

Modélisation UML



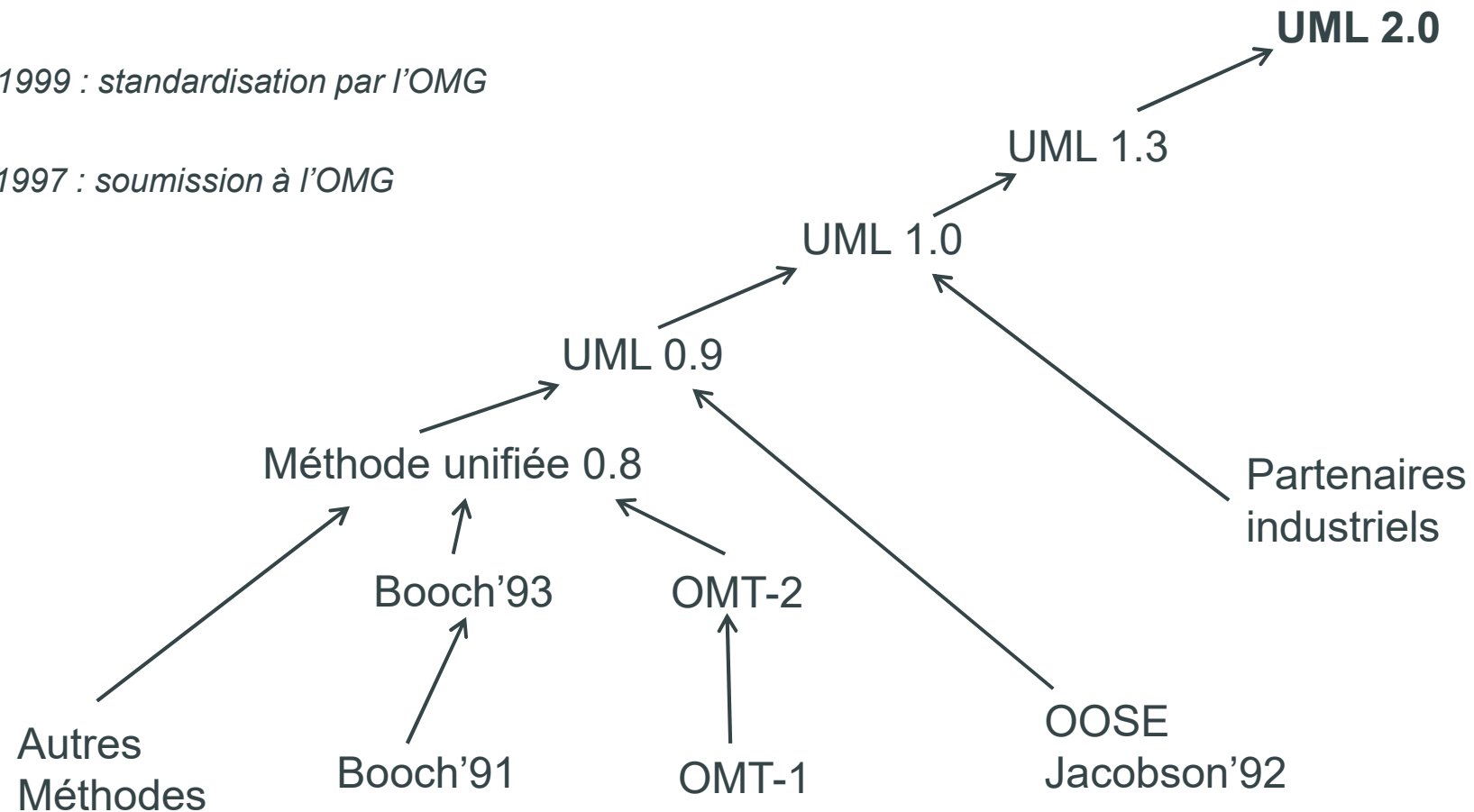
DEFINITION:

- **Langage** destiné aux phases amont de la réalisation d'un logiciel.
- Technique de modélisation unifiée issue de méthodes orientées objets plus anciennes :
 - OMT (James Rumbaugh),
 - OOSE (Ivar Jacobson),
 - OOD (Grady Booch)
- Pourquoi des **standards** ?
 - Faciliter la **communication***

HISTORIQUE

1999 : standardisation par l'OMG

1997 : soumission à l'OMG



OBJECTIFS :

