

03/04/2023

Version 2

UML

JÉRÉMY PERROUAULT

A decorative wavy line in light blue and white, flowing from the top left towards the bottom left of the slide.

PRÉSENTATION

INTRODUCTION UML

UML

- **U**nified **M**odeling **L**anguage
- Modèle qui simplifie la réalité
- Permet de
 - Visualiser le système comme il est (ou comme il devrait être, idéalement)
 - De valider le modèle avec le client
 - De spécifier les structures (données) et le comportement (fonctionnalités)
 - Fournir un guide
 - Documenter le système

UML, CE LANGAGE

- **UML** n'est pas une méthode
- **UML** est un langage de modélisation objet
- **UML** a été adopté par toutes les méthodes objets
- **UML** est dans le domaine public
- **UML** est une norme

UML

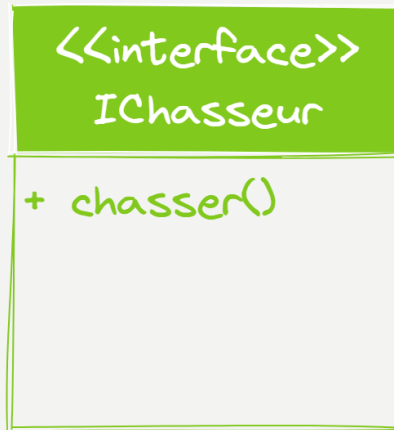
- 3 blocs de base pour construire
 - Les entités
 - Les relations
 - Les diagrammes
- Règles à observer pour utiliser ces blocs
 - Règles sémantiques
 - Règles de présentation
- Les mécanismes communs
 - Spécification
 - Présentation
 - Extension des modèles

LES ENTITÉS

classe



Interface



Collaboration



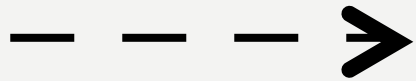
Cas d'utilisation



Acteur



LES RELATIONS



Dépendance (Dépendance entre entités)



Association (Relation entre entités)

