



CHAPITRE II: **Modélisation avec UML**



Dr S.Fellah

Modélisation avec UML

- Modélisation avec UML (historique, UML et OMG)

Modélisation avec UML (historique, UML et OMG)

- ❖ Aujourd'hui, le **standard** de **modélisation objet** est **UML** (Unified Modeling Language). Il est sous l'entière responsabilité de l'OMG (**O**bject **M**anagement **G**roup).
- ❖ UML se définit comme un langage de **modélisation graphique** destiné à :
 - ❖ **comprendre** et **décrire** des **besoins**,
 - ❖ **spécifier** et **documenter** des systèmes,
 - ❖ **tracer** des architectures logicielles,
 - ❖ **concevoir** des **solutions** et **communiquer** des points de vue.
- ❖ UML est une notation qui permet de **modéliser** un problème de **façon standard**.
- ❖ Ce langage est né de la fusion de **plusieurs méthodes existantes** auparavant, et il est devenu une **référence** en terme de **modélisation** objet, et sa connaissance devienne **indispensable** pour un développeur.
- ❖ UML **unifie** à la fois les **notations** et les **concepts**.
- ❖ Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une **sémantique précise**.

Modélisation avec UML (historique, UML et OMG)

- ❖ L'OMG (Object Management Group) est un groupement d'industriels dont l'objectif est la **standardisation** autour des **technologies objet**.
- ❖ L'OMG comprend actuellement plus de 800 membres, dont les principaux acteurs de l'industrie informatique (Sun, IBM, Microsoft, etc.), mais aussi les plus grandes entreprises utilisatrices dans tous les secteurs d'activité.

Modélisation avec UML

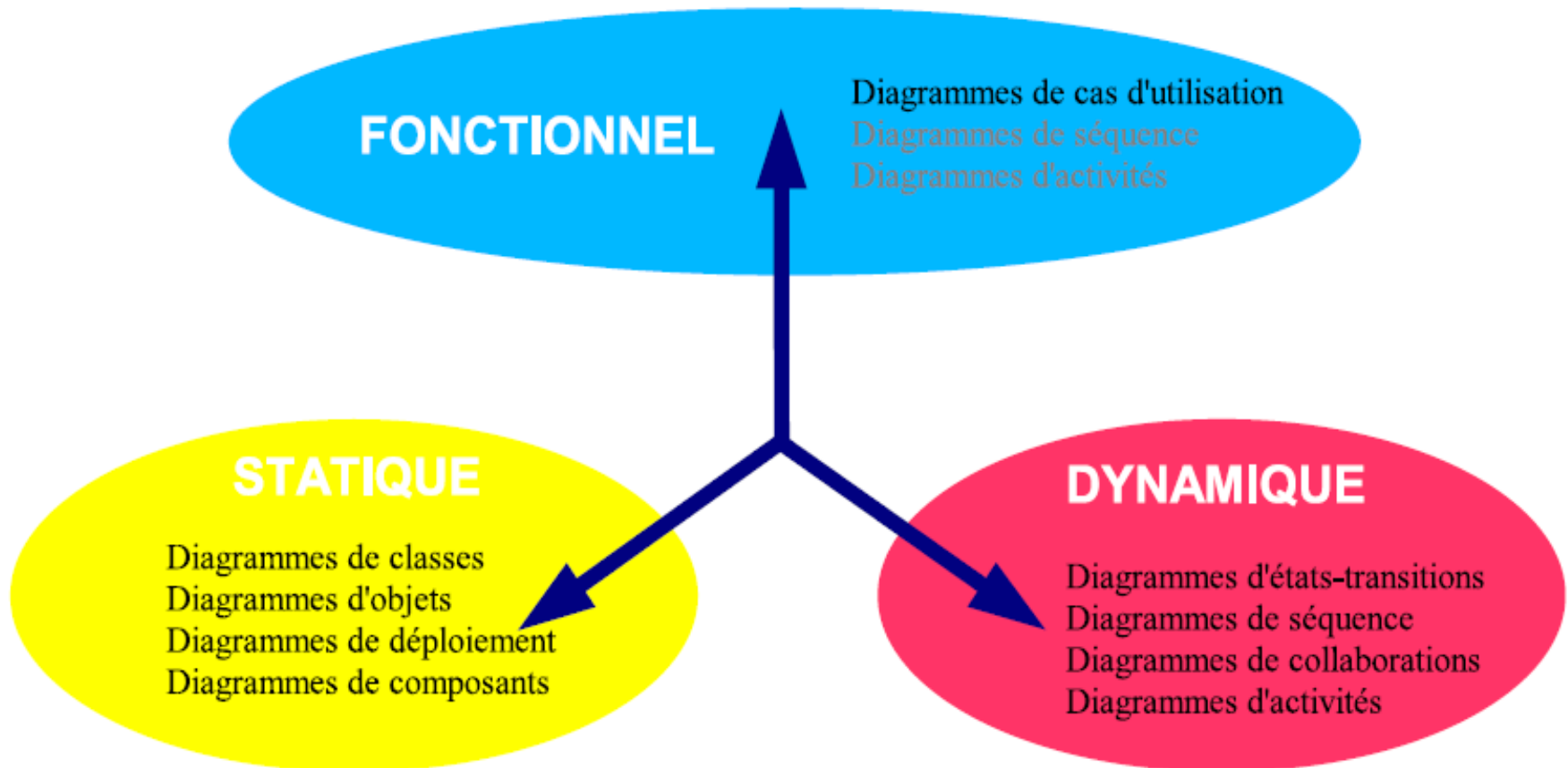
- UML est une notation, pas une méthode
- UML est un langage de modélisation objet
- UML convient pour toutes les méthodes objet
- UML est dans le domaine public

Modélisation avec UML

Les Démarches de modélisation

- UML ne définit pas le processus d'élaboration des modèles
 - Mais suggère 3 démarches :
1. **Itérative et incrémentale** : L'idée est de concevoir un prototype et de l'améliorer.
 2. **Besoins utilisateurs** : Dans ce cas ce sont les **utilisateurs qui guident la réalisation** du modèle. **Validation** des différents livrables du modèle par les utilisateurs.
 3. **Centré sur l'architecture** : en **utilisant les différentes vues d'UML**

Modélisation avec UML



Les diagrammes structurels (ou statiques)

Diagramme de classe
Diagramme d'objet

Diagramme de classe

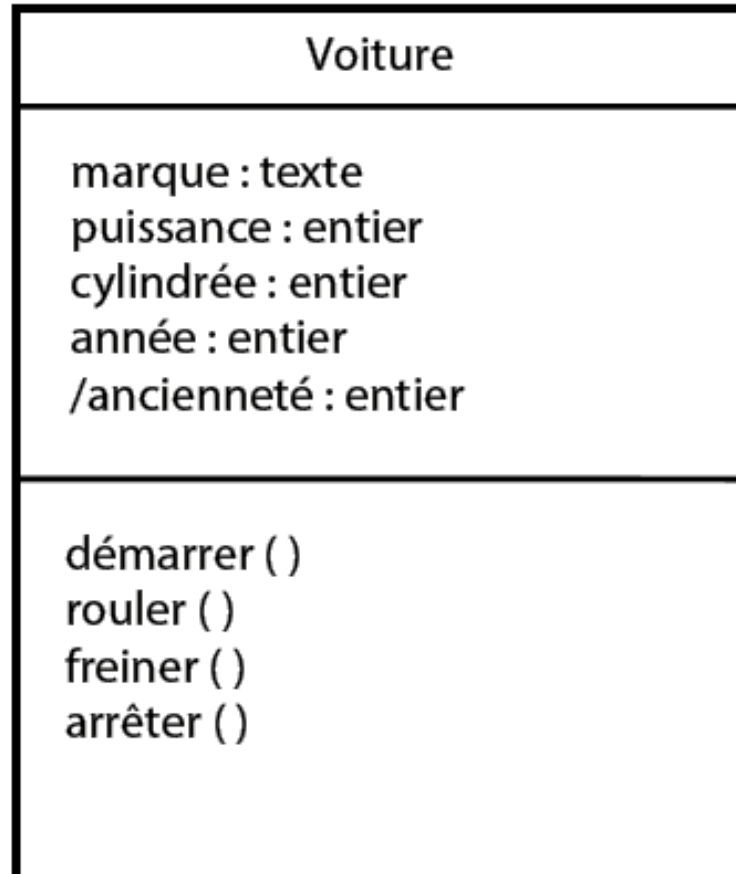
Diagramme de classe

Définition

- ❖ Le diagramme donne une représentation **statique** du système à développer.
- ❖ Cette représentation est **centrée** sur les **concepts** de **classe** et **d'association**.
- ❖ Le diagramme de classes est le **point central** dans un **développement orienté objet**.
 - En **analyse**, il a pour objectif de **décrire** la **structure** des **entités manipulées** par les utilisateurs.
 - En **conception**, le diagramme de classes **représente** la **structure d'un code orienté objet**.

Diagramme de classe

Représentation d'une classe



Représentation d'une classe

Diagramme de classe

Représentation d'un objet

Le nom d'un objet peut être désigné sous trois formes :

nom de l'objet : désignation directe et **explicite** d'un objet ;

nom de l'objet : nom de la classe : désignation **incluant** le nom de la **classe** ;

: nom de la classe : désignation **anonyme** d'un objet d'une **classe donnée**.