- Représenter des **systèmes** entiers.
- Créer un langage de **modélisation**
 - Utilisables par les hommes & machines
 - Compatible avec toutes les techniques de réalisation
 - Adapté à toutes les phases du développement
- **Indépendant** des langages de programmation

OBJECTIFS:

- **Visualiser**

- *Chaque symbole graphique a une sémantique*

- **Spécifier**

- *de manière précise et complète, sans ambiguïté*

- **Construire**

- *les classes, les relations,*

- **Documenter**

- *les diagrammes, notes, contraintes, exigences*

REMARQUES :

- UML = **Langage** (Unified Modeling Language)

*Pas une **méthode***

- Des problèmes de **communication** persistent :
 - diagrammes complexes : difficiles à comprendre
 - diagrammes simples : sujets à interprétation
 - « mal s'exprimer » : apparition d'erreurs, d'imprécisions...

LES DIAGRAMMES :

- UML utilise des **diagrammes** :
 - Possibilité de visualiser et manipuler des éléments de **modélisation**
 - Représentation **graphique** d'une séquence d'opérations ou de la structure d'un système
 - Simplicité de compréhension

LES DIAGRAMMES :

9 Diagrammes au total (UML 1.4)

4 **Statiques** (structures – données) :

- Diagramme d'objets
- Diagramme de classes
- Diagramme de composants
- Diagramme de déploiement

5 **Dynamiques** (comportement) :

- Diagramme de cas d'utilisation
- Diagramme de collaboration
- Diagramme de séquence
- Diagramme d'états-transition
- Diagramme d'activités

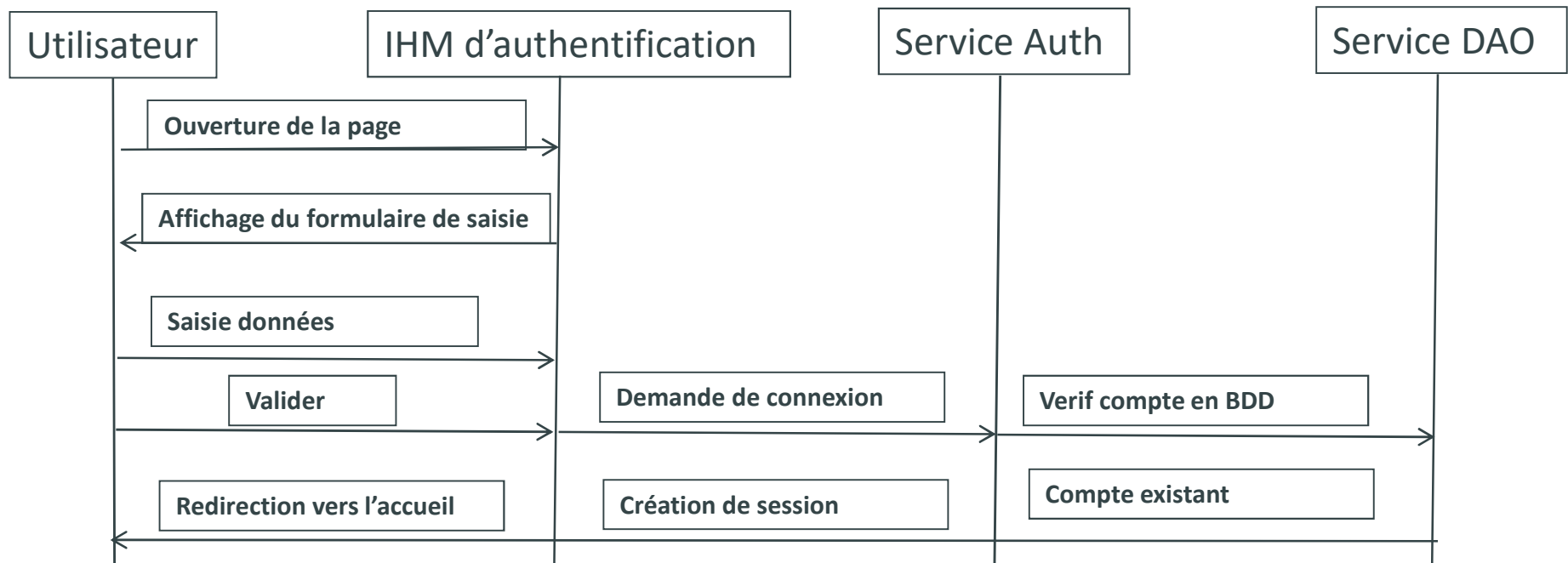
DIAGRAMMES

Découverte et analyse des besoins :

- **Diagramme de cas d'utilisation** : décrit les **fonctions** du système selon le point de vue de ses futurs **utilisateurs**
- **Diagramme de séquence** : représentation **temporelle** des interactions des objets
- **Diagramme d'activités** : pour des représentations **grossières**, ils représentent le **comportement** d'un service ou d'un cas d'utilisation.