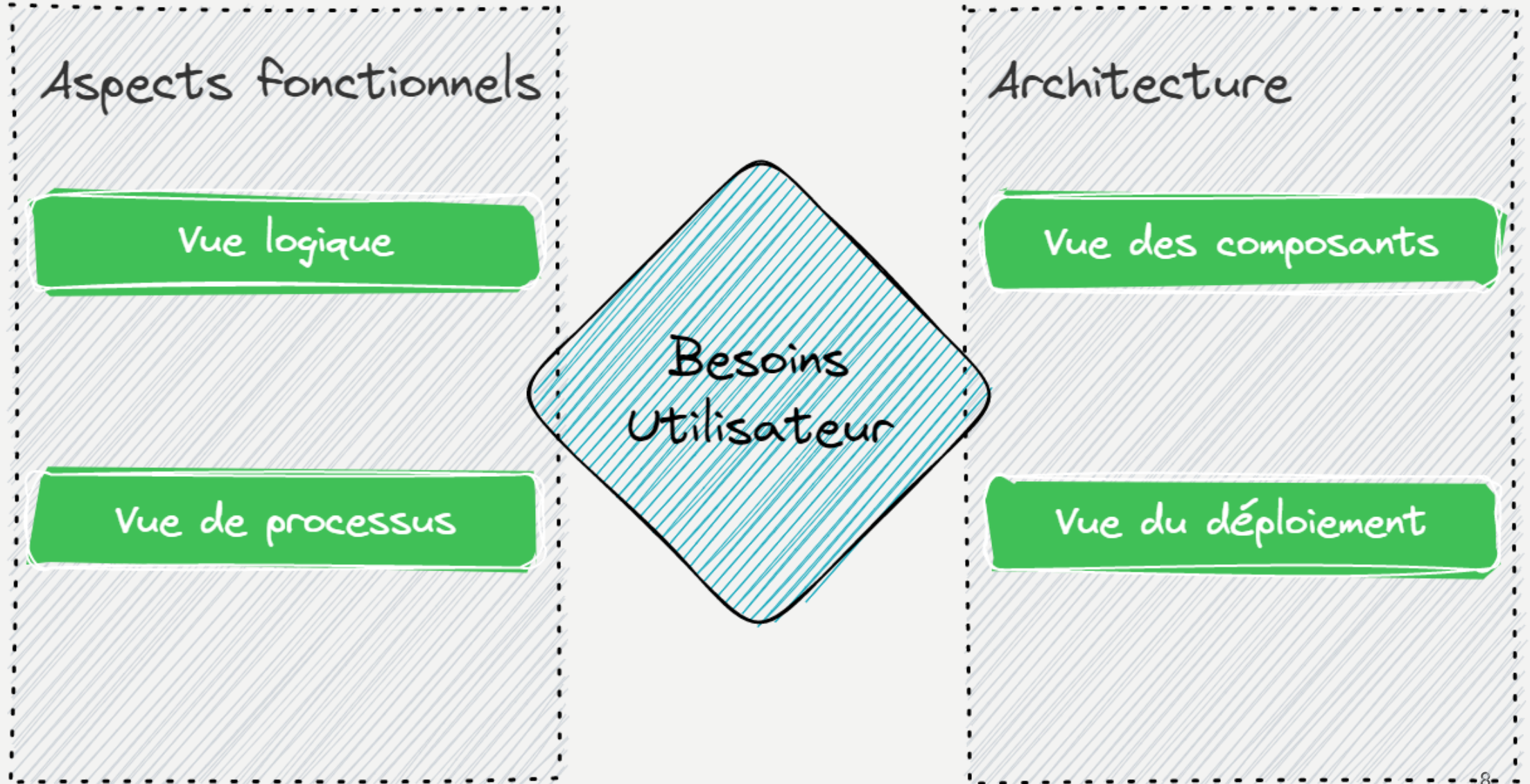


Héritage (Une classe hérite d'une classe mère)



Implémentation (Une classe implémente une interface)

LES 4+1 VUES D'UN SYSTÈME



LES DIAGRAMMES

- Diagrammes de structure (ou statiques)
 - Diagramme de classes
 - Représente les classes et interfaces
 - Diagramme d'objets
 - Représente les instances de classe (les objets)
 - Diagramme de composants
 - Représente les composants d'un point de vue physique (fichiers, bases de données)
 - Diagramme de déploiement
 - Représente des éléments matériels (périphérique, machine, réseau)
 - Diagramme de paquets
 - Regroupe des définitions
 - Diagramme de structure composite
 - Représentation sous forme de « boîte blanche » les relations entre composants d'une classe
 - Diagramme de profils
 - Spécialisation et personnalisation du domaine particulier d'un meta-modèle de référence

LES DIAGRAMMES

- Diagrammes de comportement
 - Diagramme de cas d'utilisations
 - Représente les possibilités d'interactions entre le système et les acteurs
 - Diagramme états-transitions
 - Représente sous forme de machine à états finis le comportement du système ou des composants
 - Diagramme d'activités
 - Représente sous forme de flux d'activités le comportement du système

LES DIAGRAMMES

- Diagrammes d'interaction (dynamiques)
 - Diagramme de séquence
 - Représente de façon séquentielle le déroulement des traitements et des interactions entre acteurs
 - Diagramme de communication
 - Représente de façon simplifiée un diagramme de séquence en se concentrant sur les messages échangés
 - Diagramme global d'interaction
 - Représente les enchaînements possibles entre les scénarios de séquence
 - Diagramme de temps
 - Représente les variations d'une données au cours du temps