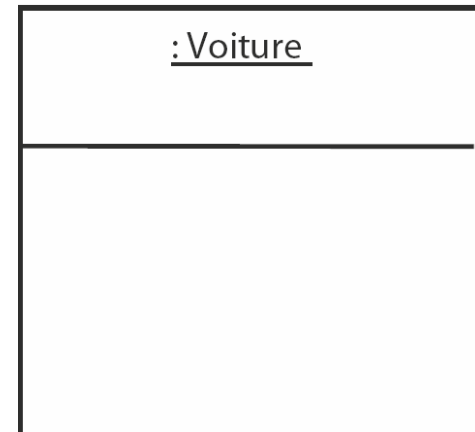
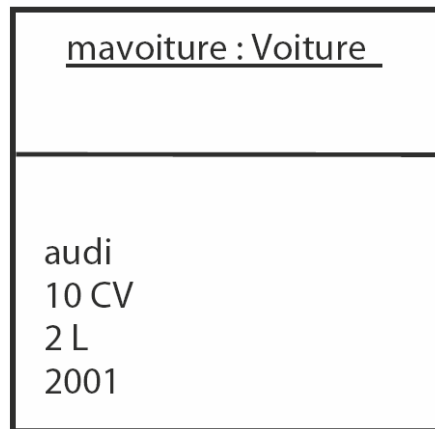
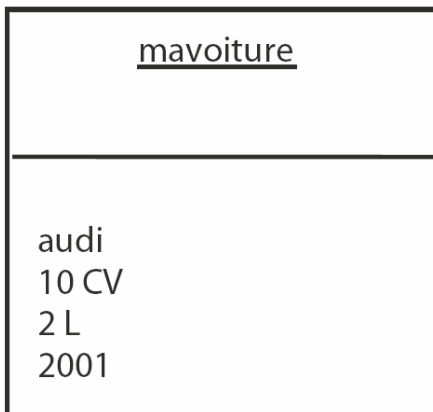
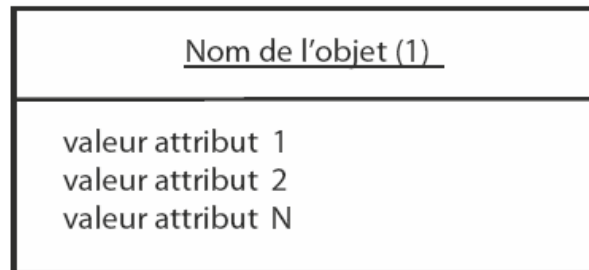


Diagramme de classe

Visibilité des attributs et opérations

- Chaque attribut ou opération d'une classe peut être de type **public**, **protégé**, **privé** ou **paquetage**.
- Les symboles +, #, - et ~ sont indiqués devant chaque attribut ou opération pour signifier le type de **visibilité** autorisé pour les **autres classes**.

Les droits associés à chaque niveau de confidentialité sont :

- **Public (+)** – Attribut ou opération **visible par tous**.
- **Protégé (#)** – Attribut ou opération **visible** seulement à **l'intérieur** de la **classe** et pour toutes les **sous-classes** de la classe.
- **Privé (-)** – Attribut ou opération seulement visible à **l'intérieur** de la **classe**.
- **Paquetage (~)** – Attribut ou opération ou classe seulement visible à **l'intérieur** du **paquetage** où se trouve la classe.

Diagramme de classe

Visibilité des attributs et opérations

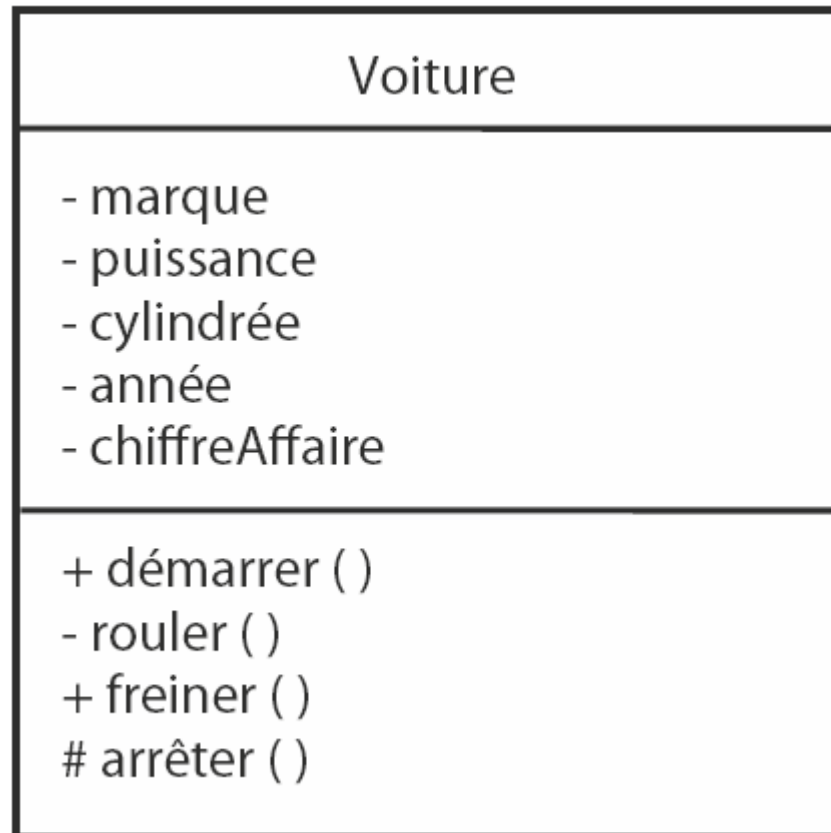


Diagramme de classe

Attributs et opérations (Attributs et méthode de classe)

- Un attribut de classe (**static**) garde une **valeur unique** et **partagée** par toutes les instances de la classe. C'est **une propriété de la classe**.
- L'**accès** à une **méthode de classe** ne nécessite pas l'**existence** d'une **instance** de cette classe.
- L'attribut « ristourne » est de type **classe** et l'opération « créer » est une opération exécutable au niveau de la classe.

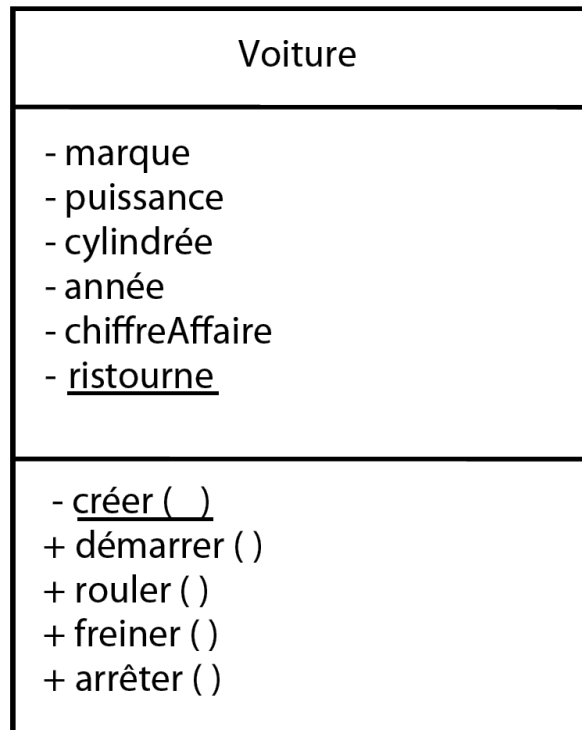


Diagramme de classe

Association

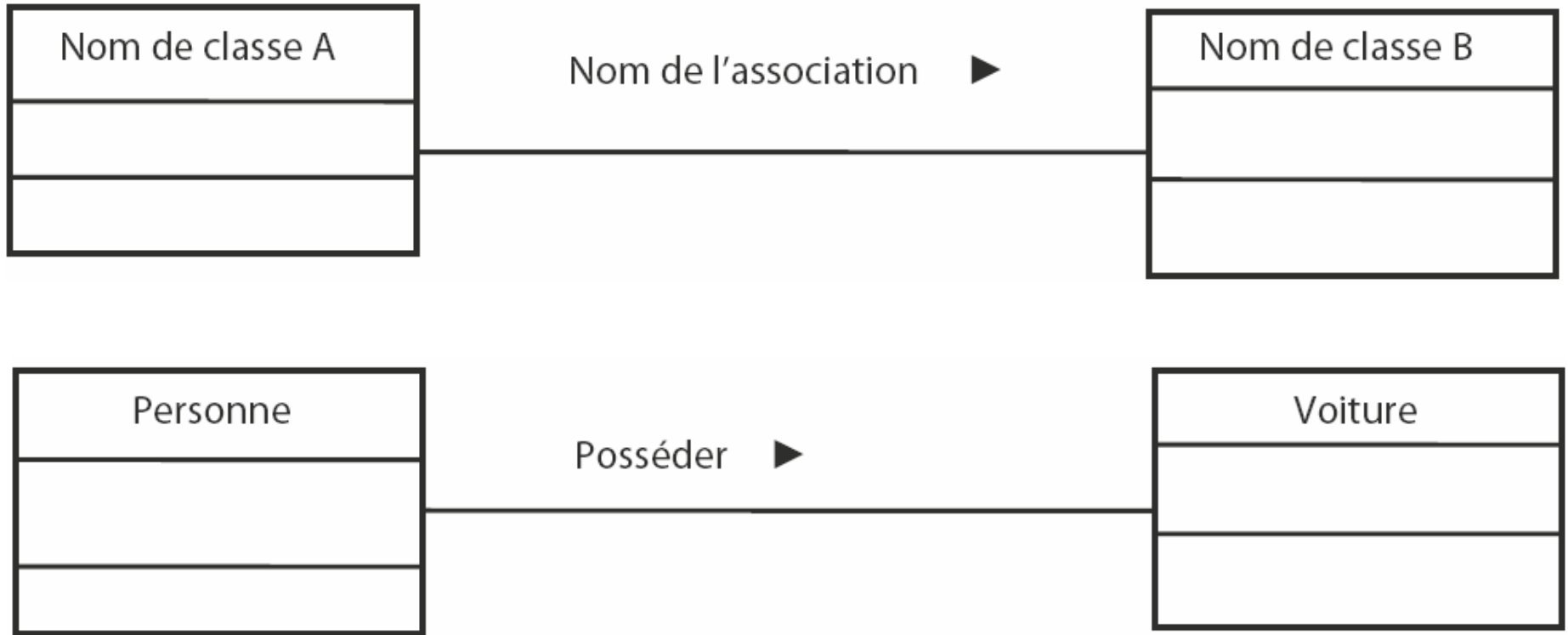


Diagramme de classe

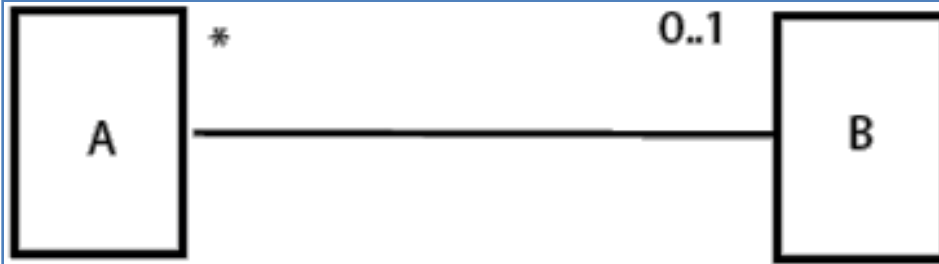
Rôle d'association

Le **rôle** tenu par une classe vis-à-vis d'une association peut être précisé sur l'association.

