

# Concepton Objet

Cours de DUT Informatique

s.Benbernou, M.Ouziri

IUT de Paris - Université Paris Descartes

Expression de contraintes avec le langage

OCL – Object Constraint Language

# Expression de contraintes en UML

## ☞ Objectif

- Exprimer des règles de cohérence des données
- Le diagramme de classes ne permet pas d'exprimer tous types de règles
- OCL : extension de UML

## ☞ Deux types de contraintes étudiés

- Contraintes statiques : multiplicités et contraintes de l'héritage
- Contraintes dynamiques : expressions logiques dans le langage OCL – Object Constraint Language

# Expression de contraintes

## Contraintes statiques

### ☞ Contrainte {ordered}



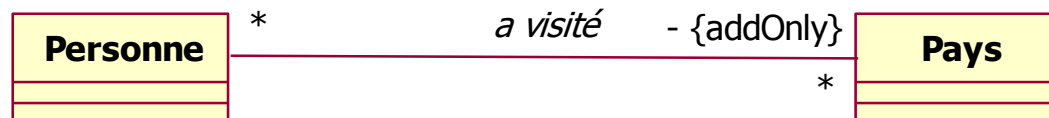
Les objets de l'association sont ordonnés

### ☞ Contrainte {frozen}



L'association n'est pas modifiable

### ☞ Contrainte {addOnly}



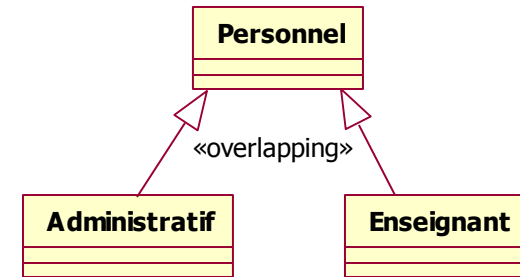
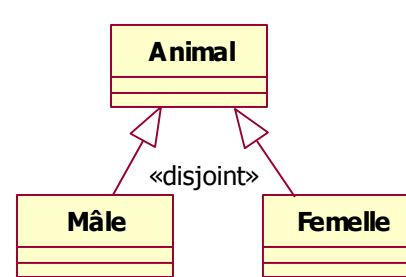
On ne peut que ajouter des objets à l'association (pas de suppression d'objets)

# Expression de contraintes

## Contraintes statiques

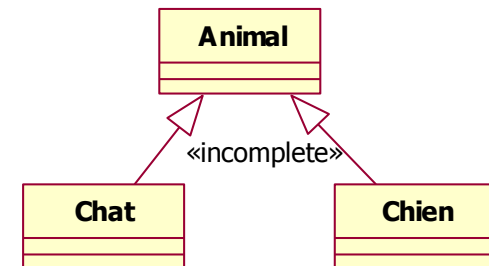
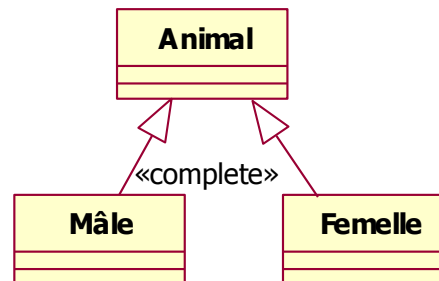
### ☞ Disjoint (*resp.* Overlapping)

Les instances des sous-classes sont disjointes (*resp.* se chevauchent)



### ☞ Complete (*resp.* Incomplete)

Toute instance de la super-classe est (*resp.* n'est pas) obligatoirement une instance d'au moins une de ses sous-classes



### ☞ Ces contraintes peuvent être combinées

# Expression de contraintes

## Le langage de contraintes OCL

### ☞ Contraintes dynamiques

- OCL – Object Constraint Language
- Syntaxe pseudo-mathématique et pseudo-algorithmique
- Formule de la logique propositionnelle
- Elle s'expriment à partir d'un objet

