

Analyse et modélisation

Séance 9: Diagramme des classes

420-A56-GG

Fortement inspiré du livre « UML 2 en action »

<https://doc.lagout.org/programmation/Databases/SQL/UML.pdf>

Étape 2 dans l'analyse

- P.4 du livre (p.16 dans le lecteur de pdf)

Introduction

- Diagramme le plus important dans toutes les méthodes orientées objet.
- Le diagramme de cas d'utilisation montre un système du point de vue des acteurs.
- Les diagrammes de cas d'utilisation modélisent à **QUOI** sert le système.
- Le système est composé d'objets qui interagissent entre eux et avec les acteurs pour réaliser ces cas d'utilisation :
 - Les diagrammes de classes permettent de spécifier **QUI** intervient à l'intérieur du système
 - Il spécifie également quels **liens** peuvent entretenir les **objets** du système

Introduction

- Le diagramme de classes montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.
- **Remarque:** un même objet peut intervenir dans la réalisation de plusieurs cas d'utilisation.
- Un diagramme de classes n'est pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

Introduction

- Il s'agit d'une vue **statique**, car on ne tient pas compte du facteur temporel dans le comportement du système.
- Il modélise les concepts du domaine d'application ainsi que les concepts internes créés dans le cadre de l'implémentation d'une application.
- Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage particulier.
- Les principaux éléments de cette vue statique sont les **classes** et leurs **relations** : **association**, **généralisation** et plusieurs types de **dépendances**.

Diagrammes statiques: Les classes

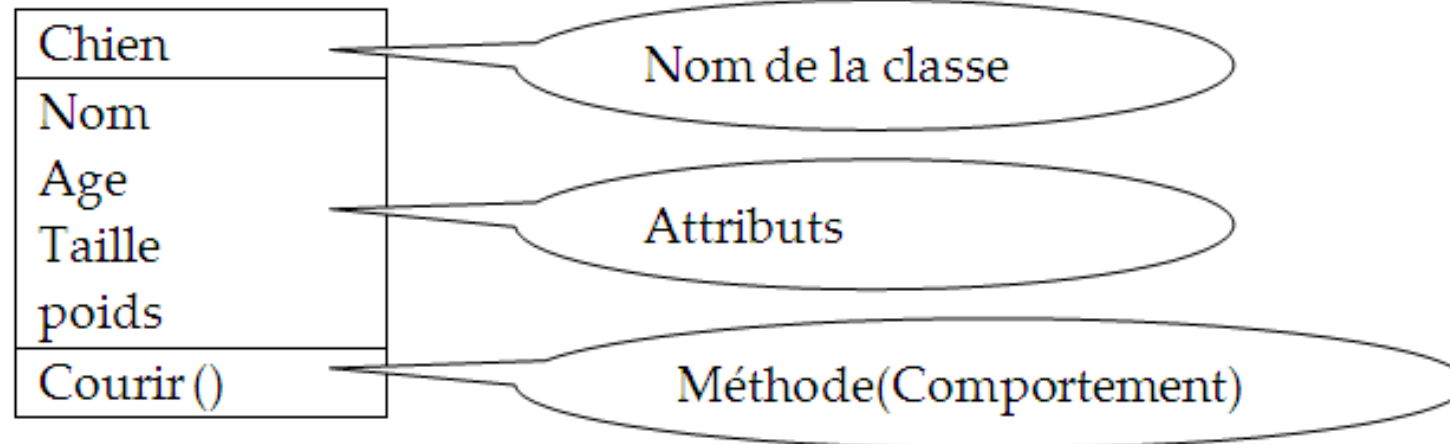
Classe : décrit un groupe d'objets ayant les mêmes propriétés(attributs), un même comportement(opération), et une sémantique commune(domaine de définition).

Un **objet** est une instance d'une classe. La classe représente l'abstraction de ses objets.

Diagrammes statiques: Les classes

- Une *instance* est une concrétisation d'un concept abstrait. Par exemple :
 - la *Mercedes* qui se trouve dans votre garage est une instance du concept abstrait *Automobile* ;
 - l'*amitié* qui lie P1 et P2 est une instance du concept abstrait *Amitié* ;
- Une classe est un concept abstrait représentant des éléments variés comme :
 - des éléments concrets (ex. : des avions),
 - des éléments abstraits (ex. : des commandes de marchandises ou services),
 - des composants d'une application (ex. : les boutons des boîtes de dialogue),
 - des structures informatiques (ex. : des tables de hachage),etc.

Formalisme général d'une classe



1

<u>MonChien:Chien</u>
<u>Fido</u>
2
55 cm
20kg

Exemple de représentation d'objets

Formalisme général d'une classe

- Personne travaille dans une entreprise ? classes

Association des classes

- Une association entre les classes représente les liens qui existent entre les instances de ces classes.
- Définir le rôle d'association



Multiplicité

- Précise le nombre d'instances qui participent à une relation. Aussi appelé **Cardinalités**.

