



Polytech Paris-Sud
Formation initiale 3^e année
Spécialité Informatique
Année 2016-2017

UML

Cours 4

Diagrammes de classes (suite)

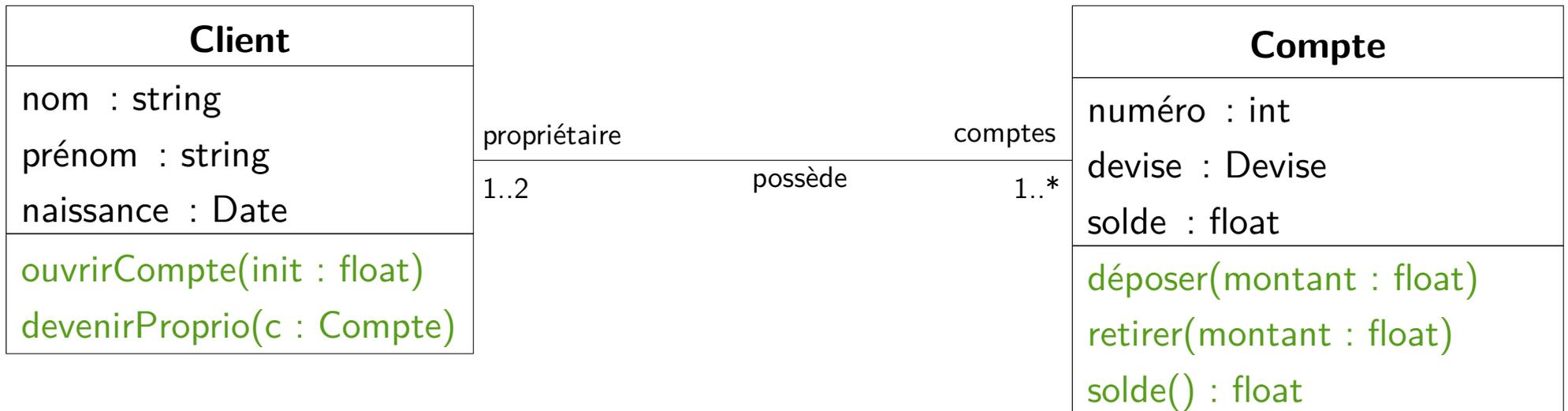
Delphine Longuet
delphine.longuet@lri.fr

<http://www.lri.fr/~longuet/Enseignements/16-17/Et3-UML>

Opérations

Opérations

- **Service** qui peut être demandé à tout objet de la classe
- **Comportement commun** à tous les objets de la classe



Opérations

Effets possibles d'une opération

- Renvoyer le **résultat** d'un calcul

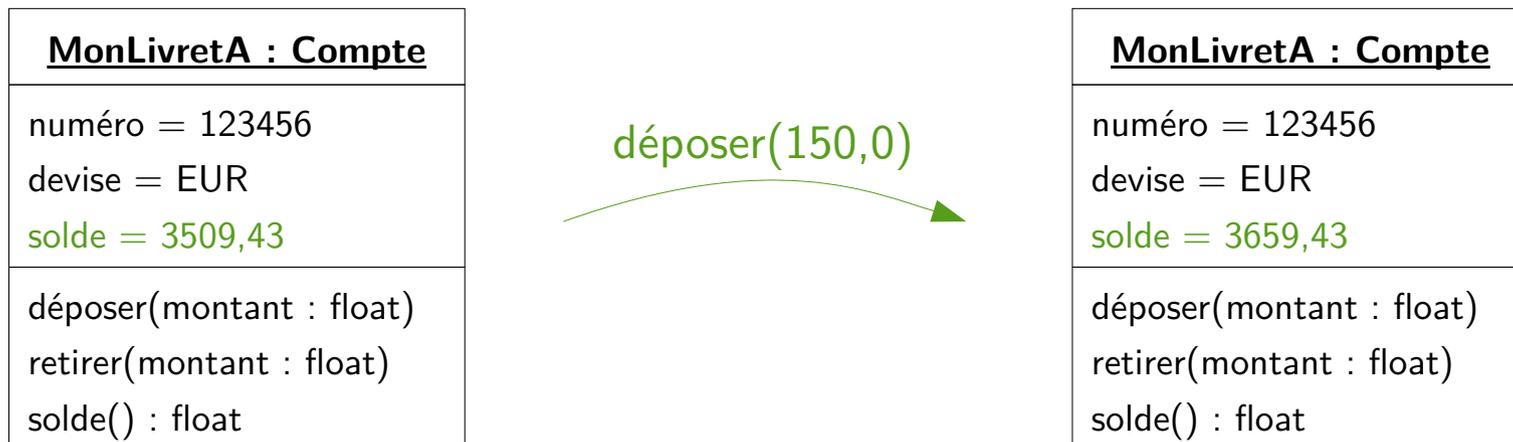
<u>MonLivretA : Compte</u>
numéro = 123456
devise = EUR
solde = 3509,43
déposer(montant : float)
retirer(montant : float)
solde() : float



Opérations

Effets possibles d'une opération

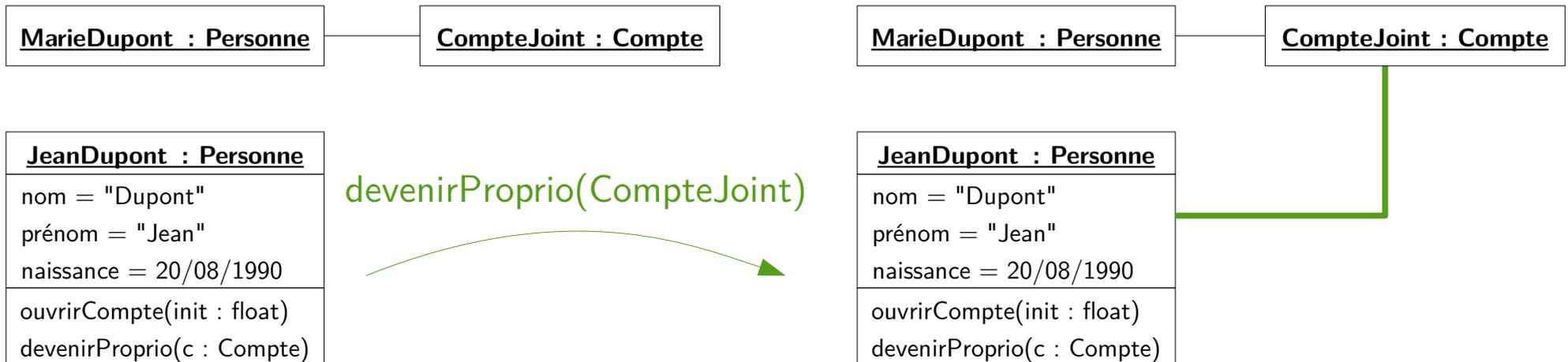
- Renvoyer le **résultat** d'un calcul
- **Modifier l'état** du système
 - modification de la valeur des attributs



