

Chapitre : Diagramme de classe

Classe, responsabilité, association, multiplicité, agrégation, composition, attribut, attribut dérivé, attribut de classe, classe d'association, qualificatif, opération, opération de classe, classification, généralisation, spécialisation, classe abstraite, principe de substitution, généralisation multiple, contrainte

π

L2GLSI-ISIMM-23/24-M.GZARA

21

21

π

Notion de l'objet

- › Un objet est une entité aux frontières bien définies
 - Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser:
 - › La chaise de mon bureau, ma voiture, commande n°12, client Mourad
- › Un objet a une identité :
 - Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.
- › Un objet encapsule un état
- › Un objet encapsule un comportement

L2GLSI-ISIMM-23/24-M.GZARA

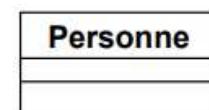
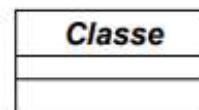
22

22

π

Notion de classe

- › Une classe représente la description abstraite d'un ensemble d'objets (abstraction de l'objet):
 - Mêmes propriétés (attributs)
 - Même comportement (opérations)
 - Une sémantique (domaine de définition)
 - Classe Voiture, classe Elève, classe Personne
- › Un objet est une instance de classe
- › Les objets d'une même classe possèdent les mêmes caractéristiques (attributs).
- › Notation UML:



L2GLSI-ISIMM-23/24-M.GZARA

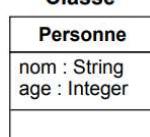
23

23

 π

Attribut

- › Un attribut est un type d'information contenu dans une classe
 - Attributs élémentaires ou atomiques : nom, âge
 - Attributs structurés: adresse
 - Attributs multivalués : une personne peut avoir plusieurs numéros de téléphones
 - Exemple :
 - › Classe Voiture
 - › Attributs: vitesse courante, numéro de l'immatriculation, nombre de cylindre, ...
- › Notation UML:



L2GLSI-ISIMM-23/24-M.GZARA

24

24

π

Attribut

- › **Visibilité/Nom Attribut : type [=valeur initiale {propriété}]**
 - Visibilité (-, +, #)
 - Nom d'attribut: nom unique dans sa classe
 - Type : type primitif
 - Valeur initiale facultative
 - Attribut dérivé : calculé à partir d'autres attributs de la classe (/nom attribut dérivé)
 - {propriétés}: valeurs marquées facultatives (ex: <<interdit>> pour mise à jour interdite)
 - Attribut peut avoir des valeurs multiples (diplômes[5])

 π

Opération

- › Décrit le comportement d'un objet :
 - Un élément de comportement (un service) contenu dans une classe
- › Fonction applicable aux objets d'une classe
- › Une méthode est l'implémentation d'une opération
- › Formalisme:
 - Uniquement le nom de l'opération
 - (ou)
 - Nom de l'opération, liste de paramètres, type de résultat

π

Opération

- › Visibilité Nom opération(paramètres) [:[type résultat] {propriétés}]:
 - Visibilité (-, +, #)
 - Nom opération: verbe représentant l'action à réaliser
 - Paramètres : nom du paramètre, type, valeur par défaut
 - Type résultat: type (s) de valeurs renvoyées
 - {propriétés} : valeurs facultatives applicables

 π

Visibilité des attributs et des opérations

- › Public (+) : attribut ou opération visible partout
- › 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

π

Attribut ou opération de niveau classe

- › Attribut qui est une constante pour toutes les instances d'une classe
- › Opération d'une classe abstraite
- › Opération <<créer>> qui peut être définie au niveau de la classe et applicable à la classe elle-même
- › Formalisme:
 - Soulignement de l'attribut ou de l'opération

 π

Classe : Formalisme général

- › Un rectangle comportant plusieurs compartiments:
 - Les trois compartiments de base
 - › Désignation de la classe
 - › Description des attributs
 - › Description des opérations
 - Deux autres compartiemnts :
 - › Description des responsabilités de la classe
 - › Description des exceptions traitées par la classe

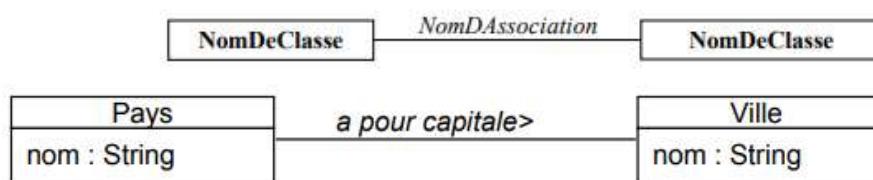
NomDeClasse
nomAttribut1 : typeDeDonnées1 = Valeur parDéfaut1
nomAttribut2 : typeDeDonnées2 = Valeur parDéfaut2
...
nomOpération1 (listeArguments1) : TypeDuRésultat1
nomOpération2 (listeArguments2) : TypeDuRésultat2
...

