

UML : Diagrammes de Classes

Objet, Classe, Attribut, Méthode
Lien, Association, Cardinalité
Généralisation, Composition, Aggrégation,
Classe associative, Association qualifiée...

Jean-Marie.Favre@imag.fr

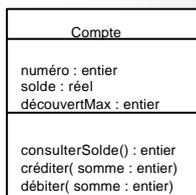
Concepts de base

- UML est basé sur différents concepts de base :
 - Objet, Classe
 - Lien, Association
 - Contrainte
- UML propose des notations et des diagrammes
 - Diagramme de classes (description au niveau modélisation, cas général)
 - Diagramme d'objets (description au niveau instance, exemples)

jfavre@imag.fr

Diagrammes de Classes 2

Notation pour les classes



Nom de la classe

Attributs
nom
type

Méthodes
nom
paramètre
type du résultat

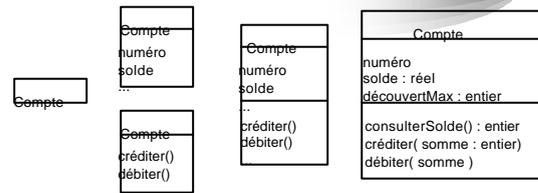
Contraintes



jfavre@imag.fr

Diagrammes de Classes 3

Notations simplifiées pour les classes



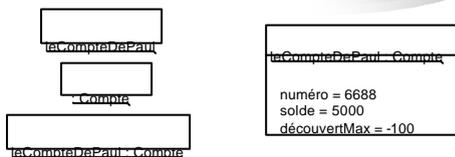
Conventions :

- les noms de classes commencent par une majuscule
- les noms d'attributs et de méthodes commencent par une minuscule

jfavre@imag.fr

Diagrammes de Classes 4

Notations pour les objets



Convention :

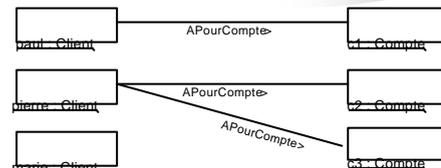
- les noms d'objets commencent par une minuscule et sont soulignés

jfavre@imag.fr

Diagrammes de Classes 5

Liens (entre objets)

Un lien indique une connexion entre deux objets



Conventions :

- les noms des liens sont des formes verbales et commencent par une majuscule
- > indique le sens de la lecture (ex : « paul APourCompte c1 »)

jfavre@imag.fr

Diagrammes de Classes 6

Rôles

Chacun des deux objets joue un **rôle** différent dans le lien



pierre assume le rôle de titulaire pour le compte c1
c1 assume le rôle de compte pour pierre

Conventions :

- choisir un groupe nominal pour désigner un rôle
- si un nom de rôle est omis, le nom de la classe fait office de nom

jrfavre@imag.fr

Diagrammes de Classes 7

Associations (entre classes)

Une association décrit un ensemble de liens de même sémantique



Diagramme de classes (modélisation)

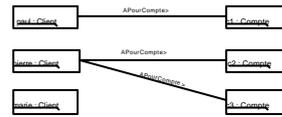


Diagramme d'objets (instances)

jrfavre@imag.fr

Diagrammes de Classes 8

Liens vs. Associations

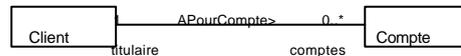
- Un lien lie deux objets
- Une association lie deux classes
- Un lien est une instance d'association
- Une association décrit un ensemble de liens
- Des liens peuvent être ajoutés ou créés pendant l'exécution, (ce n'est pas le cas des associations)

jrfavre@imag.fr

Diagrammes de Classes 9

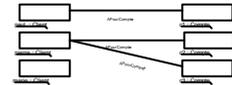
Cardinalités d'une association

- Précise combien d'objets peuvent être liés à un seul objet source
- Cardinalité minimale et cardinalité maximale ($C_{min}..C_{max}$)



« Un client a 0 ou plusieurs comptes »

« Un compte a toujours 1 et 1 seul titulaire »



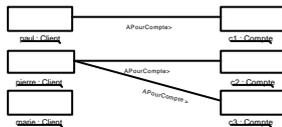
jrfavre@imag.fr

Diagrammes de Classes 10

Utiliser les rôles pour «naviguer»



paul.comptes = {c1}
pierre.comptes = {c2,c3}
marie.comptes = {}
c1.titulaire = paul
c2.titulaire = pierre
c3.titulaire = pierre

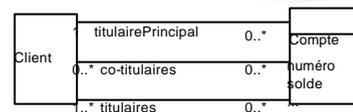


jrfavre@imag.fr

Diagrammes de Classes 11

Contraintes entre associations

Les cardinalités ne permettent pas d'exprimer toutes les contraintes...