Opérations

Effets possibles d'une opération

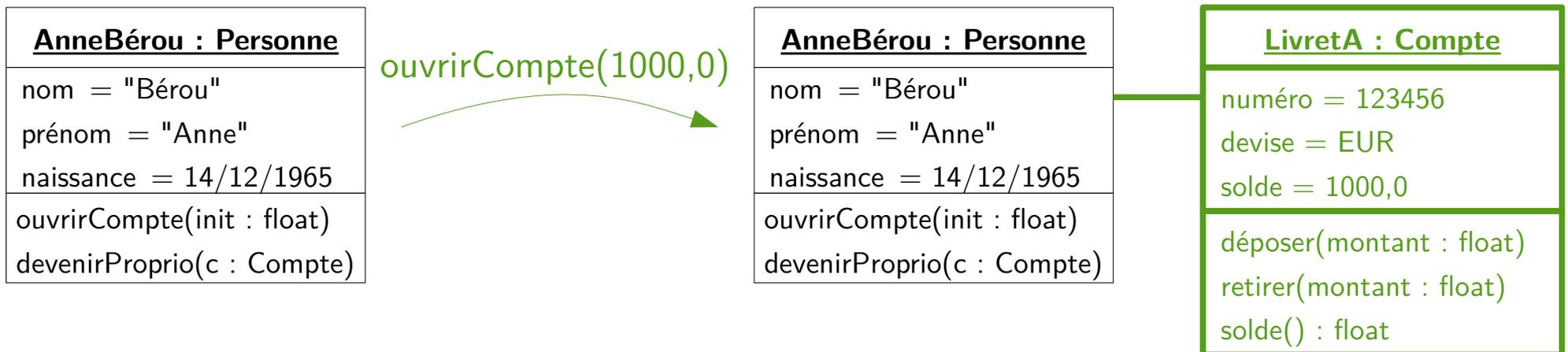
- Renvoyer le **résultat** d'un calcul
- **Modifier l'état** du système
 - modification de la valeur des attributs
 - ajout/suppressions de liens entre objets



Opérations

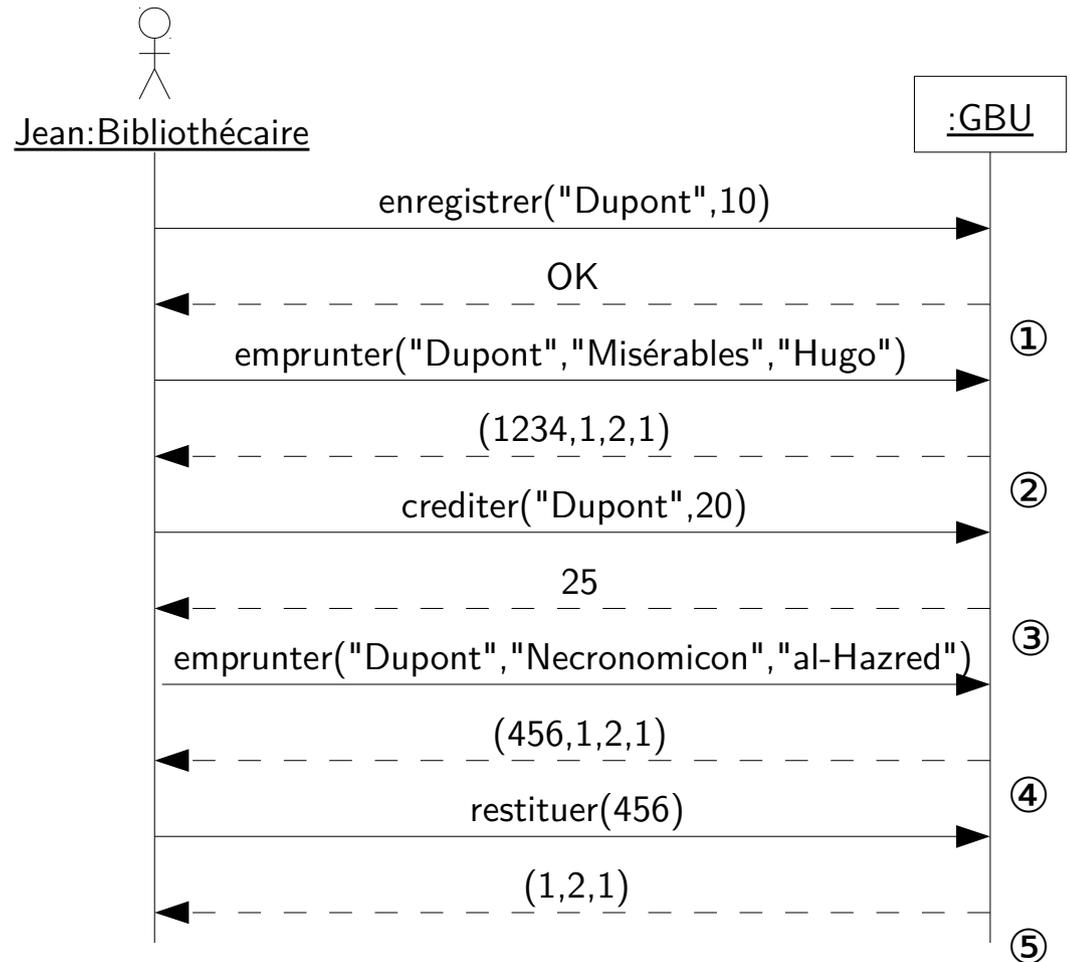
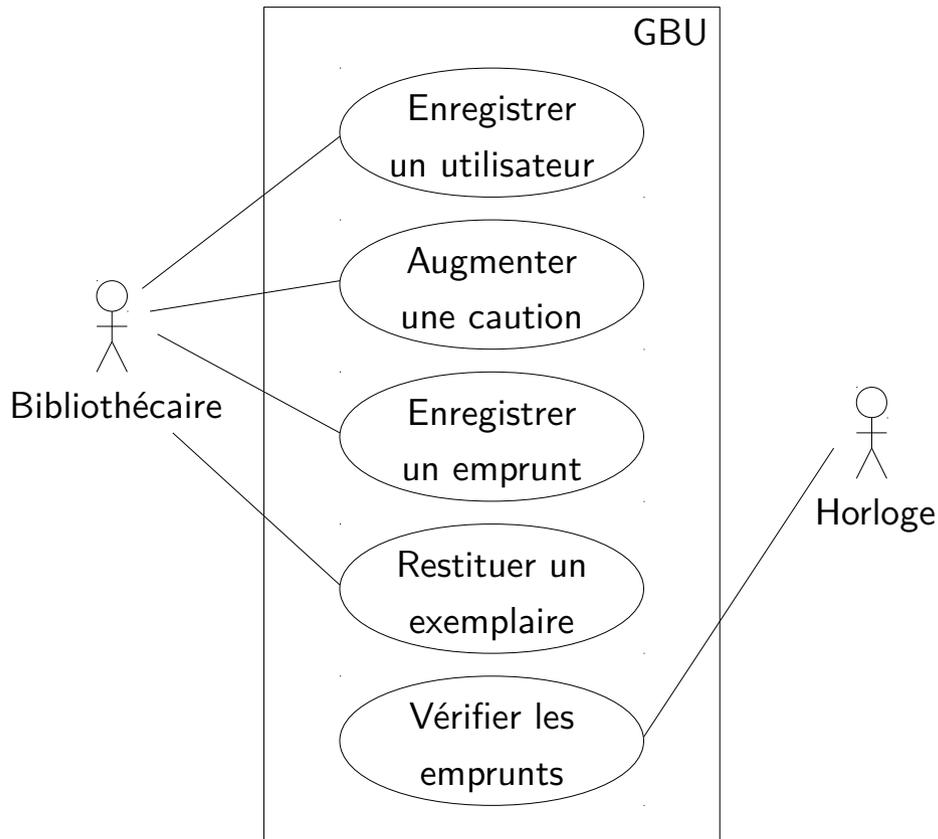
Effets possibles d'une opération

- Renvoyer le **résultat** d'un calcul
- **Modifier l'état** du système
 - modification de la valeur des attributs
 - ajout/suppressions de liens entre objets
 - création/destruction d'objets



Exercice : bibliothèque

Donner les diagrammes d'objets correspondant aux états numérotés du scénario concret.