☞ Il est possible de manipuler les classes en limitant le niveau de description à un nombre réduit de compartiments selon les objectifs de modélisation

π

Association

- › Une association représente une relation sémantique durable entre deux classes
 - Exemple:
 - › Une personne peut posséder plusieurs voitures
 - › Une voiture est possédée par une seule personne
 - Une association est bidirectionnelle
- › Notation UML:



L2GLSI-ISIMM-23/24-M.GZARA

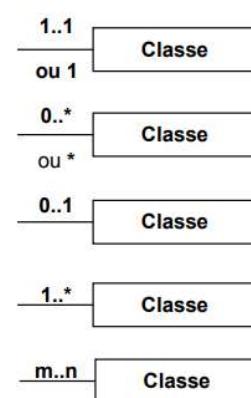
31

31

 π

Association : multiplicité et contraintes

- › Multiplicité : contrainte valable durant toute l'existence des objets
 - Une et une seule
 - Plusieurs (0 ou plus)
 - Optionnelle (0 ou 1)
 - 1 ou plus
 - De m à n
- › Autres contraintes sur association
 - {ordered}
 - {subset}
 - Contrainte de partition {xor}



L2GLSI-ISIMM-23/24-M.GZARA

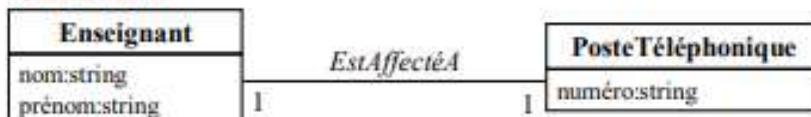
32

32

π

Association : multiplicité et contraintes

« un-à-un »



« zéro-à-un »



« plusieurs-à-plusieurs »



33

33

 π

Rôle

- › Possibilité de nommer les extrémités d'association
- › Indispensable pour les associations entre objets de même classe

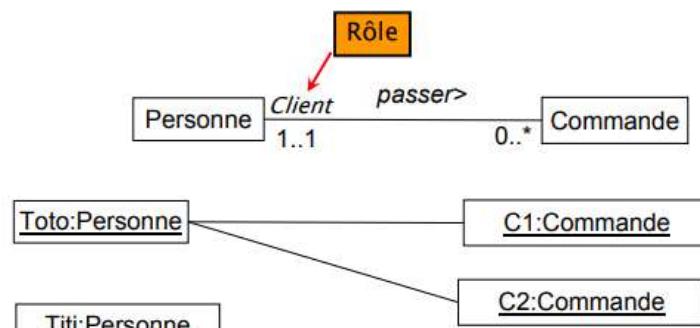
34

π

Association : multiplicité et contraintes

- › Une personne peut passer plusieurs commandes
- › Une commande est toujours passée par une et une seule personne

- L'extrémité d'une association est appelée rôle.
- Le rôle décrit comment une classe voit l'autre classe au travers d'une association



L2GLSI-ISIMM-23/24-M.GZARA

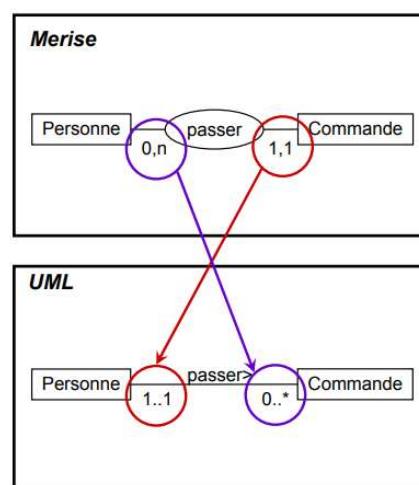
35

35

 π

Association : multiplicité et contraintes

- › Cardinalités (Merise, modèle entité association)
- › Multiplicité (UML)



36

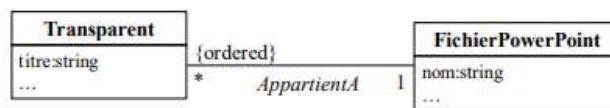
36

π

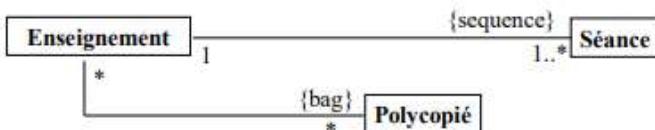
Association : multiplicité et contraintes

› Ordonnancement, bags et séquences

- Ordonnancement des objets situés à l'extrême d'une association « plusieurs »



- Bag (sac) : collection non ordonnée avec autorisation de doublons
- Séquence : collection ordonnée avec autorisation de doublons