DIAGRAMMES

- Exemple de diagrammes de séquence



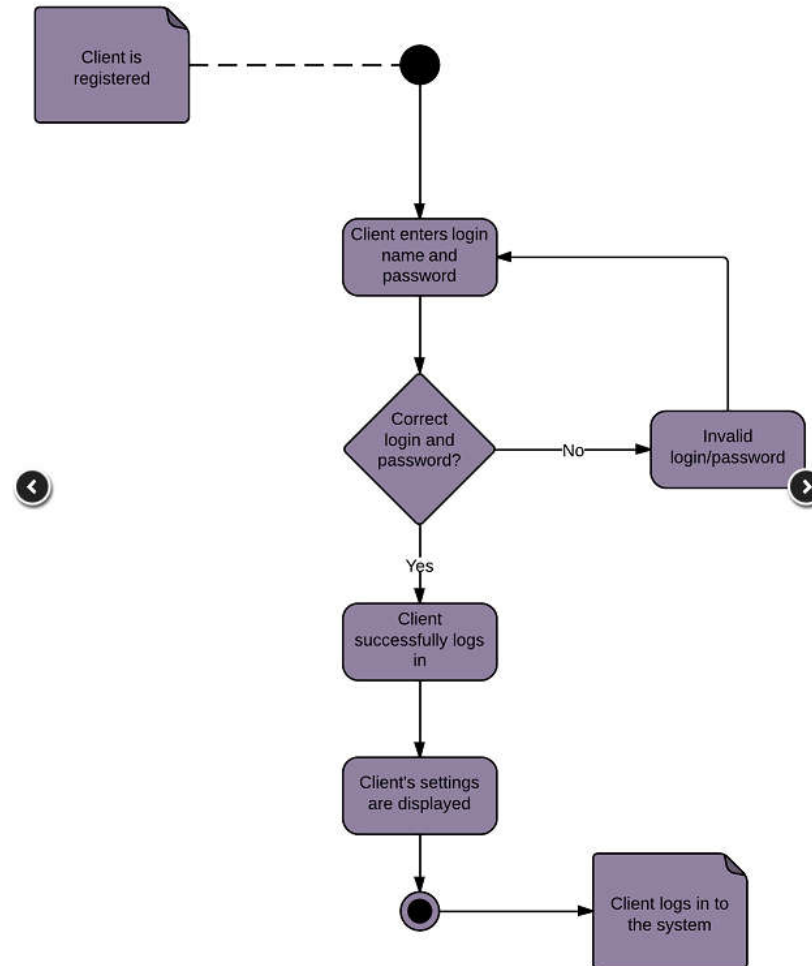
DIAGRAMMES

Spécification :

- **Diagramme de classes:** structure des données du système définies comme un ensemble de relations et de classes.
- **Diagramme d'objets:** illustration des objets et de leurs relations. Ce sont des diagrammes de collaboration simplifiés sans représentation des envois de messages.
- **Diagramme de collaboration:** représentation des interactions entre objets.
- **Diagramme d'états-transitions :** représentation du comportement des objets d'une classe en terme d'états et de transition d'états

DIAGRAMME

Exemple de diagramme d'activités:



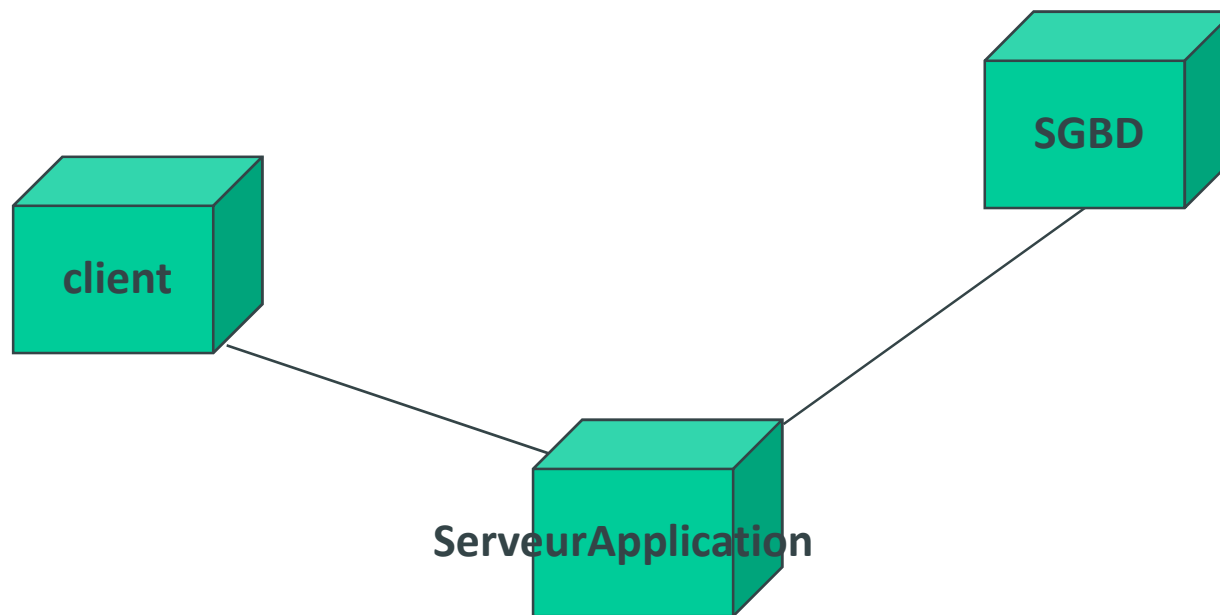
DIAGRAMME

Conception:

- **Diagramme de séquence:** représentation des interactions temporelles entre objets et dans la réalisation d'une opération.
- **Diagramme de composants:** architecture des composants physique d'une application.
- **Diagramme de déploiement:** description du déploiement des composants sur les dispositifs matériels

DIAGRAMME

Exemple de diagramme de déploiement:



UML : DIAGRAMME DE CLASSES

APPROCHE OBJET

- Représentation d'un **concept** ou d'une **chose**, ayant des **limites** et un **sens** dans un **contexte** donné.
- Physique ou Abstraite
- Caractéristiques :
 - Informations
 - Identifiant
 - Comportement
 - Messages
 - Encapsulation

UML : DIAGRAMME DE CLASSES

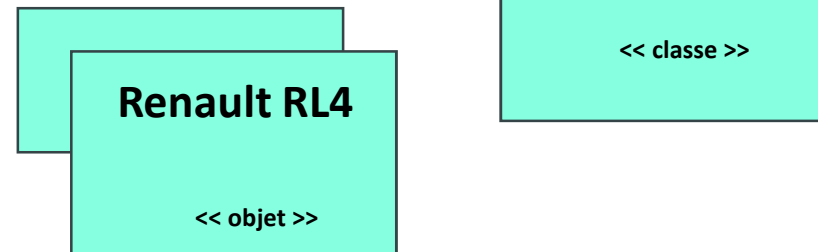
CARACTERISTIQUES

- **Informations** : attributs décrivant l'objet
- **Identifiant** : permet d'individualiser les objets entre eux
- **Comportement** : implémenter sous forme de « méthodes »
- **Messages** : moyen de communication entre objets
- **Encapsulation** : gestion des accès

UML : DIAGRAMME DE CLASSES

DEFINITION :

- **Classe :**
modélisation de l'ensemble des objets ayant :
 - attributs similaires
 - comportement en commun
 - relations communes avec d'autres objets



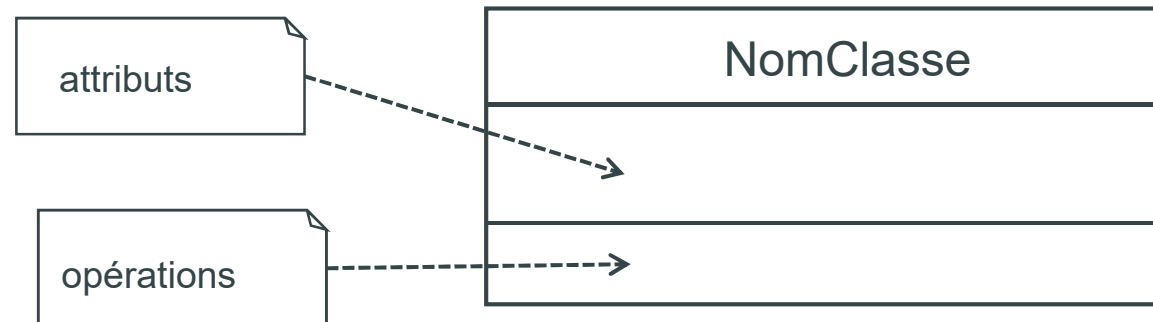
UML : DIAGRAMME DE CLASSES

NOTATION :