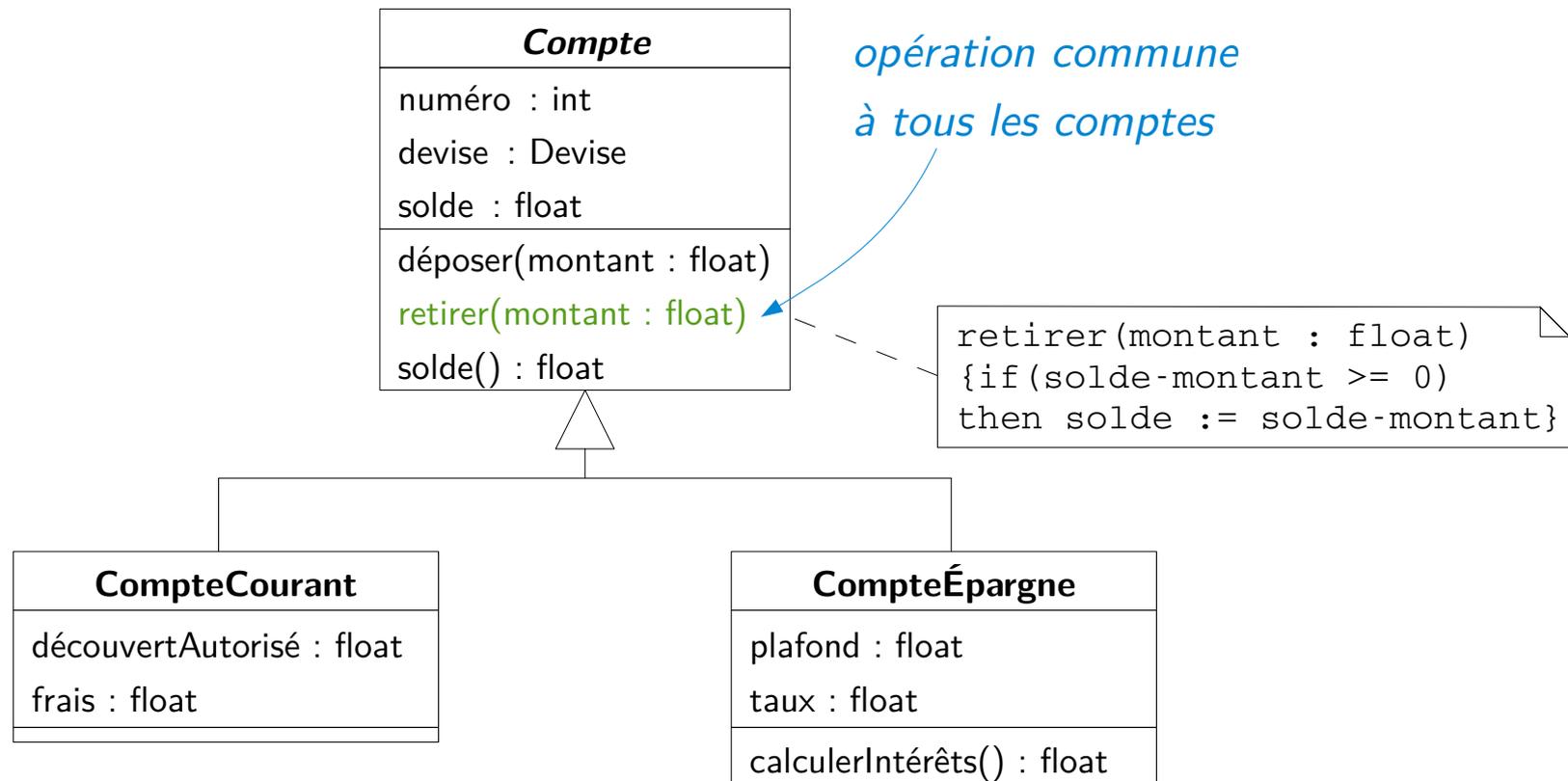
Opérations :

- enregistrer(nom,caution)
- créditer(nom,montant) : caution
- emprunter(nom,titre,auteur) : (code,travée,étagère,niveau)
- restituer(code) : (travée,étagère,niveau)

Héritage d'opération

Opération commune aux sous-classes :

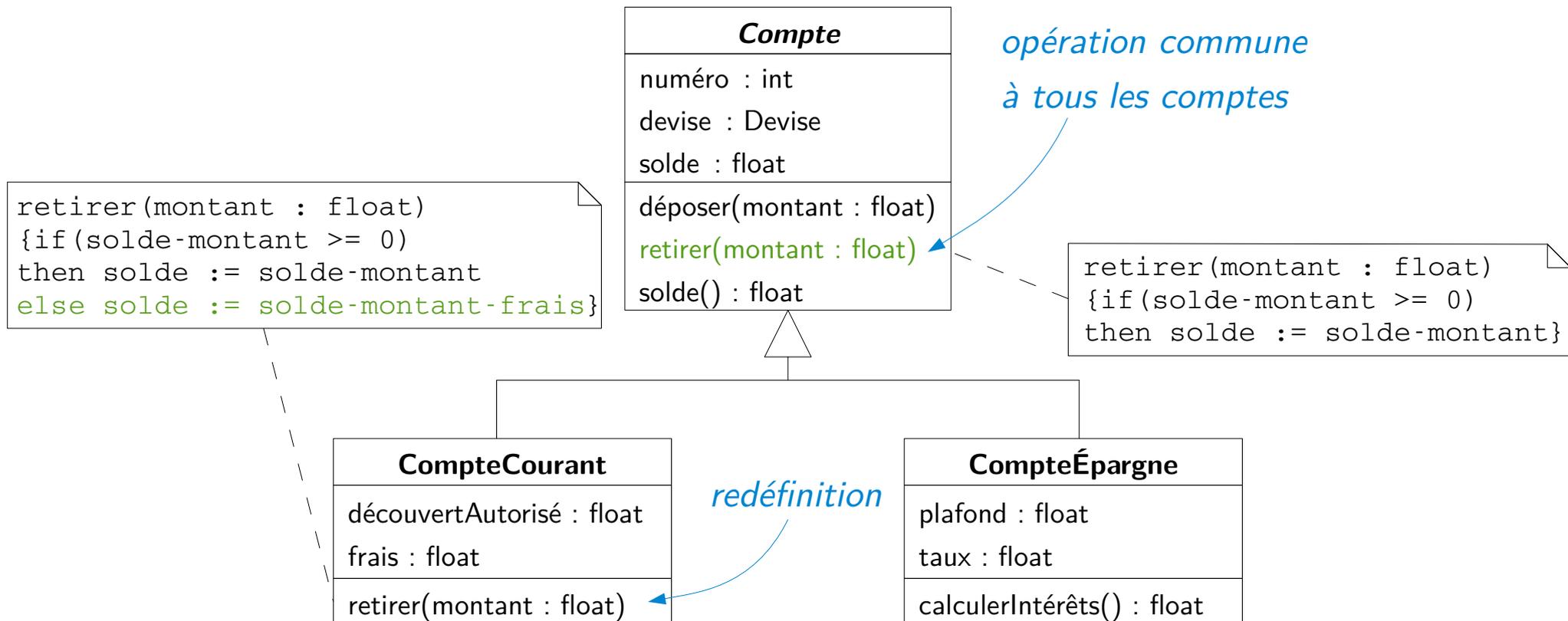
- Définition dans la super-classe



Redéfinition d'opération

Opération commune aux sous-classes :

- Définition dans la super-classe
- Possibilité de **redéfinition locale** de l'opération dans une **sous-classe** pour étendre le comportement de l'opération

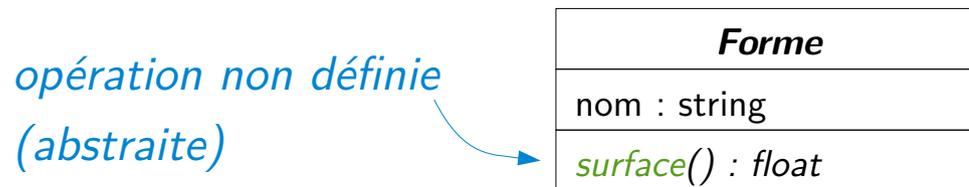


Classe abstraite

Classe sans instance car certaines opérations non définies

- Opération non définie en italique
- Nom de la classe en italique (ou stéréotype « abstract »)