L2GLSI-ISIMM-23/24-M.GZARA

37

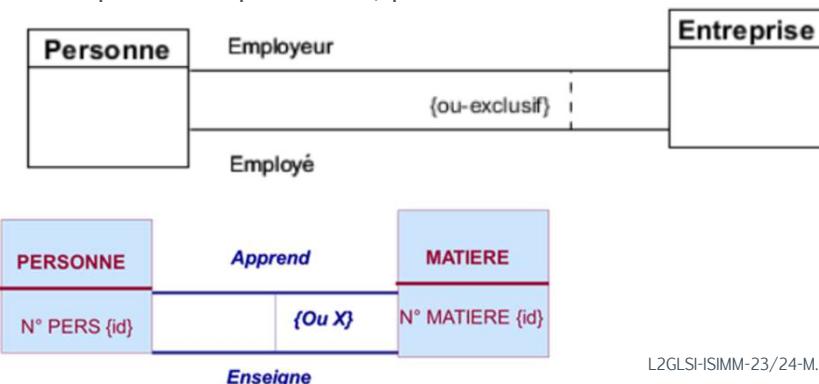
37

 π

Association : multiplicité et contraintes

› Les contraintes d'exclusion:

- Ce type de contrainte permet de modéliser le cas où pour une instance donnée d'une classe, une seule association, parmi plusieurs possibles, peut être valide à un instant donné



L2GLSI-ISIMM-23/24-M.GZARA

38

38

π

Association : multiplicité et contraintes

- › La contrainte {sous-ensemble} indique qu'une collection est incluse dans une autre collection
- › Des contraintes spécifiques au moyen du langage OCL (Object Constraint Language)

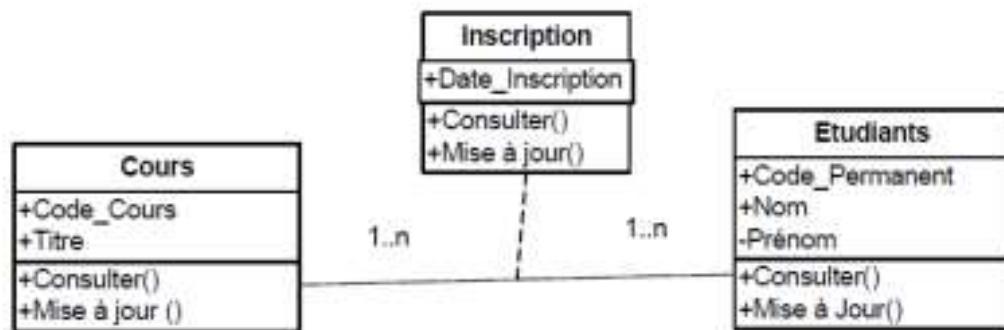
 π

Classe d'association

- › Une association ne peut posséder d'attributs, à l'exception du cas très particulier des associations qualifiées.
- › On peut définir une classe association, c'est-à-dire une association qui est également une classe.
- › C'est une classe qui réalise la navigation entre les instances d'autres classes.
- › Elle sert à connecter les classes entre elles

