

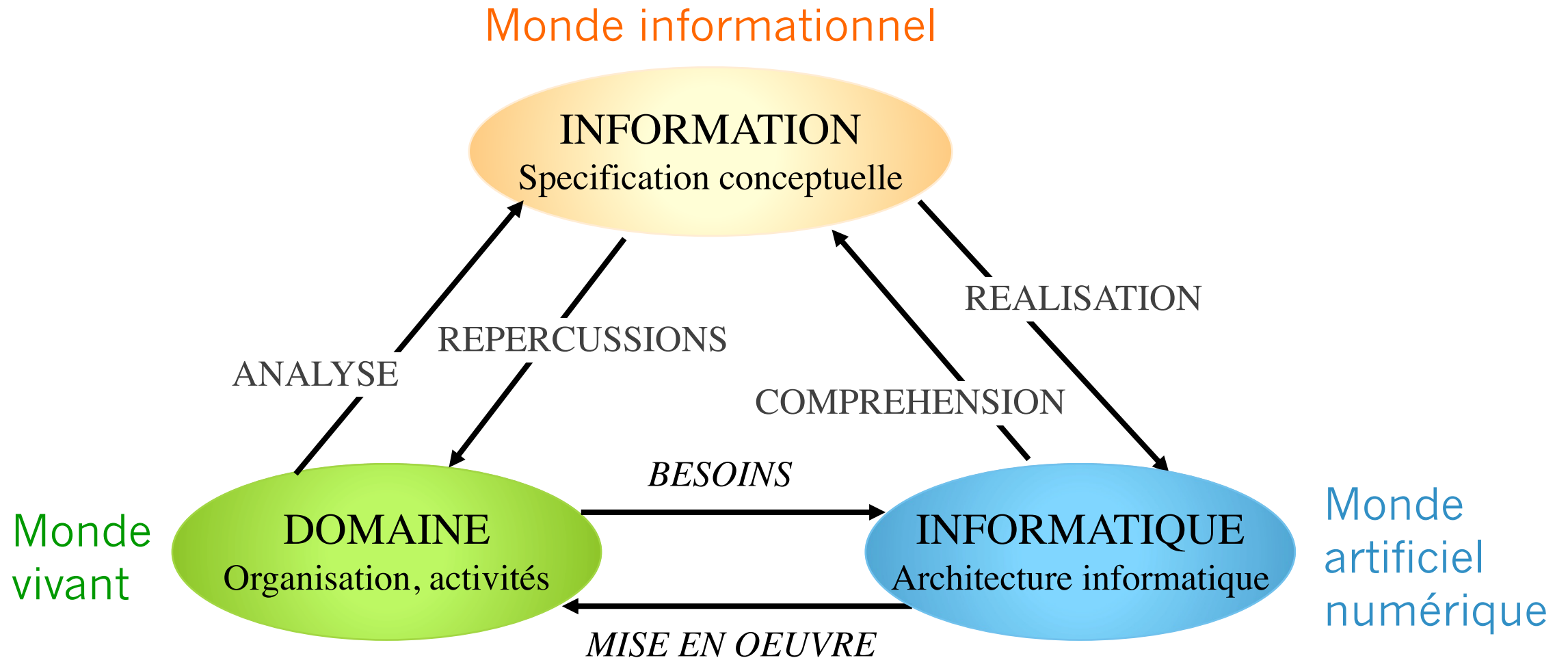
MSIS Chapitre 3

Le concept de Classe

Dans ce chapitre

- Préambule – principes de développement des SI
- Classe et ses propriétés
- Identifiants des objets d'une classe
- Représentation graphique
- Forme normale de classe