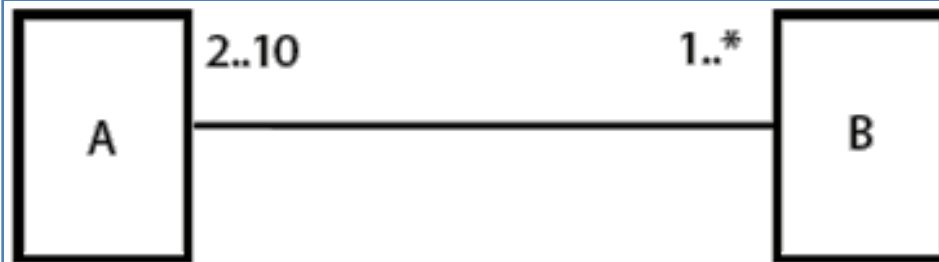


Diagramme de classe

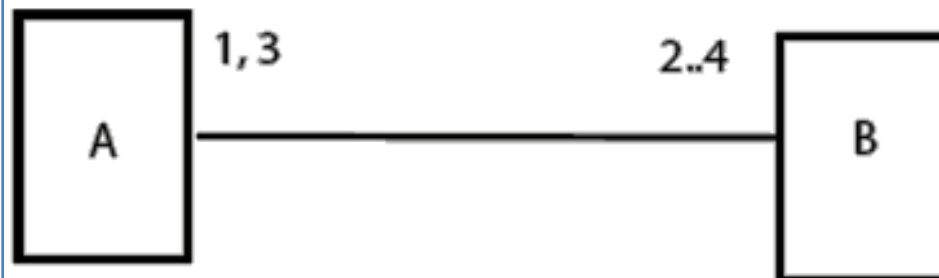
Association (Multiplicité)



- À une instance de A correspond 0 ou 1 instance de B.
- À une instance de B correspond 0 à nombre non déterminé d'instances de A.



- À une instance de A correspond 1 à un nombre non déterminé d'instances de B.
- À une instance de B correspond 2 à 10 instances de A.



- À une instance de A correspond 2 à 4 instances de B.
- À une instance de B correspond 1 ou 3 instances de A.

Diagramme de classe

Association (Multiplicité)

0..1	0 à une instance
1	une instance exactement
*	0 ou plusieurs instances
1..*	une ou plusieurs instances

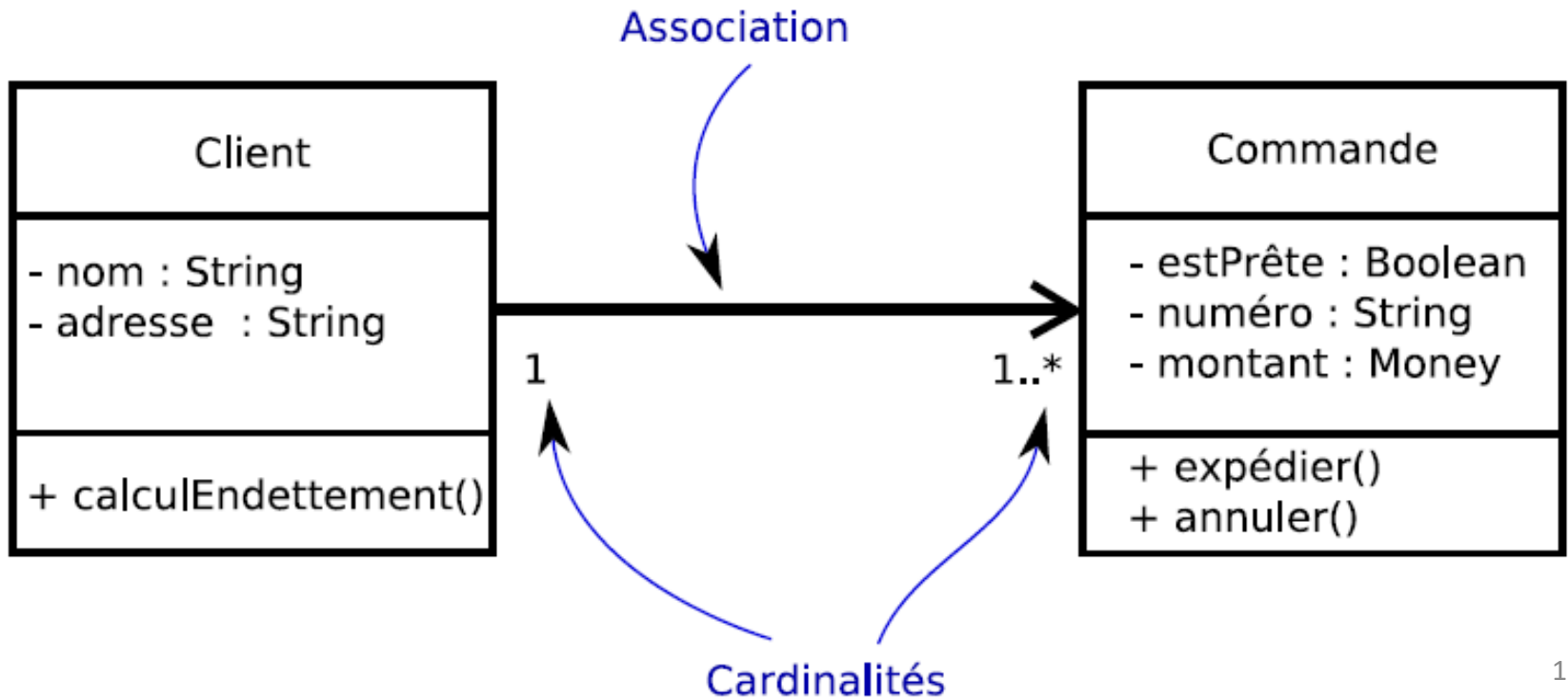
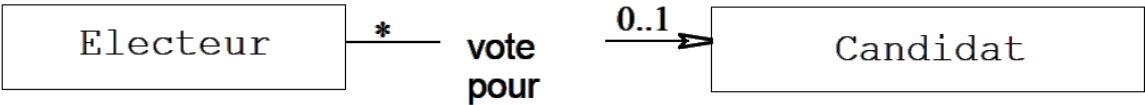


Diagramme classe

Navigabilité

- Rendre une relation unidirectionnelle.
- Permet d'indiquer que les instances d'une classe ne « connaissent » pas les instances d'une autre.



Représentations équivalentes

Diagramme de classe

Association qualifiée

- La qualification d'une relation entre deux classes permet de préciser la sémantique de l'association et de qualifier de manière restrictive les liens entre les instances.
- Seules les instances possédant l'attribut indiqué dans la qualification sont concernées par l'association

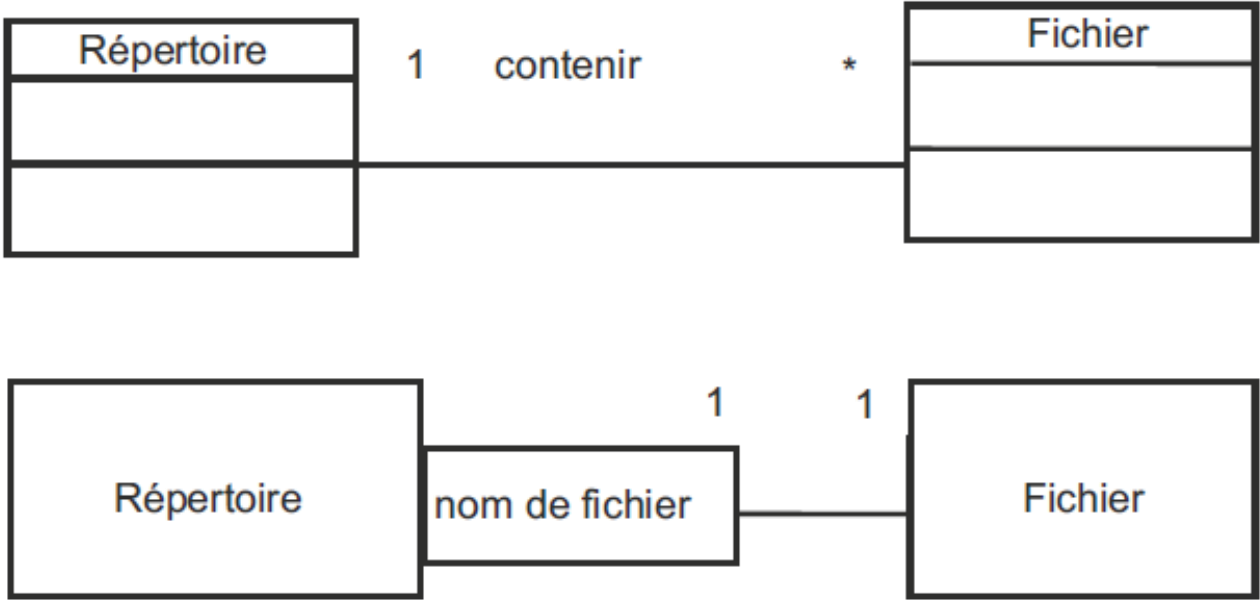


Diagramme de classe

Association (*Dépendance*)

- La **dépendance** exprime le fait qu'une classe **utilise des services** d'une **autre classe**.
- Tout changement dans la classe utilisée peut avoir un impact sur la classe qui l'utilise.