π

Classe d'association



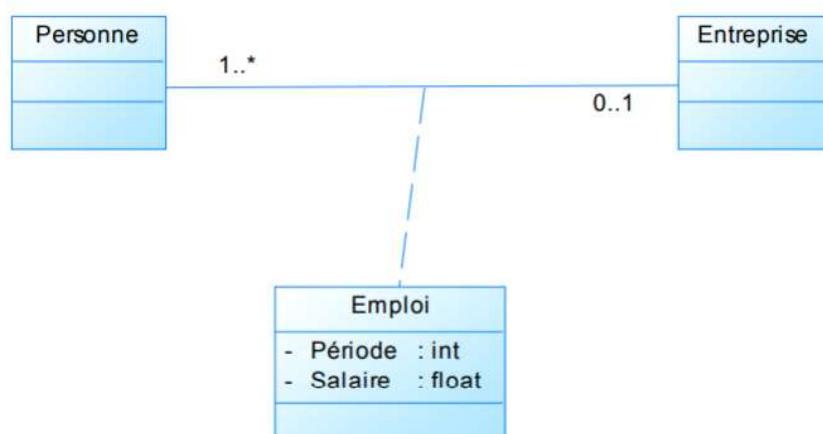
L2GLSI-ISIMM-23/24-M.GZARA

41

41

 π

Classe d'association



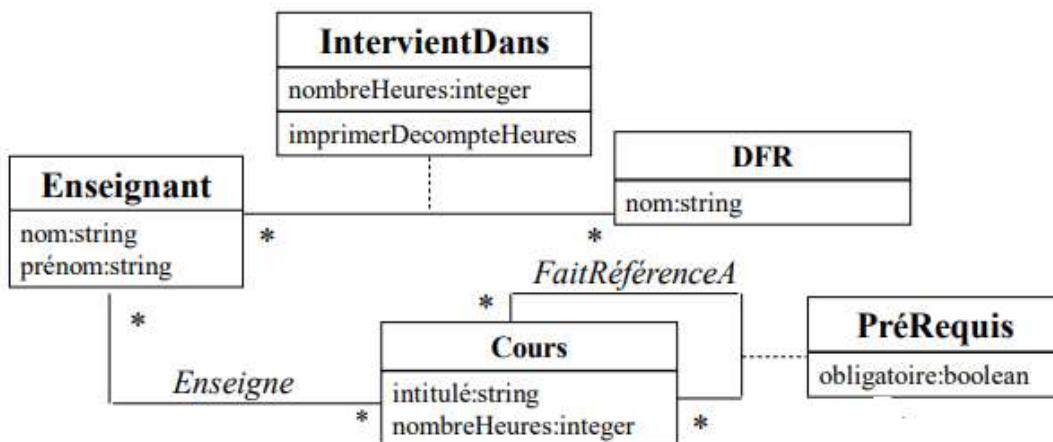
L2GLSI-ISIMM-23/24-M.GZARA

42

42

π

Classe d'association



L2GLSI-ISIMM-23/24-M.GZARA

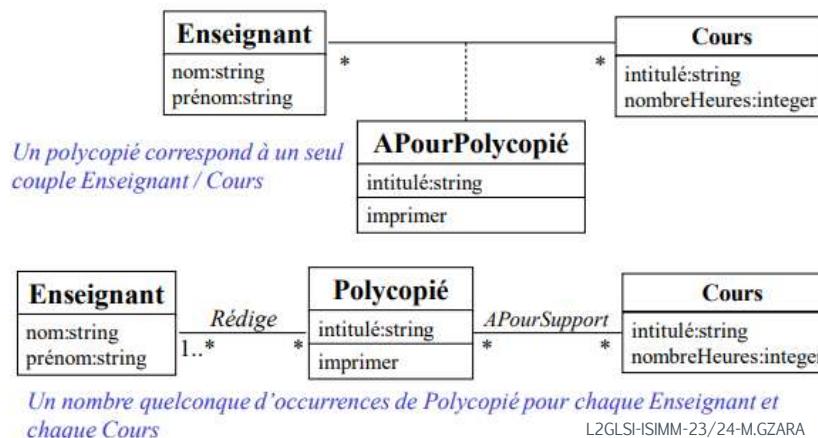
43

43

 π

Classe d'association

- › Ne pas confondre classe-association et association promue au rang de classe



L2GLSI-ISIMM-23/24-M.GZARA

44

44

π

Association qualifiée

- › Qualificateur :
 - Attribut permettant de distinguer les objets situés à l'extrémité de multiplicité « plusieurs » d'une association
 - Attribut réduisant la multiplicité « plusieurs » à « un »
- › Association qualifiée : association contenant un ou plusieurs attributs qualificateurs

L2GLSI-ISIMM-23/24-M.GZARA

45

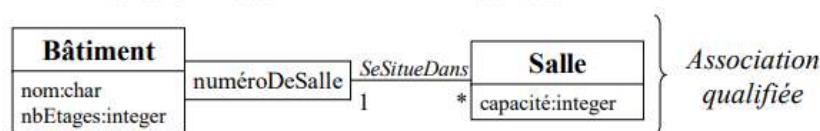
45

 π

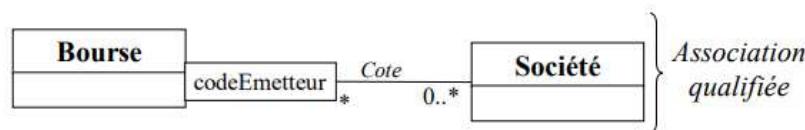
Association qualifiée

Un *numéro* de *Salle* permet d'identifier une salle unique dans un *Bâtiment* donné

Un *numéro* de *Salle* est relatif à un *Bâtiment*



Une société est cotée en bourse et a un code émetteur par Bourse



L2GLSI-ISIMM-23/24-M.GZARA

46

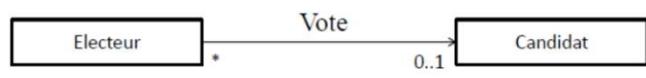
46

π

Les diagrammes de classes : Navigabilité

- › La navigabilité indique s'il est possible de traverser une association.
- › Par défaut, une association est navigable dans les deux sens **Une Association à navigabilité restreinte :**
 - Indique que les instances d'une classe ne "connaissent" pas les instances d'une autre.