Domaine de valeurs:

Intervalle fermé ex : 3...10

Valeurs exactes ex : 2,3,8

Valeur indéterminée ex : 1..*

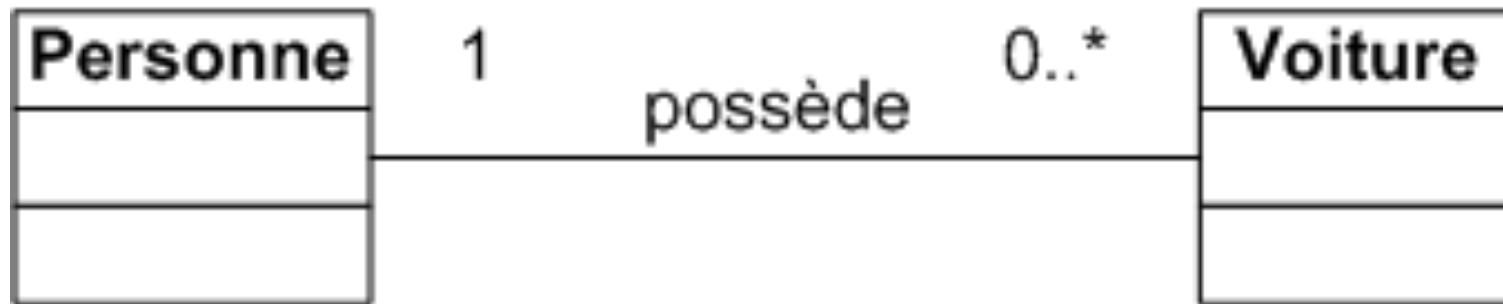
(* == 0..* aussi accepté)

Exemple



- À une instance de la Classe1 correspond 0 ou 1 instance de la Classe2
- À une instance de la Classe2 correspond 0 ou n instance de la Classe1
- EX: Diagramme de classes avec cardinalités pour : personne et voiture.

Exemple



- une personne peut posséder 0 ou * voitures
- Une voiture est possédée par une seule personne (on peut aussi avoir 0,1)

Héritage

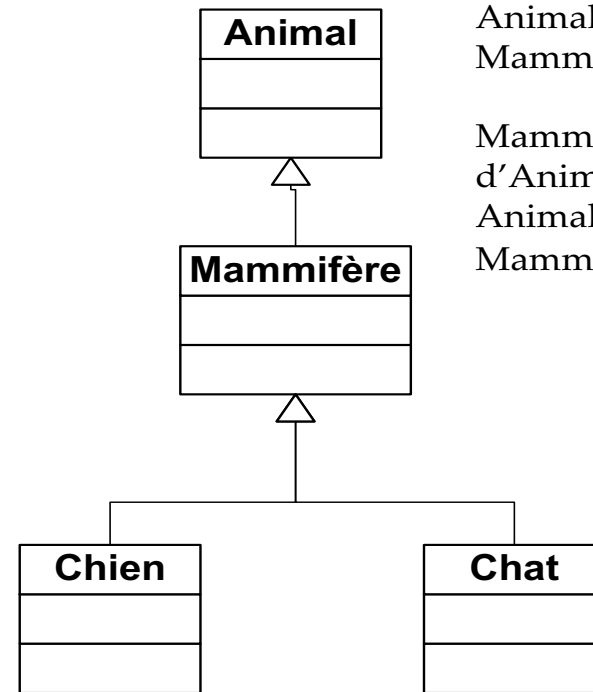
Les hiérarchies de classes permettent de gérer la complexité, en ordonnant les objets au sein d'arborescences de classes, d'abstraction croissante.

- **La Spécialisation** : étendre les propriétés d'une classe, sous forme de sous-classes, plus spécifiques.
- **La Généralisation** : factoriser les propriétés d'un ensemble de classes sous forme d'un super-classe plus abstraite.

Classification – héritage

- **L'héritage** (spécialisation et généralisation) permet la classification des objets.
- Une bonne classification est stable et extensible :
 - ne classifiez pas les objets selon des critères instables (selon ce qui caractérise leur état) ou trop vagues (car cela génère trop de sous-classes).
- Si **Y** hérite de **X**, cela signifie que "**Y est une sorte de X**". Donc ici, un **Roman** est un **Livre** qui est lui-même une **Œuvre**.