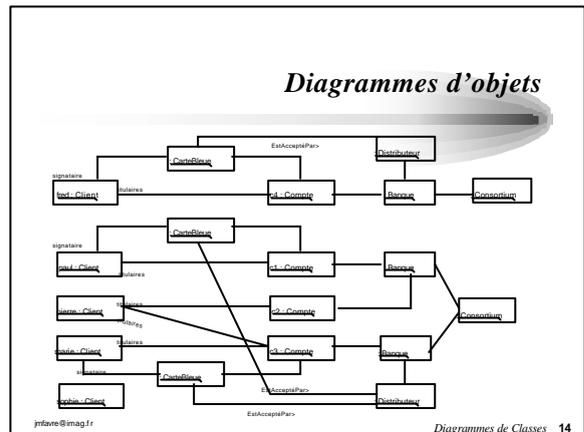
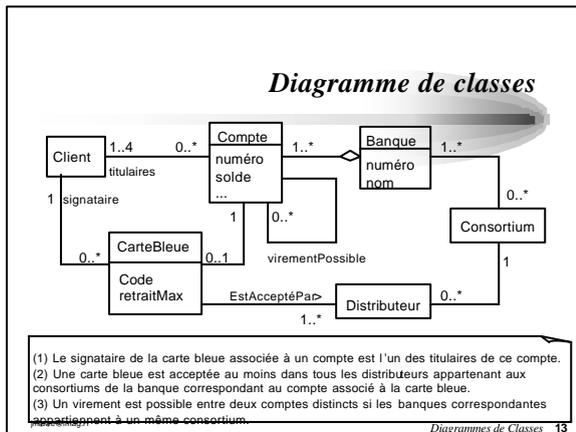


... décrire les contraintes en langue naturelle (ou en OCL le langage de contrainte d'UML)

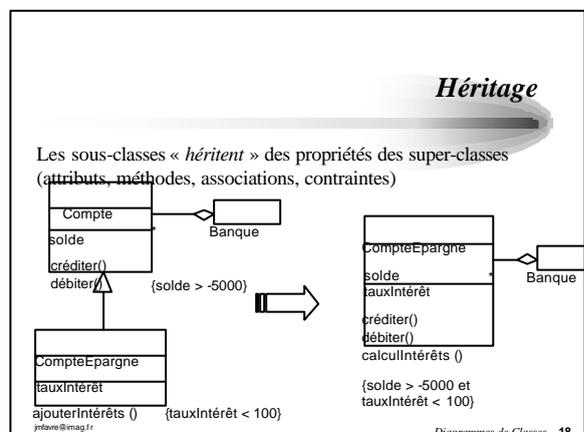
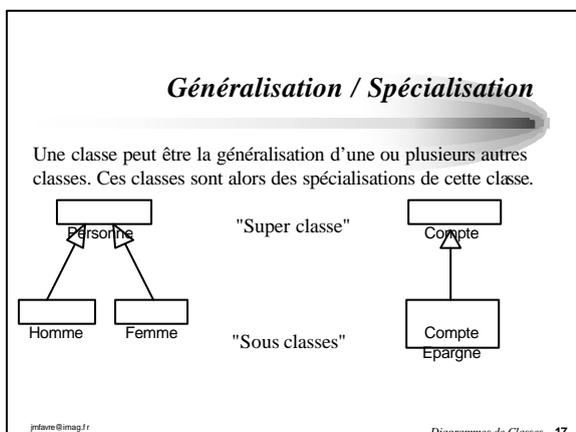
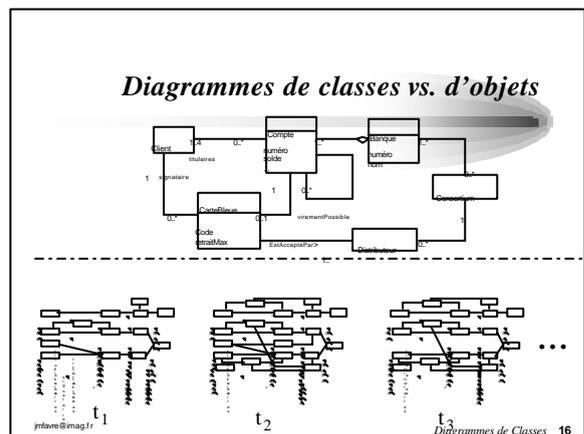
(1) Un client ne peut pas être à la fois titulaire principal et co-titulaire d'un même compte.
(2) Les titulaires d'un compte sont le titulaire principal et les co-titulaires le cas échéant