Exemple : Electeur, Candidat



L2GLSI-ISIMM-23/24-M.GZARA

47

47

 π

Esempio : Système de fichier

- › Spécification :
 - Un utilisateur possède au moins un répertoire
 - Un répertoire appartient à un et un seul utilisateur
 - Un répertoire peut contenir d'autres répertoires
 - Un utilisateur peut accéder à au moins un répertoire
 - Un répertoire peut être accédé par au moins un utilisateur
 - Réaliser le diagramme de classes correspondant



L2GLSI-ISIMM-23/24-M.GZARA

48

48

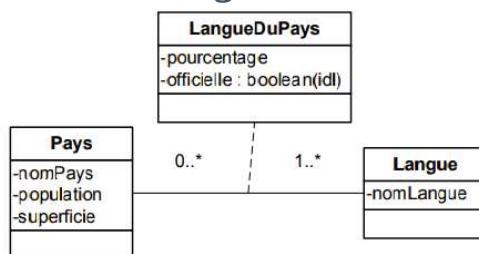
π

Exemple : pays et langues

› Spécifications :

- Un pays a un nom, un nombre d'habitants et une superficie
- Dans ce pays, certaines langues sont parlées
- Un pourcentage de la population parle une langue ;
- Une langue peut être officielle dans un pays

› Réaliser le diagramme de classes correspondant



L2GLSI-ISIMM-23/24-M.GZARA

49

49

 π

Exemple : gestion salles d'attente clinique

› Spécifications:

- Chaque salle a une capacité d'accueil limitée.
- Les salles portent un nom et sont situées à un emplacement précis de la clinique.
- Les patients sont admis normalement ou en urgence.
- Les patients sont appelés prioritairement selon l'ordre d'arrivée
- Un patient est appelé est reçu et soigné par le docteur.
- Les soins ne sont pas interrompus par l'arrivée d'un patient en urgence.

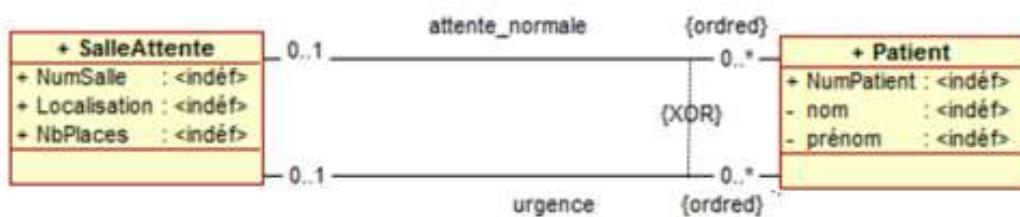
L2GLSI-ISIMM-23/24-M.GZARA

50

50

π

Exemple : gestion salles d'attente clinique



L'expression des contraintes est donnée en langage naturel ou en OCL pour enrichir encore la sémantique exprimée dans le diagramme.

L2GLSI-ISIMM-23/24-M.GZARA

51

51

 π

Agrégation et composition

- › Une agrégation exprime une relation de contenance
- › Association non symétrique
- › Une agrégation n'a pas besoin d'être nommée
 - Implicitement : <<contient>>, <<est composé de>>
 - Classe agrégat (composé), classes agrégée (composant)



Un répertoire contient des fichiers

L2GLSI-ISIMM-23/24-M.GZARA

52

52

π

Exemple : personnes et commune

› Spécification :

- Une personne peut travailler pour plusieurs sociétés
- Une société a un et un seul PDG
- Une personne peut être PDG de plusieurs sociétés
- Une personne peut posséder plusieurs voitures
- Une voiture a un seul propriétaire
- Une personne peut avoir des enfants
- Une personne peut être le maire d'une commune
- Une personne peut être le député d'une circonscription
- Une commune appartient à une circonscription

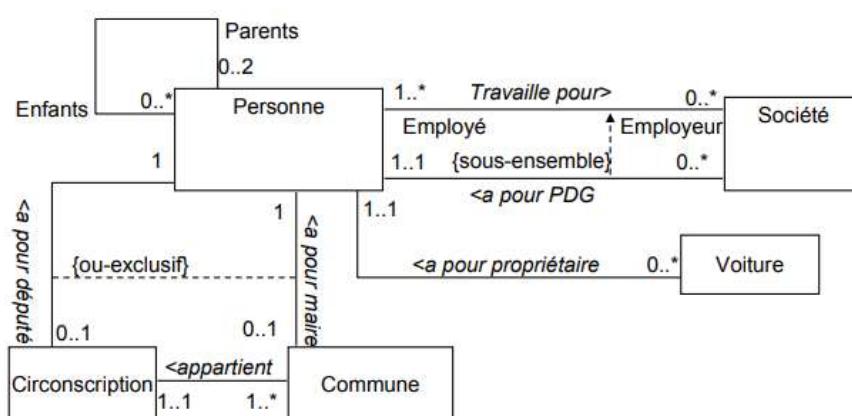
L2GLSI-ISIMM-23/24-M.GZARA

53

53

 π

Exemple : personnes et commune



L2GLSI-ISIMM-23/24-M.GZARA

54

54

π

Agrégation et composition

- › Une composition est une agrégation plus forte:
 - Un élément ne peut appartenir qu'à un seul agrégat composite
 - › Agrégation non partagée
 - La destruction de l'agrégat composite entraîne la destruction de tous ses éléments
 - › Le composite est responsable du cycle de vie des parties,