- Comme une relation de dépendance peut représenter plusieurs types de relation, les **stéréotypes** sont utilisés pour indiquer la nature exacte de la dépendance.

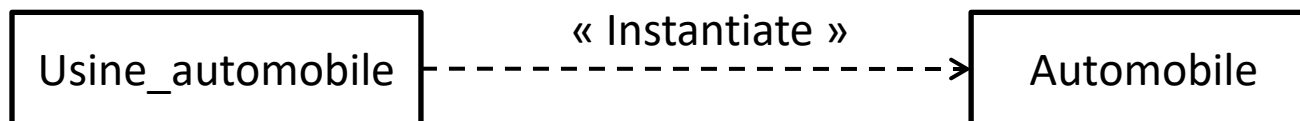
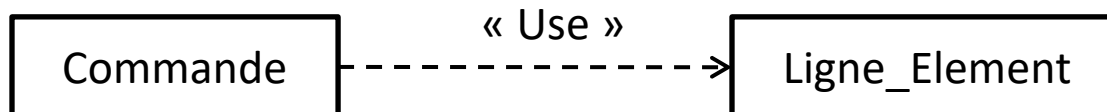


Diagramme de classe

Association de dimension supérieure à 2 et classe-association

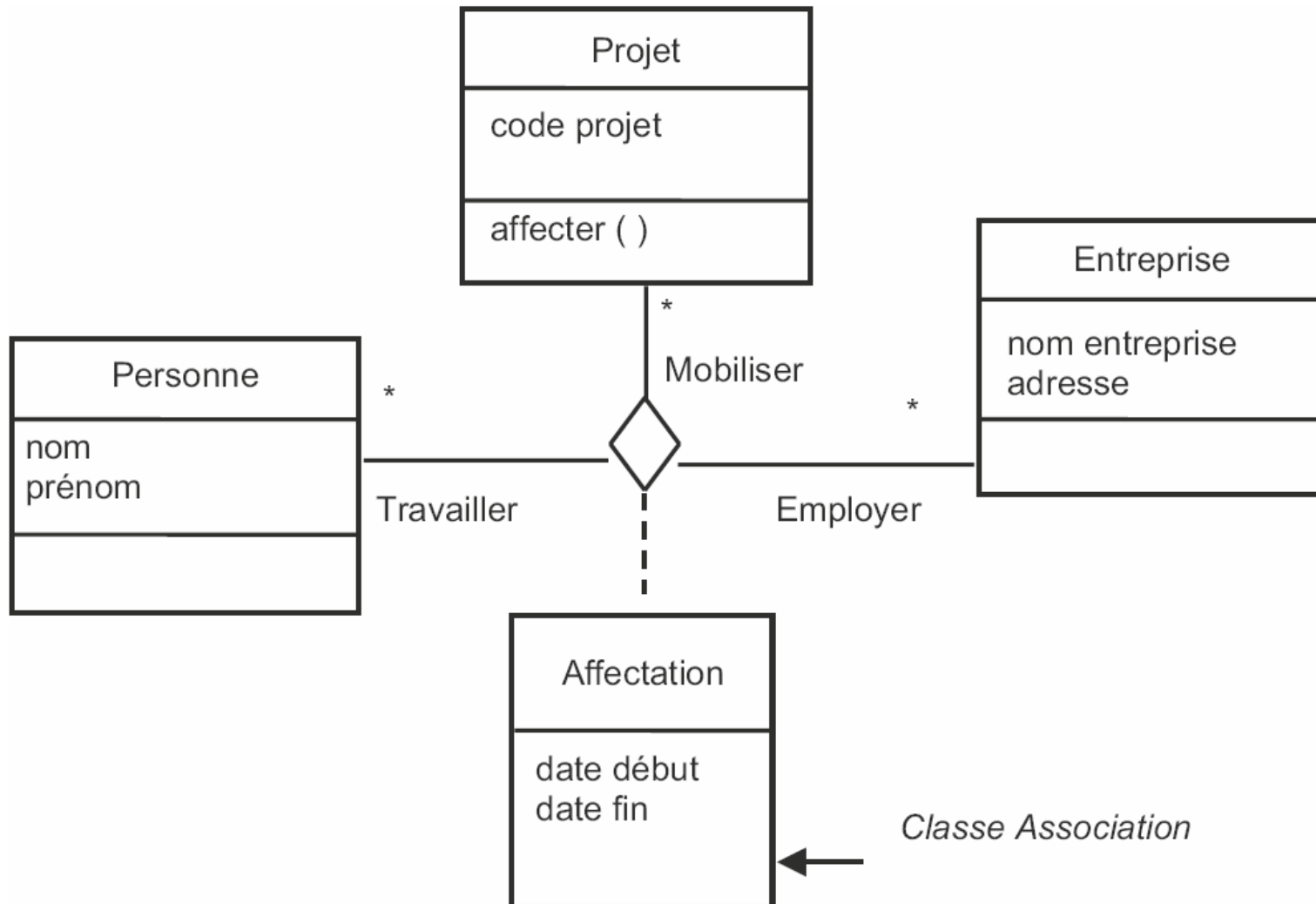


Diagramme de classe

Agrégation

L' **Agrégation** entre classes est une relation **bidirectionnelle**, qui exprime un couplage plus fort que la simple association :

L'agrégation permet d'exprimer des relations de type **maître-esclave, ou contenant-contenu,**

Diagramme de classe

Agrégation

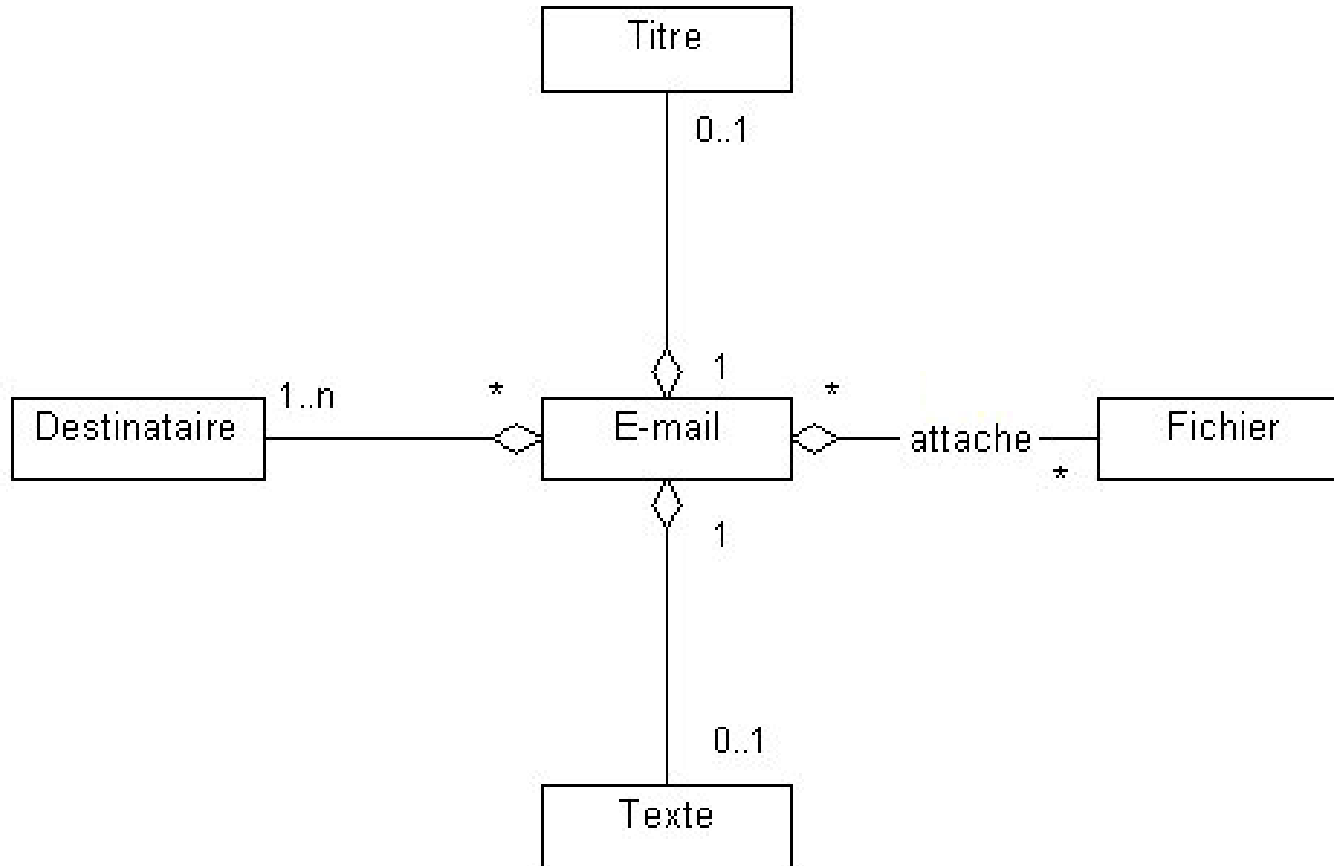


Diagramme de classe

Composition

- La **composition** est une **Agrégation forte**
- Les **éléments agrégés** ne peuvent **pas** être **partagés**
- Le **cycle de vie** des éléments et de l'agrégat sont **liés**.