LES DIAGRAMMES

Diagramme	Etape projet
Diagramme de cas d'utilisation	Spécifications, cahier des charges, cahier fonctionnel
Diagramme de séquences	
Diagramme d'activités	
Diagramme de classes	Conception architecturale, cahier technique
Diagramme d'objets	
Diagramme de communication	
Diagramme de déploiement	

LES DIAGRAMMES

Diagramme	Etape projet
Diagramme de packages	Besoins de l'utilisateur
Diagramme de cas d'utilisation	
Diagramme de classes	Vue logique
Diagramme d'objets	
Diagramme de séquences	Vue des processus
Diagramme d'activités	
Diagramme de collaboration	
Diagramme d'états-transitions	
Diagramme global d'interaction	
Diagramme de temps	
Diagramme de structure composite	Vue des composants
Diagramme de composants	
Diagramme de déploiement	Vue de déploiement



CLASSES

LES DIAGRAMMES DE CLASSE

DIAGRAMME DE CLASSES

- Une classe permet de représenter quelque chose
 - Définition de ses attributs, des différentes informations, données
 - Définition de ses actions, ses fonctionnalités, son comportement
- Exemple :
 - Une personne a un nom, un prénom, un age, et peut marcher
 - Ses attributs sont nom, prenom, age
 - Ses comportements sont marcher()
 - Une réunion a un nom, une date, un lieu, une liste des personnes, et peut être démarrée ou annulée
 - Ses attributs sont nom, date, lieu, listePersonnes
 - Ses actions sont demarrer() et annuler()

DIAGRAMME DE CLASSES

- Chaque attribut et chaque comportement d'une **classe** a une *visibilité*, une *portée*
 - Elle peut être publique notée +
 - Elle peut être privée notée -
 - Elle peut être protégée notée #
 - Dans ce cas, l'information est visible uniquement par les **classes** de la même hiérarchie
- Chaque attribut est d'un type
- Chaque comportement peut (éventuellement) retourner un type
 - On note ces types par : type