Une pièce contient des murs

L2GLSI-ISIMM-23/24-M.GZARA

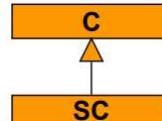
55

55

 π

Généralisation, super-classe, sous-classe

- › Relation de généralisation:
 - Relier une super-classe (classe plus générale) à une ou plusieurs autres classes plus spécialisées (sous-classes)
 - Les sous-classes <>hérivent<> des propriétés de leur super-classe
 - Les sous-classes peuvent comporter des propriétés spécifiques
- › Exemple :
 - Classe : moyen de transport : désignation, âge, vitesse limite, prix HT, ...
 - Voiture: numéro d'immatriculation, nombre de portes, couleur
 - Bateau : nombre de voiles
 - Avion: altitude,



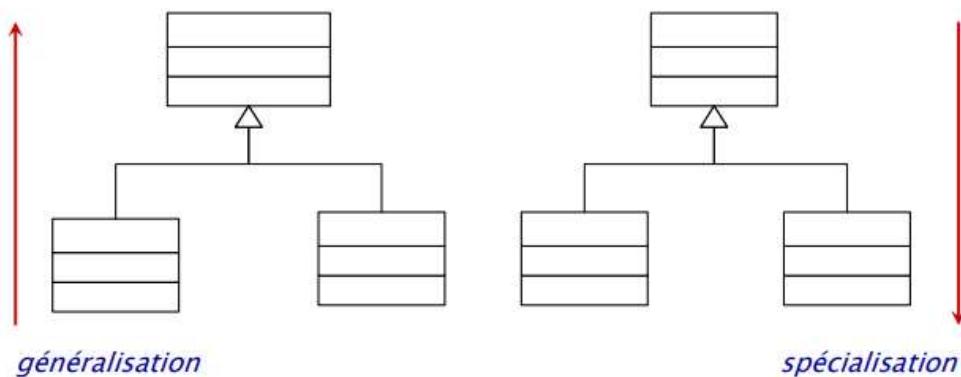
L2GLSI-ISIMM-23/24-M.GZARA

56

56

π

Généralisation, super-classe, sous-classe



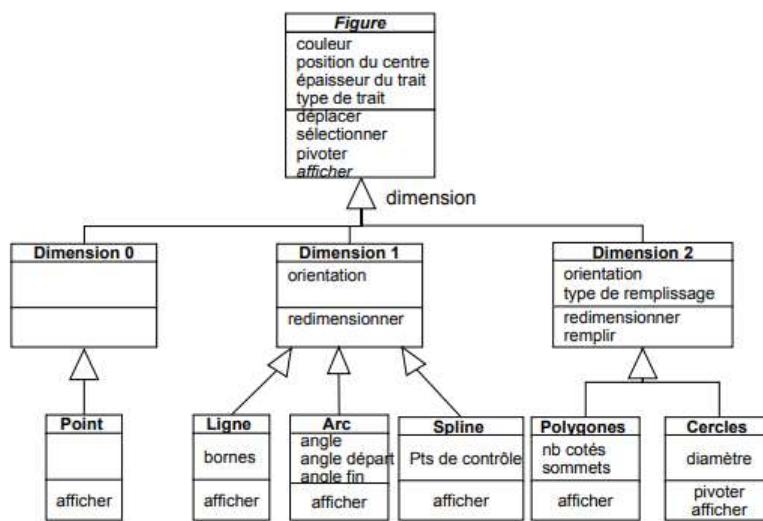
L2GLSI-ISIMM-23/24-M.GZARA

57

57

 π

Généralisation, super-classe, sous-classe



58

58

π

Association généralisation: contrainte

- › Une généralisation incomplète indique qu'il existe d'autres sous-classes qui peuvent être introduites dans la généralisation.
- › Une généralisation complète indique que l'ensemble des sous-classes a été répertorié
- › Par défaut, la généralisation est dite complète et disjointe, i.e. une instance est au plus instance d'une seule des sous-classes
- › La contrainte overlapping indique qu'une instance de la super-classe est l'instance de deux ou plusieurs sousclasses.

 π

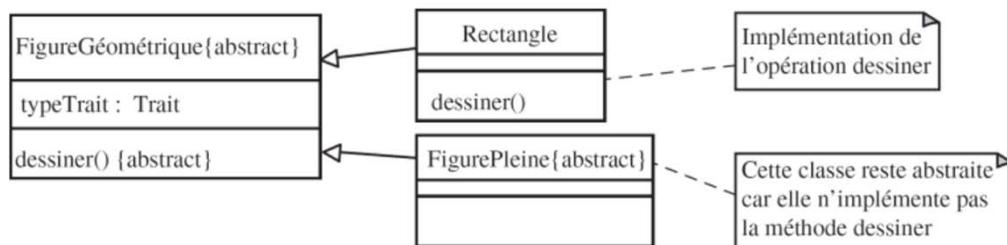
Encapsulation

- › L'encapsulation est un principe de conception consistant à protéger le cœur d'un système des accès intempestifs venant de l'extérieur.
- › En UML, utilisation de modificateurs d'accès sur les attributs ou les classes :
 - Public ou << + >> : propriété ou classe visible partout
 - Protected ou <<#>> : propriété ou classe visible dans la classe et par tous ses descendants.
 - Private ou <<- >> : propriété ou classe visible uniquement dans la classe Package, ou <<~>> : propriété ou classe visible uniquement dans le paquetage
 - Il n'y a pas de visibilité par défaut .