jrfavre@imag.fr

Diagrammes de Classes 12

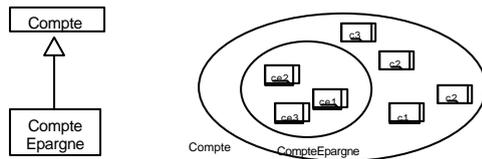


- ### Diagrammes de classes vs. d'objets
- Un diagramme de classes
 - définit l'ensemble de tous les états possibles
 - les contraintes doivent toujours être vérifiées
 - Un diagramme d'objets
 - décrit un état possible à un instant t, un cas particulier
 - doit être conforme au modèle de classes
 - Les diagrammes d'objets peuvent être utilisés pour
 - expliquer un diagramme de classe (donner un exemple)
 - valider un diagramme de classe (le "tester")
- Diagrams de Classes 15



Vision ensembliste

Vision ensembliste :
 tout objet d'une sous-classe appartient également à la super-classe
 => inclusion des ensembles d'objets

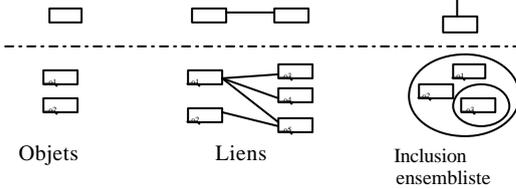


jrfavre@imag.fr

Diagrammes de Classes 19

Synthèse des concepts de base

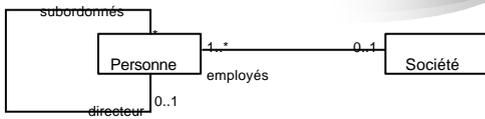
- Classe
 - attribut
 - méthode
- Association
 - rôle
 - cardinalité
- Généralisation



jrfavre@imag.fr

Diagrammes de Classes 20

Exemple

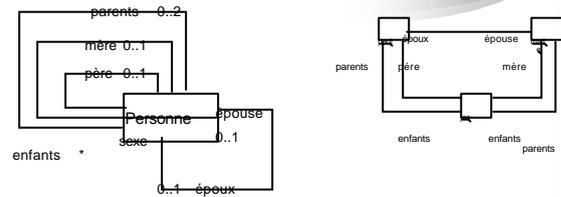


- (1) Le directeur de tout employé est employé dans la même société.
- (2) Dans toute société il y a au moins une personne qui n'est dirigée par personne (le directeur de la société).
- (3) Une personne ne peut pas être directeur d'elle même

jrfavre@imag.fr

Diagrammes de Classes 21

Exemple



jrfavre@imag.fr

Diagrammes de Classes 22

Diagrammes de classes: concepts additionnels

Concepts à utiliser à bon escient,
 lorsque nécessaire
 et uniquement lorsque nécessaire...

Rappels

- UML est un ensemble de notations
- De nombreuses utilisations possibles
- Notation simple vs. complexe
- Notation plus ou moins formelle
- Notation extensible

jrfavre@imag.fr

Diagrammes de Classes 24

Savoir s'adapter

S'adapter

- au niveau d'abstraction
- au domaine d'application
- à ses collègues
- aux outils utilisés
 - pour écrire des diagrammes UML,
 - pour générer du code, des schémas de bases de données, etc.
- ...

