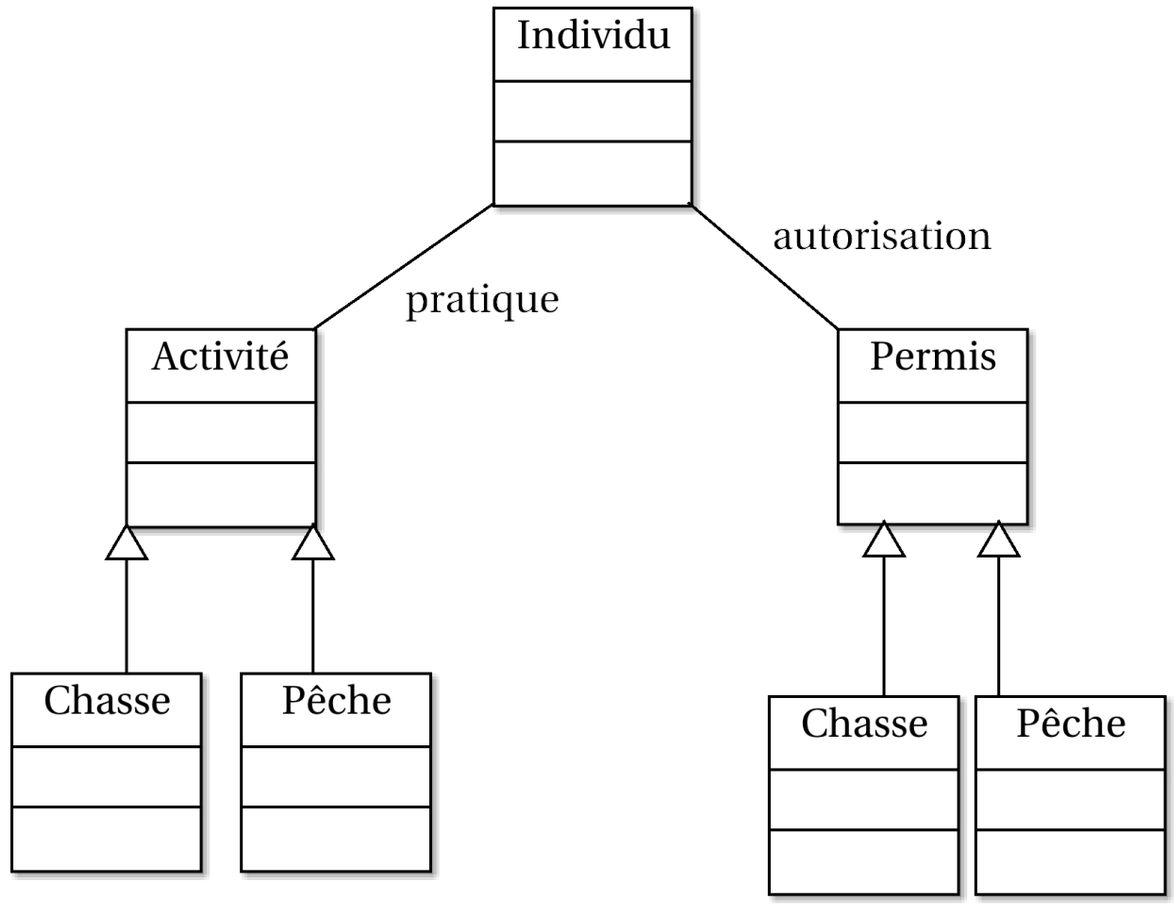
Relations ?

- La **covariance**
  - indépendance des dimensions d'une unique classification
    - solution : généralisation multiple



