

**UNIVERSITE ASSANE SECK DE
ZIGUINCHOR**

UFR Sciences et Technologies



Diagramme de Classes et Diagramme d'Objets

**L2 Informatique et Ingénierie
2021-2022**

© Bassirou DIENE

0

Diagramme de Classe (DCL)

1

Diagramme de classes (DCL) (1/2)

- ❑ Le DCL est Considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation
- ❑ Diagramme de cas d'utilisation montre le système du point de vue des acteurs,
- ❑ Le diagramme de classes montre la structure interne du système
- ❑ Le DCL fournit une représentation abstraite des objets du système qui doivent interagir pour réaliser les fonctions du système (les cas d'utilisation).
- ❑ Le DCL montre la structure statique du système
 - Pas de prise en compte du facteur temporel dans le comportement du système.

B. DIENE - 2021/2022

2

2

Diagramme de classes (DCL) (2/2)

- ❑ Le DCL permet de modéliser les classes du système et leurs relations, indépendamment d'un langage de programmation particulier.
- ❑ Le DCL décrit la vue statique du système en terme:
 - de **classes**: description formelle d'un ensemble d'objets ayant une sémantique et des propriétés communes.
 - de **relations entre classes**
- ❑ Pour créer un diagramme de classes, il faut :
 - Identifier et décrire les classes
 - Identifier et décrire les relations qui existent entre ces classes

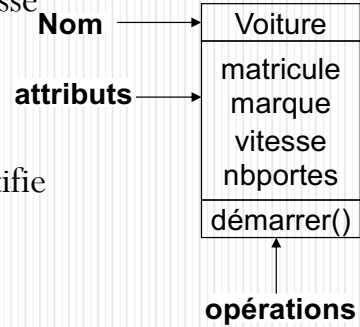
B. DIENE - 2021/2022

3

3

Représentation d'une classe

- ❑ Une classe est représentée à l'aide d'un rectangle comportant trois compartiments de base
- ❑ Le nom
 - Doit évoquer le concept qui décrit la classe
 - Commence par une majuscule
- ❑ Les attributs
 - Décrivent les caractéristiques des objets
 - Captent l'information qui décrit et identifie une instance spécifique d'une classe
- ❑ Les opérations
 - décrivent ce qu'une classe peut faire et quels services offrent-elles aux autres classes.



B. DIENE - 2021/2022

4

4

Attributs d'une classe

- ❑ Un *attribut* est une propriété commune à tous les objets d'une classe.
- ❑ Ils représentent les données encapsulées dans les objets d'une classe.
- ❑ Chaque attribut est défini par un **nom**, un **type de données**, une **visibilité** et peut être initialisé.
- ❑ Le nom de l'attribut doit être unique dans la classe
- ❑ La syntaxe de la déclaration d'un attribut est la suivante :
<visibilité> [/] <nom_attribut>:<Type>

B. DIENE - 2021/2022

5

5

Méthodes d'une classe

- ❑ **Méthode:** fonction assurée par une classe
 - Une méthode peut être de 2 types:
 - ✓ de **classe:** Elle ne peut manipuler que des attributs de la classe et ses propres paramètres.
 - ✓ **abstraite:** on connaît son entête mais pas la manière dont elle peut être réalisée (i.e. on connaît sa déclaration mais pas sa définition).
 - doit être redéfinie dans une autre classe avant d'être utilisée
 - **Unicité**
 - ✓ Une opération (même nom et mêmes types de paramètres) doit être unique dans une classe.

B. DIENE - 2021/2022

6

6

Visibilité des attributs et des opérations

- ❑ Il existe trois niveaux de visibilité:
 - **Public (+)** : attribut ou opération visible par tous.
 - **Protégé (#)** : attribut ou opération visible seulement à l'intérieur de la classe et par les sous classe de la classe.
 - **Privé (-)** : attribut ou opération visible à l'intérieur de la classe.
- ❑ Les attributs sont en général inaccessibles tandis que les opérations sont accessibles (principe de l'encapsulation)

ClasseX
-attribut_1: int -attribut_2: string
+set_attribut_1(int): void +get_attribut_1(): int +set_attribut_2(string): void +get_attribut_2(): string