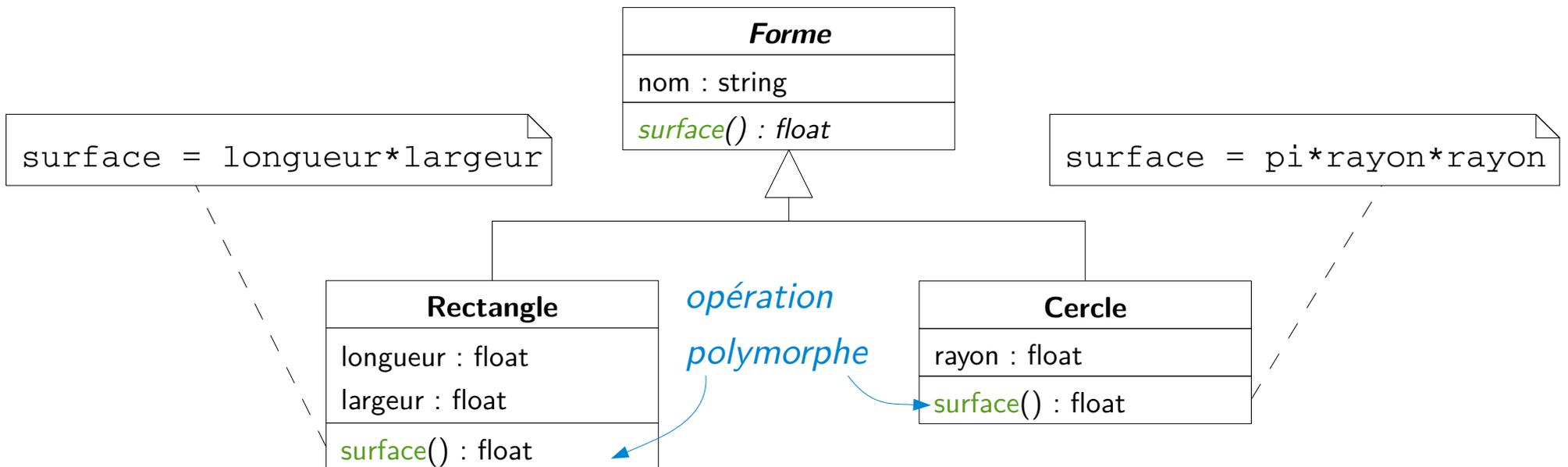
Exemple : On ne peut pas calculer la surface d'une forme sans savoir de quelle forme il s'agit



Polymorphisme

Contexte : Définition d'une opération abstraite dans les **classes** héritant d'une **classe abstraite**

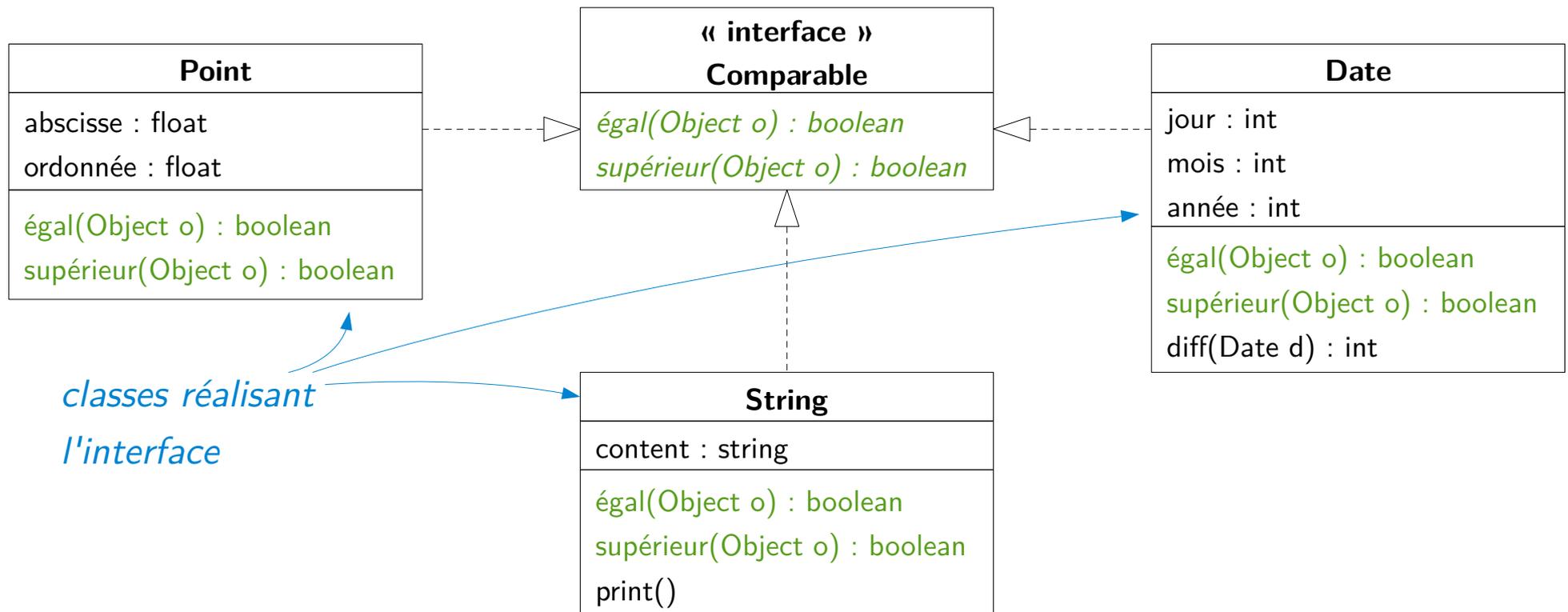
Opération polymorphe : Opération définie dans différentes sous-classes mais opération **spécifique** à la sous-classe



Interface

Liste d'opérations constituant un **contrat** à respecter par les classes réalisant l'interface

- Pas une classe, ne peut pas servir à créer des objets
- Toutes les opérations sont abstraites