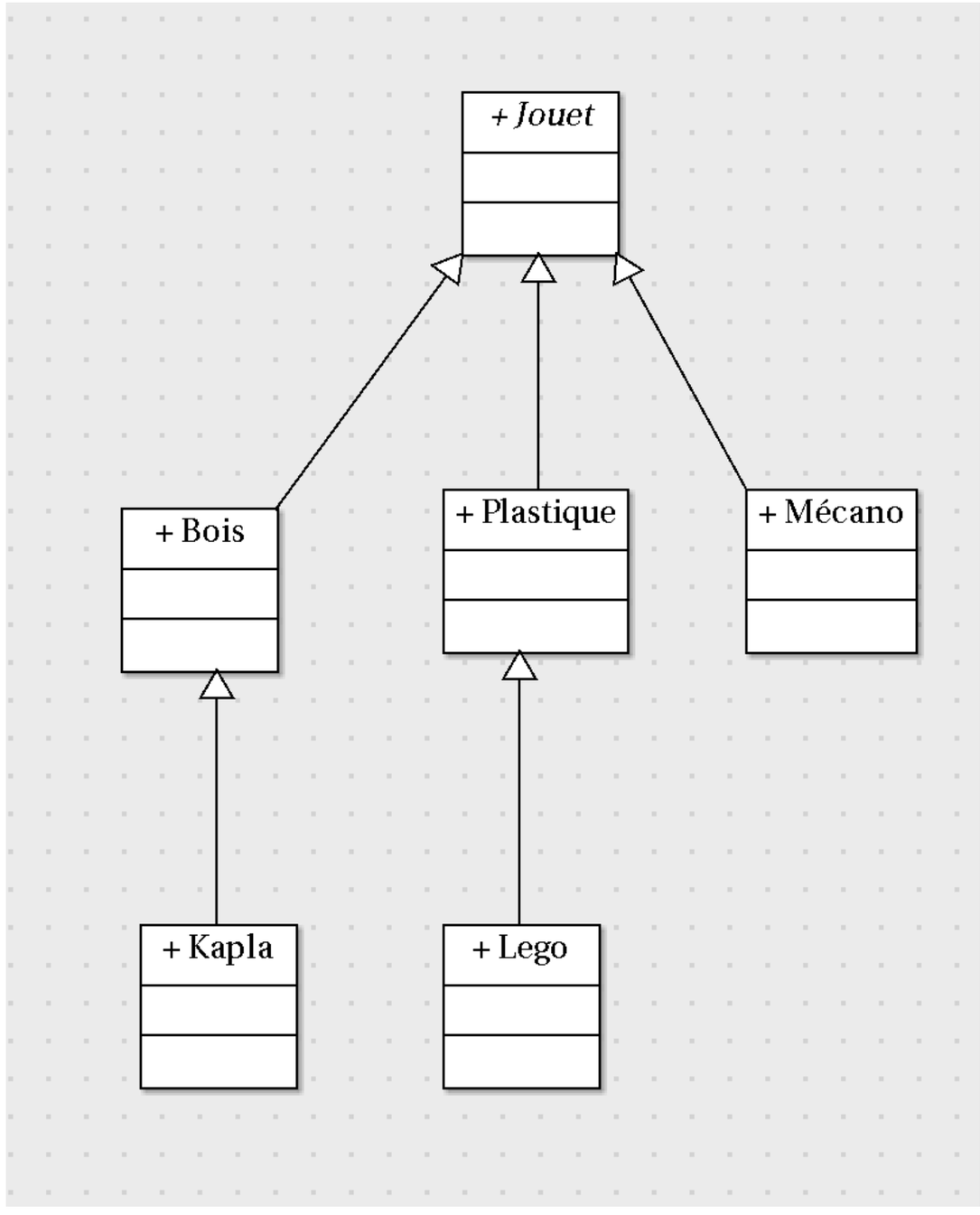
Relations ?

- La **covariance** non réductible
  - indépendance des classifications, parallélisme conceptuel