B. DIENE - 2021/2022

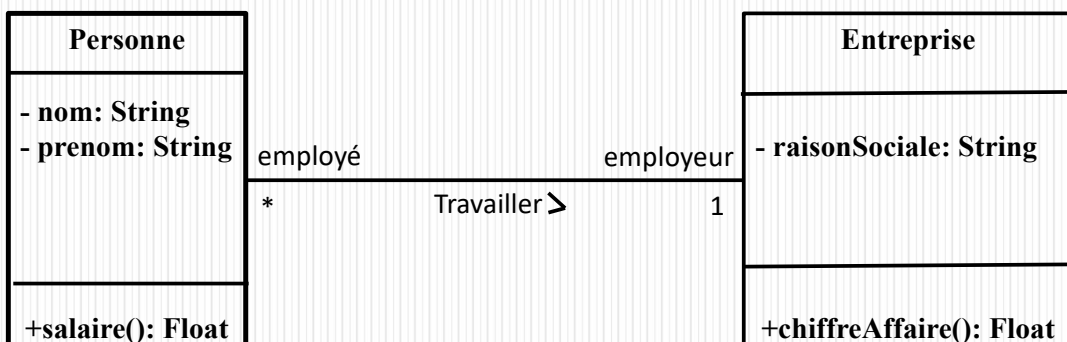
7

7

Associations entre classes

- ❑ Une association représente une relation sémantique entre les objets d'une classe.
- ❑ Une association est représentée par un trait plein entre les classes associées.
 - Elle est complétée par un nom
 - Avec une précision du sens de lecture en cas d'ambiguïté
 - Chaque extrémité de la relation indique le rôle de la classe dans l'association et précise le nombre d'objets de la classe qui y interviennent (multiplicité)

Associations entre classes



Types d'associations

- ❑ Les relations suivantes peuvent exister entre des classes :
 - **L'association simple:** lien entre instances de classes.
 - **La généralisation/spécialisation:** factorisation des propriétés communes à plusieurs classes.
 - **L'agrégation:** lien de type ensemble/élément
 - **La composition:** cas particulier de l'agrégation avec un couplage fort.

B. DIENE - 2021/2022

10

10

Multiplicité ou Cardinalité^{1/2})

- ❑ C'est un ensemble de valeurs indiquant le nombre possible d'instances de la classe destination du rôle qui peuvent être reliées à une instance de la classe origine du rôle.
- ❑ Elle peut être associée à une terminaison **d'association simple, d'agrégation ou de composition.**

1	Un et un seul
0..1	zéro ou un
n	n (entier naturel)
m..n	De m à n (entier naturel)
*	plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs

B. DIENE - 2021/2022

11

11

Multiplicité ou Cardinalité(2/2)

- ❑ **Association binaire:** la multiplicité sur la terminaison cible contraint le nombre d'objets de la classe cible pouvant être associés à un seul objet donné de la classe source (la classe de l'autre terminaison de l'association).
- ❑ **Association n-aire:** la multiplicité apparaissant sur le lien de chaque classe s'applique sur une instance de chacune des classes, à l'exclusion de la classe-association et de la classe considérée.

NB: Pour les habitués du modèle entité/relation, les multiplicités sont en UML « à l'envers » (par référence à Merise) pour les associations binaires et « à l'endroit » pour les n-aires avec $n > 2$.

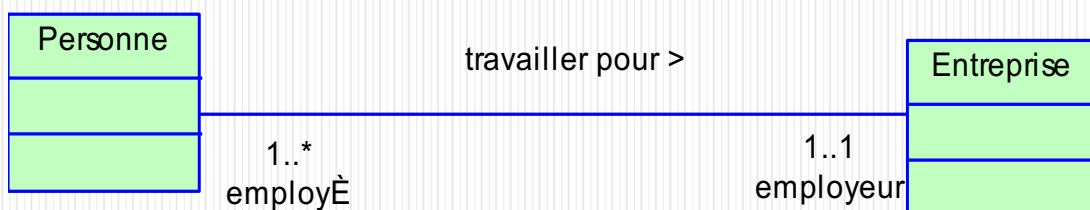
B. DIENE - 2021/2022

12

12

Documentation d'une association (1/2)

- ❑ Une association binaire est matérialisée par un trait plein entre les classes associées.
- ❑ Elle peut être ornée d'un **nom**, avec éventuellement une précision du sens de lecture $>$ ou $<$.
- ❑ Le **rôle** tenu par une classe vis-à-vis d'une association peut être précisé (indispensable pour les associations réflexives).