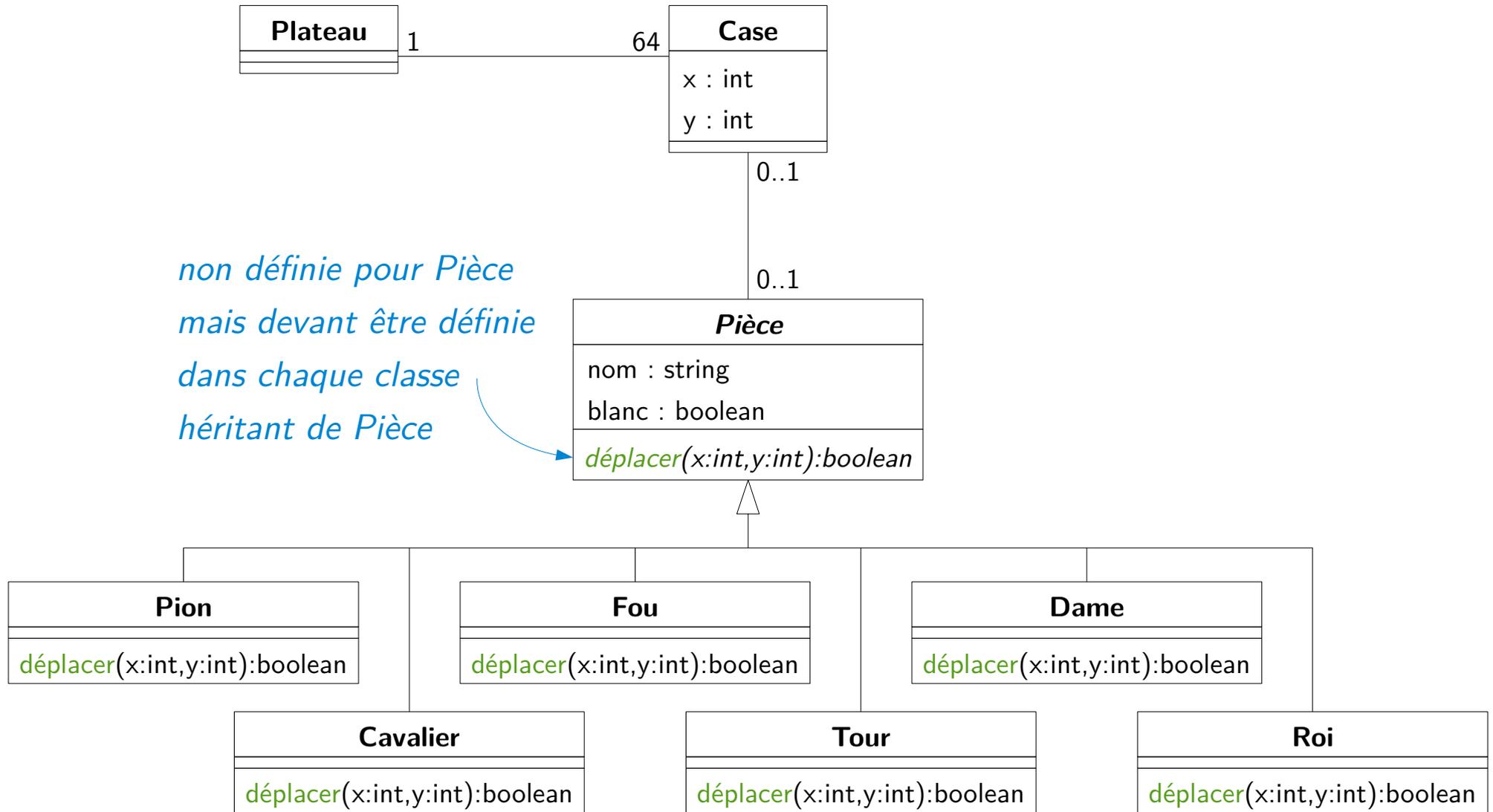
Exemple : échecs

Donner un diagramme de classes modélisant un jeu d'échecs simplifié (plateau + pièces).

Déplacement des pièces :

- **Pion** : une case en avant
- **Cavalier** : en L, deux cases puis une perpendiculairement
- **Fou** : en diagonale
- **Tour** : horizontalement ou verticalement
- **Dame** : dans toutes les directions
- **Roi** : une case dans toutes les directions

Exemple : échecs



Notations avancées

Attributs et opérations de classes

- Attribut de classe : **valeur commune** à toutes les instances
- Opération de classe : **opération sur la classe** elle-même
(pas besoin d'une instance pour être appelée)
- Soulignés dans la classe
- (`static` en Java)

Article
- référence : int
- prixHT : float
- <u>nbInstances</u> : int
+ calculerPrixTTC(taxe : float) : float
+ <u>compterInstances()</u> : int

Notations avancées

Visibilité

- + : **public**, accessible à toutes les classes
- # : **protégé**, accessible uniquement aux sous-classes
- : **privé**, inaccessible à tout objet hors de la classe

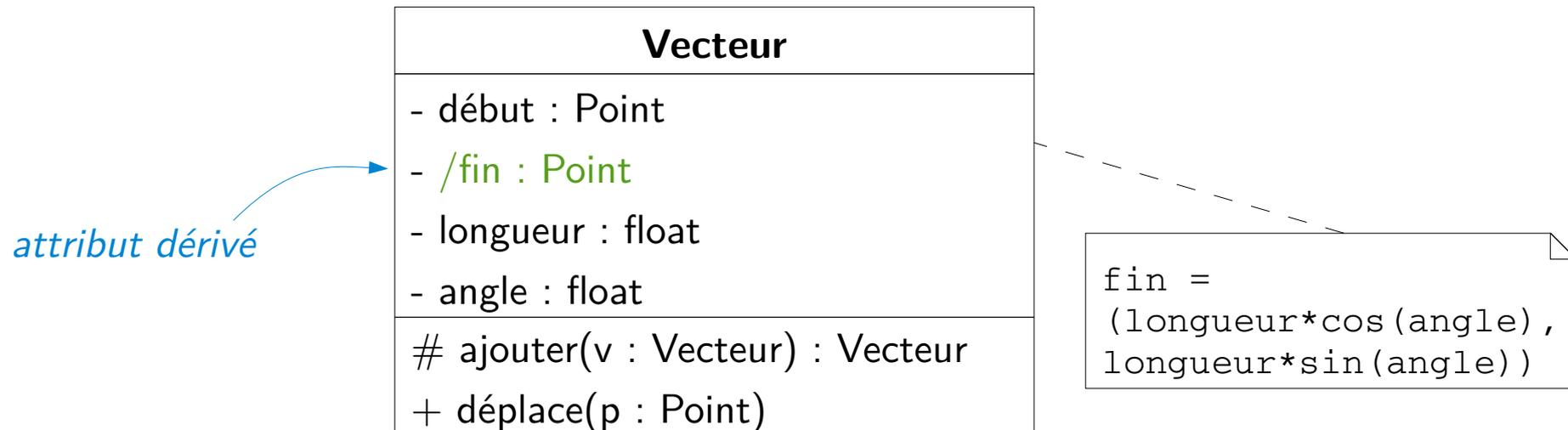
Pas de visibilité par défaut

Vecteur
- début : Point
- /fin : Point
- longueur : float
- angle : float
ajouter(v : Vecteur) : Vecteur
+ déplace(p : Point)

Attribut dérivé

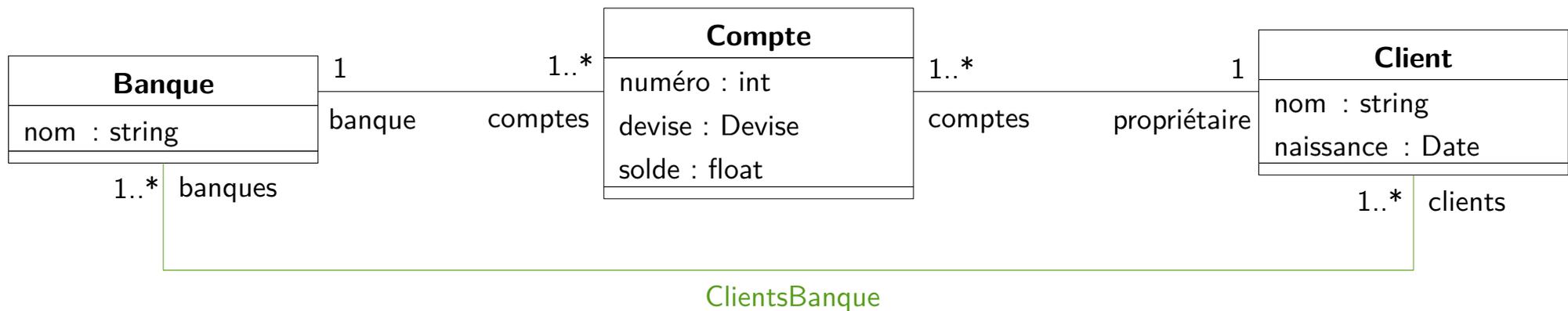
Peut être calculé à tout moment à partir d'autres informations du système

- Notation : /attribut
- Peut nécessiter des informations de plusieurs classes



Association dérivée

Redondance d'une association : Association n'apportant pas d'information supplémentaire, se demander si nécessaire