Classification - héritage



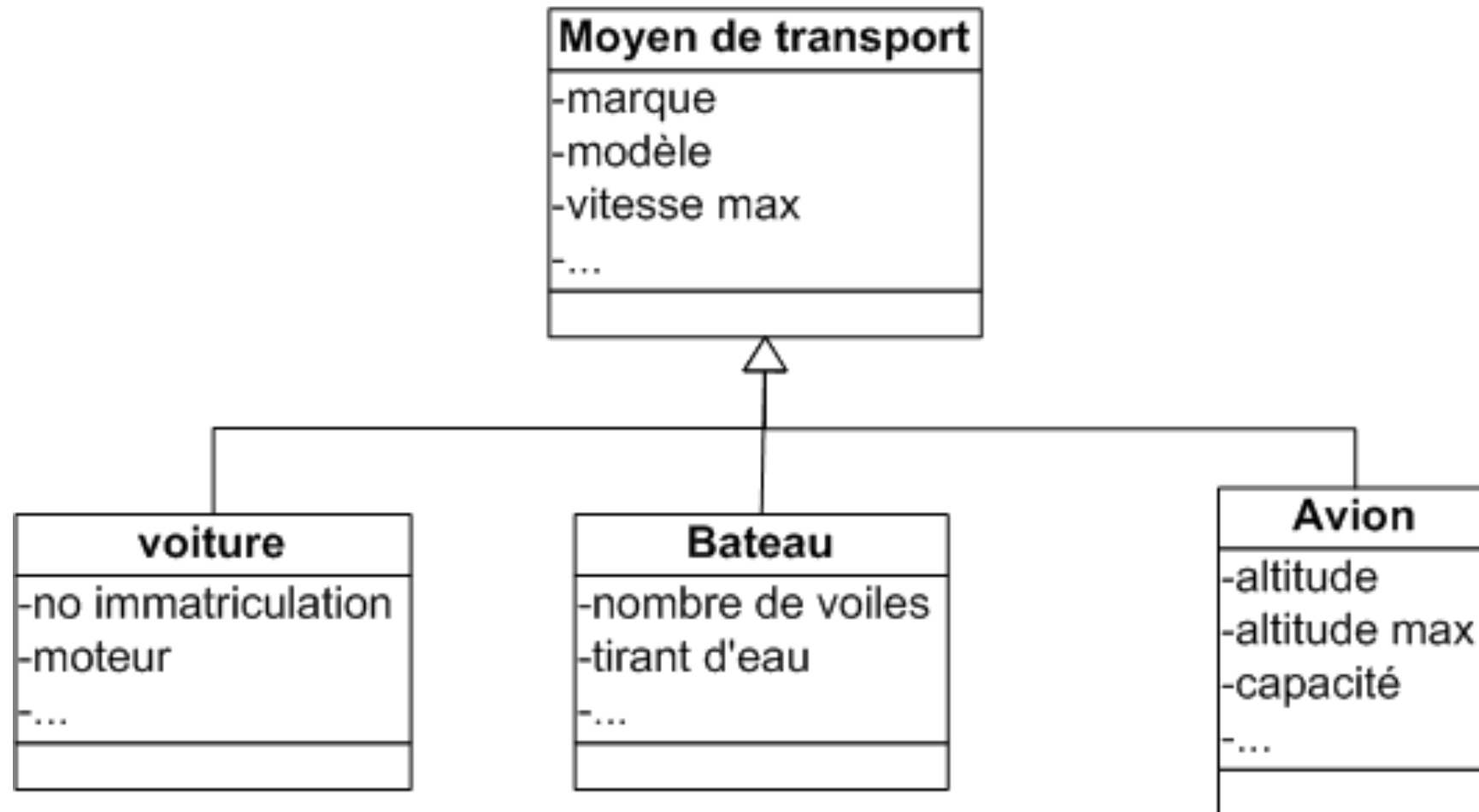
Animal est une surclasse de Mammifère,
Mammifère est une sous-classe d'Animal,

Mammifère devient une spécialisation
d'Animal.
Animal devient une généralisation de
Mammifère

Mammifère est une surclasse de Chien et Chat,
Chien et Chat sont des sous-classes de Mammifère

Chien et Chat devient une spécialisation de
Mammifère.
Mammifère devient une généralisation de Chien et
Chat