DIAGRAMME DE CLASSES

+ publique
- privé
protégé

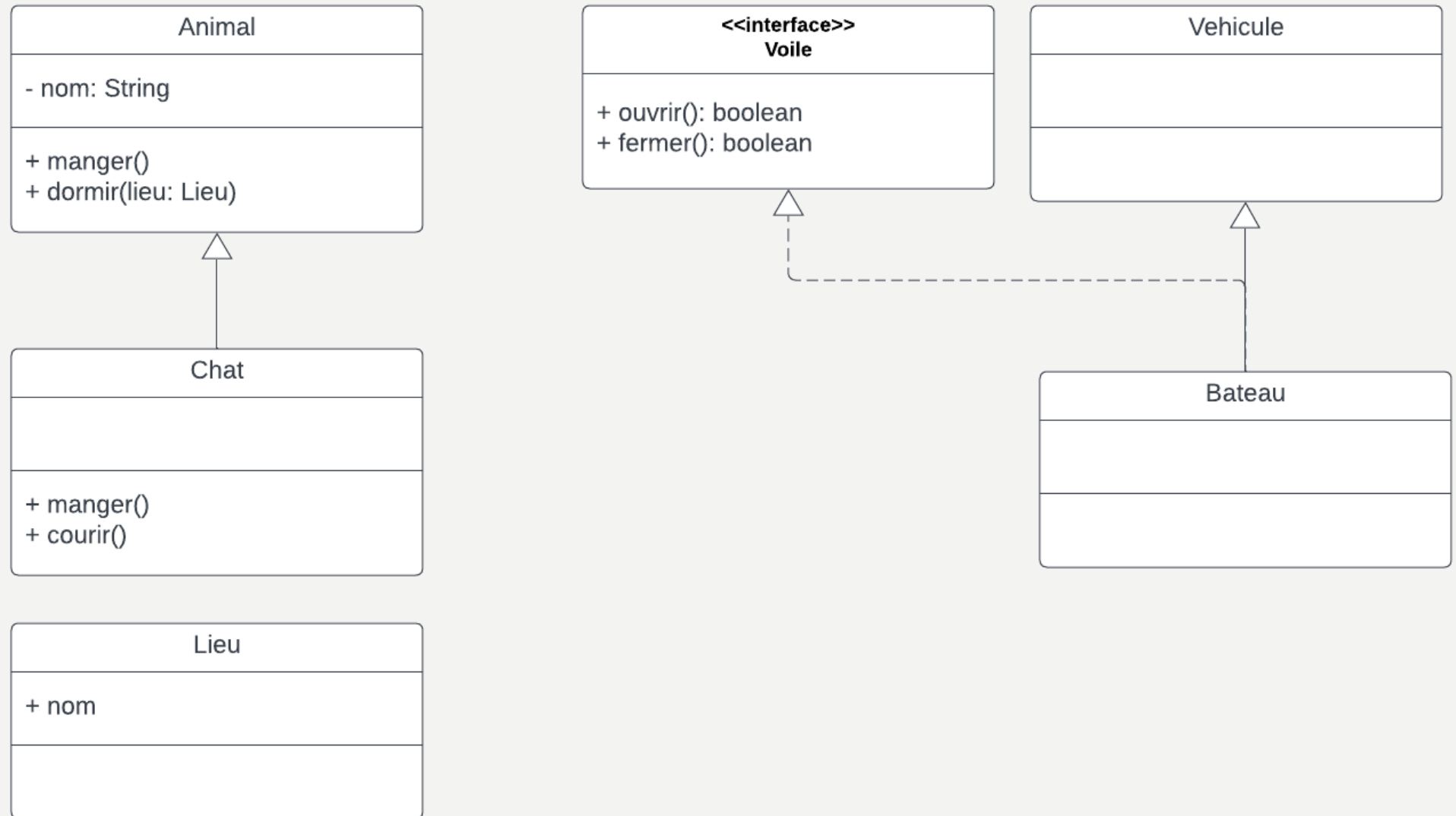
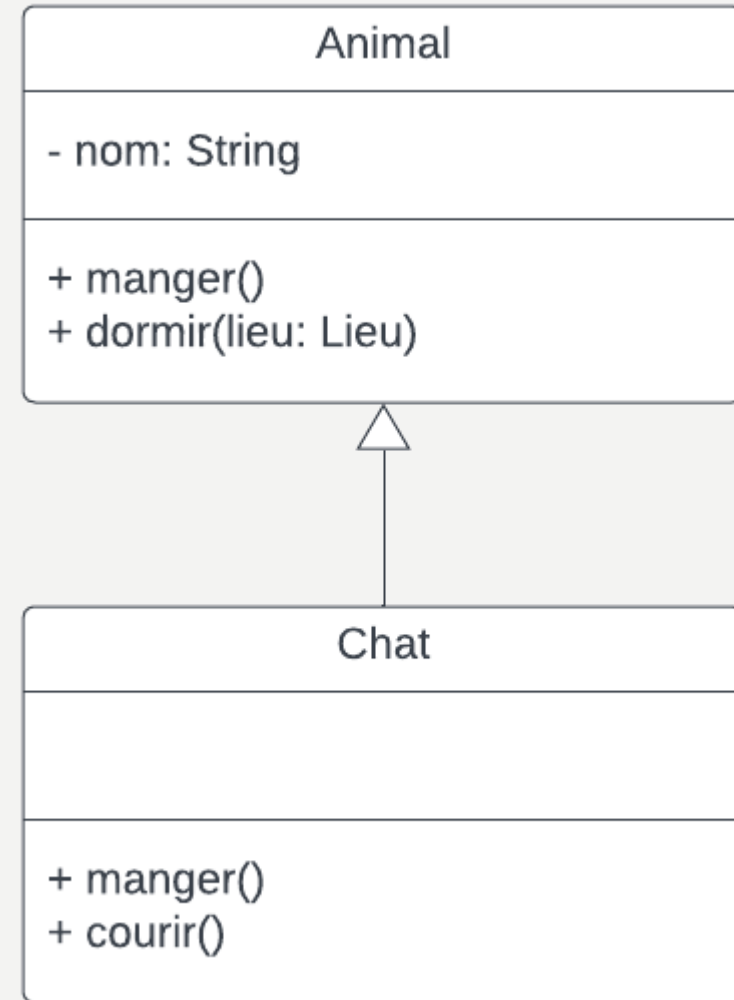