Savoir lire vs. savoir écrire
Ingénierie vs. Retro-ingénierie

jrfavre@imag.fr

Diagrammes de Classes 25

+ # - Visibilité des éléments

- Restreindre l'accès aux éléments d'un modèle
- Contrôler et éviter les dépendances entre paquetages
 - + public visible
 - # protégé visible dans la classe et ses sous-classes
 - privé visible dans la classe uniquement

- N'a pas de sens dans un modèle abstrait
- Utile lors de la conception.
- N'utiliser que lorsque nécessaire.

jrfavre@imag.fr

Diagrammes de Classes 26

Déclaration d'attributs

[visibilité] nom [card] [: type] [= valeur-initiale] [{ props... }]

exemples:

- age
- - age : Integer = 0
- # age [0..1] : Integer
- # numsecu : Integer {frozen}
- # motsClés [*] : String {addOnly}

- Adapter le niveau de détail au niveau d'abstraction

jrfavre@imag.fr

Diagrammes de Classes 27

Déclaration d'opérations

[visibilité] nom [(params)] [: type] [{ props... }]

params := [in / out / inout] nom [: type] [=default] [{ props... }]

exemples:

- getAge()
- + getAge() : Integer
- - updateAge(in date : Date) : Boolean
- + getAge() : Integer {isQuery}

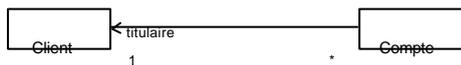
- Adapter le niveau de détail au niveau d'abstraction

jrfavre@imag.fr

Diagrammes de Classes 28

Navigation

Association unidirectionnelle
On ne peut naviguer que dans un sens



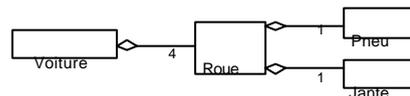
Ce détail n'est a priori utile que lors de la conception
En cas de doute, ne pas mettre de flèche !!!

jrfavre@imag.fr

Diagrammes de Classes 29

Composition

Notion intuitive de "composants" et de "composites"



composition = cas particulier d'association
+ contraintes décrivant la notion de "composant"...

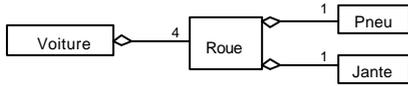
jrfavre@imag.fr

Diagrammes de Classes 30

Composition

Contraintes liées à la composition :

1. Un objet *composant* ne peut être que dans un seul objet *composite*
2. Un objet *composant* n'existe pas sans son objet *composite*
3. Si un objet composite est détruit, ses composants aussi



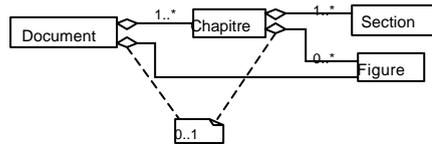
jrfave@imag.fr

Diagrammes de Classes 31

Composition

Contraintes:

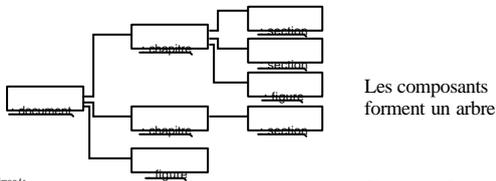
1. Un objet *composant* ne peut être que dans un seul objet *composite*
2. Un objet *composant* n'existe pas sans son objet *composite*
3. Si un objet composite est détruit, ses composants aussi



jrfave@imag.fr

Diagrammes de Classes 32

Composition

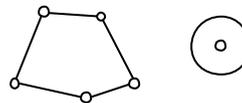
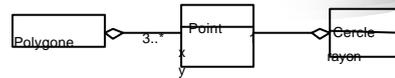


Les composants forment un arbre

jrfave@imag.fr

Diagrammes de Classes 33

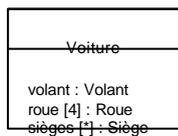
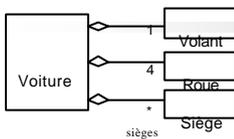
Composition