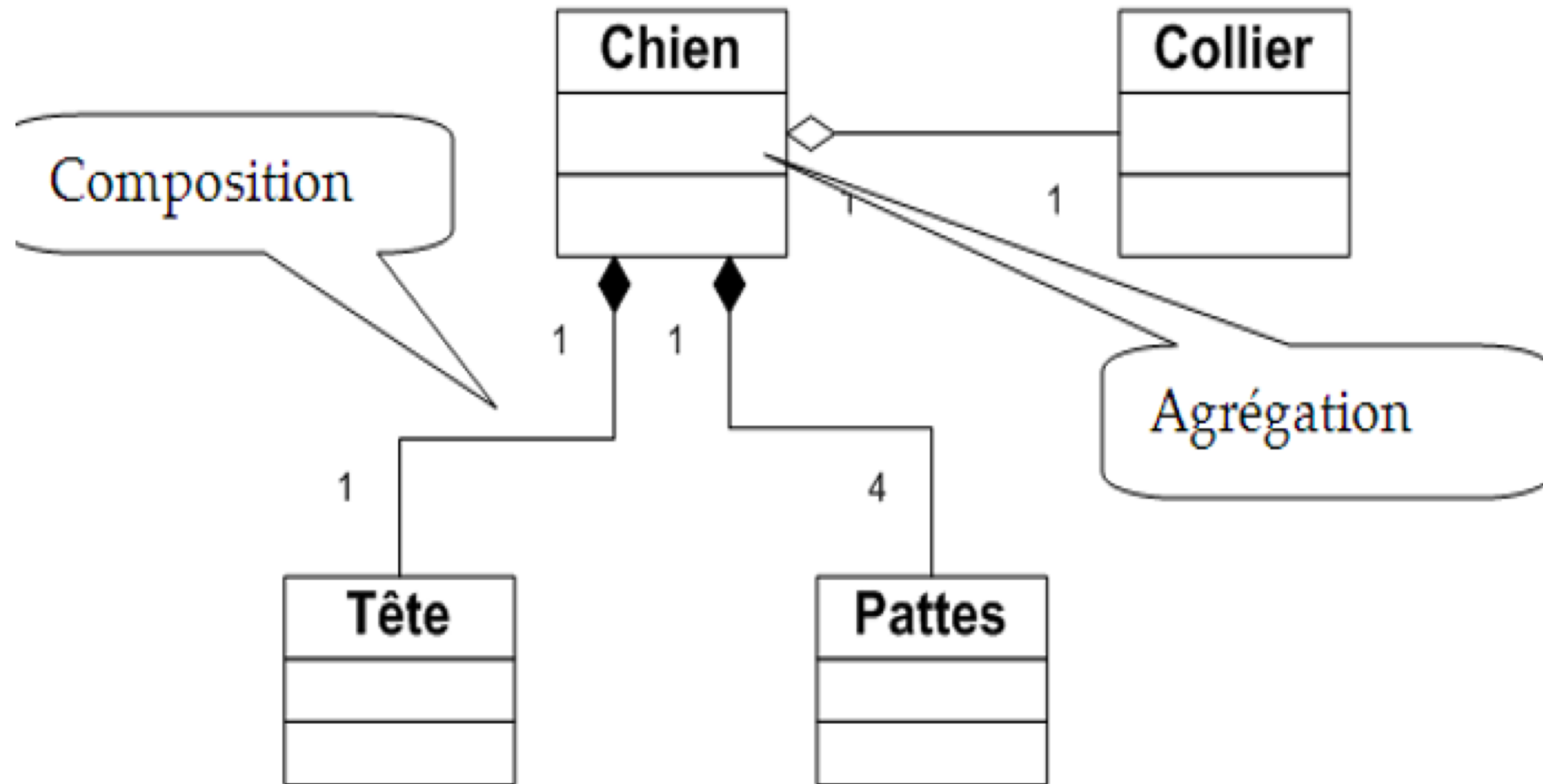
Classification - héritage



Composition (et agrégation)

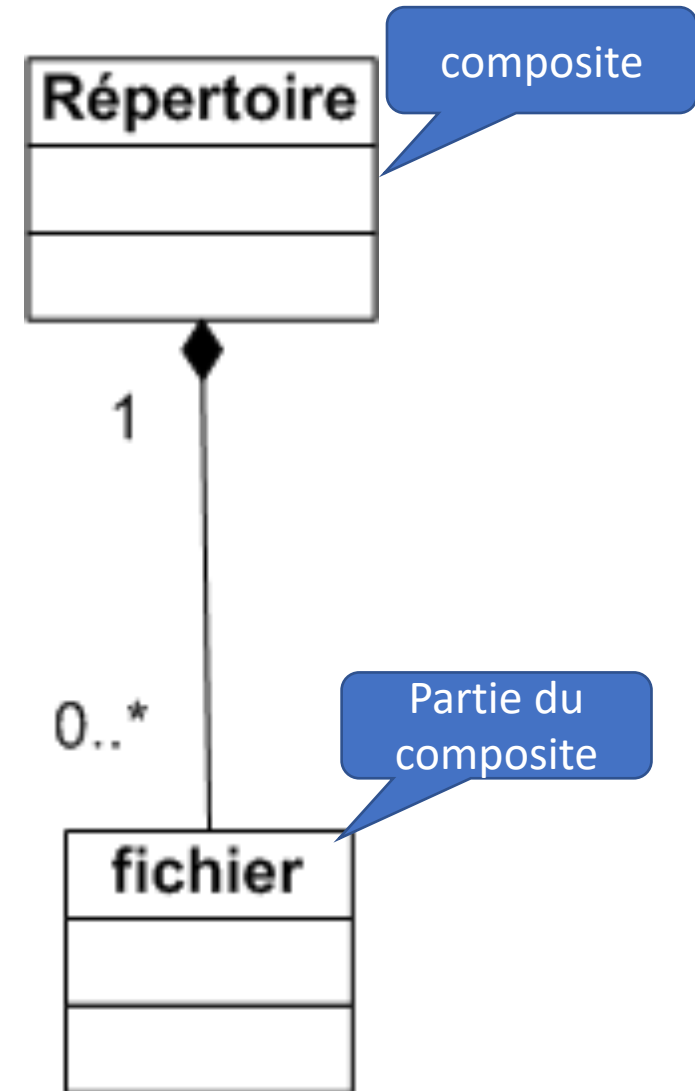
- La composition (forte, classe d'objet ne pouvant être partagé, objet obligatoire pour bien définir l'objet).
 - La destruction du composé (composite) entraîne la destruction des composants (partie du composite).
- L'agrégation (composition faible, facultatif, l'objet peut être partagé avec une autre classe d'objet).
 - La destruction de l'objet n'entraîne pas la destruction des agrégats.

