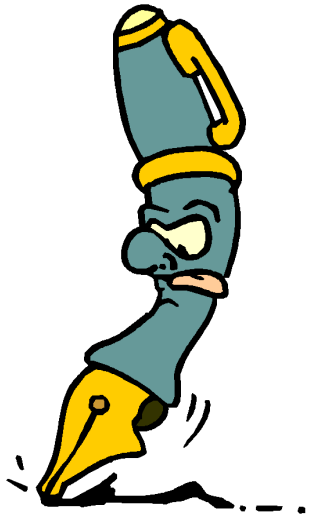
Object Constraint Language

Corrigé des exercices

Philippe Collet

COO – L3 Info & MIAGE

Novembre 2012

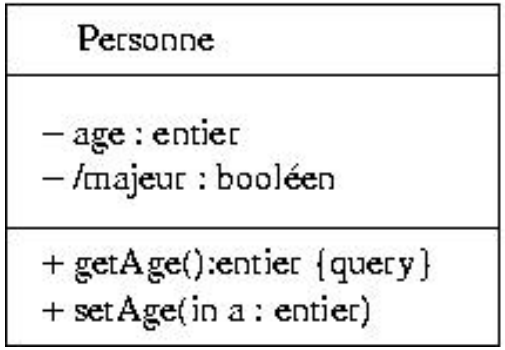


Université Nice Sophia Antipolis

Exercice

- **Ajoutez un attribut mère de type Personne dans la classe Personne.**

- **Ecrivez une contrainte précisant**
  - que la mère d'une personne ne peut être cette personne elle-même
  - et que l'âge de la mère doit être supérieur à celui de la personne

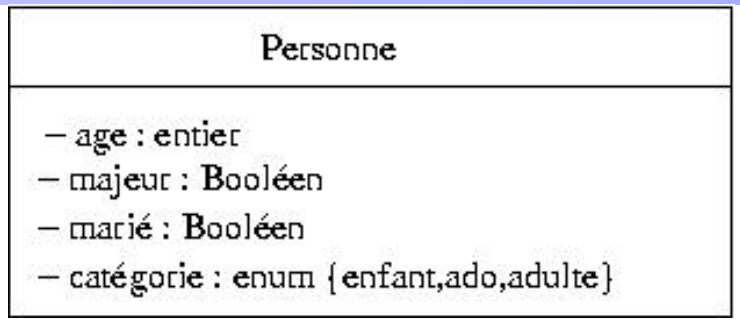



```
context Personne inv:
self.mere <> self and self.mere.age > self.age
```

Exercice

Avec la classe Personne « étendue »

- Indiquez qu'une personne mariée est forcément majeur




```
context Personne inv:
marié  implies majeur
```

- Trouvez une version plus compacte de l'expression suivante

```
context Personne inv majeurIf:
if age >=18 then majeur=vrai
else majeur=faux endif
```

```
context Personne inv:
majeur =  age >= 18
```

(mauvais) exercice

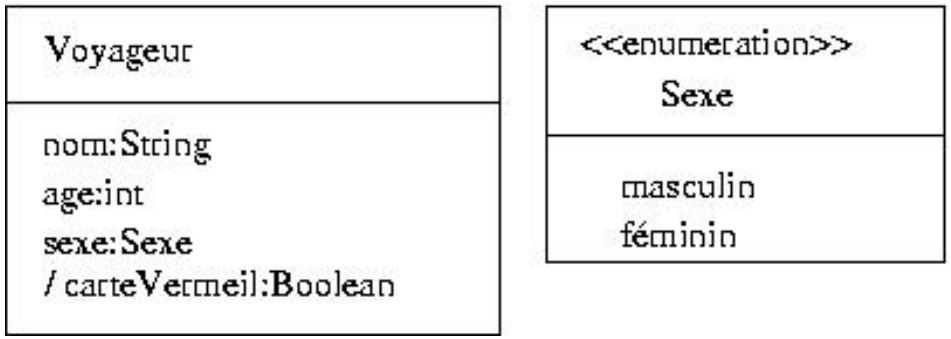
- **En supposant l'existence**
  - d'un attribut hauteur dans la classe Rectangle
  - d'une méthode hauteur():Réel dans Polygone