jrfave@imag.fr

Diagrammes de Classes 34

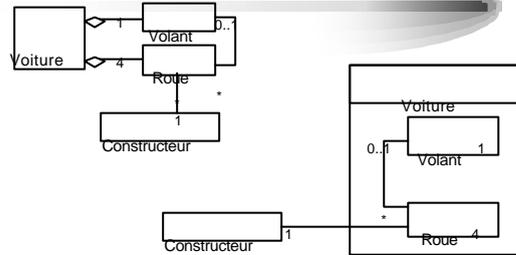
Autres notations pour la Composition



jrfave@imag.fr

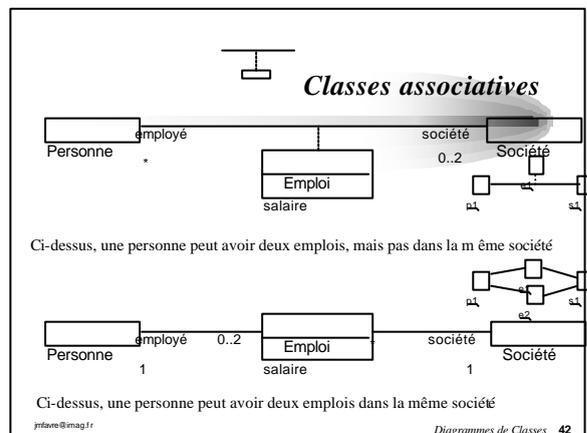
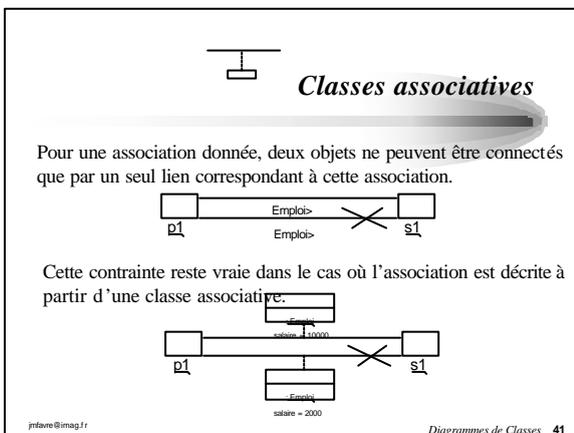
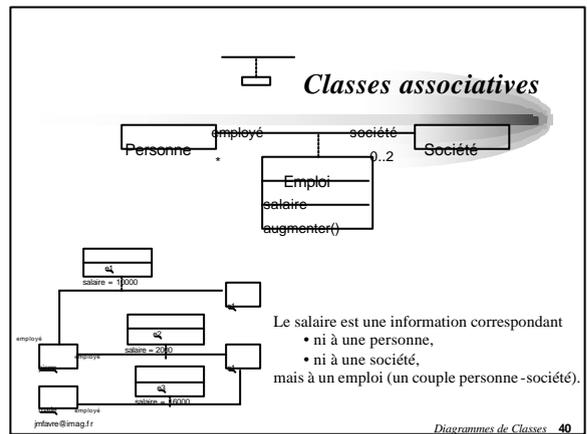
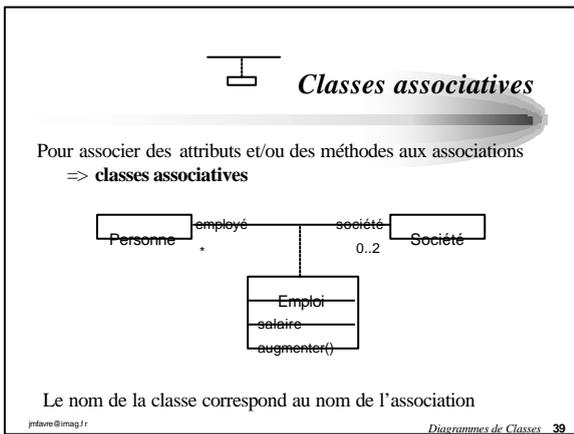
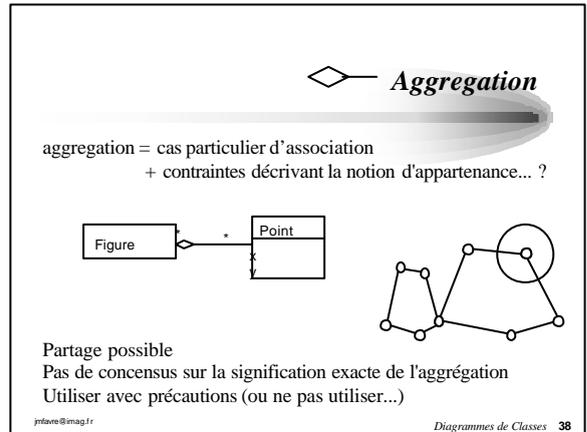
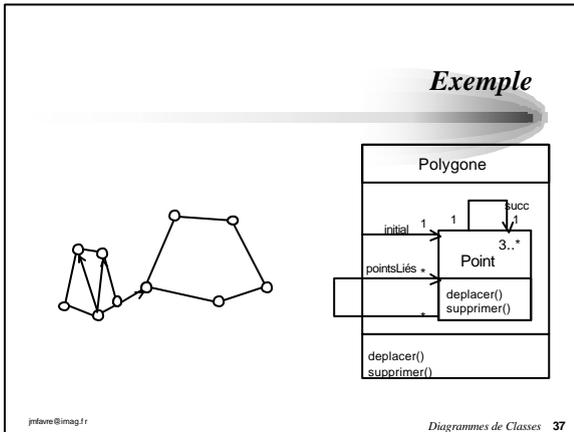
Diagrammes de Classes 35