DIAGRAMME DE CLASSES

- Un héritage entre une classe mère et une classe fille



EXERCICE

- Modéliser les classes `Animal` et `Chat` dans un diagramme de classe
 - `Chat` est un `Animal`
 - Un animal a un `nom` et un `age`, et il peut `manger`
 - Un chat peut `griffer`

DIAGRAMME DE CLASSES

- *Agrégation*
 - Relation qui indique que la **classe** A a un attribut de la **classe** B
 - Schématisé par un losange vide
- *Composition*
 - Agrégation forte
 - La **classe** A ne peut pas exister sans la **classe** B
 - Schématisé par un losange plein

DIAGRAMME DE CLASSES

- La Personne a une relation agrégée avec le Chat
 - Elle peut exister sans le Chat, avoir plusieurs Chats
- Le Chassis a une relation de composition avec la Voiture
 - Le Chassis est un élément indissociable de la Voiture
 - La voiture a besoin du chassis
 - Si le chassis disparaît, la voiture aussi

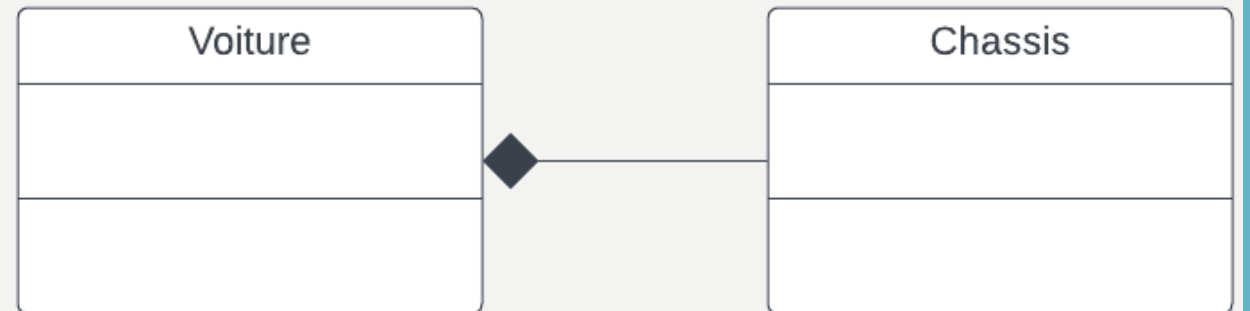
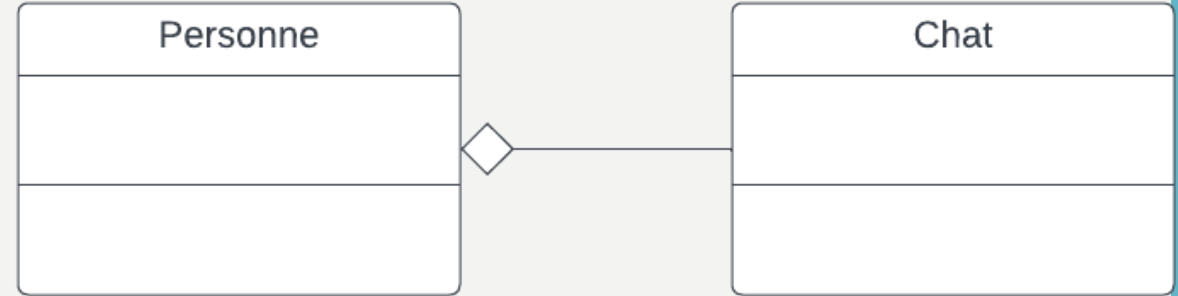