π

Classe abstraite

- Une méthode est dite abstraite lorsqu'on connaît son entête mais pas la manière dont elle peut être réalisée.
- Il appartient aux classes enfant de définir les méthodes abstraites.
- Une classe est dite abstraite lorsqu'elle définit au moins une méthode abstraite ou lorsqu'une classe parent contient une méthode abstraite non encore réalisée.



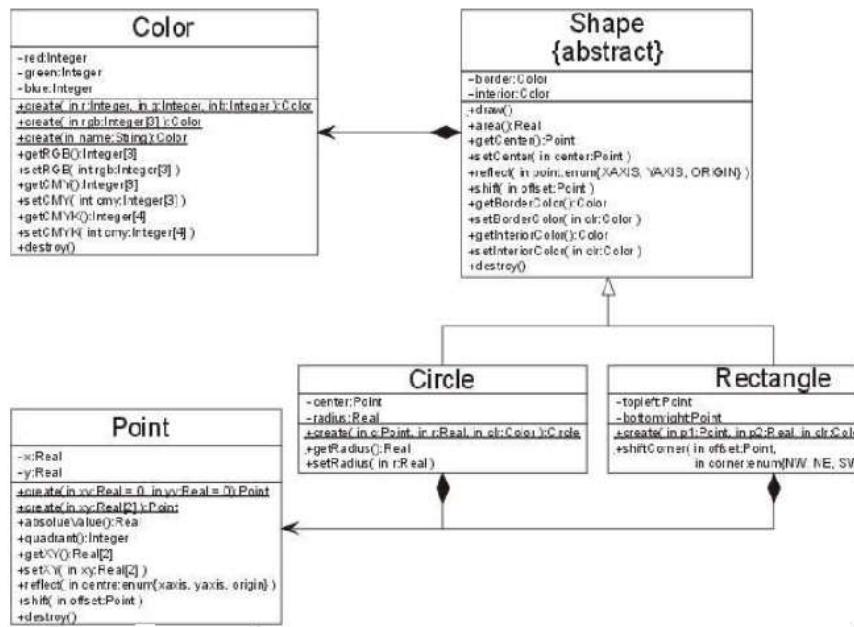
L2023-2024 - 24-01-2024

01

61

 π

Classe abstraite



62

62

π

Package

- › Mécanisme général de regroupement d'éléments en UML
 - Regrouper des classes et des associations
 - Les packages sont des espaces de nommage:
 - › Deux éléments ne peuvent pas porter le même nom au sein du même package
 - › Deux éléments appartenant à des packages différents peuvent porter le même nom
- › Structuration d'un modèle en packages:
 - Cohérence :
 - › Regrouper les classes proches de point de vue sémantique
 - Independence :
 - › Minimiser les relations entre packages, soit, les relations entre classes de packages différents

Critère intéressant: Évaluer les durées de vie des instances de concepts et rechercher l'homogénéité

L2GLSI-ISIMM-23/24-M.GZARA

63

63

 π

Interface



- › Le rôle d'une interface est de regrouper un ensemble d'opérations assurant un service cohérent offert par un classeur et une classe en particulier.
- › Une interface est définie comme une classe, avec les mêmes compartiments.
- › On ajoute le stéréotype <<interface>> avant le nom de l'interface.
- › On utilise une relation de type réalisation entre une interface et une classe qui l'implémente.
- › Les classes implémentant une interface doivent implémenter toutes les opérations décrites dans l'interface

L2GLSI-ISIMM-23/24-M.GZARA

64

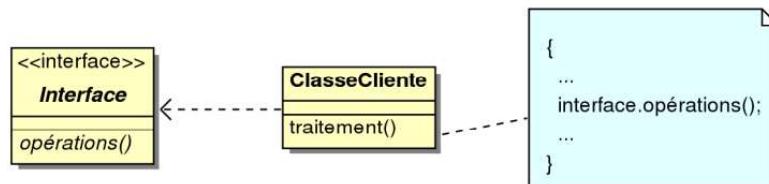
64

π

Interface

› Classe cliente d'une interface:

- Quand une classe dépend d'une interface (interface requise) pour réaliser ses opérations, elle est dite classe cliente de l'interface
- On utilise une relation de dépendance entre la classe cliente et l'interface requise.
- Toute classe implémentant l'interface pourra être utilisée