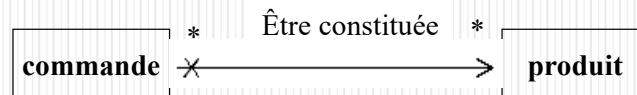
B. DIENE - 2021/2022

13

13

Documentation d'une association (2/2)

- ❑ **Navigabilité:** indique s'il est possible de traverser une association.
- ❑ La navigabilité est représentée graphiquement par une flèche du côté de la terminaison navigable et on empêche la navigabilité par une croix du côté de la terminaison non navigable



- la terminaison du côté de la classe *Commande* n'est pas navigable : cela signifie que les instances de la classe *Produit* ne sont pas constituées d'une liste d'objets du type *Commande*.
- Inversement, la terminaison du côté de la classe *Produit* est navigable : chaque objet commande est constitué d'une liste de produits.

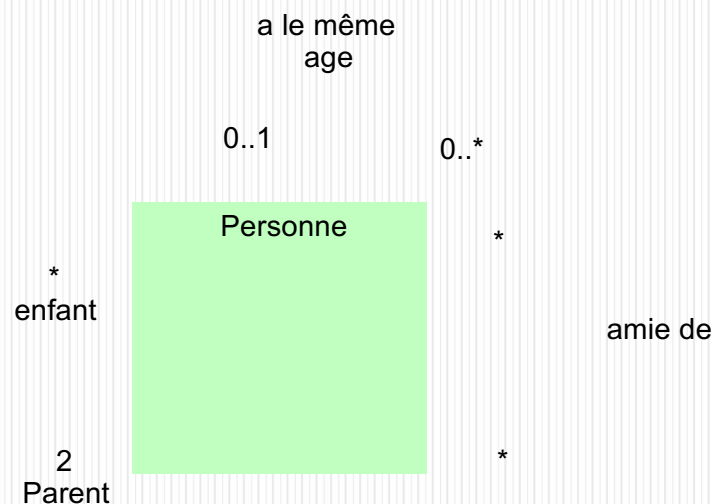
B. DIENE - 2021/2022

14

14

Association réflexive

- ❑ Une association est dite réflexive quand les 2 extrémités pointent vers le même classeur.
- ❑ Elle a pour principale fonction de structurer les objets d'une même classe.



B. DIENE - 2021/2022

15

15

Classe-association

- ❑ Une association peut avoir ses propres propriétés.
- ❑ Ses propriétés ne sont disponibles dans aucune des classes qu'elle lie.
- ❑ Une classe association possède à la fois les caractéristiques d'une association et celles d'une classe.
- ❑ Elle se connecte à deux ou plusieurs classes et peut posséder des attributs et des opérations.
- ❑ Elle est caractérisée par un trait discontinu entre la classe et l'association qu'elle représente