- Tout nom de classe commence par une majuscule

NomAttribut : type = ValeurInitiale

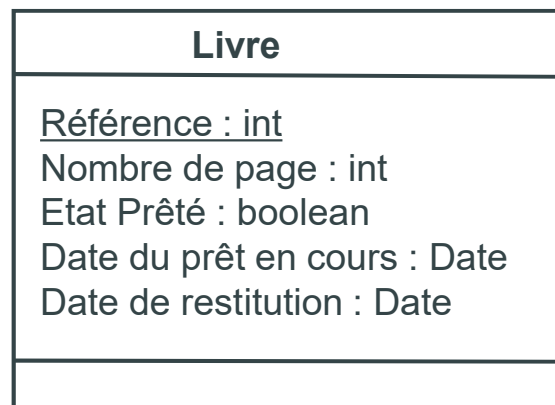
NomOpération(ListeArguments) : typeRetour



UML : DIAGRAMME DE CLASSES

ATTRIBUT :

- Donnée gérée par la classe (donc pour toutes les instances)
- Valeur unique par instance (mais la même peut être dans plusieurs occurrences)
- Identifiant est un attribut :
 - particulier, autant de valeurs que d'instances
 - valeurs déterminent les instances de la classe



UML : DIAGRAMME DE CLASSES

OPERATIONS - METHODES:

- **Opération** : effectuée par l'objet
- **Méthode** : effectuée par la classe
- **Signature** (notation) : *NomOperation(arg1,arg2) : TypeOutput*
- **Comportement** : effets observables des opérations

LIVRE
<u>Référence</u> Nombre de page Date du prêt en cours
<u>Créer(arg1,arg2...) : bool</u> <u>Prêter(arg1,arg2...) : bool</u>



UML : DIAGRAMME DE CLASSES

OPERATIONS :

- **Peut :**
 - Accéder à ses propres attributs
 - Invoquer une autre opération de son propre objet
 - Invoquer une autre opération d'un autre objet (même classe ou autre classe)
- **Ne peut pas:**
 - Accéder directement aux attributs d'un autre objet

UML : DIAGRAMME DE CLASSES

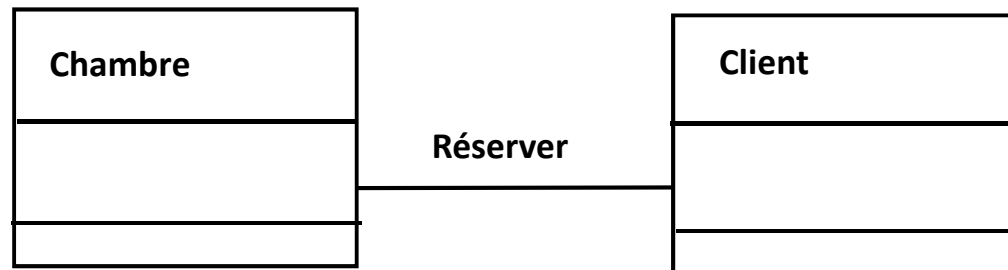
LIENS & ASSOCIATIONS:

- **Lien :**
 - connexion physique ou conceptuelle entre instances de classes
- **Association:**
 - Instance du lien (ex : rédiger, créer)
 - L'invocation d'une méthode est une association
 - Représentation
 - Unidirectionnelle 
 - Bidirectionnelle 

UML : DIAGRAMME DE CLASSES

TYPES ASSOCIATIONS:

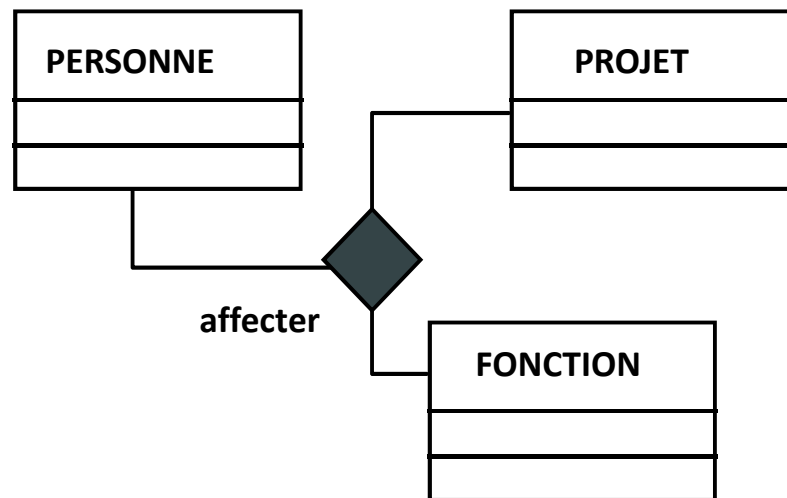
- **Association binaire** : associe 2 classes



UML : DIAGRAMME DE CLASSES

TYPES ASSOCIATIONS:

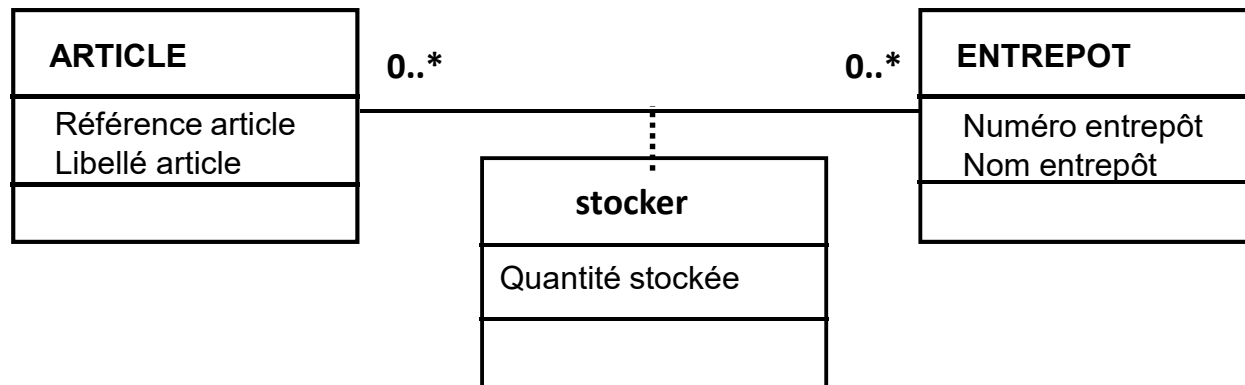
- **Association n-aire** : associe plus de 2 classes



UML : DIAGRAMME DE CLASSES

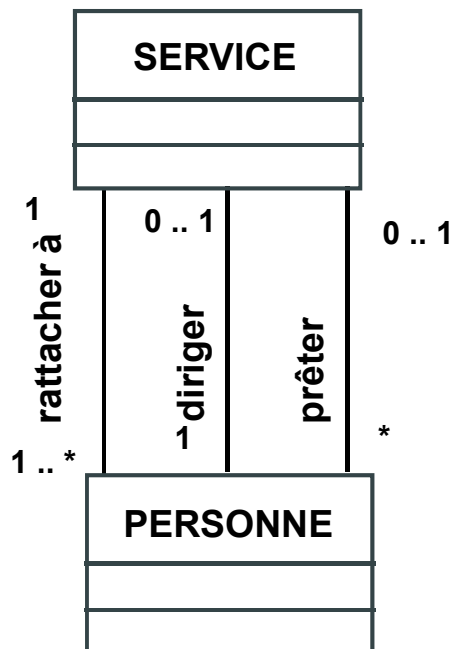
CLASSE D'ASSOCIATION:

- Définition : Classe qui a les caractéristiques d'une association ou Association qui a les caractéristiques d'une classe.
- Nécessaire quand les attributs ou les opérations sont liées à l'ensemble des classes de l'association
- Utile pour simplifier



UML : DIAGRAMME DE CLASSES

MULTIPLICITES (cardinalités):



1	Un et un seul
0..1	Zéro ou Un
M..N	De M à N (entiers naturels)
0..*	De Zéro à plusieurs
*	
1..*	De Un à plusieurs
N	Exactement N (entier naturel)