Principes de développement des SI



Modèles pour chaque dimension

Information

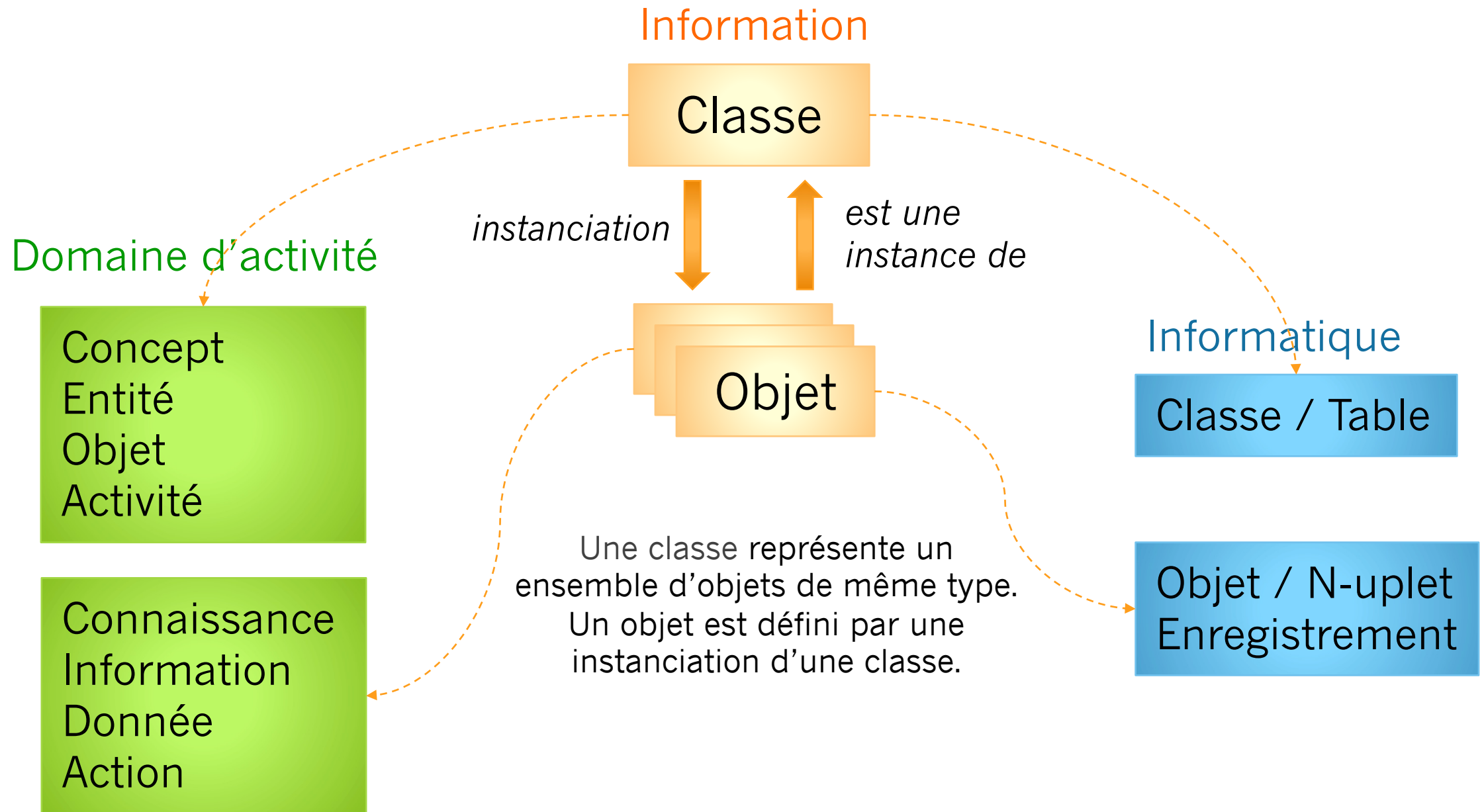
Entité-Association
Relationnel
Objet

Métier / organisation

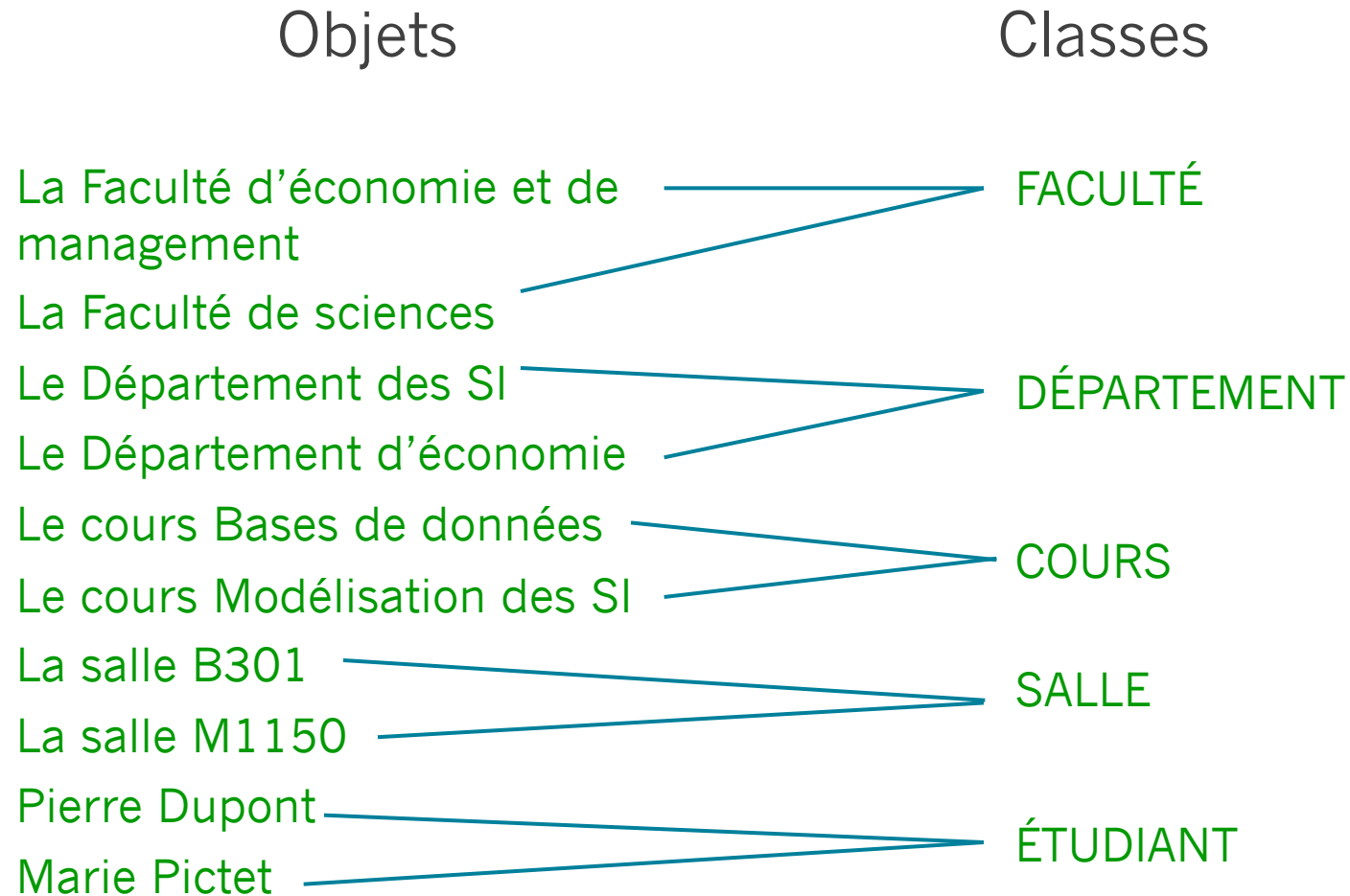
Business
Processus métier
Rôles
Règles de gestion

Informatique

Relationnel
Objet



Principe abstraction – instanciación



Définition

Classe (d'objets) – un type abstrait de données qui définit la structure d'un ensemble d'objets de même nature – un schéma informationnel comprenant :

- les propriétés (**attributs** et **opérations**) communes à ses objets,
- un mécanisme (**instanciation**) permettant de créer les objets ayant les propriétés de la classe.

Un **objet** est une instanciation d'une classe.

classe = attributs + méthodes (opérations) + instanciation

Propriétés d'une classe

Attributs, méthodes, états

Propriétés d'une classe

Les objets d'une classe partagent en commun un schéma informationnel comprenant :

- des **attributs** associés à des domaines de valeurs
pour lesquels les objets vont prendre une ou plusieurs valeurs de leur domaine
- des **méthodes**
qui, appliquées aux objets de la classe, transforment les valeurs qu'ils prennent pour les attributs
- des **états**
que des méthodes ou des traitements font prendre aux objets

Attribut

Attribut – une représentation d'une propriété associée à une classe d'objets.