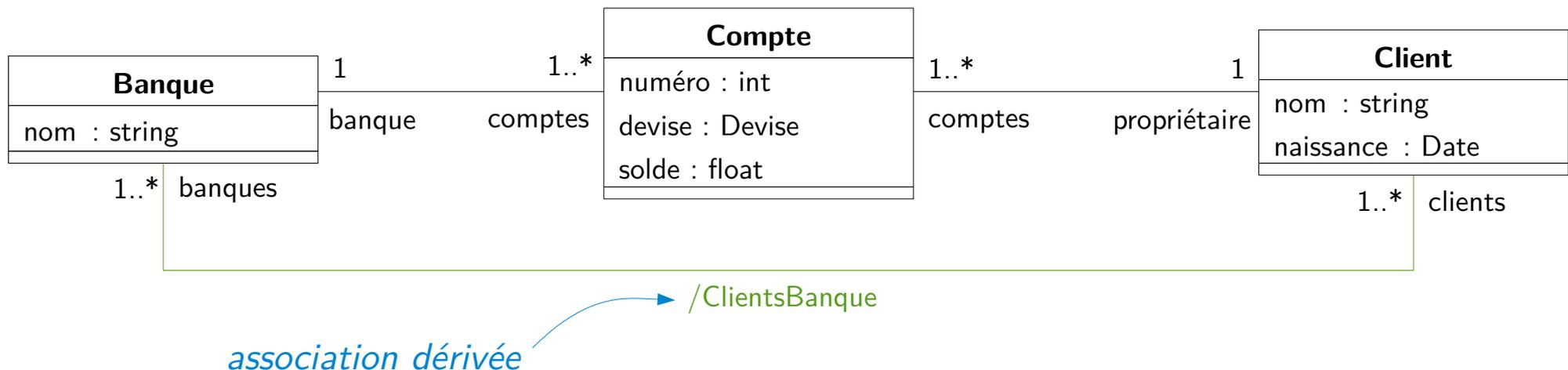
Association dérivée

Redondance d'une association : Association n'apportant pas d'information supplémentaire, se demander si nécessaire

Si nécessaire, la faire apparaître comme **association dérivée**

Association dérivée : Peut être calculée à tout moment à partir d'autres informations du système

Notation : **/association**



Limites du diagramme de classes

Diagramme de classes représente la **structure du système** en termes d'**objets** et de **relations entre ces objets**

Ne permet pas de représenter :

- **Valeurs autorisées** des attributs
- **Conditions** sur les associations
- **Relations** entre les attributs ou entre les associations

Expression des contraintes liées au diagramme :

- **Notes** dans le diagramme
- **Texte** accompagnant le diagramme
- **OCL** : langage de contraintes formel associé à UML

Contraintes, invariants

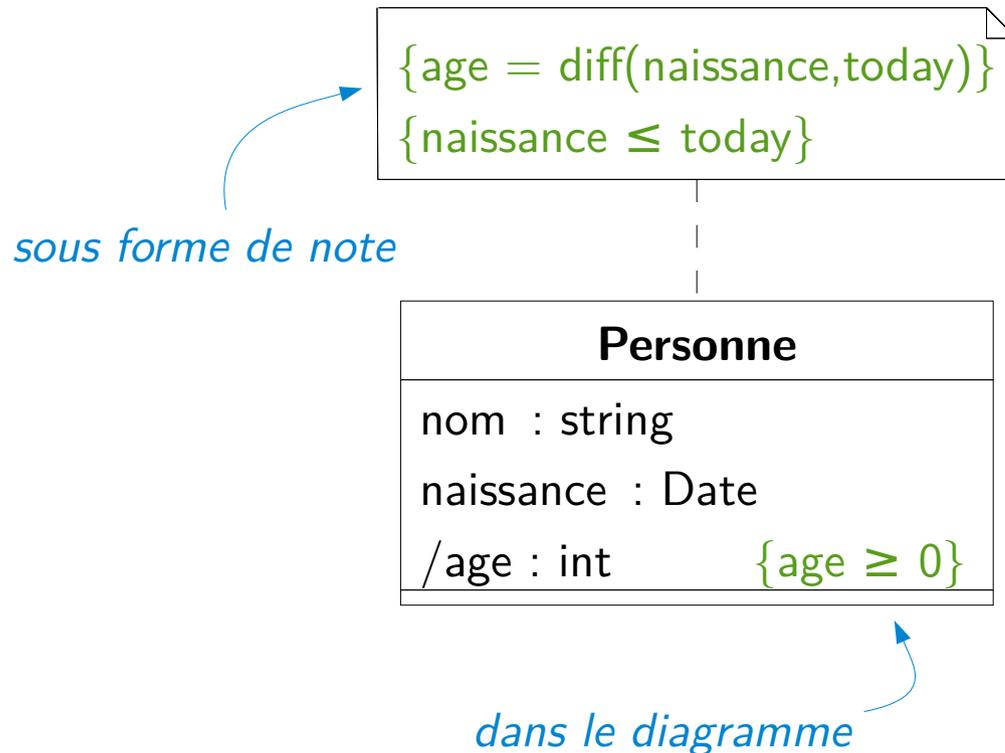
Propriétés :

- Portant sur les éléments du modèle
- Doivent être vérifiées à tout instant
- En général, restriction sur les diagrammes d'objets possibles à partir du diagramme de classes
- Héritage des contraintes de la super-classe vers les sous-classes

Contraintes présentes dans le diagramme :

- Type des attributs
- Multiplicités des associations

Contraintes sur les attributs

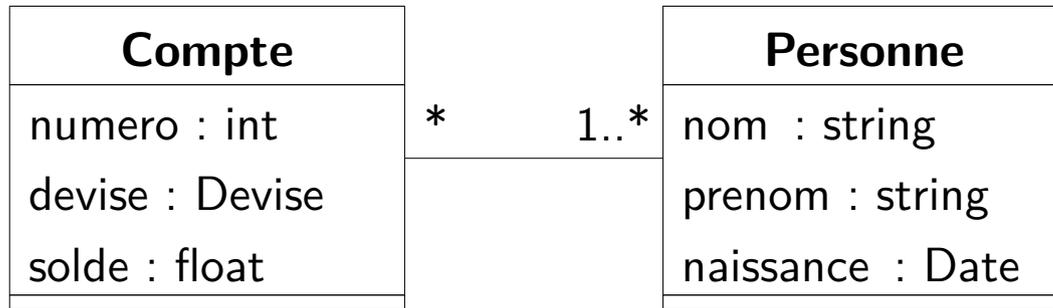


Contraintes sur la classe Personne :

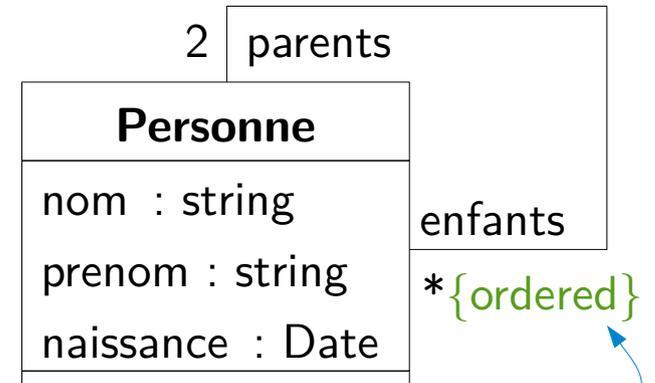
- L'âge est toujours positif
- L'âge est calculé comme la différence de la date de naissance et la date d'aujourd'hui
- La date de naissance est inférieure ou égale à la date d'aujourd'hui