Composition (et agrégation)



Lien « composition »

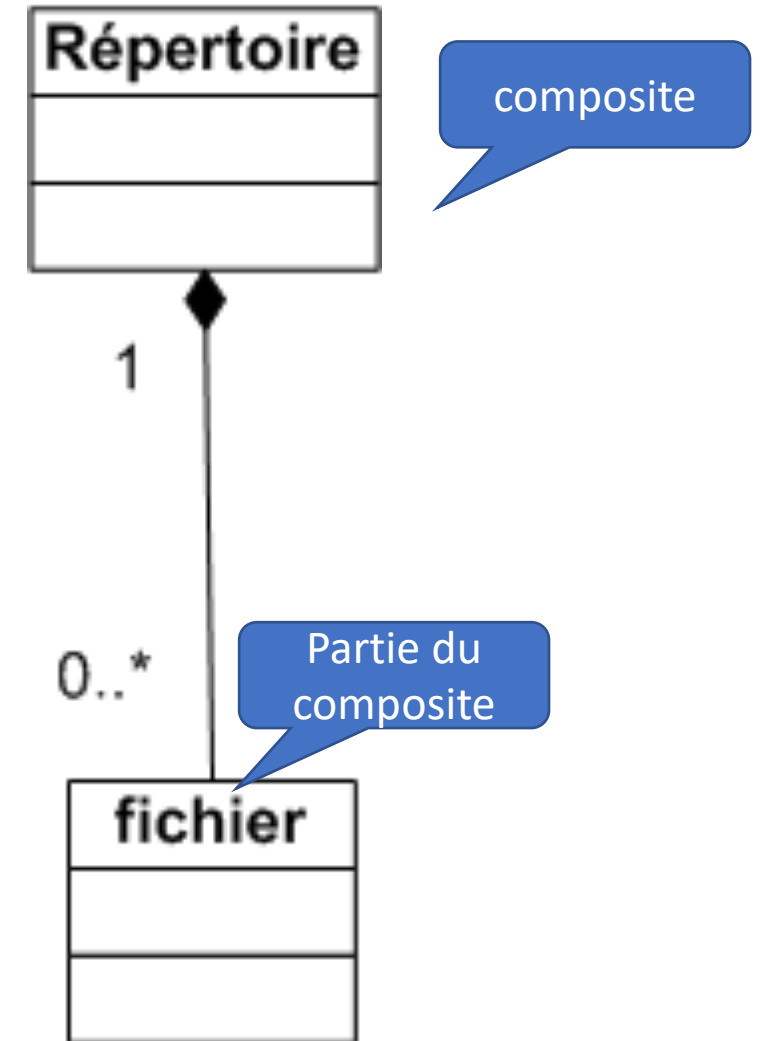
Un répertoire contient des fichiers.



Lien « composition »

- 1^{er} critère : la multiplicité ne doit pas être supérieur à un du côté du composite. C'est correct puisqu'un fichier appartient à un seul répertoire.
- 2^{ième} critère : le cycle de vie des parties dépend de celui du composite. Ici la destruction d'un répertoire entraîne la destruction des fichiers qu'il contient.

C'est donc un lien de composition.



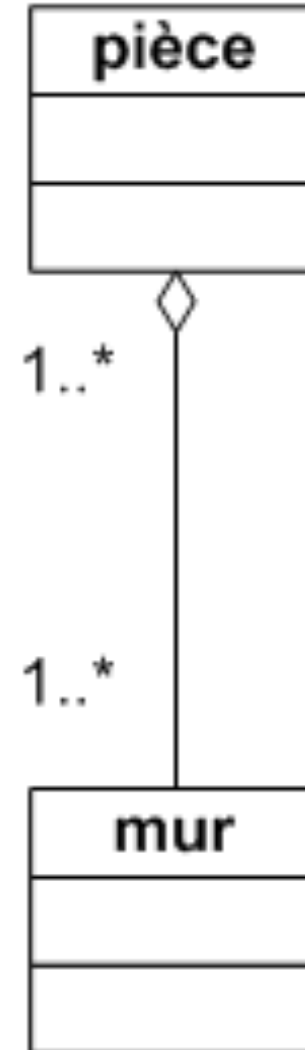
Lien agrégat

Une pièce contient des murs.

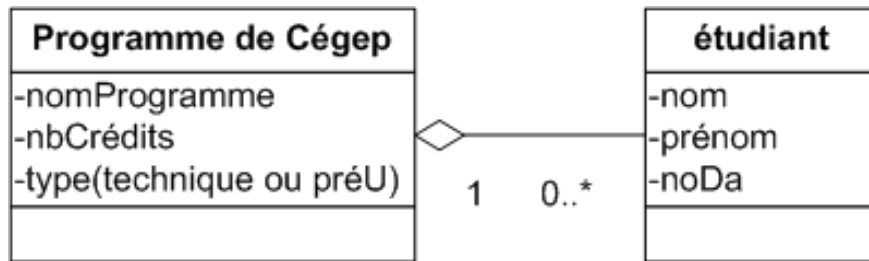
1^{er} critère à vérifier : la multiplicité ne doit pas être supérieur à un du côté du composite.

C'est faux! un mur peut appartenir à plusieurs pièces....ce n'est pas un lien de composition mais un lien d'agrégation

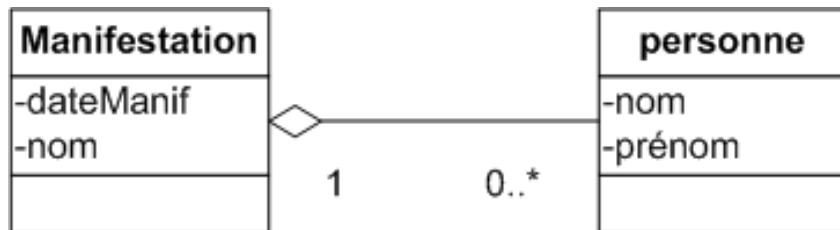
(On suppose aussi ici qu'une pièce peut avoir un seul mur circulaire....)