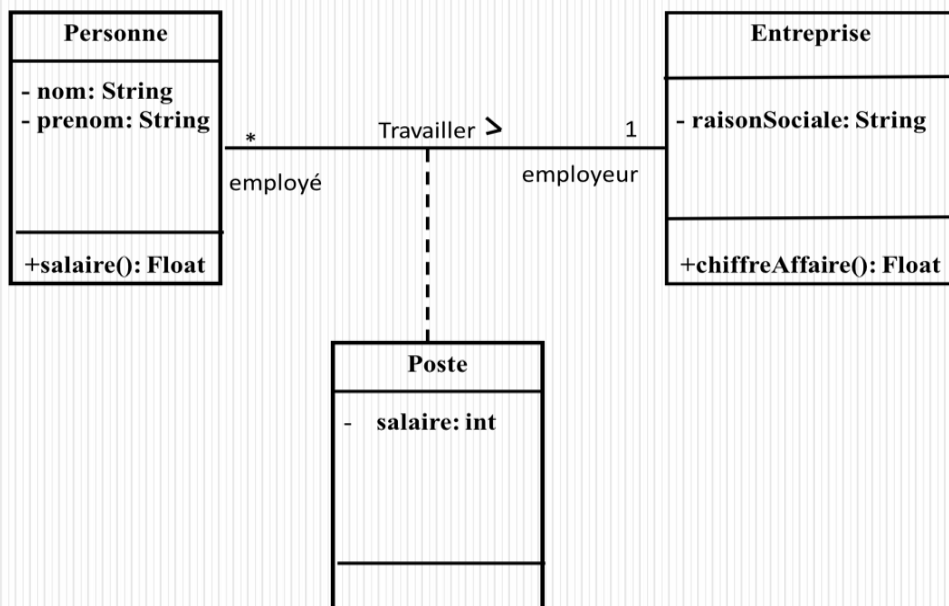
B. DIENE - 2021/2022

16

16

Classe-association

- ❑ **Exemple:** Les personnes qui travaillent dans une entreprise occupent un poste. Le salaire d'un employé est lié au poste.



B. DIENE - 2021/2022

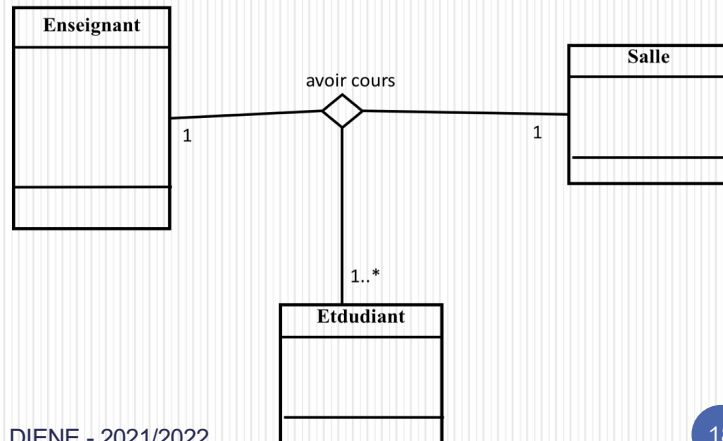
17

17

Association n-aire

- Une association peut relier plus de deux classes; elle est dite dans ce cas n-aire
 - 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.

Exemple
d'association
ternaire



B. DIENE - 2021/2022

18

18

Agrégation (1/2)

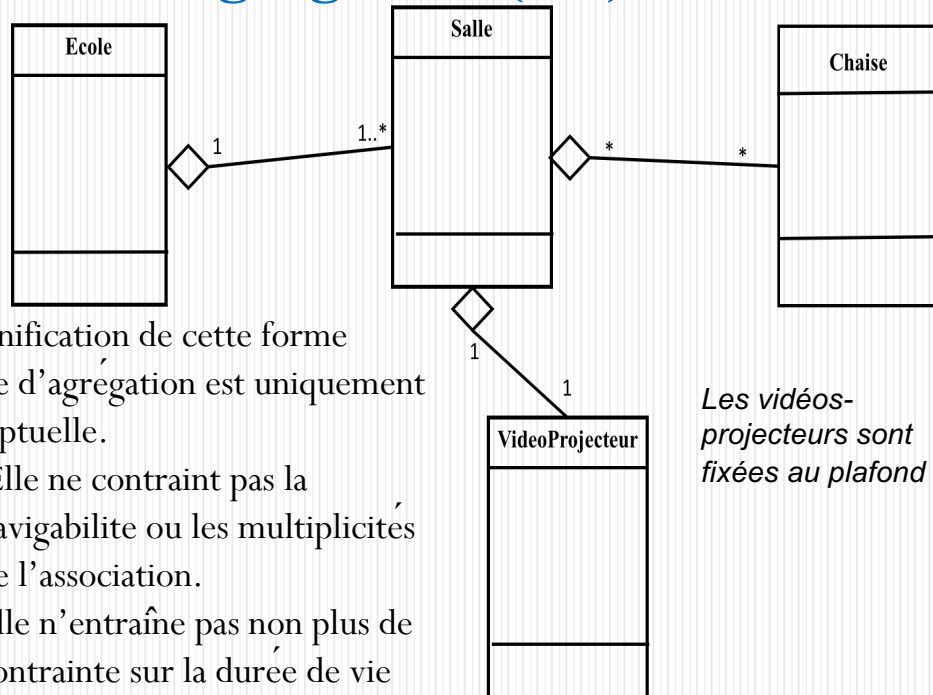
- C'est une **association asymétrique** dans laquelle **l'une** des **extrémités** joue un rôle **prédominant** par rapport à **l'autre**.
- Elle représente une relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble
- Elle peut notamment (mais pas nécessairement) exprimer :
 - qu'une classe (un "élément") fait partie d'une autre ("l'agrégat"),
 - qu'un changement d'état d'une classe, entraîne un changement d'état d'une autre,
 - qu'une action sur une classe, entraîne une action sur une autre.
- Pour la représenter, on ajoute un losange vide du côté de l'agrégat