- Tout attribut a un nom
les attributs de la classe VÉHICULE :
NoVéhicule, Catégorie, Marque, NoChâssis, NoMoteur, DateCirculation
- Un attribut est une fonction qui associe une valeur à un objet.
NoVéhicule = 56
Catégorie = 'tram'
Marque = 'Alstom'
NoChâssis = 'AWE34DF56'
NoMoteur = '2354-34'
DateCirculation = '12/10/2020'

Domaine d'un attribut

Le **domaine d'un attribut** définit l'ensemble de valeurs possibles qu'un attribut peut prendre pour cet attribut.

- Tout attribut a un et un seul domaine.
- Plusieurs attributs peuvent avoir le même domaine.

Si A est un attribut de la classe C , et o un objet de C , alors :

- **Dom(A)** définit l'ensemble des valeurs que les objets de C peuvent prendre pour A
ex.: **Dom(NoVéhicule) = Mot**
- **$o[A]$** (autre notation usuelle $o.A$) désigne les valeurs que prend l'objet o pour l'attribut A
ex.: **véhicule1[NoVéhicule] = 56**

Domaine d'un attribut

En modélisation conceptuelle

Mot
Texte
Ordonné (Unité)
Numérique (Unité)
 entier, entier positive, décimal, réel
Booléen
Date

Dom(NoVéhicule) = Mot

Dom(Catégorie) = Mot

Dom(DateCirculation) = Date

Dom(Nom) = Texte

En informatique

Char
Varchar
Numeric (Decimal ou Dec)
Integer (ou Int)
Float, Real
Boolean
Date

Opérations selon le type de domaine

Le type d'un domaine indique les opérations qui sont définies sur ses valeurs

- Texte : aucune
- Mot : $= \neq$
- Ordonné : $= \neq > \geq \leq <$
- Numérique : $= \neq > \geq \leq < + - \times \div$
- Booléen : ET OU NON
- Date : $= \neq > \geq \leq <$

Définition des domaines

Pour définir un nouveau domaine, on précise ses valeurs :

- par énumération dans le cas de domaines de type mot ou mot ordonné ou booléen

Dom COULEUR : Mot {rouge, bleu, jaune, vert}

- avec des intervalles de définition s'il est de type numérique ou mot ordonné

Dom NOJOUR : Numérique entier [1, 31]

- par un formalisme mathématique ou algorithmique

Dom NOTE : Numérique décimal [1, 6] arrondi à 2 chiffres après virgule

Exemples de définition des domaines

- Dom MOIS : Mot {janvier, février, mars, avril, mai, juin, juillet, août, septembre, octobre, novembre, décembre}
- Dom ANNÉE : Mot ordonné [1900, -]
- Dom TAILLE : Mot ordonné {'XS', 'S', 'M', 'L', 'XL'}
- Dom HEURE : Mot ordonné [0, 23]
- Dom MINUTE : Mot ordonné [0, 59]
- Dom HORAIRE : Mot ordonné composé de HEURE et MINUTE
- Dom DURÉE : Numérique entier (unité seconde)
- Dom POIDS : Numérique entier positive (unité grammes)

Classe - Attribut - Domaine

Classe CL (A1, A2, A3...)

VEHICULE (NoVéhicule, Catégorie, Marque, NoChâssis, NoMoteur, DateCirculation)

v1 est l'objet de la classe VEHICULE

Si o est un objet de CL, alors :

$o[A1]$ (autre notation : $o.A1$) désigne les valeurs que prend l'objet o pour l'attribut A1

$v1[NoVéhicule] = 56$ ou $v1.NoVéhicule = 56$

$o[A1] \in \text{dom}(A1)$ désigne que l'objet o prend une ou plusieurs valeurs du domaine de l'attribut A1

si $\text{Dom}(Catégorie) = \text{Mot } \{\text{tramway, bus, trolleybus}\}$

$\forall v1 \in \text{Véhicule } v1[Catégorie] = \text{tramway} \mid \text{bus} \mid \text{trolleybus}$

Attribut monovalué vs multivalué

Attribut monovalué – l'attribut à une seule valeur pour un objet.

Attribut multivalué – l'attribut à plusieurs valeurs pour un objet.