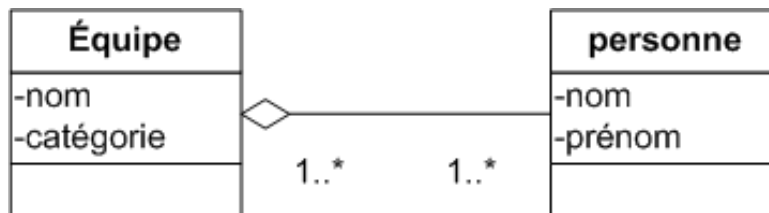
Exemple de lien d'agrégat



Un programme a(est composé) zéro ou plusieurs étudiants
Un étudiant est inscrit dans un seul programme

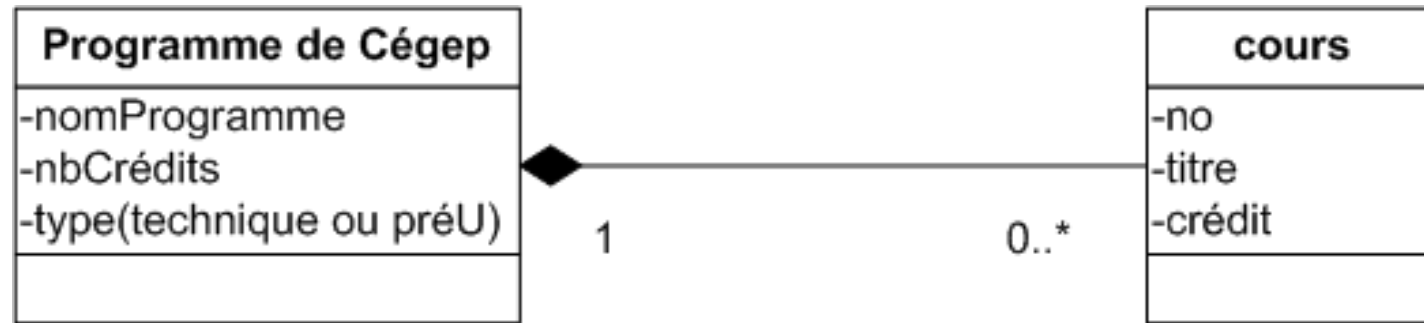


Une manifestation contient(est composé) zéro ou plusieurs personnes
Une personne est dans la manifestation



Une équipe contient(est composé) une ou plusieurs personnes
Une personne est dans une ou plusieurs équipes

Exemple de lien de composition



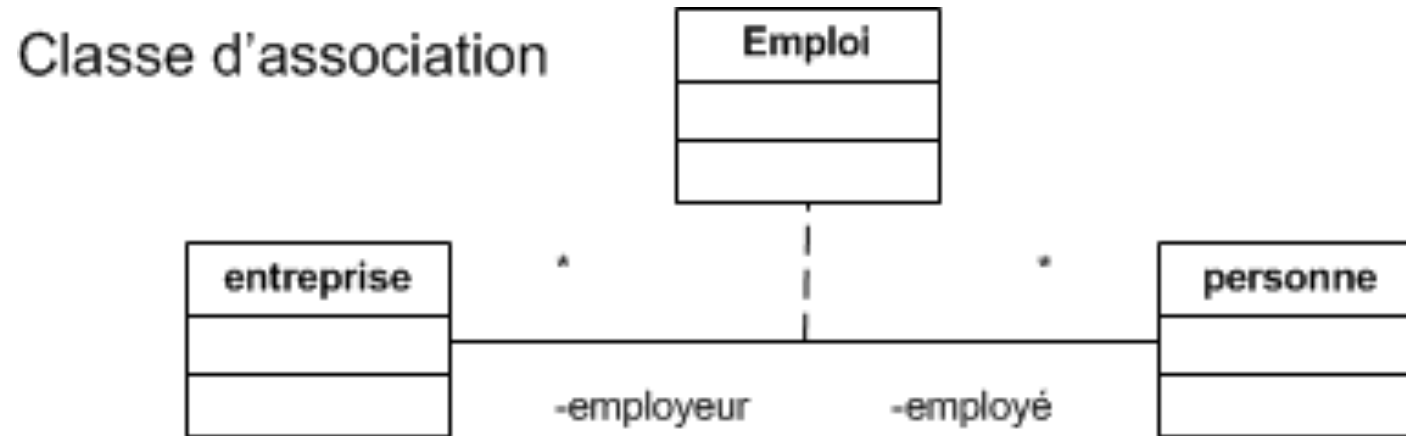
Un programme est composé de zéro à plusieurs cours ... un cours ne peut appartenir qu'à un seul programme! Le cours n'existe que si le programme existe. Si le programme est détruit les cours aussi. C'est donc un lien de composition

Attention, tout cela est un point de vue!