B. DIENE - 2021/2022

19

19

Agrégation (2/2)



- La signification de cette forme simple d'agrégation est uniquement conceptuelle.
 - Elle ne contraint pas la navigabilité ou les multiplicités de l'association.
 - Elle n'entraîne pas non plus de contrainte sur la durée de vie des parties par rapport au tout.

Les vidéos-projecteurs sont fixées au plafond

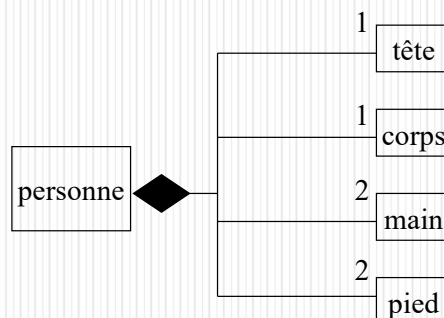
B. DIENE - 2021/2022

20

20

Composition (1/2)

- C'est un cas particulier de l'agrégation avec une contrainte de durée de vie entre la classe composite et la ou les classes composées.
- La destruction de l'agrégat implique automatiquement la destruction de tous ses composants.
- Elle implique une contrainte sur la valeur de la multiplicité du côté de la classe composite qui doit être 0 ou 1.



Règles obligatoires pour la composition

- La suppression du composite entraîne la suppression des composants
- Les attributs du composite sont utilisés dans les composants
- Un composant ne peut pas être en relation avec d'autres classes externes au composite.

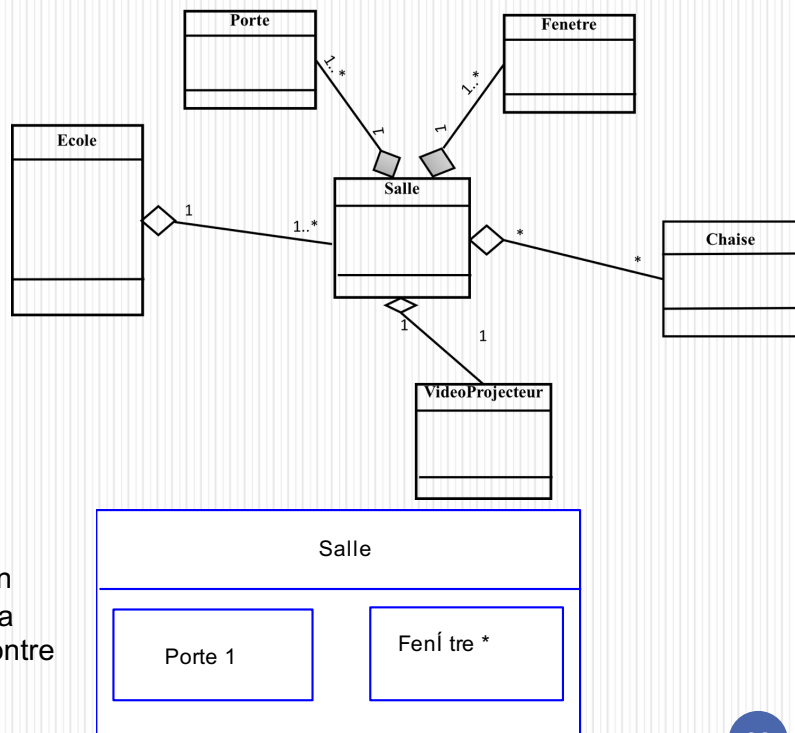
B. DIENE - 2021/2022

21

21

Composition (2/2)