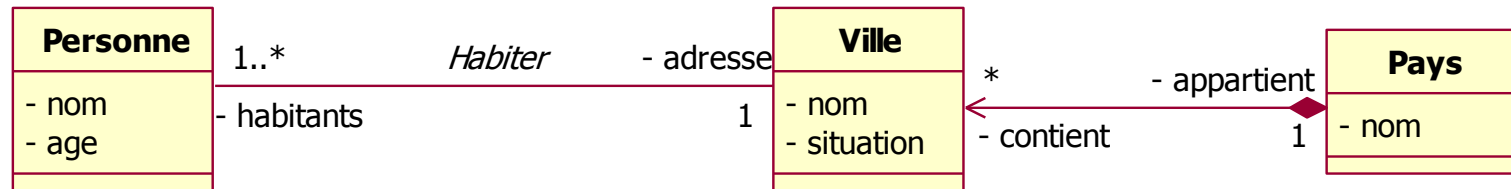
### ☞ Syntaxe

- Contrainte = Contexte + Invariant
  - Contexte : objet concerné par la contrainte
  - Invariant : expression (mathématique ou algorithmique) de la contrainte
- Self : pour désigner l'objet concerné par la contrainte (*this*)
- Utilise les opérateurs logiques (ET, OU, IMPLIES, etc.)

# Expression de contraintes

## Le langage de contraintes OCL

☞ Quelques éléments du vocabulaire OCL



– Pour un objet de la classe *Personne* :

*self* : référence un objet de la classe *Personne*

*self.nom* : référence le nom d'une personne (valeur de l'attribut `nom`)

*self.adresse* : référence la ville où habite la personne (type *Ville*)

*self.adresse.appartient* : référence le pays où habite la personne (type *Pays*)

– Pour un objet de la classe *Ville* :

*self.nom* : référence le nom de la ville

*self.habitants* : référence la liste des (objets) personnes habitants la ville (type *Set*)

# Expression de contraintes

## Le langage de contraintes OCL

### ☞ Contraintes simples sur les attributs d'une classe

- Un étudiant a au moins 16 ans

*context : Etudiant*

*invariant : self.age ≥ 16*

- Les étudiants en apprentissage ont au moins 20 ans

*context : Etudiant en App*

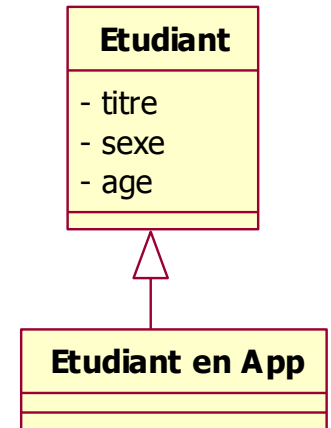
*invariant : self.age ≥ 20*

- Si le sexe est Féminin alors le titre est *Mme* ou *Melle* et si le titre est Masculin alors le titre est *Mr*

*context : Etudiant*

*invariant : self.sexe = Féminin IMPLIES (self.titre = Mme OR self.titre = Melle)*

*self.sexe = Masculin IMPLIES self.titre = Mr*



# Expression de contraintes

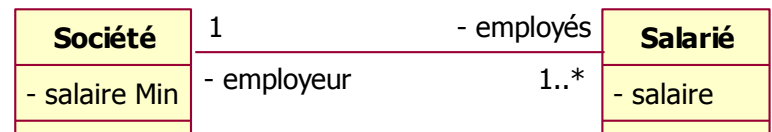
## Le langage de contraintes OCL

### ☞ Contraintes sur les associations simples (multiplicité max. à 1)

- Les employés touchent un salaire supérieur au salaire minimum fixé pour leur société