Classes d'association

- Une association peut apporter de nouvelles informations (attributs et méthodes) qui n'appartiennent à aucune des deux classes qu'elle relie et qui sont spécifiques à l'association.
- Ces nouvelles informations peuvent être représentées par une nouvelle classe attachée à l'association via un trait en pointillés.

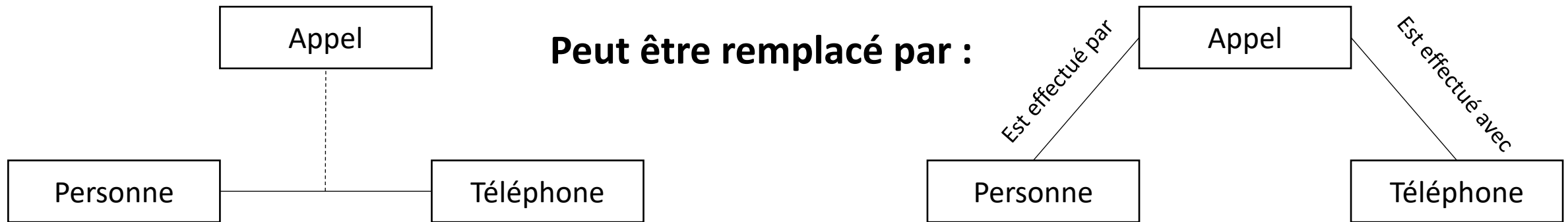
Classes d'association



Il existe une instance d'Emploi pour chaque lien entre objets des classes Entreprise et Personne

Classes d'association

- **Exemple:** Lorsqu'une personne utilise un téléphone, il faut pouvoir mesurer la durée de l'appel et savoir à quel moment il a lieu afin de le tarifier. Nous ajoutons donc deux attributs durée et période tarifaire qui n'appartiennent ni à la classe Personne ni à la classe Téléphone. Ces deux attributs sont mis dans une nouvelle classe (la classe Appel) qui est attachée à l'association.

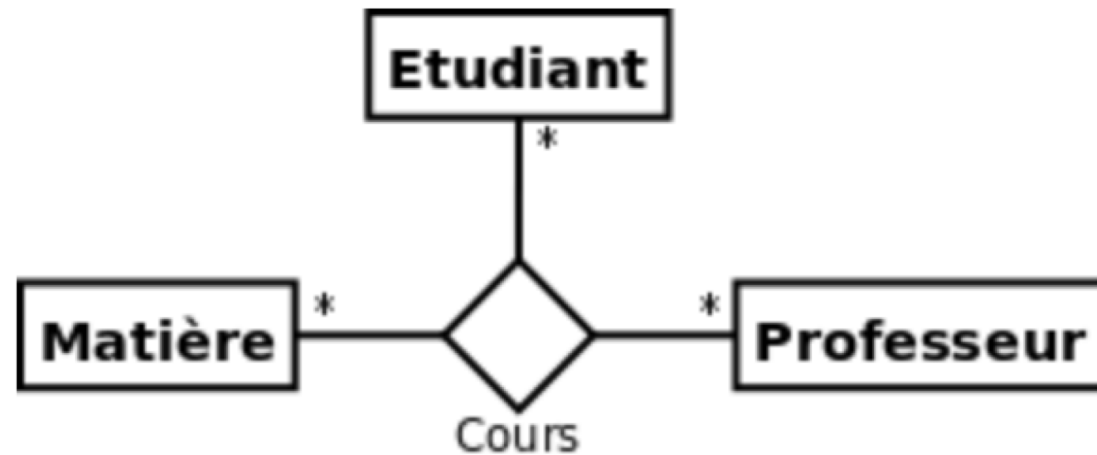


Association n-aire

- Une association **n-aire** lie plus de deux classes. La ligne pointillée d'une classe-association peut être reliée au losange par une ligne discontinue pour représenter une association **n-aire** dotée d'attributs, d'opérations ou d'associations.
- On représente une association n-aire par un grand losange avec un chemin partant vers chaque classe participante. Le nom de l'association, le cas échéant, apparaît à proximité du losange.
- Pas très utilisée !

Association n-aire

- **Exemple 1 :**



Association n-aire

- **Exemple 2** : Une séance de cinéma peut correspondre à l'association ternaire de 3 classes