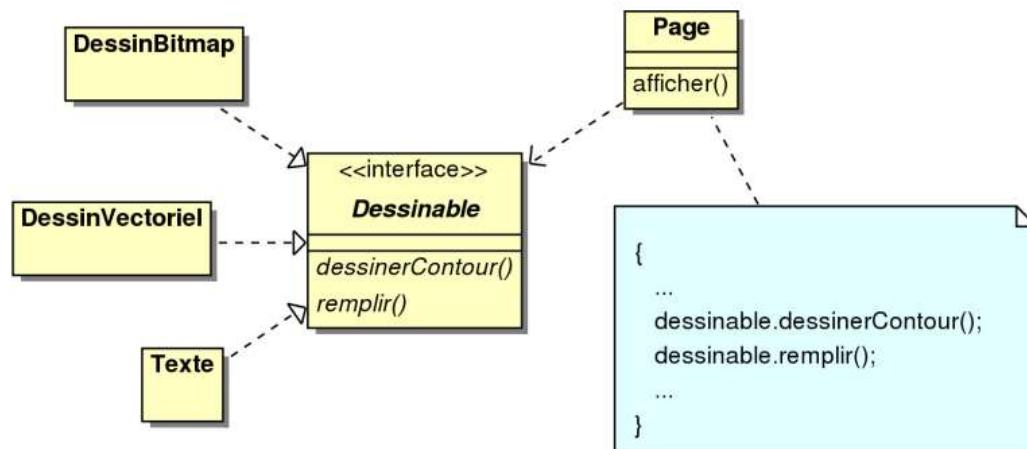
L2GLSI-ISIMM-23/24-M.GZARA

65

65

 π

Interface



L2GLSI-ISIMM-23/24-M.GZARA

66

66

π

Eléments dérivés

- › Les attributs dérivés peuvent être calculés à partir d'autres attributs et des formules de calcul.
- › Les attributs dérivés sont symbolisés par l'ajout d'un <</>> devant leur nom.
- › Lors de la conception, un attribut dérivé peut être utilisé comme marqueur jusqu'à ce que vous puissiez déterminer les règles à lui appliquer.
- › Une association dérivée est conditionnée ou peut être déduite à partir d'autres autres associations. On utilise également le symbole <</>>.

L2GLSI-ISIMM-23/24-M.GZARA

67

67

 π

Attributs de classe

- › Par défaut, les valeurs des attributs définis dans une classe diffèrent d'un objet à un autre.
- › Parfois, il est nécessaire de définir un attribut de classe qui garde une valeur unique et partagée par toutes les instances.
- › Graphiquement, un attribut de classe est souligné

L2GLSI-ISIMM-23/24-M.GZARA

68

68

π

Opérations de classe

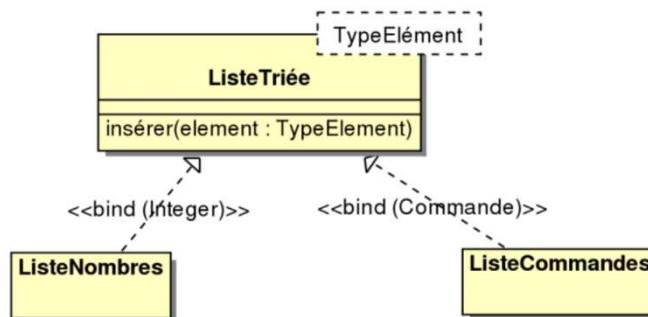
- › Semblable aux attributs de classe
- › Une opération de classe est une propriété de la classe, et non de ses instances.
- › Elle n'a pas accès aux attributs des objets de la classe.

 π

Classe paramétrée

- › Pour dérir une classe générique et paramétrable en fonction de valeurs et/ou de types :
 - Déinition d'une classe paramétrée par des éléments spéciés dans un rectangle en pointillés ;
 - Utilisation d'une dépendance stéréotypée <<bind>> pour dérir des classes en fonction de la classe paramétrée.

Java5 : généricité
C++ : templates



π

Le diagramme de classe

- › Le diagramme de classe exprime de manière générale la structure statique d'un système
- › Le modèle des classes et des objets décrit les objets persistants du système d'information
- › Le diagramme de classes est le diagramme le plus important dans toutes les méthodes orientés objets.
- › Le développement du modèle statique constitue la deuxième activité dans l'étape d'analyse
- › Une classe ne doit pas avoir trop de responsabilités
- › Ne pas confondre objet logique et objet physique (une entité et sa description)

 π

Comment construire un diagramme de classes

- › Lister les noms
 - Un substantif (mot du domaine) est un candidat potentiel pour
 - › Une entité
 - › Un attribut
 - Supprimer :
 - › Synonymes
 - › Polysèmes
 - › Noms superflus (inutiles ou hors domaine)
 - Procéder par itérations
- › Identifier les classes
 - Regroupent des attribut