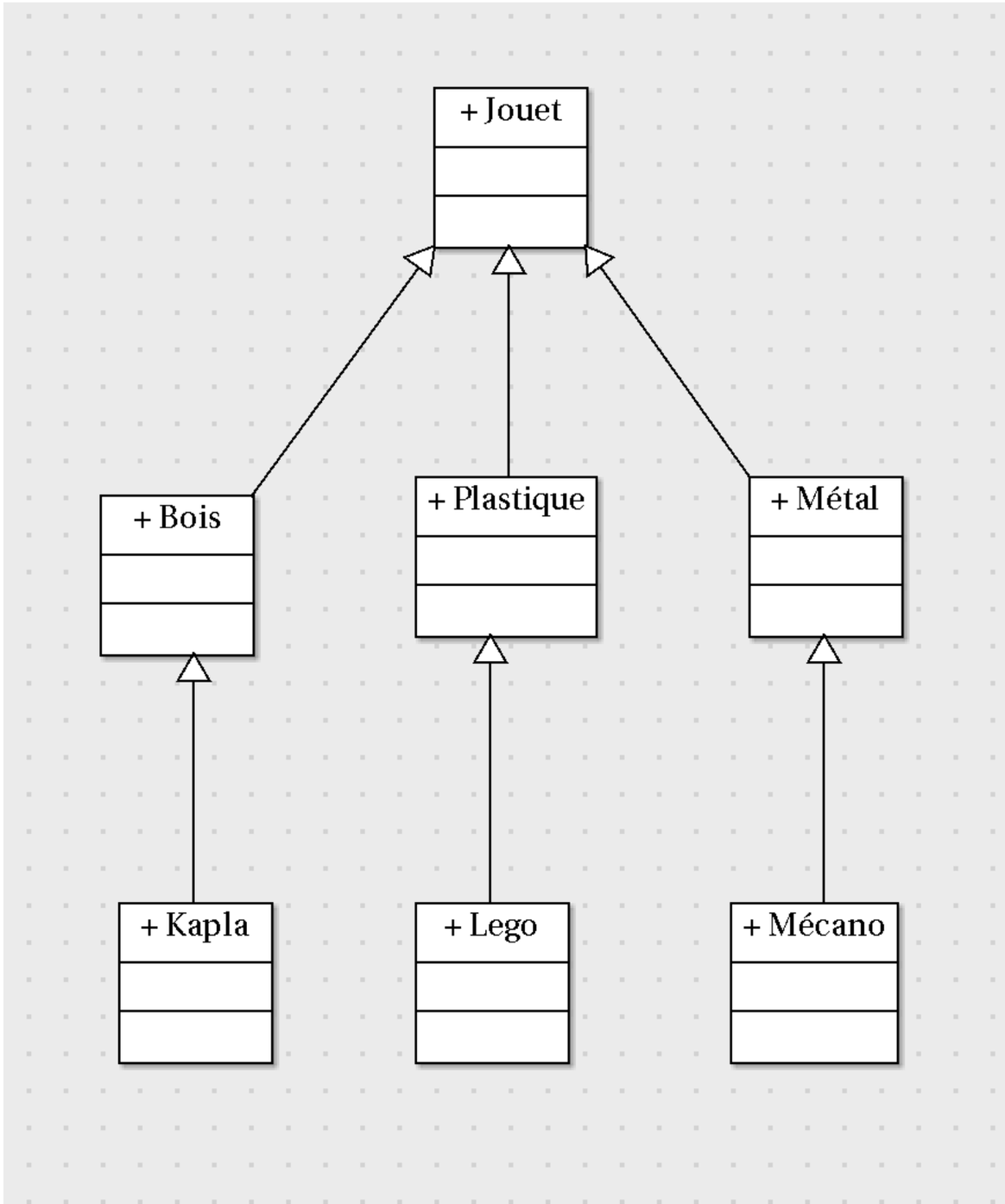
Relations ?