Associations intra vs. inter composites

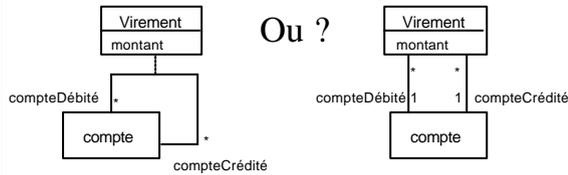


jrfave@imag.fr

Diagrammes de Classes 36



Classes associatives

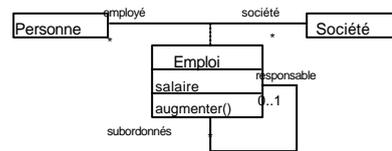


jrfave@imag.fr

Diagrammes de Classes 43

Classes associatives

Les classes associatives sont des associations mais aussi des classes. Elles ont donc les mêmes propriétés et peuvent par exemple être liées par des associations.



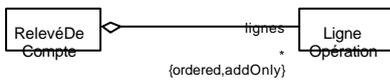
jrfave@imag.fr

Diagrammes de Classes 44

Contraintes sur les associations

Contraintes prédéfinies sur les associations. Par exemple :

- { frozen } : fixé lors de la création de l'objet, ne peut pas changer
- { ordered } : les éléments de la collection sont ordonnés
- { addOnly } : impossible de supprimer un élément

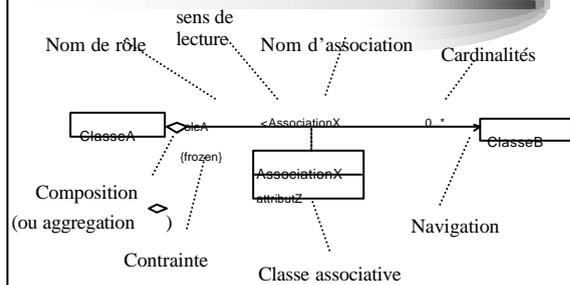


Il est possible de définir de nouvelles contraintes

jrfave@imag.fr

Diagrammes de Classes 45

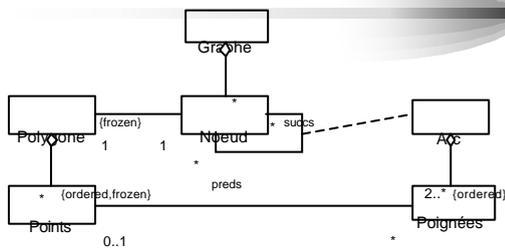
Synthèse sur les associations



jrfave@imag.fr

Diagrammes de Classes 46

Exemple

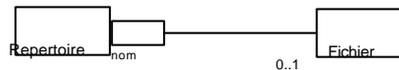


jrfave@imag.fr

Diagrammes de Classes 47

Associations qualifiées

Un **qualifieur** est un attribut ou un ensemble d'attributs dont la valeur sert à déterminer l'ensemble des instances associées à une instance via une association.