dans un document annexe

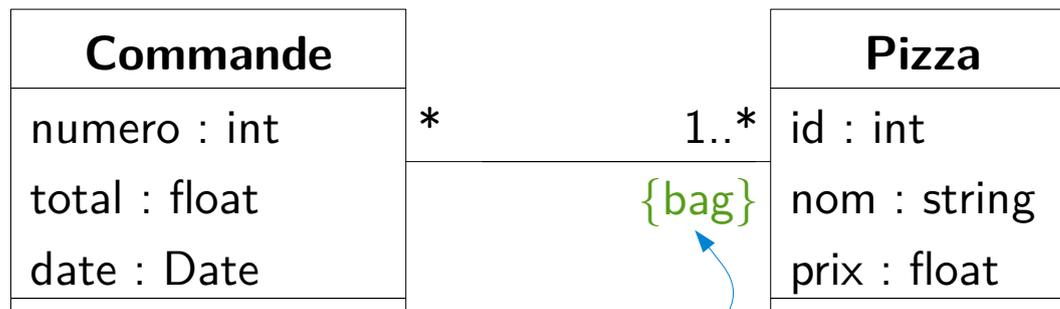
Contraintes associées à la multiplicité



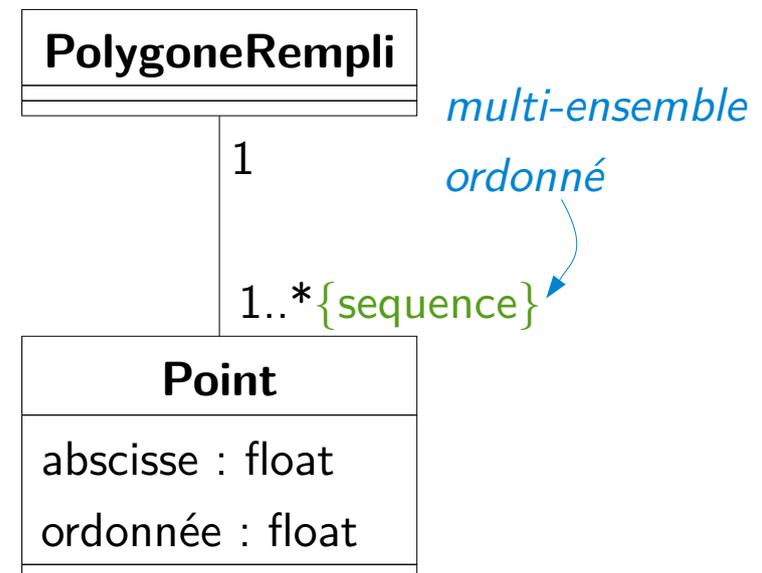
Par défaut : **ensemble non ordonné**



ensemble ordonné

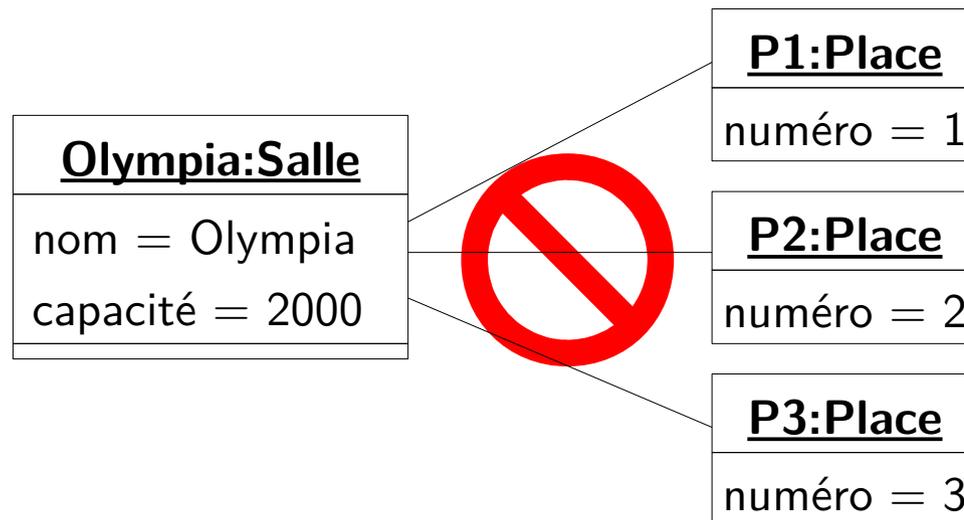
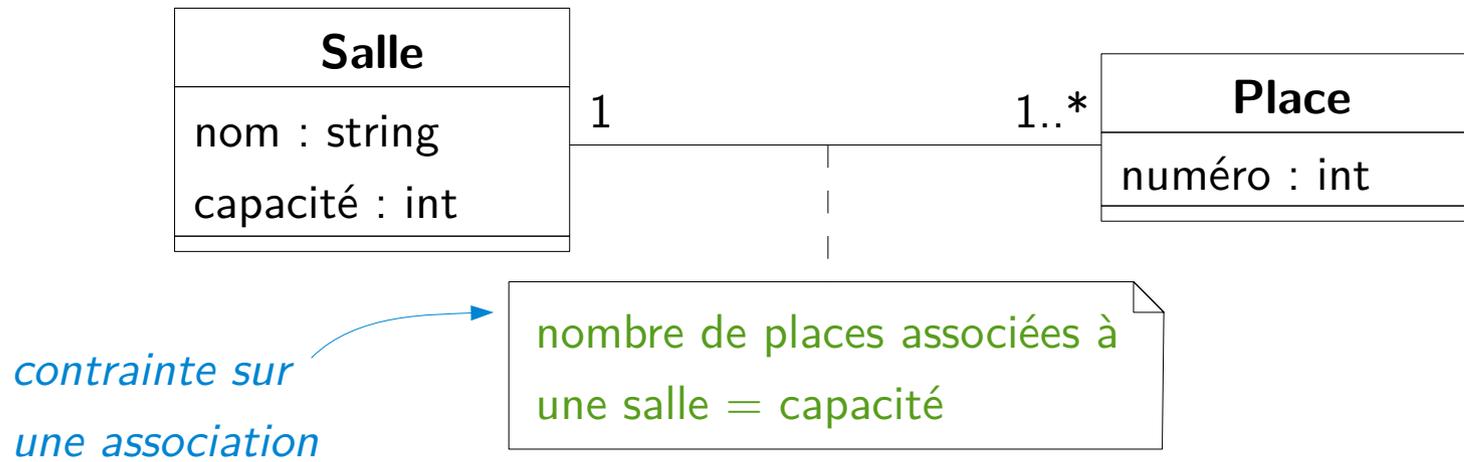