UML : DIAGRAMME DE CLASSES

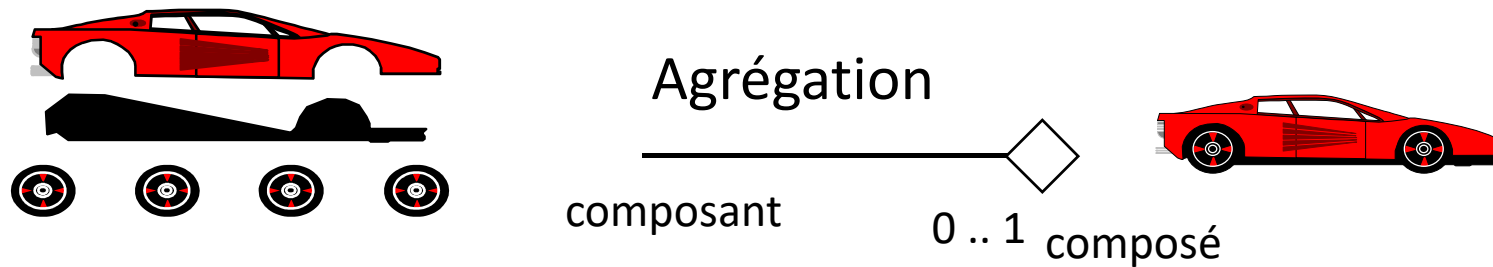
CONCEPTS AVANCES...

UML : DIAGRAMME DE CLASSES

AGREGATION

- **DEFINITION :**

Forme d'association qui spécifie la relation entre un **ensemble** ou « **composé** » et sa **partie** ou « **composant** ».

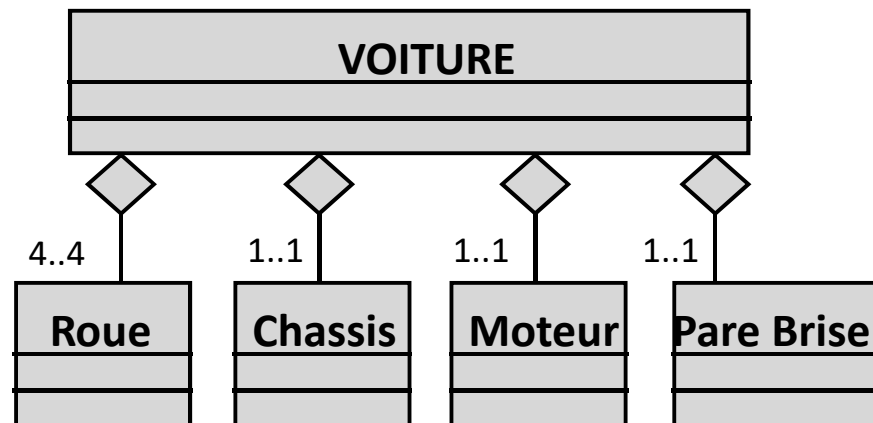


Contrainte D'intégrité Fonctionnelle **FAIBLE**

UML : DIAGRAMME DE CLASSES

AGREGATION

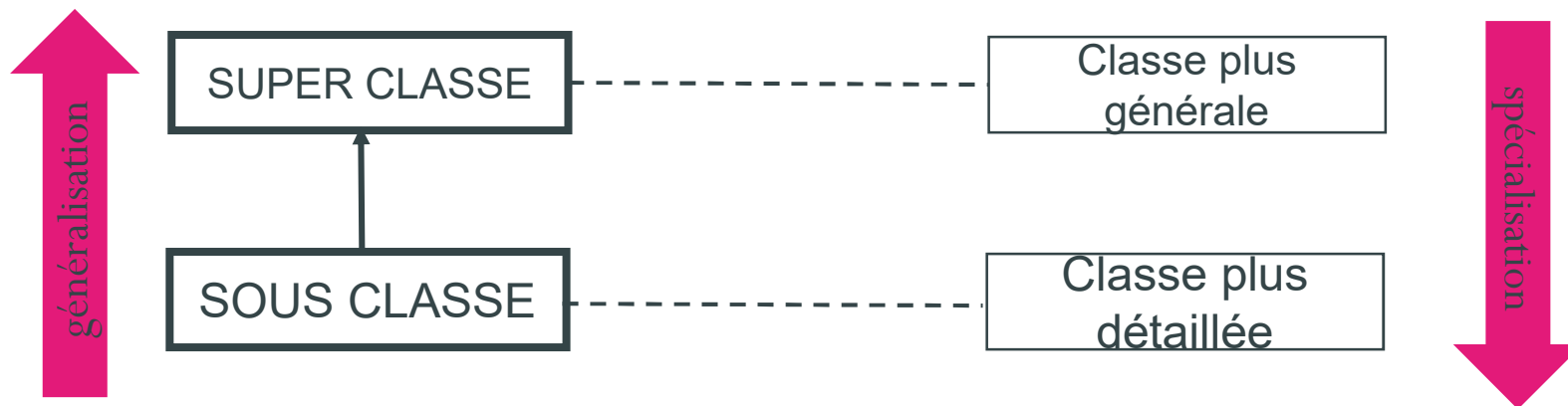
- La partie (composant) peut exister sans l'agrégat (composé)
- Si le composé (agrégat) disparaît, le composant (partie) peut continuer d'exister