- **Ecrivez un invariant dans Polygone disant que le résultat de hauteur():Réel vaut hauteur pour les polygones qui sont des rectangles, sinon 0**

```
context p : Polygone::hauteur() inv:
if p.oclIsKindOf(Rectangle)
        then result=p.oclAsType(Rectangle).hauteur
        else result=0
endif
```

- **Ceci un exemple de très mauvaise conception objet !**

Exercice

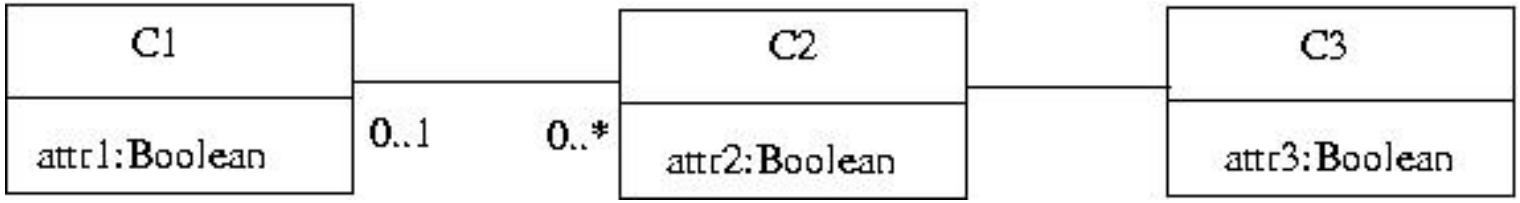


- **Ecrivez la contrainte qui caractérise l'attribut dérivé carteVermeil**
  - Un voyageur a droit à la carte vermeil si c'est une femme de plus de 60 ans ou un homme de plus de 65 ans.

```
context Voyageur inv :
carteVermeil = ((age >= 65) or
                ((sexe = Sexe::féminin) and (age >= 60)) )

-- Cette contrainte peut également s'écrire avec derive.
```

Navigation - ambiguïtés



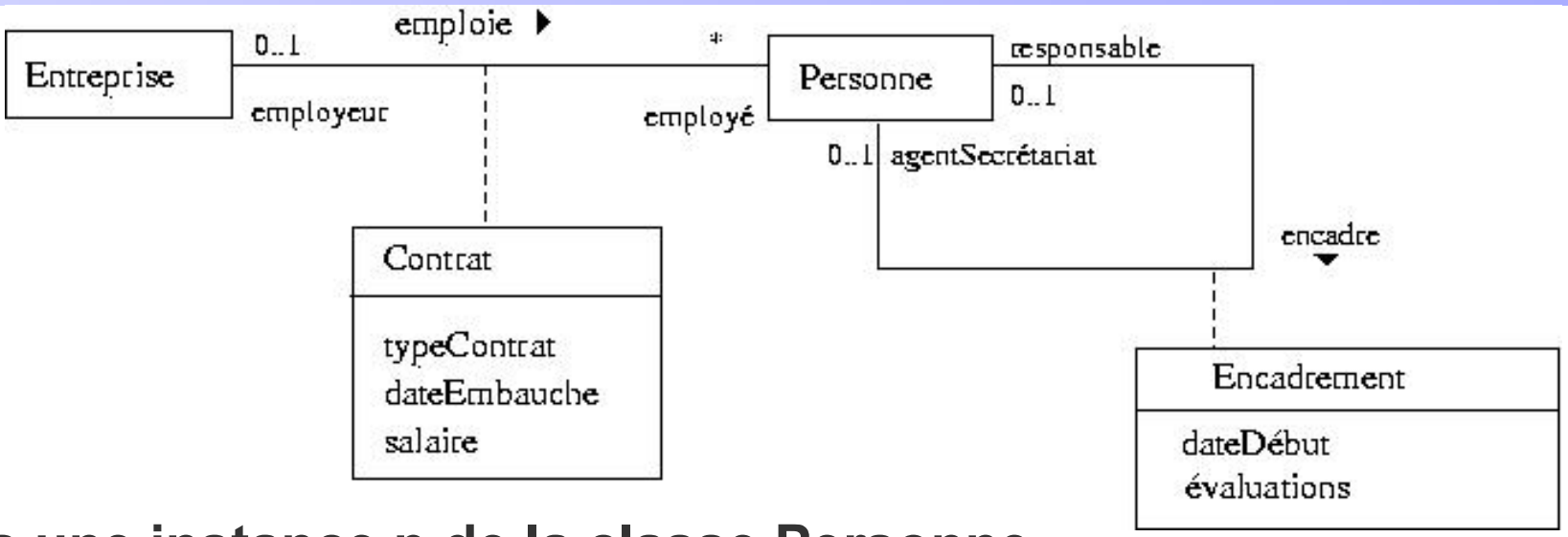

```
context C1 inv :
c2.attr2=c2.c3.attr3

context C2 inv :
attr2=c3.attr3
```