- Le **déséquilibre**
  - solution : introduire de l'**uniformité**



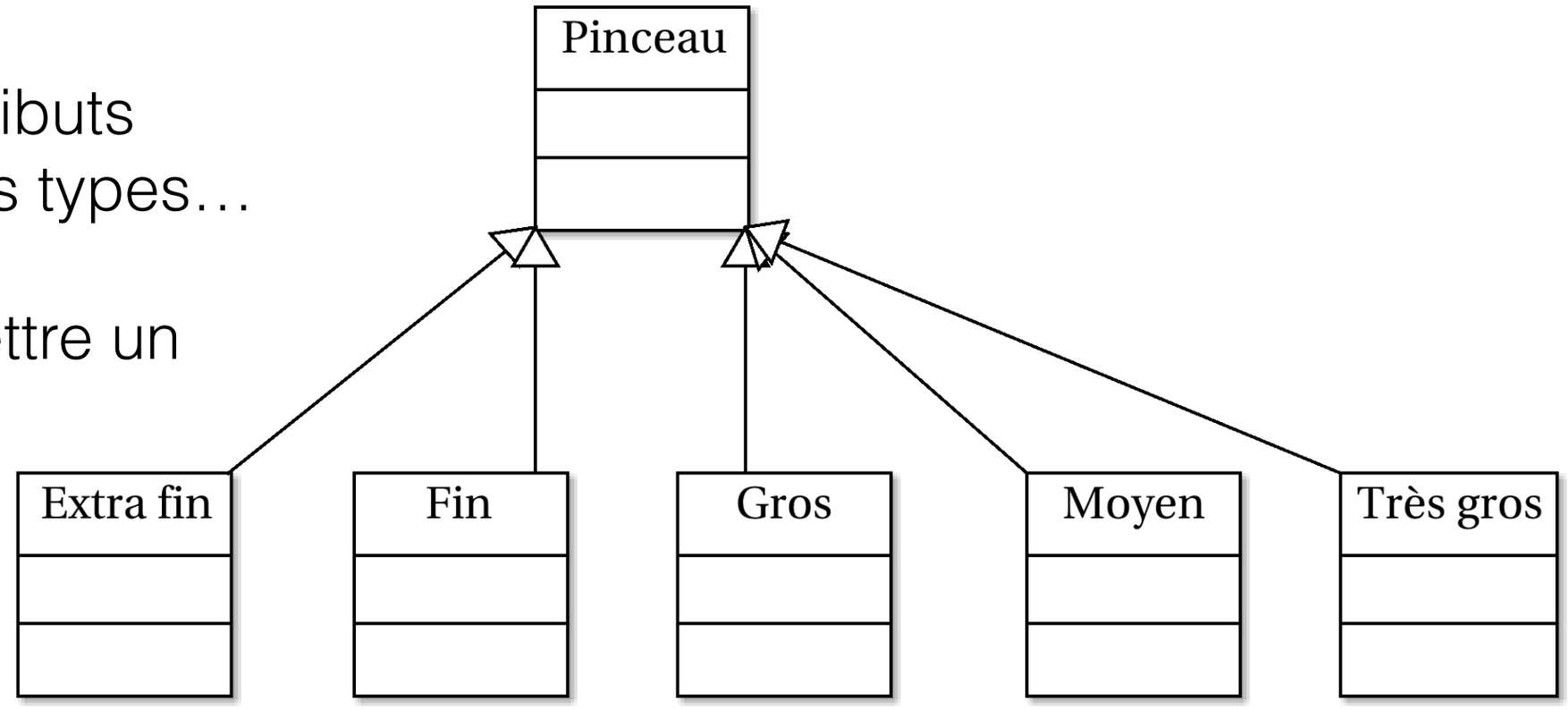
Relations ?

- Le **déséquilibre**
  - solution : introduire de l'**uniformité**



Relations ?