*context : Salarié*

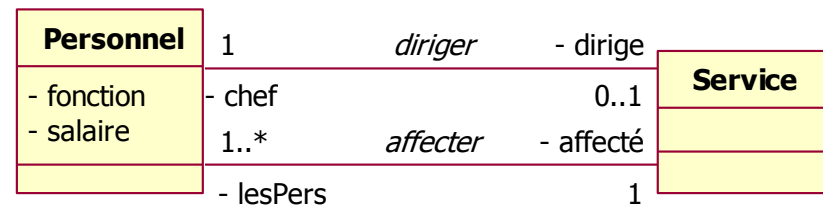
*invariant : self.salaire ≥ self.employeur.salaire Min*



- Les services sont dirigés par des cadres

*context : Service*

*invariant : self.chef.fonction = Cadre*



- Les personnels d'un service touchent un salaire inférieur à celui de leur chef de service
- Tous les cadres dirigent un service

# Expression de contraintes

## Le langage de contraintes OCL

### ☞ Contraintes sur les associations multiples (multiplicité max. à \*)

- La contrainte est exprimée sur un rôle de type *Collection*
- Utiliser les primitives de collections (voir slide suivant)

### ☞ Exemple

- Le salaire minimum d'une société doit être inférieur au salaire de tous ses employés



*context* : Société

*invariant* :  $self.salaire\ Min < self.employés.salaire$  Faux !

car *self.employés* est de type *Collection* (*Set* ou *List*)

# UML

## DC – Diagramme de classes

### ☞ Quelques primitives s'appliquant sur les collections

`isEmpty` : retourne vrai si la collection est vide

`size` : retourne le nombre d'éléments de la collection

`forall` : retourne vrai si tous les éléments vérifient la contrainte spécifiée

`exists` : retourne vrai si au moins un élément respecte la contrainte spécifiée et faux sinon

`includes` (`excludes`) : retourne vrai si la collection contient l'élément spécifié

`includesAll` : retourne vrai si la collection contient tous les éléments spécifiés

`union/intersection` : retourne l'union/intersection avec une autre collection

`select` : retourne le sous-ensemble de la collection dont les éléments vérifient la contrainte spécifiée

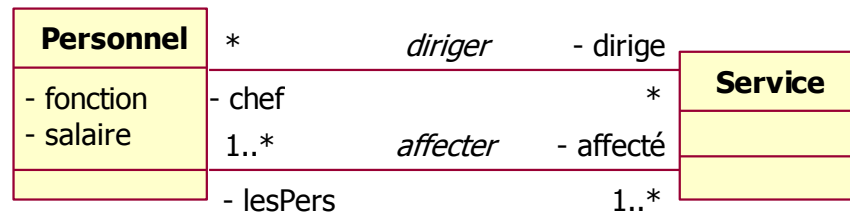
`reject` : idem mais ne garde que les éléments ne vérifiant pas la contrainte

# Expression de contraintes

## Le langage de contraintes OCL

☞ Exemples de contraintes sur des collections : *isEmpty* et *size*

– Les techniciens ne peuvent pas être dirigeants de service



*context : Personnel*

*invariant : self.fonction = technicien IMPLIES self.dirige.isEmpty()*

– Les cadres ne sont affectés que à un seul service au maximum et les techniciens à trois

*context : Personnel*

*invariant : self.fonction = cadre IMPLIES self.affecté.size() <= 1*

*self.fonction = technicien IMPLIES self.affecté.size() <= 3*

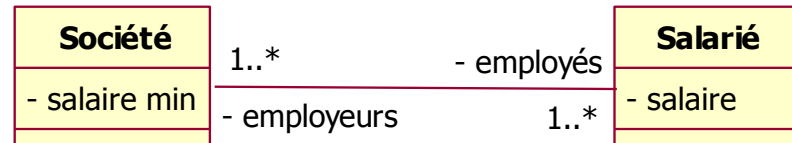
# Expression de contraintes

## Le langage de contraintes OCL

☞ Exemples de contraintes sur des collections : *forAll* et *exists*

- Le salaire minimum d'une société doit être inférieur au salaire de tous ses employés