Notation pour attribut multivalué: **Attribut***

CHAUFFEUR (NoAVS, Nom, Prénom, Adresse, **NumTél***, DateNaiss, Sexe, TotalHeures)

- l'attribut **Nom** ne peut avoir qu'une valeur
- l'attribut **NumTél** peut avoir plusieurs valeurs

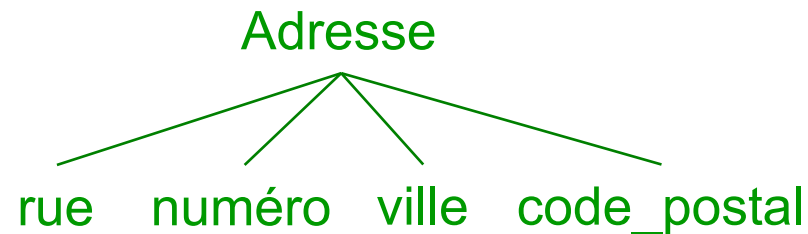
Attribut simple vs composite

Attribut simple – la valeur de l'attribut est un objet atomique ne pouvant être décomposé.

Les attributs Nom, Prénom de la classe CHAUFFEUR

Attribut composite : la valeur de l'attribut est composée de plusieurs valeurs atomiques.

L'attribut Adresse de la classe CHAUFFEUR



Attribut dérivé

Attribut dérivé – la valeur de l'attribut est calculée en fonction des autres attributs.

Notation pour attribut dérivé: **Attribut/**

CHAUFFEUR (NoAVS, Nom, Prénom, Adresse, NumTél, DateNaiss, Sexe, TotalHeures, **Age/**)

Age est un attribut dérivé car l'âge du chauffeur peut être calculé à partir de sa date de naissance.

VÉHICULE (NoVéhicule, Catégorie, Marque, NoChâssis, NoMoteur, DateCirculation, **Durée-de-Circulation/**)

La durée de circulation d'un véhicule peut être calculée à partir de la date de la mise en circulation.

Attribut : valeur obscure

Valeur obscure : valeur inconnue, valeur impossible...

- Pour chaque attribut A de la classe CL, il faut indiquer si un objet de CL peut prendre une valeur obscure pour A.
- Une valeur qui n'est pas obscure est claire.
- Notation: **Attribut⁻**

VÉHICULE (**NoVéhicule⁻**, Catégorie, Marque, NoChâssis, NoMoteur, **DateCirculation⁻**)

Si les attributs **NoVéhicule** et **DateCirculation** admettent une valeur obscure, alors on peut créer un objet de **VÉHICULE** qui n'a pas de valeur pour les attributs **NoVéhicule** et **DateCirculation**.

Par exemple, le véhicule a été acheté mais n'a pas encore été immatriculé ni mis en circulation.

CHAUFFEUR (NoAVS, Nom, Prénom, Adresse, NumTél, DateNaiss, **Sexe⁻**, TotalHeures)

Attribut permanent

Attribut permanent – un attribut A d'une classe est permanent si la valeur claire prise pour A par tout objet de la classe ne peut pas être modifiée.

Notation: **Attribut⁼**

VÉHICULE (**NoVéhicule⁼**, **Catégorie⁼**, **Marque⁼**, **NoChâssis⁼**, NoMoteur, DateCirculation⁻)

Pour tout objet de VÉHICULE les valeurs prises pour NoVéhicule, Catégorie, Marque et NoChassis ne peuvent pas être modifiées alors que celles prises pour son NoMoteur et sa DateCirculation peuvent être modifiées.

CHAUFFEUR (**NoAVS⁼**, Nom, Prénom, Adresse, NumTél, **DateNaiss⁼**, Sexe⁻, TotalHeures)

Méthode d'une classe

Méthode d'une classe – une opération qui concerne les objets de cette classe.

Méthodes primitives :

CRUD: Create – Retrieve – Update – Delete

Créer – Retrouver – Modifier – Supprimer

VÉHICULE (NoVéhicule =, Catégorie=, Marque=, NoChâssis=, NoMoteur, DateCirculation=)

Méthodes primitives

- créer (véhicule : Véhicule)
- retrouver (véhicule : Véhicule)
- modifier (véhicule.nomoteur : Véhicule.NoMoteur)
- supprimer (véhicule : Véhicule)
- activer (véhicule : Véhicule)
- désactiver (véhicule : Véhicule)

Méthodes complexes

- contrôler (véhicule : Véhicule)
- réparer (véhicule : Véhicule)

État

État – un attribut mis-à-jour seulement par des transactions (ou traitements).