On peut compléter l'exemple sur l'agrégation en indiquant qu'une salle est composée d'une porte et de plusieurs fenêtres



NB: La composition étant structurelle, on peut alors adopter la représentation ci-contre

B. DIENE - 2021/2022

22

22

Généralisation et héritage

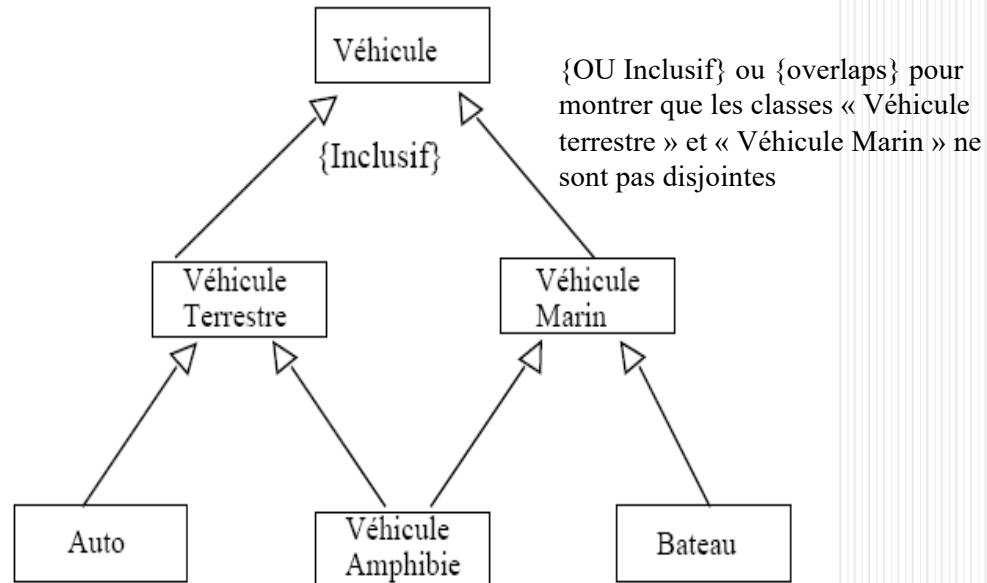
- ❑ La **généralisation** est la relation qui existe entre une classe générale (super classe) et plusieurs autres classes (sous-classes) plus spécifiques.
- ❑ Les attributs et les opérations d'une super classe sont transmis aux sous-classes par **héritage**.
- ❑ Une sous-classe possède toutes les propriétés de la super classe, mais elle ne peut accéder aux **propriétés privées** de celle-ci.
- ❑ Toutes les associations de la super classe s'appliquent aux sous-classes.
- ❑ Une instance d'une sous-classe peut être utilisée partout où une instance de sa super classe est attendue.
- ❑ Une classe peut avoir plusieurs parents, on parle alors **d'héritage multiple**.

B. DIENE - 2021/2022

23

23

Héritage multiple



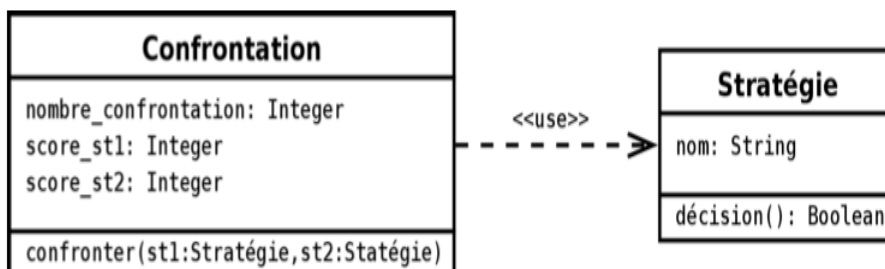
B. DIENE - 2021/2022

24

24

Relation de dépendance

- Relation unidirectionnelle exprimant une dépendance sémantique entre éléments du modèle.
 - Indique que la modification de la cible implique le changement de la source.
 - Représentée par un trait discontinu orienté + souvent un stéréotype pour mieux expliciter le lien sémantique.