*multi-ensemble
(chaque élément peut
apparaître plusieurs fois)*

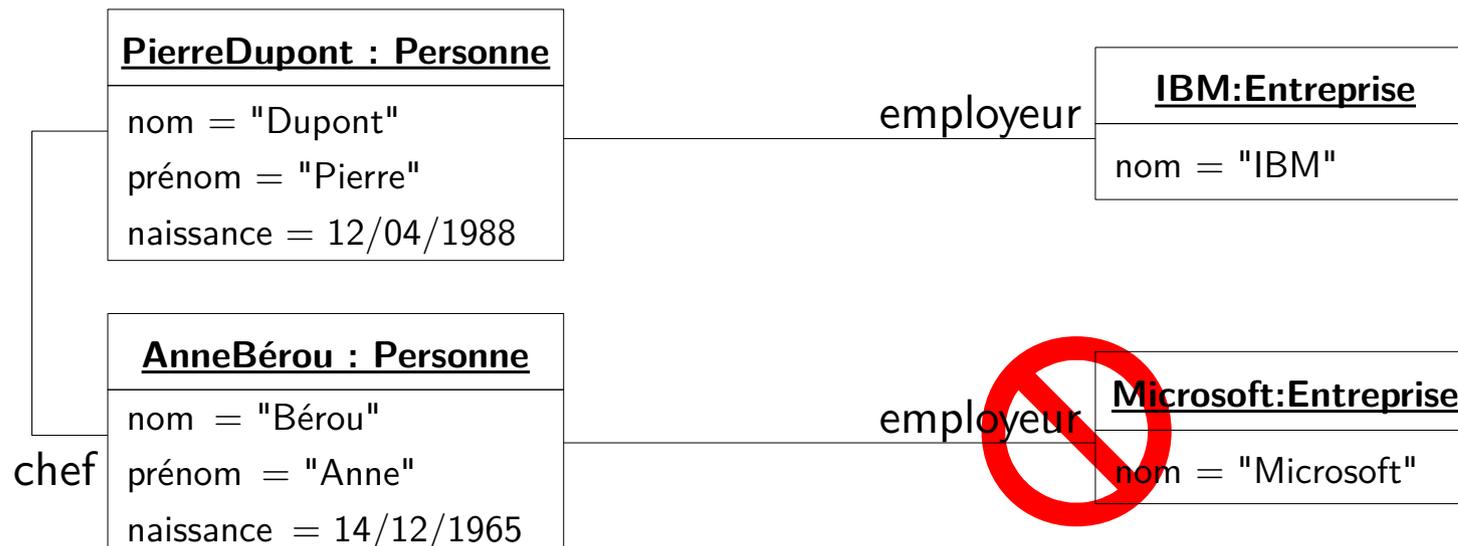
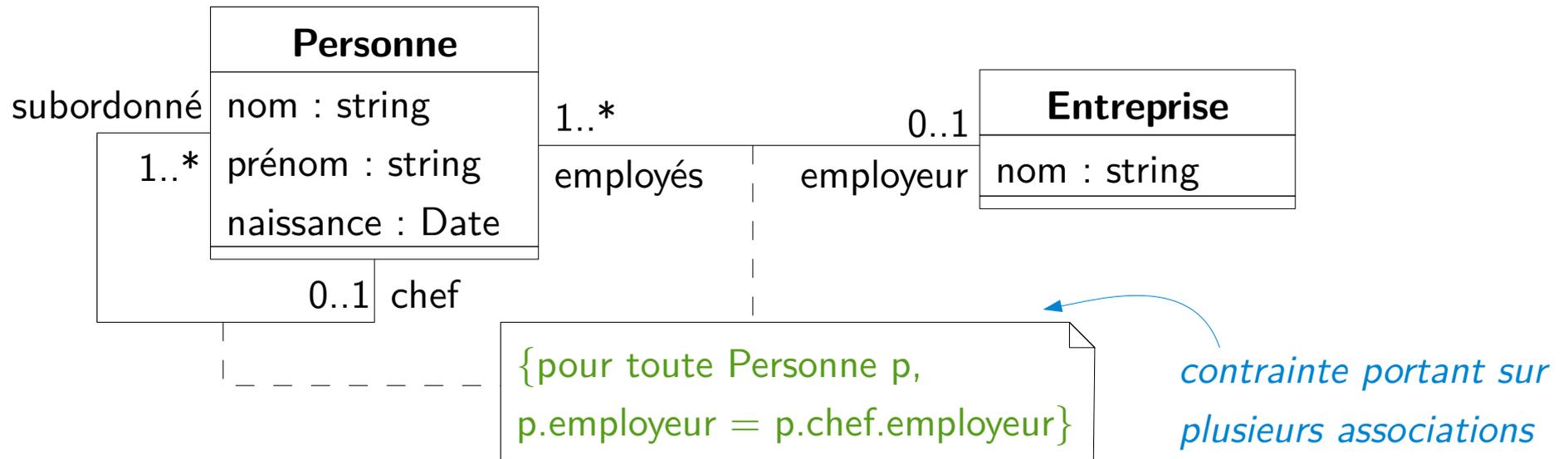


multi-ensemble ordonné

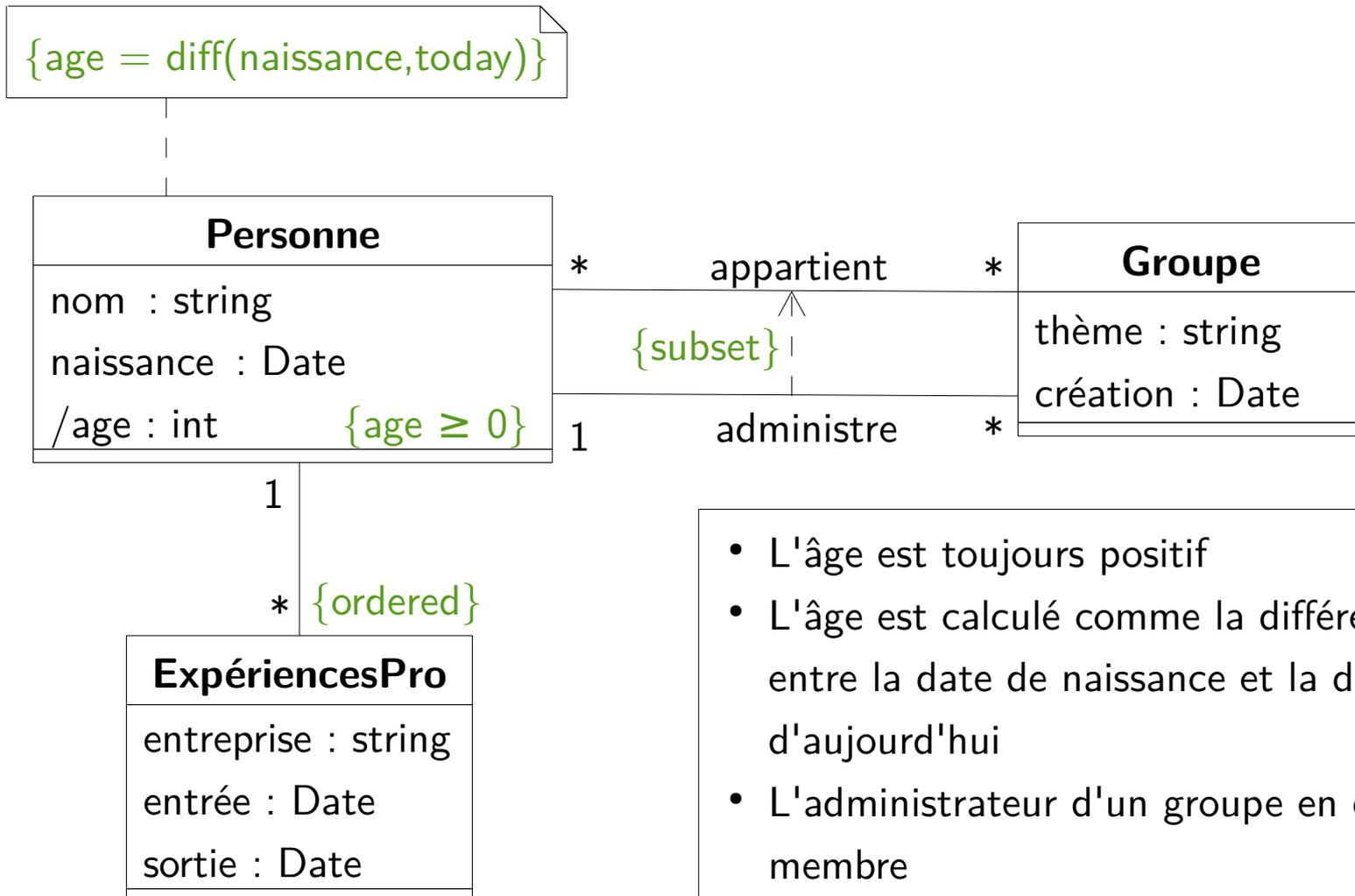
Contraintes sur les associations



Contraintes sur les associations



Contraintes, invariants



- L'âge est toujours positif
- L'âge est calculé comme la différence entre la date de naissance et la date d'aujourd'hui
- L'administrateur d'un groupe en est membre
- On a accès aux expériences professionnelles dans l'ordre de la date d'entrée