Un état donne une trace du travail d'une méthode sur un objet.

VÉHICULE (NoVéhicule =, Catégorie=, Marque=, NoChâssis=, NoMoteur, DateCirculation=)

Si **DateCirculation** est un attribut contrôlé par les méthodes *activer* et *désactiver*,

- activer (véhicule) fait passer sa **DateCirculation** d'une valeur obscure à une valeur claire, la date du jour,
- désactiver (véhicule) la fait passer de claire à obscure,

alors **DateCirculation** est un attribut de type état de la classe **VÉHICULE**.

Les objets d'une classe et leur identification

La notion d'objet

Objet – une instantiation d'une classe

Objet – une représentation d'un objet du monde réel (concret ou abstrait), perçu par le concepteur comme ayant une existence propre, et à propos duquel on veut enregistrer des informations.

CHAUFFEUR (NoAVS, Nom, Prénom, Adresse, NumTél, DateNaissance, Sexe, EtatCivil)

Exemple d'objet de la classe CHAUFFEUR :

chauffeur1 ('123.244', 'Legrand', 'Pierre', 'Rue de Carouge 15 Genève',
'0221234567', '15/04/1992', , 'marié')

Comment retrouver un objet parmi tous les objets de sa classe ?

- Représentation informatique : un identificateur d'objet (oid) système.
- Représentation conceptuelle : on désigne un ensemble d'attributs qui forme un **identifiant (clé) obligatoire**.
 - La clé est définie au niveau de la classe.
 - Tous les objets de la classe doivent être identifiables de manière unique grâce à cette clé.

Identifiant / Clé obligatoire

Un ensemble d'attributs K de la classe CL forme un **identifiant (clé) obligatoire** de CL si

- tous les attributs de K sont **monovalués**, **clairs** et **permanents**;
- deux objets $o1$ et $o2$ de CL ne peuvent pas prendre les mêmes valeurs pour K : $o1[K] \neq o2[K]$;
- K est **minimal** : il n'existe aucun sous-ensemble de K vérifiant la propriété précédente.



Toute classe doit avoir au moins un identifiant obligatoire

Identifiant / Clé obligatoire

Notation formelle

Classe (attribut_clé1 attribut_clé2 ... // attribut1, attribut2, ...)



La clé obligatoire peut être
composée de plusieurs attributs



Les autres attributs
de la classe

VÉHICULE (NoChâssis // NoVéhicule, Catégorie, Marque, NoMoteur, DateCirculation)

CHAUFFEUR (NoAVS // Nom, Prénom, Adresse, NumTél, DateNaissance, Sexe, EtatCivil)

SALLE-DE-COURS (NomBâtiment NoSalle // TypeSalle, Capacité, VidéoProjecteur, ...)

Identifiant / Clé

Un ensemble d'attributs K de la classe CL est un **identifiant (clé)** de CL si

- tous les attributs de K sont **monovalués**, et **permanents**;
- deux objets $o1$ et $o2$ de CL ne peuvent pas prendre les mêmes valeurs pour K ;
- K est **minimal** : il n'existe aucun sous-ensemble de K vérifiant la propriété précédente.

VÉHICULE (NoChâssis= / NoVéhicule -= // Catégorie, Marque, NoMoteur, DateCirculation)

NoChâssis est un identifiant obligatoire.

NoVéhicule est aussi un identifiant de la classe VÉHICULE mais il n'est pas obligatoire.



Tout identifiant obligatoire est un identifiant !

Toute classe peut avoir plusieurs identifiants dont au moins un est obligatoire.

Identifiant / clé primaire

Une classe peut avoir plusieurs identifiants / clés et même plusieurs clés obligatoires.

- Au niveau conceptuel il est important d'identifier toutes les clés.
- Au niveau informatique, si la classe admet plusieurs clés obligatoires, il faut en choisir une, appelée **clé primaire**.