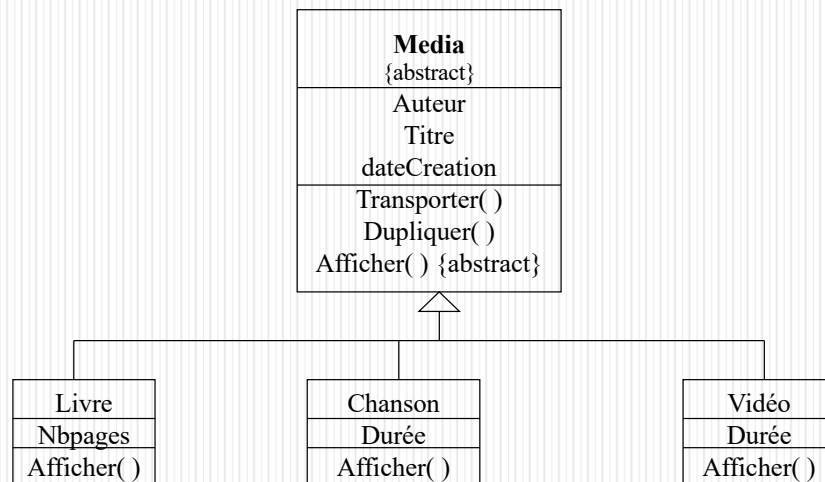


25

25

Classe abstraite

- ❑ Une classe est abstraite si elle dispose d'un attribut ou d'une méthode abstraite
- ❑ Une classe abstraite n'est pas instanciable.

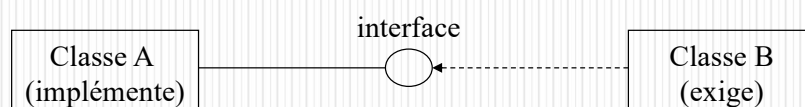


26

26

Interface (1/3)

- ❑ Elle décrit le comportement visible d'une classe.
- ❑ Elle n'est pas une classe réelle mais une liste de services accessibles par les autres classes.
- ❑ Le comportement visible d'une interface est décrit par des **opérations abstraites** dont la **visibilité est publique**
- ❑ Une interface est représentée par un petit **cercle ayant un nom**
- ❑ Une classe qui utilise l'interface (implémentée par une autre classe) est connectée via une relation de dépendance vers le cercle représentatif de cette interface.



B. DIENE - 2021/2022

27

27

Interface (2/3)

- ❑ Pour montrer les opérations dans une interface, on la spécifie comme une classe avec le stéréotype «*interface*».
- ❑ Une interface doit être réalisée par au moins une classe. La relation est représentée par un trait discontinu terminé par une flèche triangulaire stéréotypée par «*realize*».
- ❑ Une classe (classe cliente de l'interface) peut dépendre d'une interface (interface requise). La liaison sera représentée par une relation de dépendance et stéréotypée par «*use*».

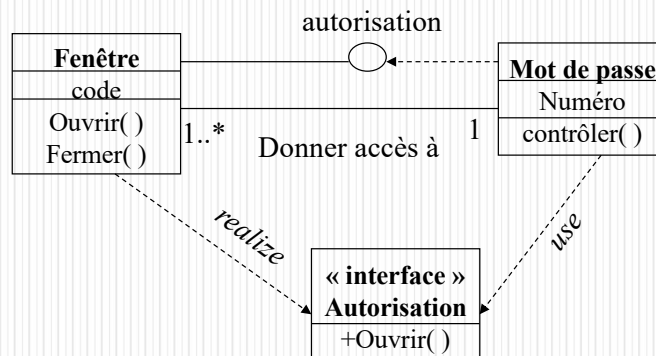
B. DIENE - 2021/2022

28

28

Interface (3/3)

- ❑ Exemple:



B. DIENE - 2021/2022

29

29

Élaboration d'un DCL

- ❑ **Trouver les classes du domaine étudié:** Les classes correspondent généralement à des concepts ou des substantifs du domaine.
- ❑ **Trouver les associations entre classes:** les associations correspondent souvent à des verbes, ou des constructions verbales, mettant en relation plusieurs classes, comme « est composé de », « pilote », « travaille pour ».