DIAGRAMME DE CLASSES

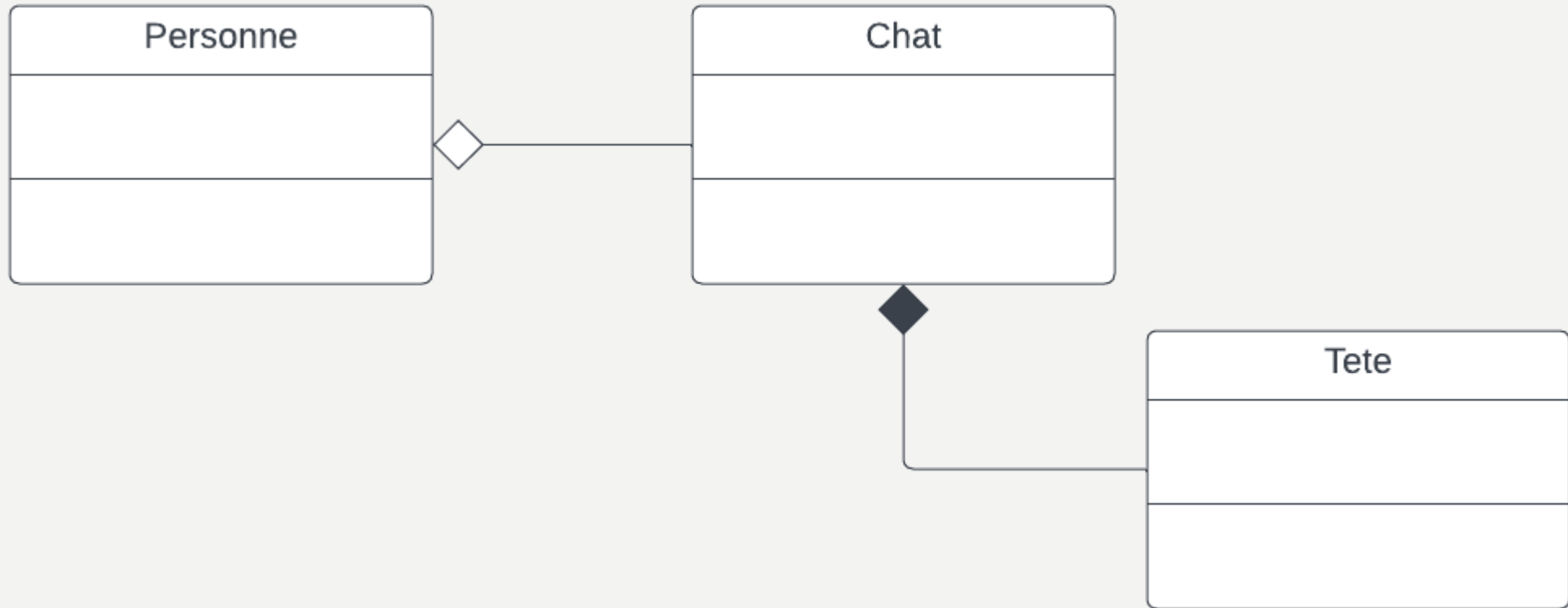


DIAGRAMME DE CLASSES

- Connexion entre 2 **classes**
- Distinguer association classique avec ou sans sens de lecture



DIAGRAMME DE CLASSES

- Préciser le rôle d'une **classe** vis-à-vis de l'autre
 - Pas besoin de le faire lorsque le nom est trivial pour le pas alourdir le schéma inutilement