Notation formelle complète d'une classe :

Classe (clé primaire / clé / ... // autres attributs)

Représentation graphique des classes

Notation de diagramme de classes UML

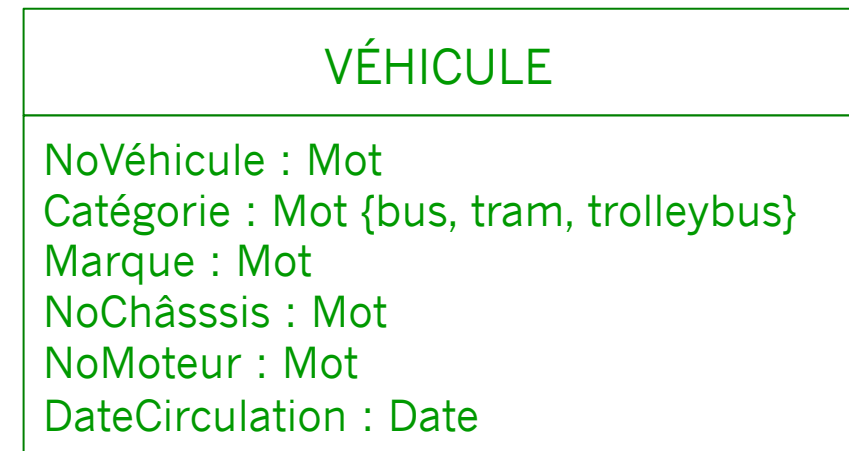
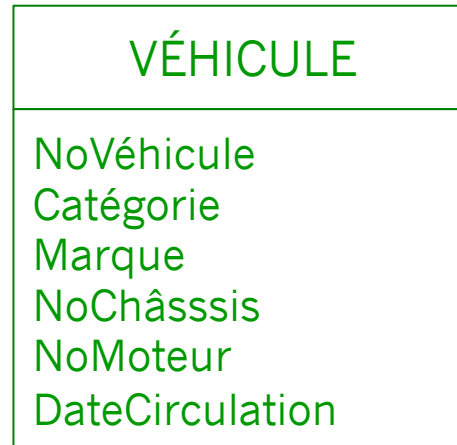
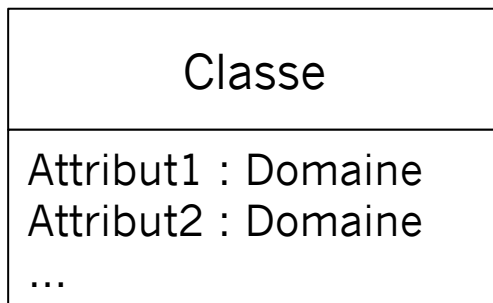
Représentation graphique de classes

Plusieurs niveaux de détails

Niveau 1 : le nom de la classe uniquement



Niveau 2 : le nom de la classe et les attributs avec ou sans domaines



Représentation graphique de classes

Plusieurs niveaux de détails

Niveau 3 : le nom de la classe, les attributs, la clé et les méthodes

Classe
Attributs clé Attribut1 : Domaine Attribut2 : Domaine
Méthode1 (signature) Méthode2 () ...

VÉHICULE
NoVéhicule NoChâssis (pk) Catégorie Marque NoMoteur DateCirculation
Créer () Supprimer () Modifier () Activer () Désactiver () Contrôler ()