Association n-aire

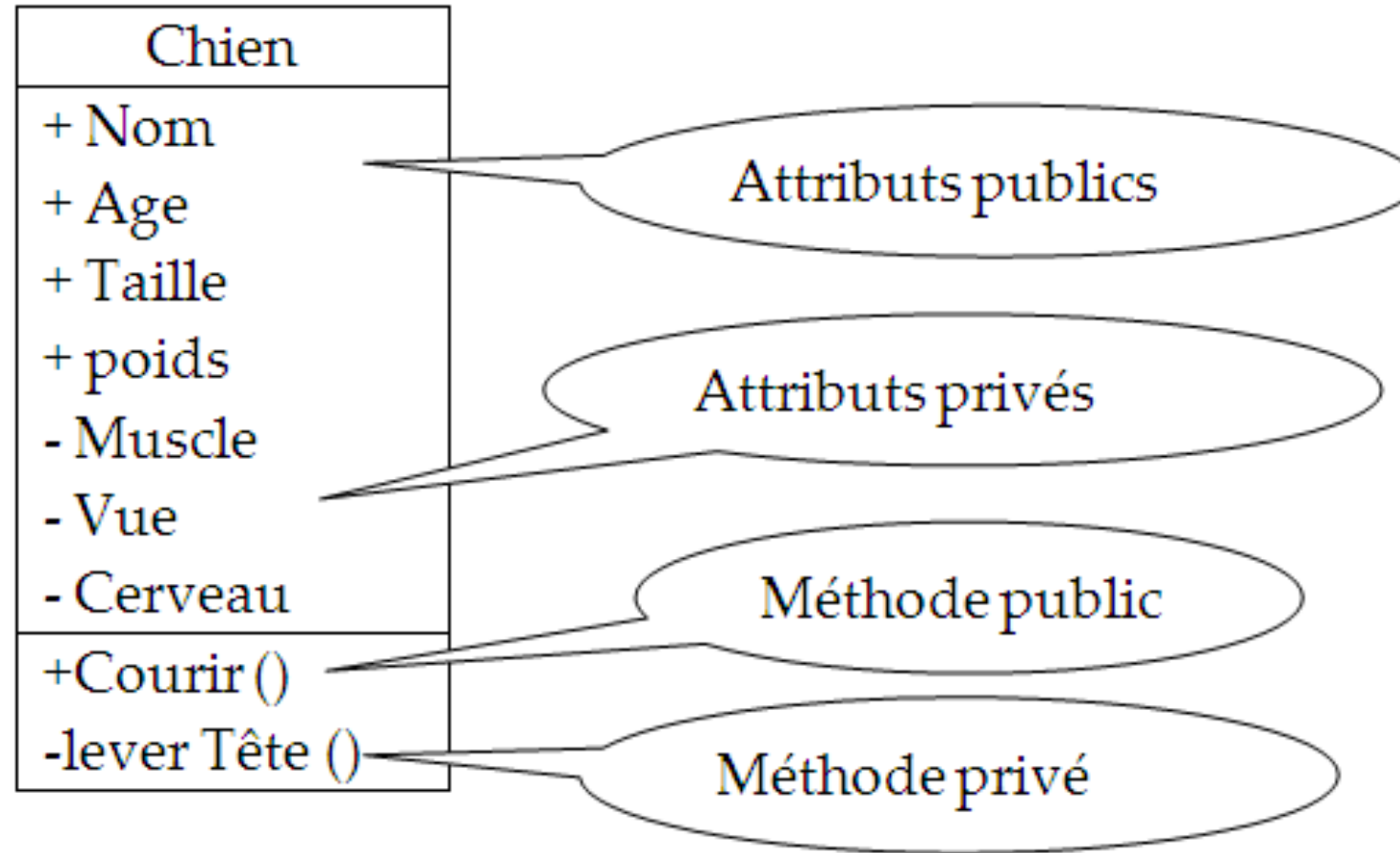
- **Exemple 3** : Un cours peut correspondre à l'association ternaire de 3 classes



Visibilités des attributs et opérations

- **Public +** : élément non encapsulé visible de tous. Accessible aux autres objets,
- **Protégé #** : élément encapsulé visible dans les sous-classes de la classe
- **Privé -** : élément encapsulé visible seulement dans la classe. Donc attributs ou méthodes utilisé(e)s pour le **traitement interne** de l'objet seulement.

Exemple



Exemple

Visio : Choisir la catégorie Opération
pour inscrire les méthodes

Chien
+attributPublic #attributProtégé -attributPrivé
+méthodePublic() #méthodeProtégée() -méthodePrivée()

Exercices d'application

- Pour chaque exemple ci-dessous, indiquez si la relation présentée est une généralisation, une agrégation ou une association :
 - Un pays a une capitale
 - Une transaction boursière est un achat ou une vente
 - Les fichiers contiennent des enregistrements
 - Une personne utilise un langage de programmation dans un projet
 - Les modems et les claviers sont des périphériques d'entrées/sorties

Exercices d'application

- Pour chaque situation ci-dessous, proposez un diagramme de classes.
 - Une librairie vend des livres, caractérisés par leur auteur et leur nombre de pages ; certains livres possèdent également d'autres caractéristiques : une fourchette des âges pour les livres pour enfants, et la discipline et le niveau pour les livres scolaires.
 - On considère une entreprise, et on suppose qu'un chef dirige plusieurs salariés (les subordonnés) et que le chef est lui-même un salarié.
 - On considère une université, et les personnes y travaillant qui peuvent être des étudiants ou des enseignants.

Exercices d'application

- Dans une société de transport, on voudrait gérer les bus de ramassage scolaire et les conducteurs. Un lycéen est un enfant, il est caractérisé par son nom, son âge et son sexe. Les informations qui caractérisent le conducteur sont les mêmes que pour le lycéen, avec en plus le numéro de son permis. Quant au bus, on a besoin de connaître son numéro d'immatriculation, sa date de mise en service, nombre d'années de service, et le poids total.
- Un bus est composé d'une carrosserie (poids, couleur), de 6 roues (pression, diamètre), de plusieurs sièges (couleur) pour passagers, plusieurs vitres (épaisseur, poids).
- Présentez le diagramme de classes.