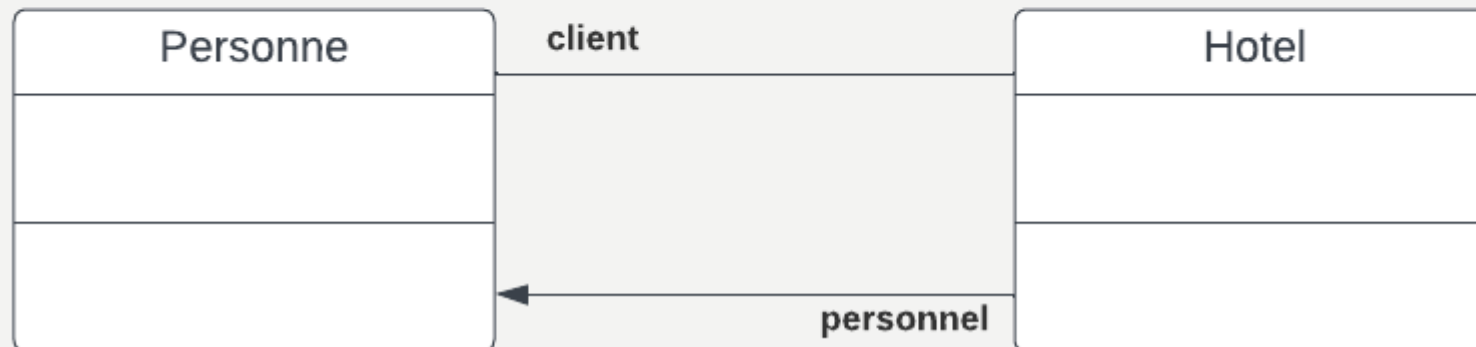
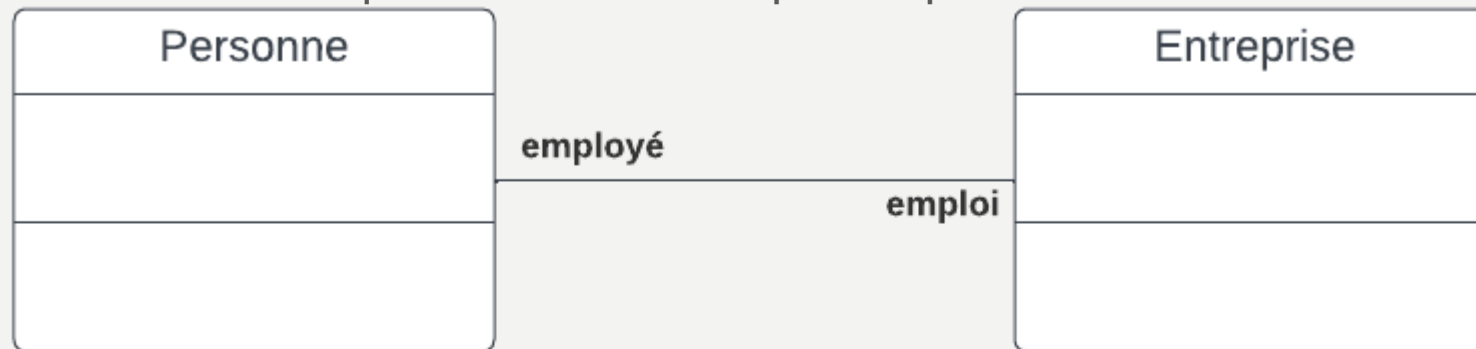


DIAGRAMME DE CLASSES

- Pour représenter un nombre d'éléments participant à la relation
 - On utilise les **cardinalités**
- x exactement x (entier > 0)
 - 7, 11
- $x..y$ de x à y (entier $y \geq x$)
 - 1..1, 1..10, 0..n
- $*$ (ou n) de 0 à plusieurs
 - 0..*, 0..n
- $x..*$ (ou $x..n$) de x à plusieurs ($x \geq 0$)
 - 0..*, 5..n, 10..*