UML : DIAGRAMME DE CLASSES

LA CLASSIFICATION :

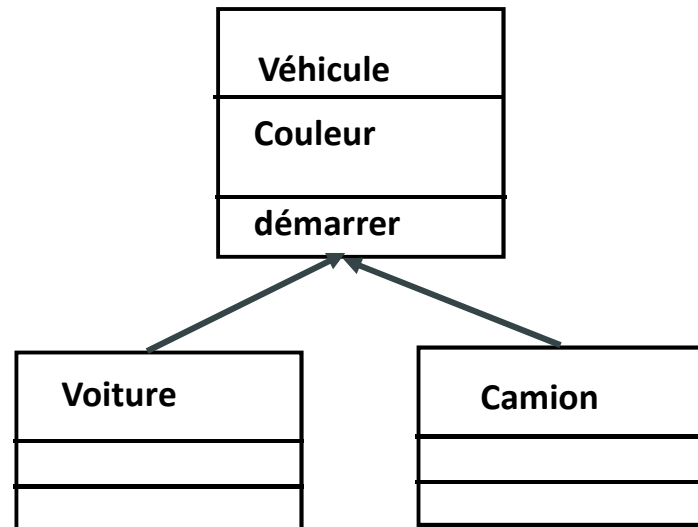
- 2 Approches :
 - La généralisation
 - La spécialisation



UML : DIAGRAMME DE CLASSES

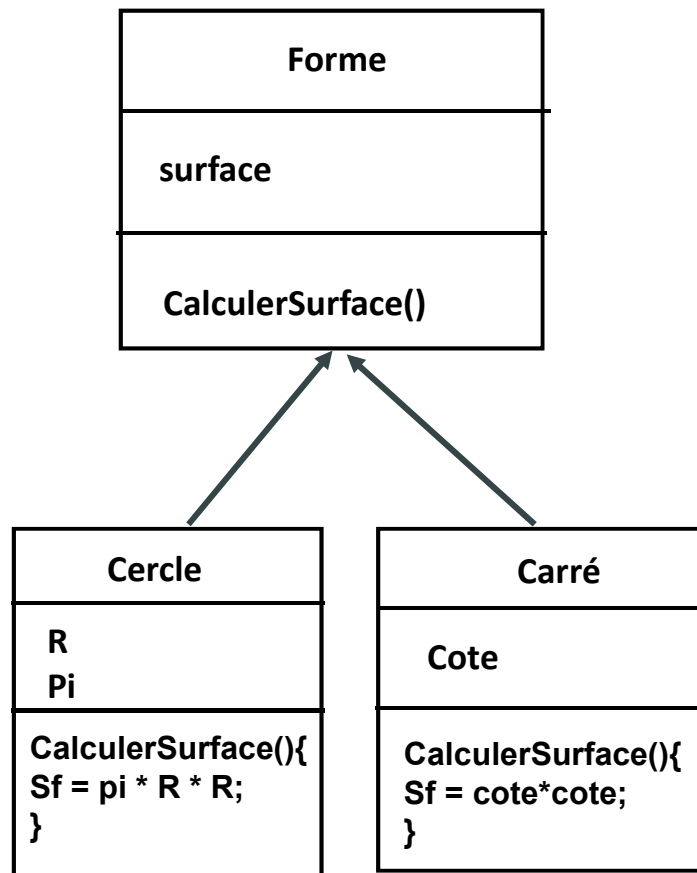
GENERALISATION:

- Factoriser les éléments communs (attributs, opérations, contraintes) d'un ensemble de classes dans une classe plus générale appelée Super-Classe (ou classe mère).



UML : DIAGRAMME DE CLASSES

POLYMORPHISME



- Une opération est polymorphe lorsque :
 - Elle est appelée sous forme générique au niveau de la Super-Classe
 - Elle est spécifiée au niveau de chaque Sous-Classe