Attention: méfiez vous de certains attributs qui sont en réalité des relations entre classes.

B. DIENE - 2021/2022

30

30

Élaboration d'un DCL

- ❑ **Trouver les attributs des classes:** les attributs correspondent souvent à des substantifs, ou des groupes nominaux. N'espérez pas trouver tous les attributs dès la construction du diagramme de classes.
- ❑ **Organisation et simplification du modèle:** éliminer les classes redondantes en utilisant l'héritage
- ❑ **Vérifier les chemins d'accès aux classes.**
- ❑ **Itérer et raffiner le modèle.** Un modèle est rarement correct dès sa première construction. La modélisation objet est un processus non pas linéaire mais itératif

B. DIENE - 2021/2022

31

31

Diagramme d'Objet (DOB)

32

Diagramme d'objets (1/3)

Unified Modeling Language

- Un diagramme d'objets représente des objets (i.e. instances de classes) et leurs liens (i.e. instances de relations) pour donner une vue de l'état du système à un instant donné
- Un diagramme d'objets permet selon les situations:
 - d'illustrer le modèle de classes (en montrant un exemple qui explique le modèle),
 - de préciser certains aspects du système (en mettant en évidence des détails imperceptibles dans le diagramme de classes),
 - d'exprimer une exception (en modélisant des cas particuliers, des connaissances non généralisables . . .),
 - de prendre une image (snapshot) d'un système à un moment donné.

Objet nommé d'une classe anonyme

nom de l'objet

Objet nommé de la classe

nom de l'objet:Classe

Objet anonyme

:Classe

Groupe d'objets

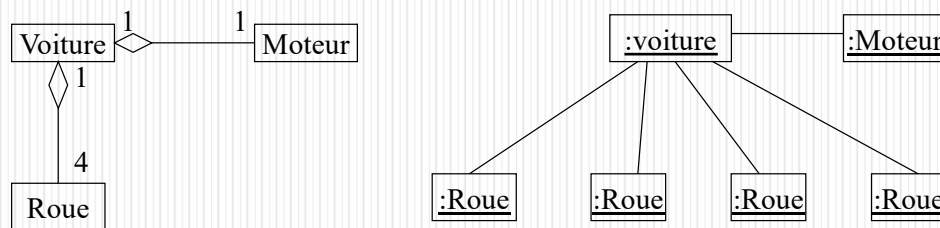
:Classe

33

33

Diagramme d'objets (2/3)

- ❑ Dans un DOB, le compartiment des opérations n'est pas utile, mais les attributs doivent recevoir des valeurs.
- ❑ Dans un DOB, les relations du diagramme de classes deviennent des liens et s'il y a un nom, il est souligné
- ❑ Naturellement, on ne représente pas les multiplicités des extrémités des liens (elles valent toujours 1).



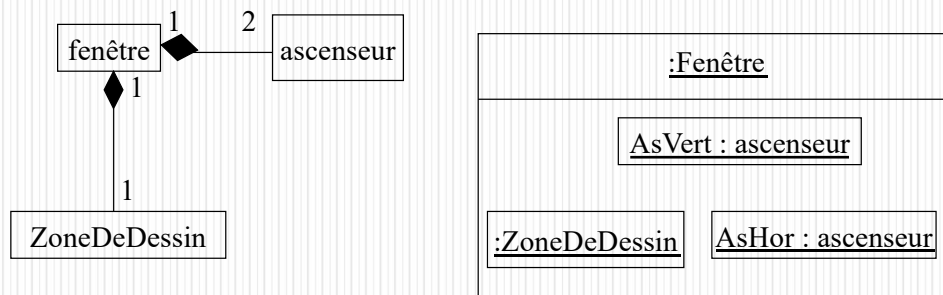
B. DIENE - 2021/2022

34

34

Diagramme d'objets (3/3)

- ❑ Il est possible de représenter les objets composés de sous objets au moyen d'un objet composite.
- ❑ L'objet composite se présente comme un objet habituel avec la différence que les attributs sont remplacés par les sous objets.
- ❑ Exemple



B. DIENE - 2021/2022

35

35

FIN