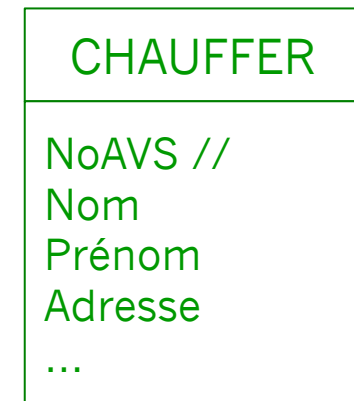
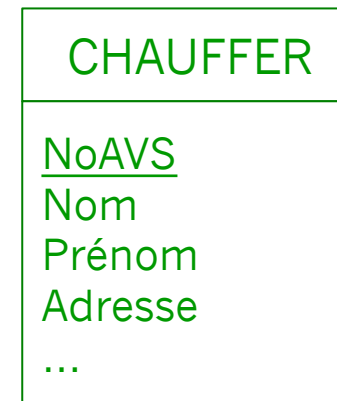
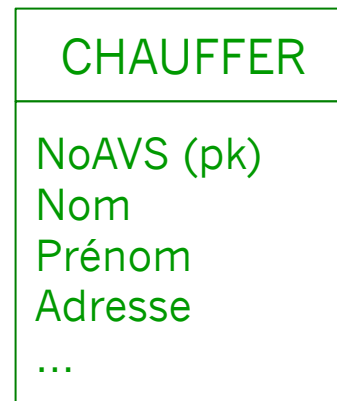
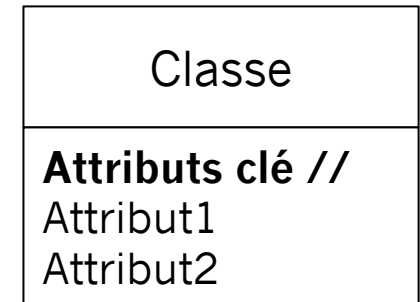
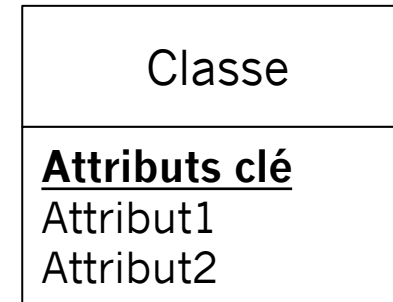
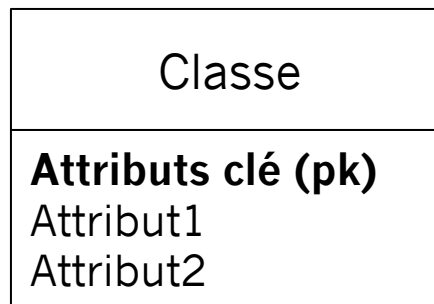
Pour désigner la clé
pk – primary key (clé primaire)

Représentation graphique de classes

Plusieurs niveaux de détails

Problème : certains outils de modélisation ne permettent pas de désigner la clé

Plusieurs manières
pour indiquer la clé



Forme normale de classe

Forme normale de classe

Dans un modèle relationnel, une classe est en **forme normale** si :

- elle a une clé,
- tous ses attributs sont atomiques (simples) et
- tous ses attributs sont monovalués.

La classe

CHAUFFEUR (NoAVS // Nom, Prénom, DateNaissance, Adresse)
est en forme normale.

Normalisation de classes

La classe

CHAUFFEUR (NoAVS // Nom, Prénom, DateNaissance, Adresse (Rue, Numéro, Code_postal, Ville), NumTél*, Salaire*)

n'est pas en forme normale.

Comment la normaliser?

→ Par transformation/extraction des attributs composites et multivalués.

Normalisation de classes

Que faire avec un attribut multivalué ?

Créer une nouvelle classe pour chaque attribut multivalué

... mais attention à la définition de la clé

TEL-CHAUFFEUR (NoAVS, NumTél // TypeTél)

SALAIRE-CHAUFFEUR (NoAVS, Salaire //)

Peut-on connaître le salaire de chaque moi? Non

Solution :

SALAIRE-CHAUFFEUR (NoAVS, Date // Salaire)

Normalisation de classes

Que faire avec un attribut composite ?

- Si l'attribut composite est monovalué, alors chaque composant devient un attribut à part entière.

CHAUFFEUR (NoAVS // Nom, Prénom, DateNaiss, Rue, Numéro, Code_postal, Ville)

- Si l'attribut composite est multivalué, il faut créer une autre classe

ADRESSE-CHAUFFEUR (NoAVS Rue Numéro Code_postal Ville // TypeAdresse)

! Les formalismes à utiliser dans ce cours

Modélisation
graphique

Avec ou sans les
domaines des attributs

Classe

Classe
Attributs clé // Attributs non-clé

VÉHICULE

VÉHICULE
NoChâssis / NoVéhicule // Catégorie Marque NoMoteur DateCirculation

Description formelle (modèle relationnel)

Classe (clé primaire / autre clé / ... // autres attributs)

VÉHICULE (NoChâssis / NoVéhicule // Catégorie, Marque, NoMoteur, DateCirculation)