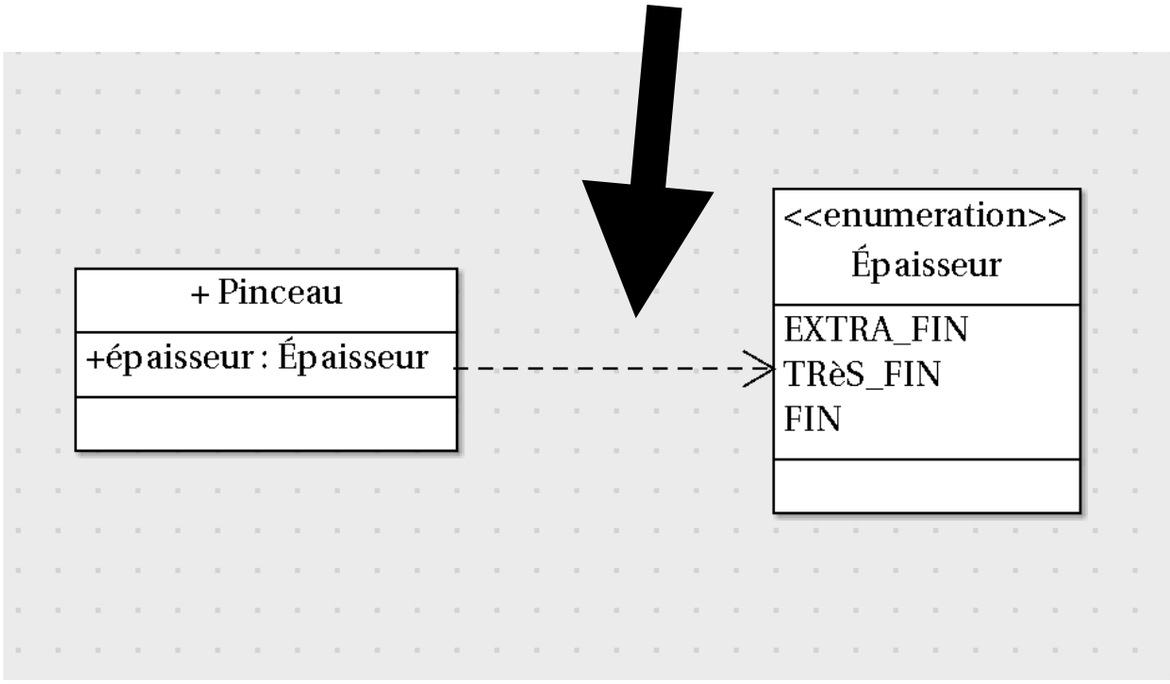
- La **dérive conceptuelle**
  - quand des attributs deviennent des types…
    - solution : mettre un attribut!



Relations ?

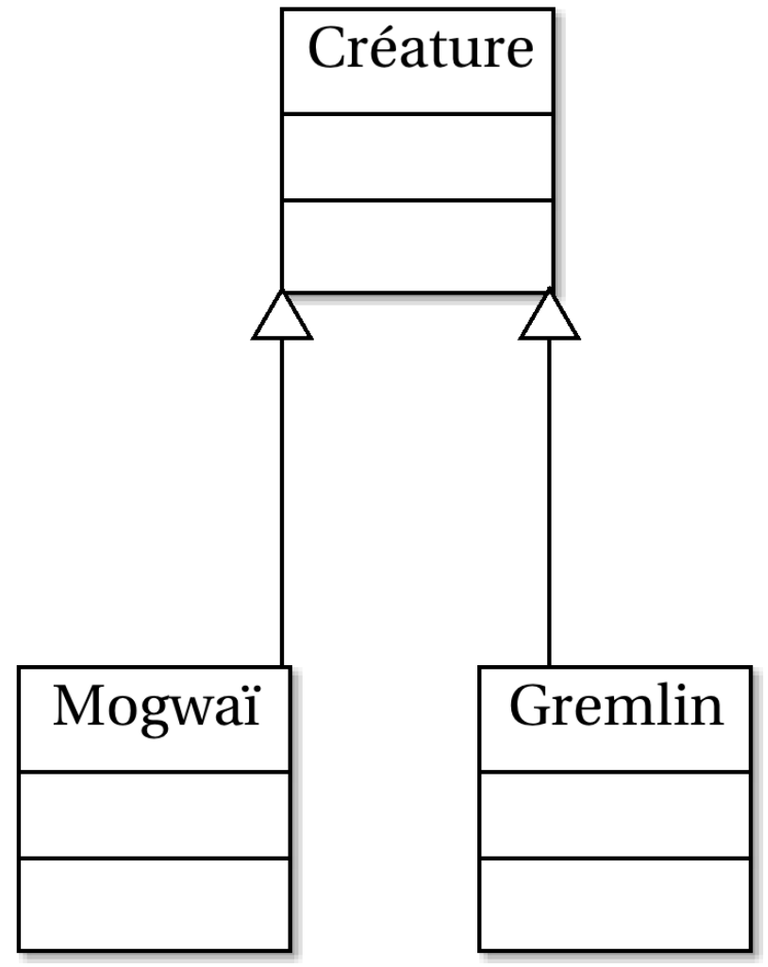
- La **dérive conceptuelle**
  - quand des attributs deviennent des types…
    - solution : mettre un attribut!

Dépendance



Relations ?

- La **mutation**



Relations ?

- La **mutation**
  - solution : séparation du modèle et de l'apparence