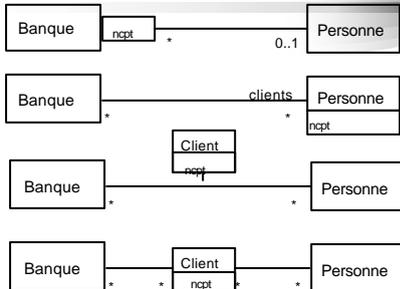


Les attributs du qualifieur sont des attributs de l'association.

jrfave@imag.fr

Diagrammes de Classes 48

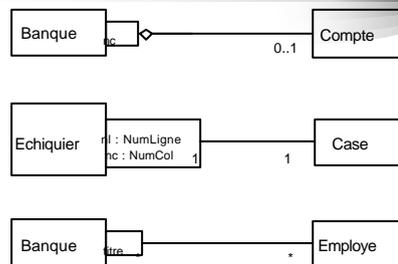
Associations vs. Associations qualifiées



jrfavre@imag.fr

Diagrammes de Classes 49

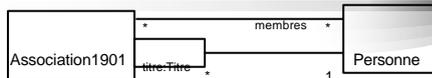
Cardinalité des Associations Qualifiées



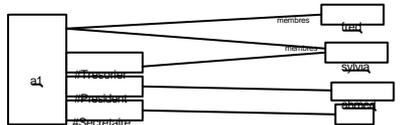
jrfavre@imag.fr

Diagrammes de Classes 50

Exemple



Titre = enum (President, Tresorier, Secretaire)

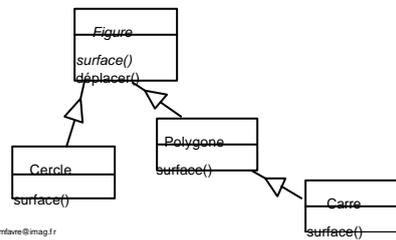


jrfavre@imag.fr

Diagrammes de Classes 51

Héritage et redéfinition

Une sous classe peut redéfinir une propriété, à condition toutefois de rester compatible avec la définition originale

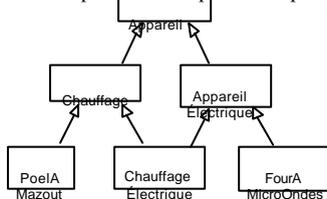


jrfavre@imag.fr

Diagrammes de Classes 52

Héritage multiple

Une classe peut hériter de plusieurs super-classes



Certains langages de programmation interdisent l'héritage multiple

jrfavre@imag.fr

Diagrammes de Classes 53

Points liés à l'héritage et à la classification

Les modèles orientés-objets ne font pas tous les mêmes hypothèses

Héritage simple vs. héritage multiple

Une classe peut-elle hériter de plusieurs classes ?

Classification simple vs. classification multiple

Un objet peut-il être simultanément instance de plusieurs classes ?

Classification statique vs. classification dynamique

Un objet peut-il changer de classe pendant l'exécution ?

jrfavre@imag.fr

Diagrammes de Classes 54

Points liés à l'héritage et à la classification

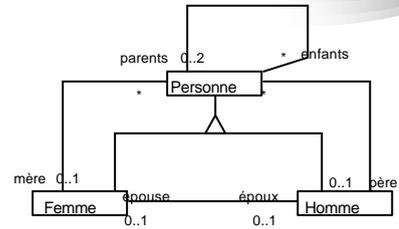
Sauf si le contraire est indiqué explicitement, en UML les hypothèses par défaut sont :

- Héritage multiple
une classe peut hériter de plusieurs classes
- Classification simple
un objet est instance d'une seule classe
- Classification statique
un objet est créé à partir d'une classe donnée et n'en change pas

jrfave@imag.fr

Diagrammes de Classes 55

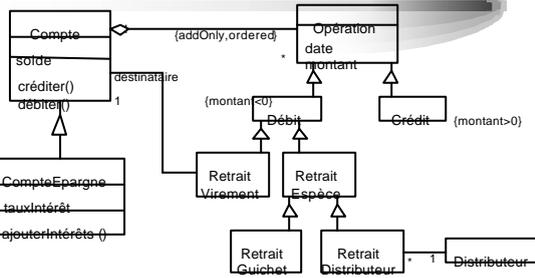
Exemple



jrfave@imag.fr

Diagrammes de Classes 56

Exemple



jrfave@imag.fr

Diagrammes de Classes 57