 Les pages sont **physiquement contenues** dans le livre

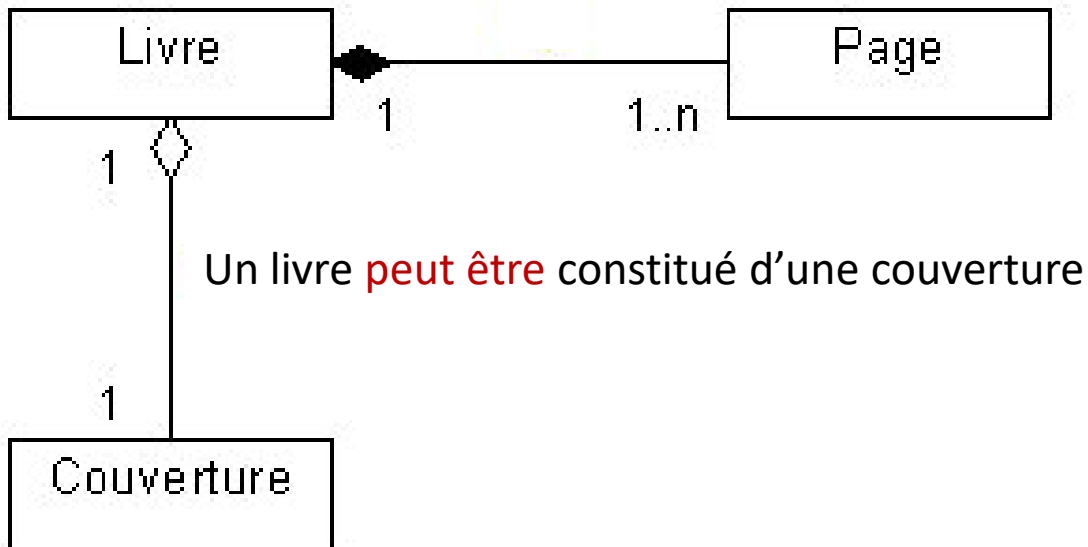


Diagramme de classe

Héritage

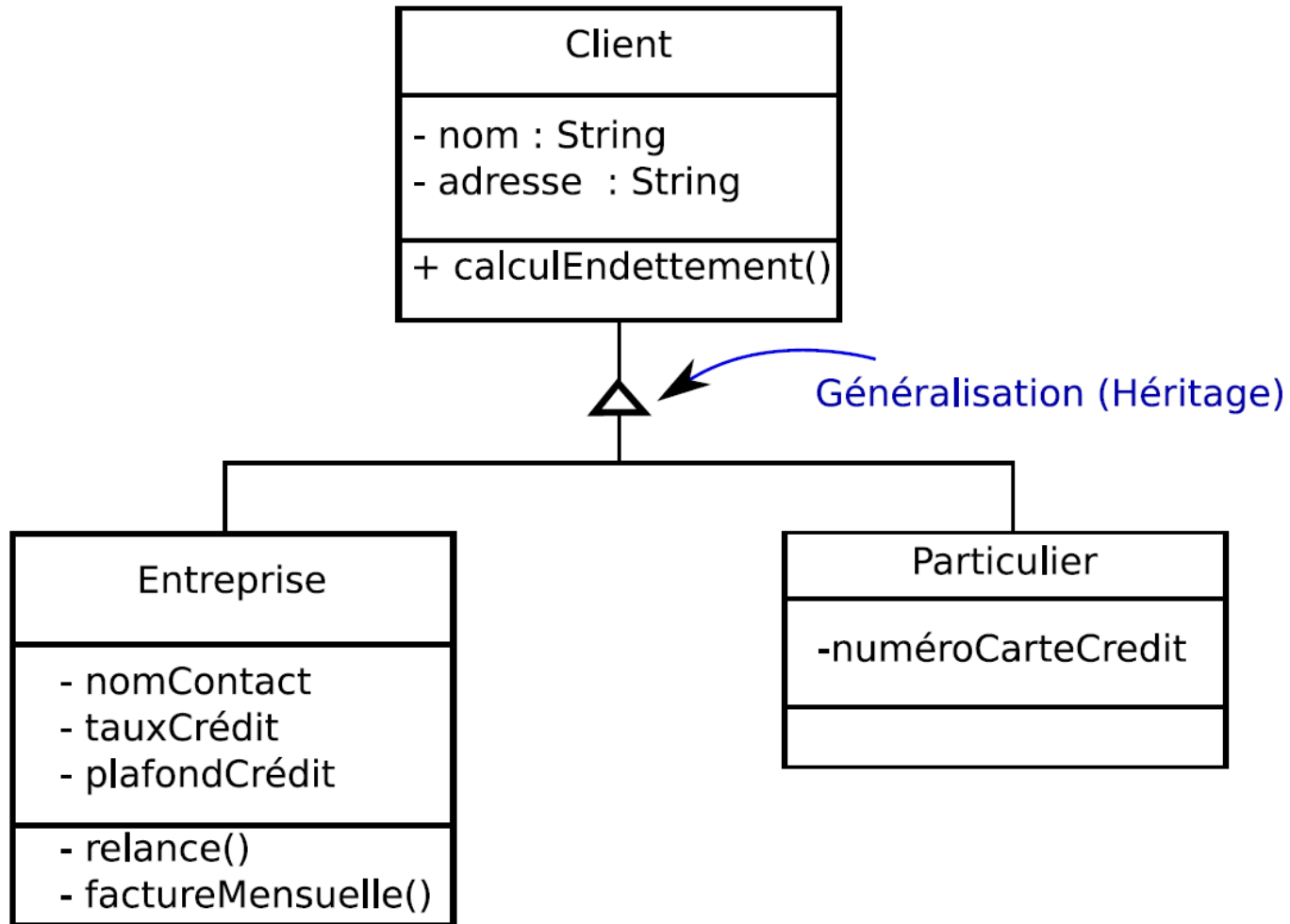


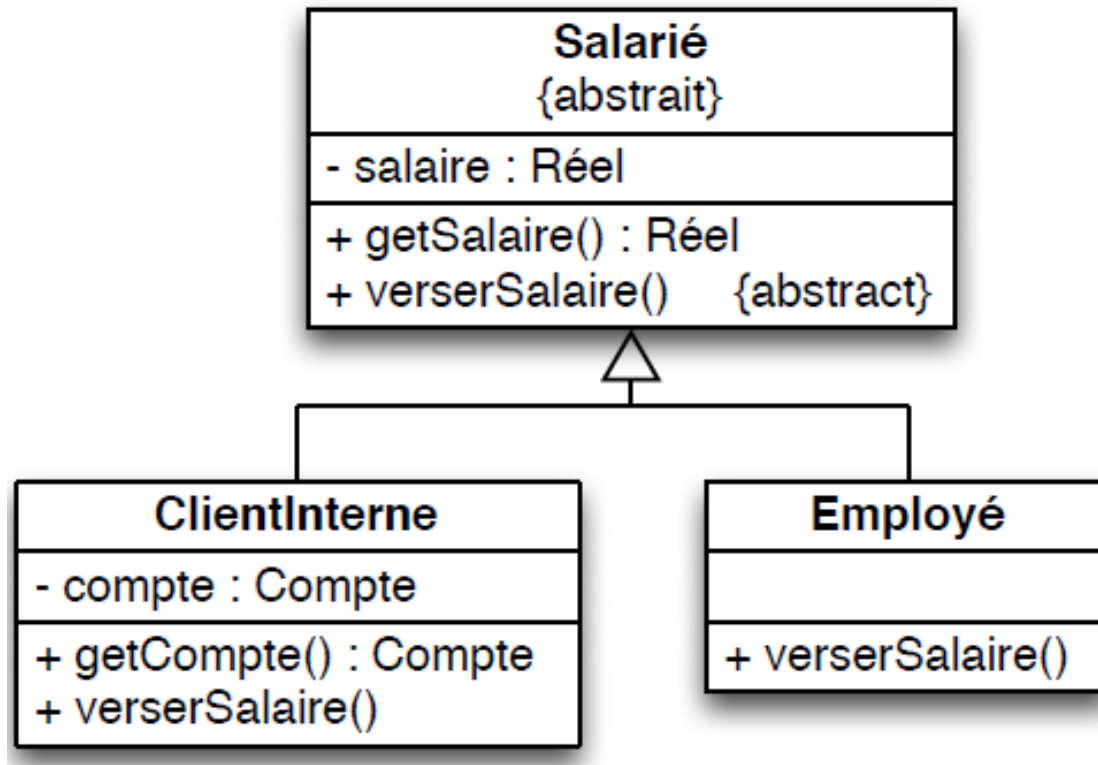
Diagramme de classe

Classe abstraite

- Une classe abstraite est une classe qui **ne peut pas être directement instanciée**.
- Une classe abstraite **encapsule des attributs et méthodes** qui peuvent **être utilisés** par les instances des classes qui en héritent.
- Elle encapsule des **méthodes dites abstraites**, dont le **traitement n'est pas défini** dans la classe.
- Les **classes héritant** d'une classe abstraite **doivent obligatoirement redéfinir ses méthodes abstraites**.

Diagramme de classe

Classe abstraite



Virement sur le compte interne ou l'envoi d'un chèque

Diagramme de classe

interface

- Une **interface** décrit des objets mais **uniquement** en terme de **méthodes abstraites**.
- Une interface **ne peut contenir ni attribut, ni méthode implémentée**.
- Le terme **d'héritage n'est pas utilisé** entre classes et interfaces : on dit qu'une **classe implémente** une interface.
- Une classe peut **implémenter plusieurs interfaces différentes**.
- Lorsqu'une classe implémente une interface, **elle doit redéfinir toutes ses méthodes abstraites**.
- Une **interface peut hériter d'une autre interface**.