DIAGRAMME DE CLASSES

- Un hôtel héberge plusieurs personnes (0 à plusieurs)
- Une personne est hébergée dans un seul hôtel (à la fois)

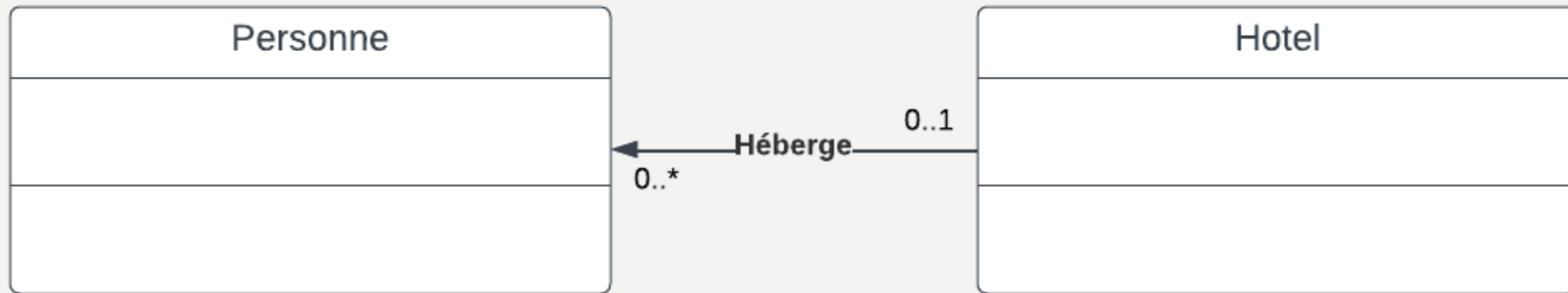
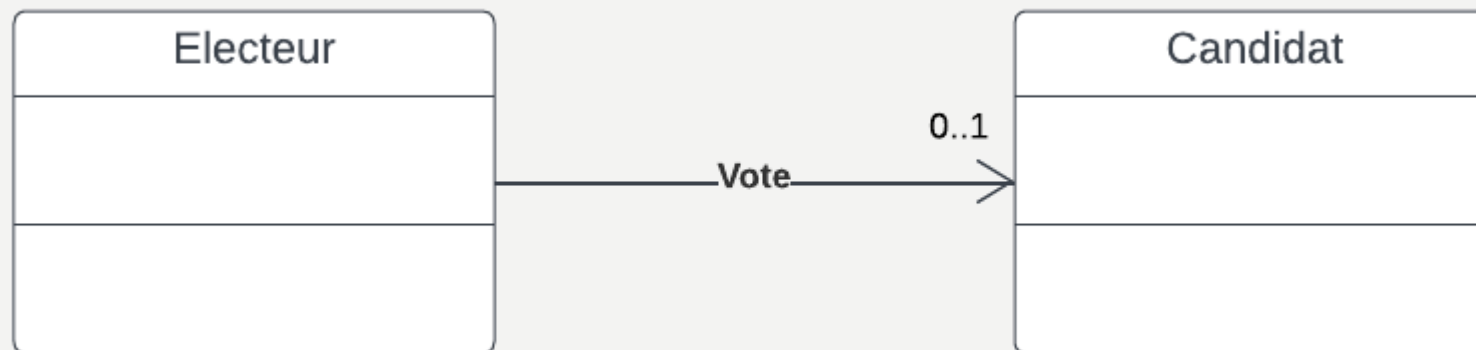


DIAGRAMME DE CLASSES

- Par défaut, une association est navigable dans les deux sens
- La réduction à une direction « unidirectionnelle » indique que les instances d'une classe ne connaissent pas les instances d'une autre
- Un électeur connaît le candidat, pas l'inverse n'est pas vrai
 - Donc, un candidat n'aura pas d'attribut électeur ou liste d'électeurs
 - Mais un électeur aura un attribut candidat



EXERCICE