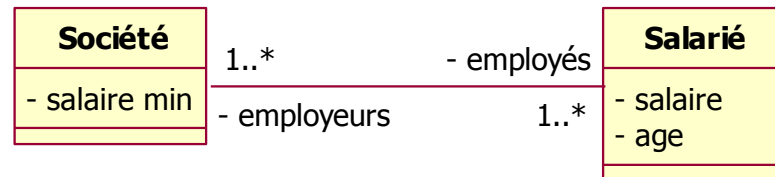
*context* : *Société*



*invariant* : *self.employés.forAll (s:Salarié | self.salaire Min < s.salaire)*

- Chaque société emploie au moins un salarié sénior (de plus de 50 ans)

*context* : *Société*

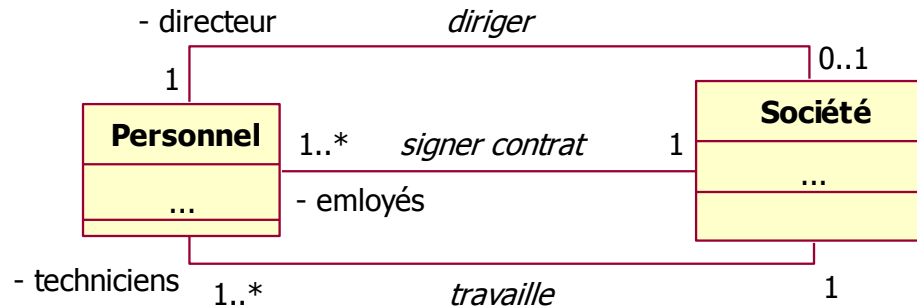


*invariant* : *self.employés.exists (s:Salarié | s.age > 50)*

# Expression de contraintes

## Le langage de contraintes OCL

☞ Exemples de contraintes sur des collections : *includes* et *includesAll*



– Les sociétés signent obligatoirement un contrat avec leur dirigeant

*context* : Société

*invariant* : *self.employés.Includes (self.directeur)*

– Les sociétés signent obligatoirement un contrat avec tous leurs techniciens

*context* : Société

*invariant* : *self.employés.IncludesAll (self.techniciens)*

# Expression de contraintes

## Le langage de contraintes OCL

☞ Exemples de contraintes sur des collections : *select* et *reject*



– Chaque société emploie au moins 10 salariés séniors

*context* : Société

*invariant* : [ *self.employés.select* (s:Salarié | s.age > 50) ].size() > 10

– Une société n’emploie que des salariés jeunes (de moins de 35 ans)

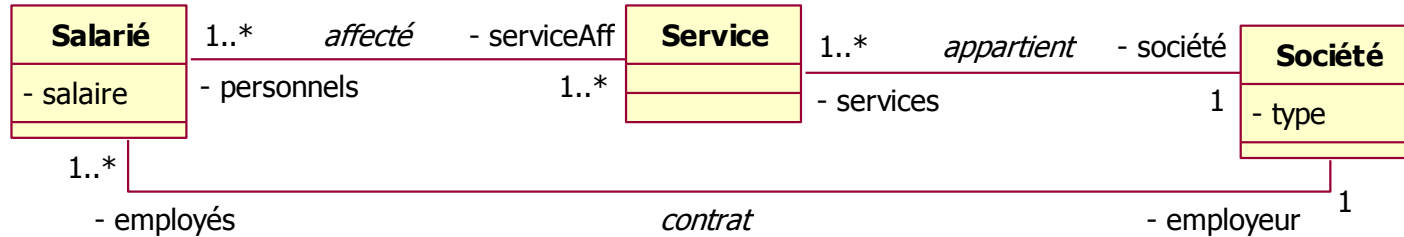
*context* : Société

*invariant* : [ *self.employés.reject* (s:Salarié | s.age < 35) ].isEmpty()

# Expression de contraintes

## Le langage de contraintes OCL

### ☞ Exercice



- L’effectif d’une société individuelle ne peut pas dépasser 15 salariés
- Un personnel est affecté aux services de la société qui l’emploie
- Le nombre de services d’une société est strictement inférieur à son effectif