Diagramme d'objets

Diagramme d'objets

Définition

- Un **diagramme d'objets** est une **instance** d'un **diagramme de classes** et illustre **l'état** d'un système a un **moment donné**.
- Exprimant la **Structure statique**
- Les diagrammes d'objets s'utilisent principalement :
 - pour **montrer** un **contexte** e.g., **avant** ou **après** une **interaction**,
 - pour **faciliter** la **compréhension** des **structures** de données **complexes**.

Ex: structures récursives

Diagramme d'objets

Rôle

- Il **représente** les **objets** d'un système (les instances des classes) et leurs **liens** (les instances des associations) à un **instant donné**.
- Il donne une vue **figée** du système à un **moment précis**.
- A un diagramme de **classe** correspond une **infinité** de diagrammes **d'objets**
- Nous nous **servirons** du diagramme **d'objet** pour donner des **exemples** qui permettront d'affiner le diagramme de classe et de **mieux le comprendre**.

Diagramme d'objets

Représentation graphique

- Un objet est représenté dans un **rectangle** dans lequel figure le **nom** de l'objet (**souligné**) et **éventuellement** la **valeur** de **un ou plusieurs** de ses **attributs**.

nom de l'objet

nom de l'objet:Classe

:Classe

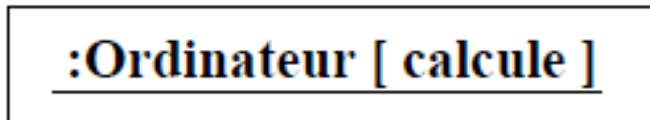
:Personne

- Des groupes d'objets instances d'une même classe peuvent se représenter.
 - Un message envoyé vers un groupe est reçu par tous les objets du groupe.

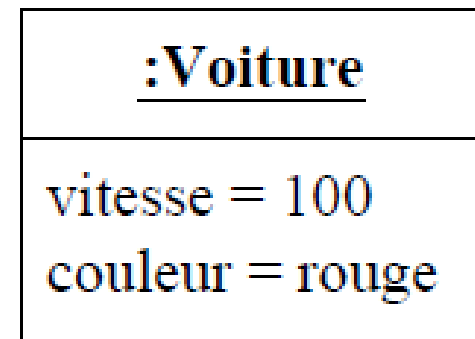
Diagramme d'objets

Représentation graphique

- ❖ **L'état** d'un objet est déterminé par les **valeurs** de ses **attributs** :
il est possible de **nommer** un **état** afin d'indiquer clairement dans quel **état** se trouve un objet.
- ❖ *Les représentations des objets peuvent contenir des attributs significatifs.*



Représentation de l'**état** d'un objet

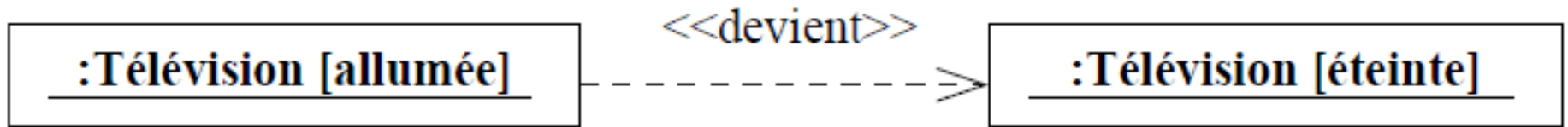


Représentation des **attributs significatifs**

Diagramme d'objets

Représentation graphique

Possibilité de modéliser les **changements d'états** des objets



- Possibilité d'utiliser des **liens stéréotypés** (exemple : la copie d'objets)

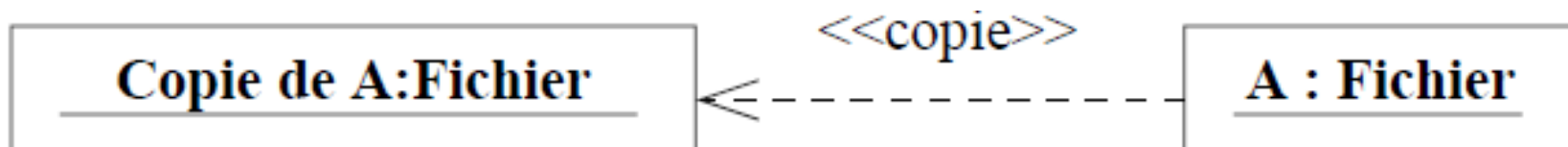
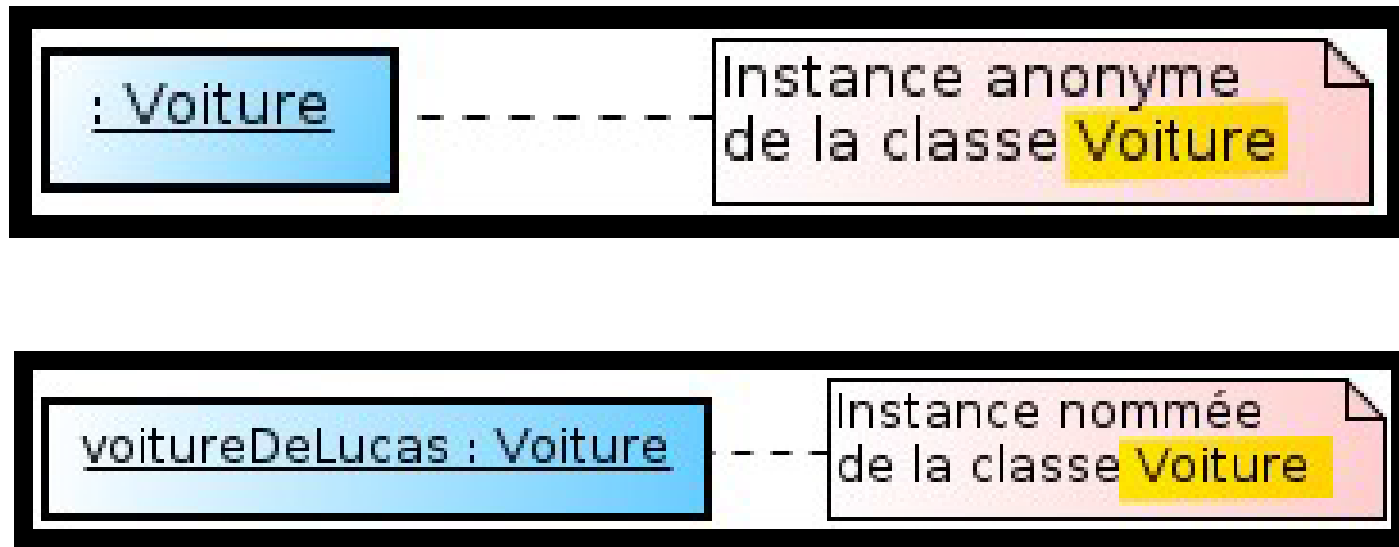


Diagramme d'objets

Représentation graphique

Exemple



voitureDeLucas est appelé le nom (identifiants) d'objet.

Diagramme d'objets

Représentation graphique

Exemple

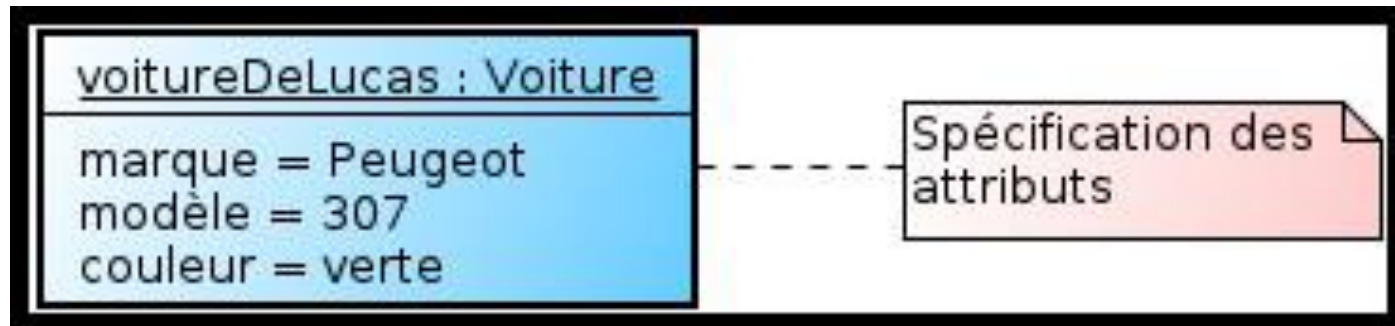
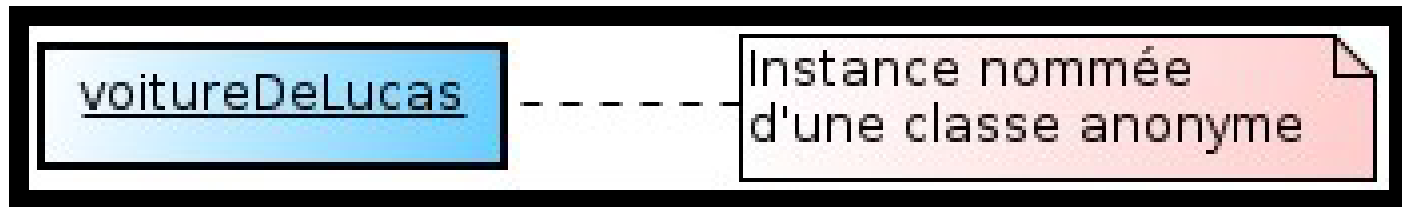
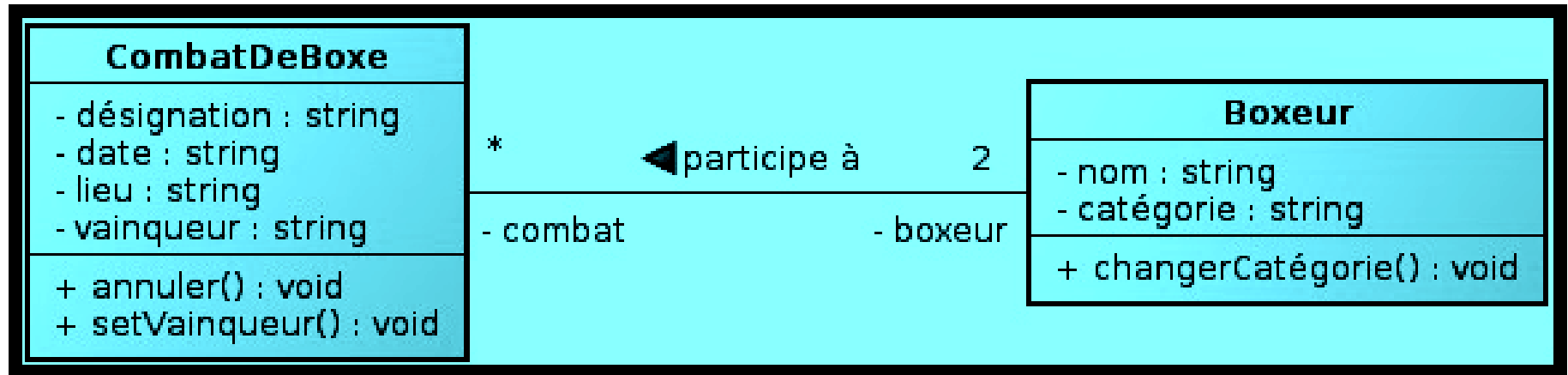


Diagramme d'objets

Exemple



Boxeurs:

Ali , lourd et Freman, lourd.

Combat: Championnat du monde le 24/09/1974 à Kinshasa le vainqueur Mohamed Ali

Diagramme d'objets

Exemple

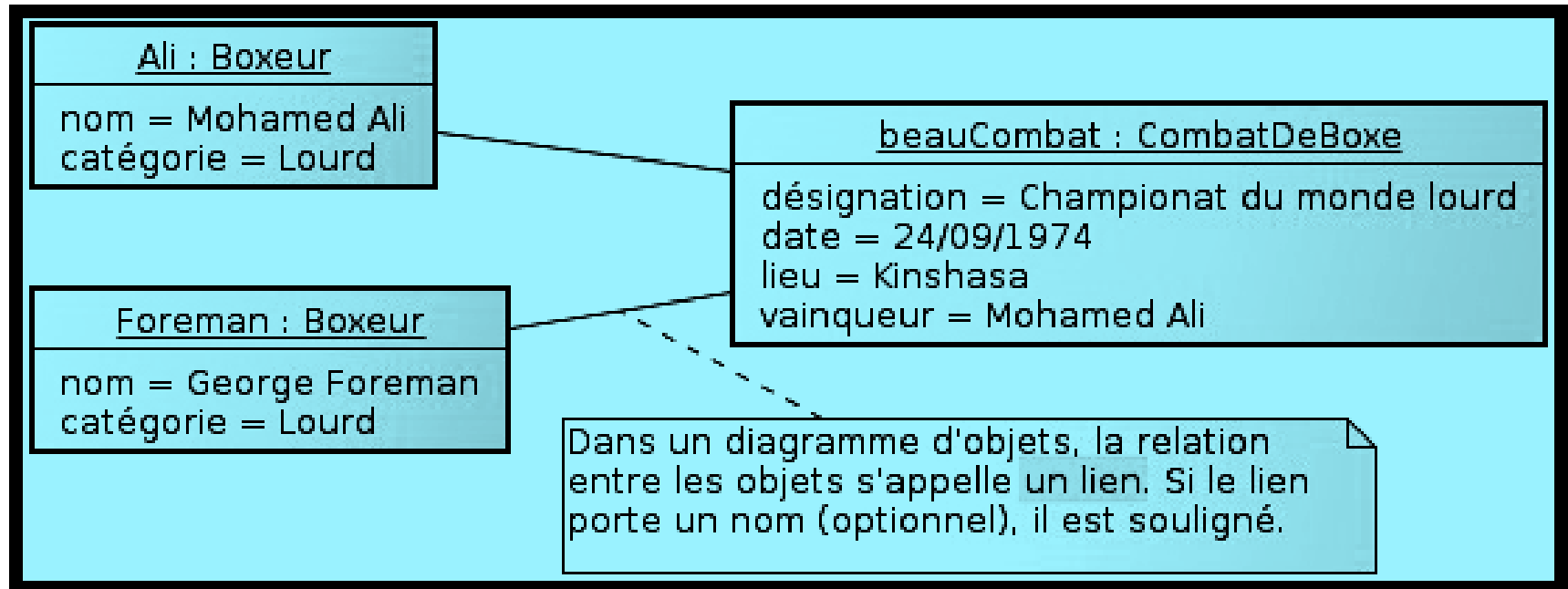


Diagramme d'objets

Représentation graphique

Lien

- Les objets sont reliés par des instances d'associations : les **liens**.
- Un **lien** représente une relation entre objets **à un instant donné**.
- *ATTENTION* : la **multiplicité** des extrémités des liens est toujours de **1**.

Exemple : représentation de la structure générale d'une voiture

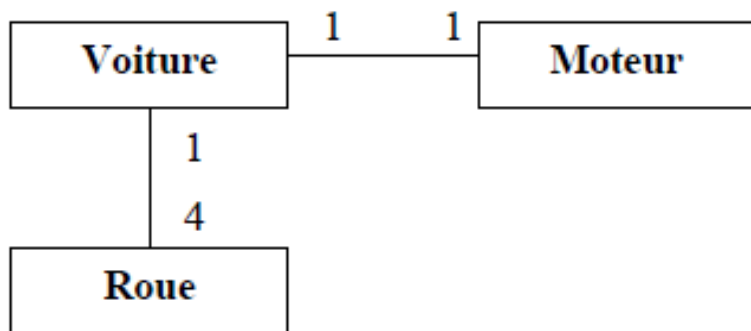


Diagramme de classe

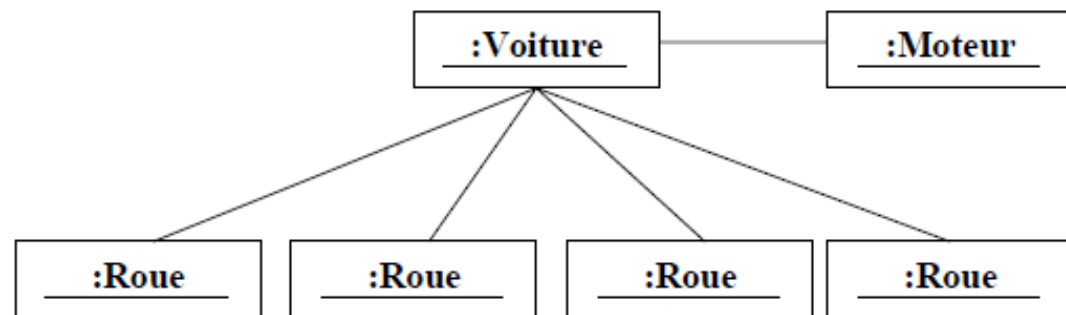


Diagramme d'objet

Diagramme d'objets

Représentation graphique

Lien

- Les *rôles* des *associations* peuvent être représentés explicitement :

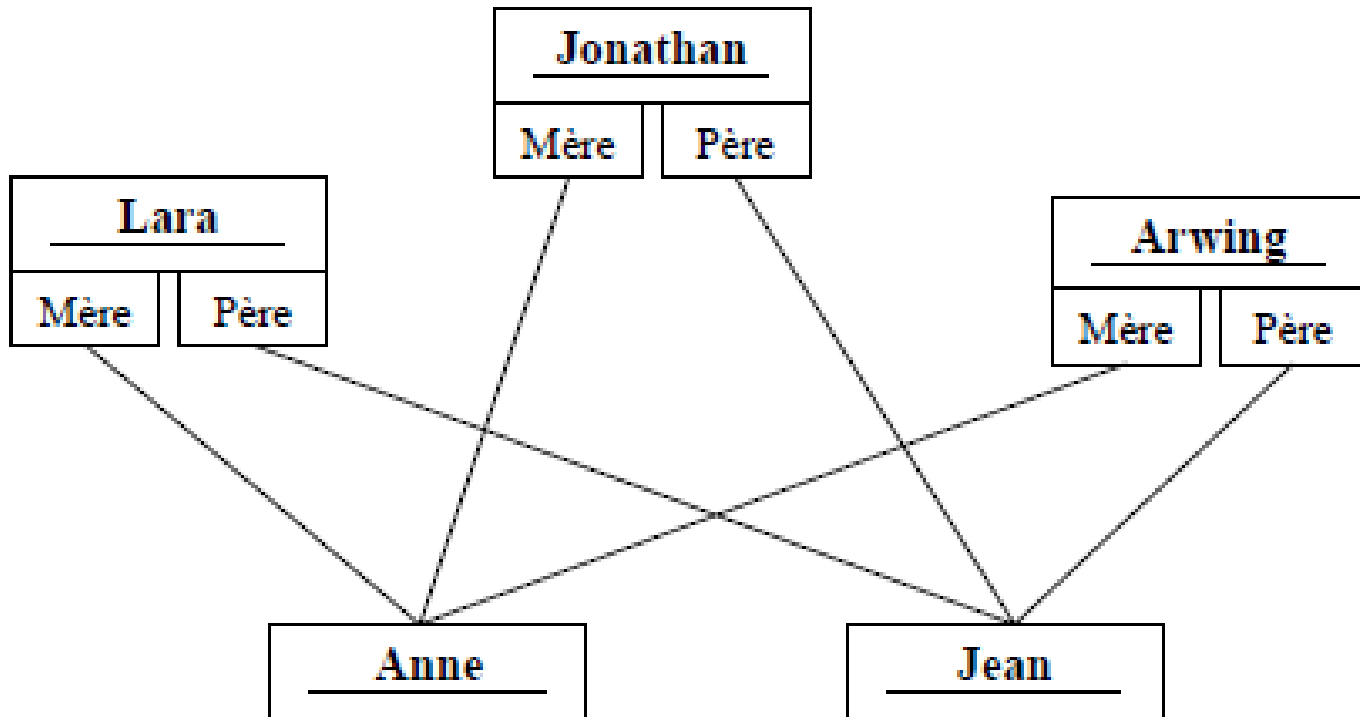


Diagramme d'objets

Représentation graphique

❖ Objet composite

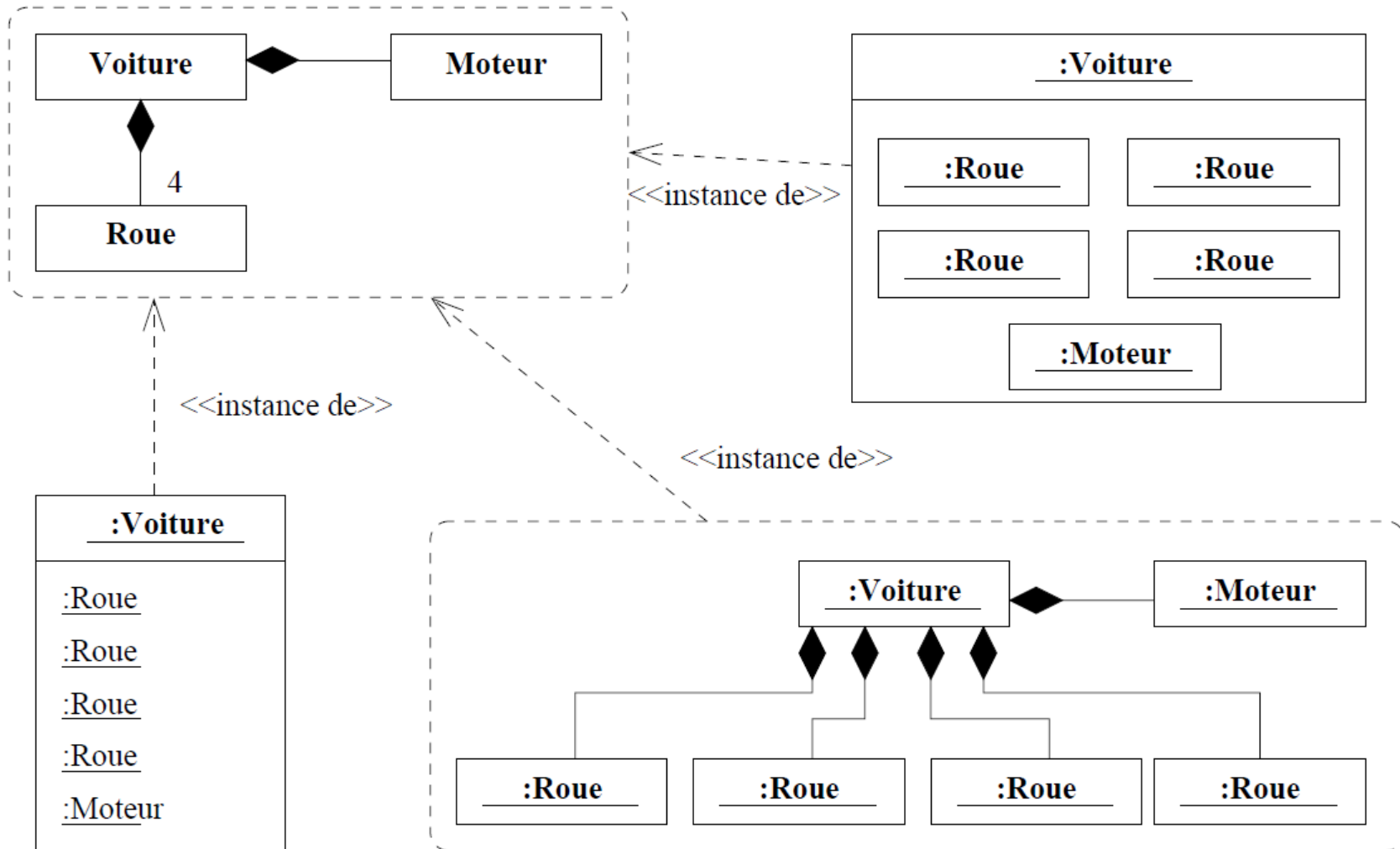


Diagramme d'objets

Représentation graphique (Exemple)

Exemple d'un système de sécurité

- *Il s'agit d'un système de sécurité limitant les accès à des parties d'un bâtiment à l'aide de cartes magnétiques.*

Diagramme d'objets

- Le système gère un seul bâtiment contenant trois portes.
- Le système peut être géré par une personne nommée Tolkien.
- Deux utilisateurs peuvent accéder au bâtiment :
 - Gandalf a accès à la première (8h-18h) et seconde porte (12h-24h).
 - Bilbon a accès à la troisième porte toute la journée.

Exemple 2

Diagramme de classe

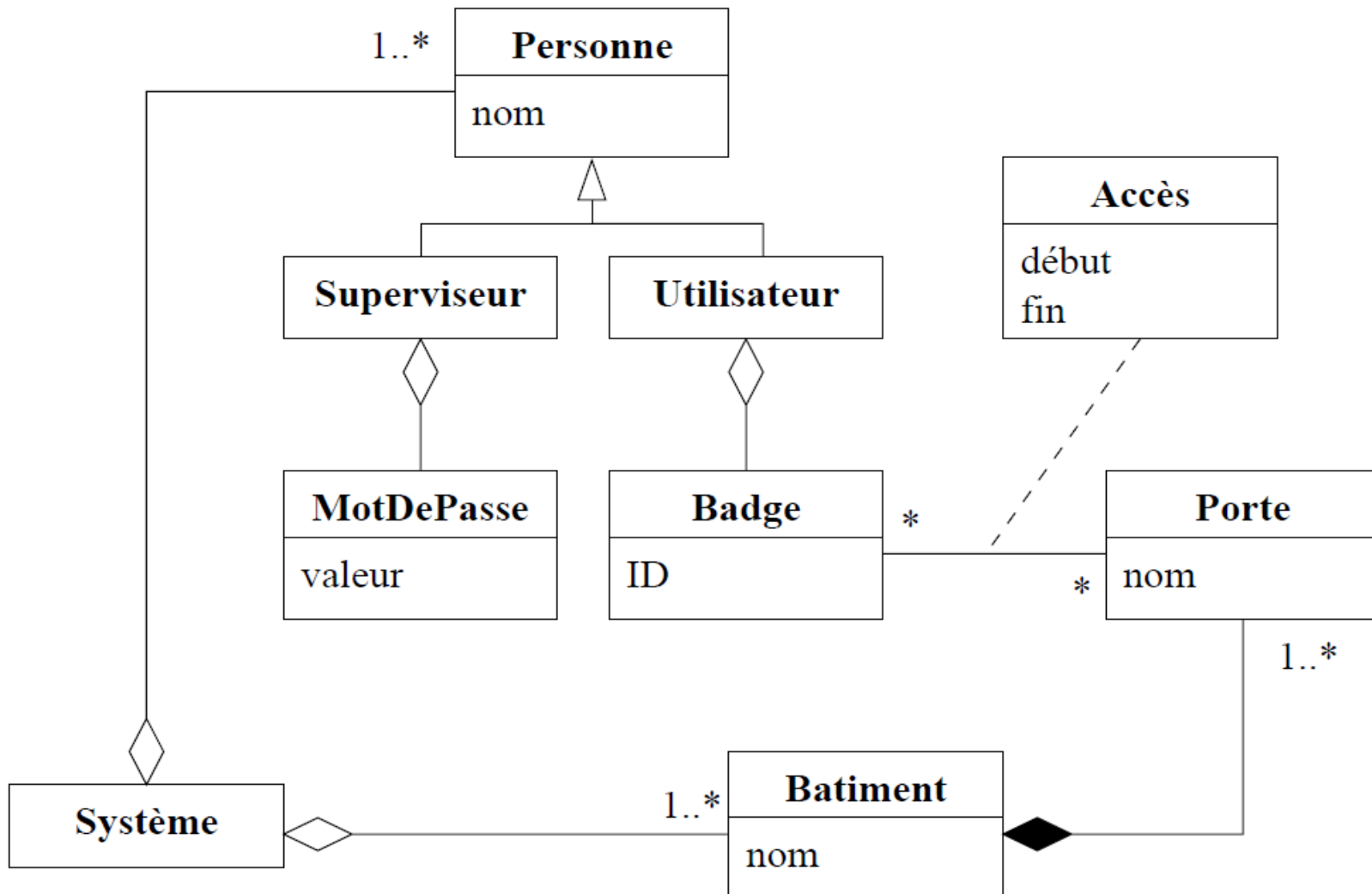


Diagramme d'objets

Exemple 2

Diagramme d'objets

- *Le système gère un seul bâtiment contenant trois portes.*
- *Le système peut être géré par une personne nommée Tolkien.*
- *Deux utilisateurs peuvent accéder au bâtiment :*
 - Gandalf a accès à la première (8h-18h) et seconde porte (12h-24h)
 - Bilbon a accès à la troisième porte toute la journée.

Diagramme d'objets

Exemple 2

Diagramme d'objets