- **Les deux contraintes ci-dessus sont-elles équivalentes ?**

NON : la première dit que pour des instances de C2 et C3 liées avec une instance de C1, les attributs attr2 et attr3 sont égaux, mais n'impose rien à des instances de C2 et C3 non liées à une instance de C1 (et il y en a à cause de la multiplicité).
La deuxième dit que pour tout couple d'instances de C2 et C3 liées, ces deux attributs sont égaux.

Exercice

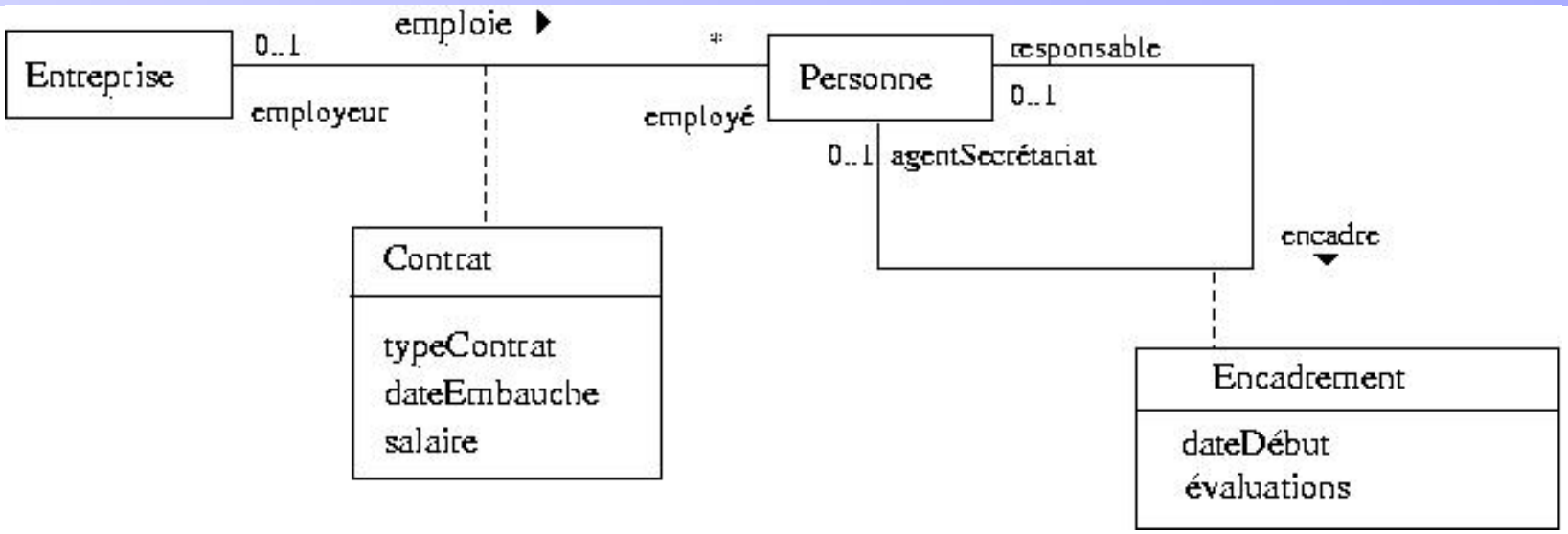


- **Depuis une instance p de la classe Personne**
  1. Comment naviguer vers l'objet Encadrement de son responsable ?
  2. Comment naviguer vers l'objet Encadrement de son agentSecrétariat ?

```
context p : Personne  …
   p.encadrement[responsable]        -- 1

   p.Encadrement[agentSecrétariat]   -- 2
```

Exercice

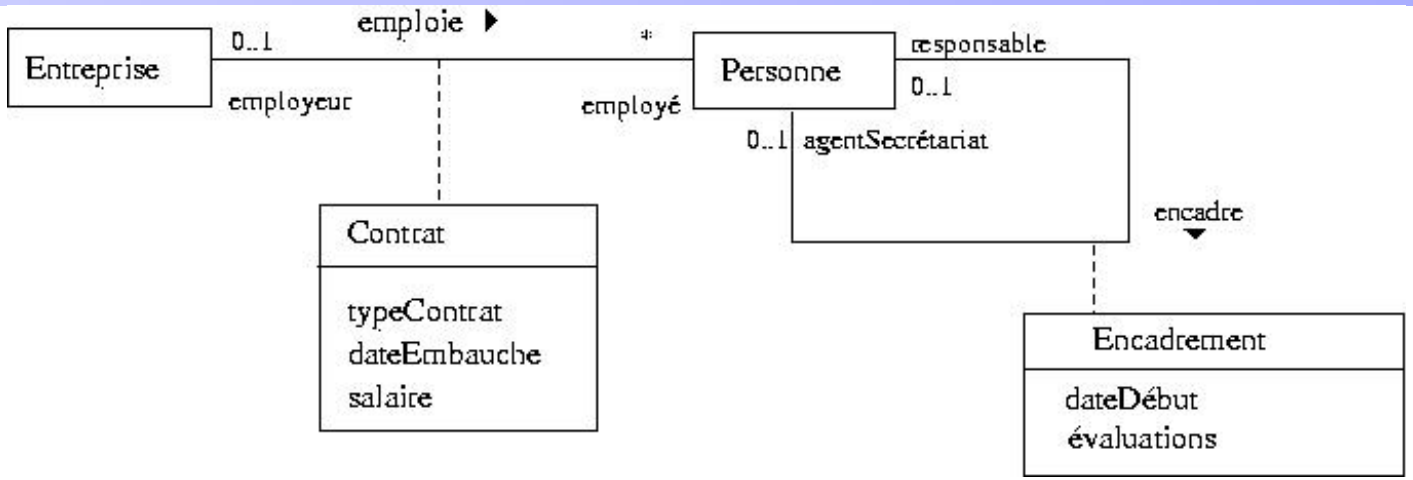


- Le salaire d'un agent de secrétariat est inférieur à celui de son responsable ?
- Un agent de secrétariat a un type de contrat ‘agentAdministratif‘ (String) ?

```
context e : encadrement inv :
e.responsable.contrat.salaire >= e.agentSecrétariat.contrat.sal
aire

context e : encadrement inv :
e.agentSecrétariat.contrat.typeContrat='agentAdministratif'
```

Exercice



De l'aide ?
Un diagramme d'objets

- Un agent de secrétariat a une date d'embauche antérieure à la date de début de l'encadrement (on suppose que les dates sont des entiers)

```
context e : Encadrement inv :
e.agentSecrétariat.contrat.dateEmbauche <= e.dateDebut
```

- Même chose dans le contexte de la classe Personne

```
context p : Personne inv :
p.agentSecrétariat.contrat.dateEmbauche
            <= p.encadrement[agentSecrétariat].dateDebut
```

Exercice

- Imaginez une classe Etudiant, disposant de 3 notes et munie d'une opération mention qui retourne la mention de l'étudiant sous forme d'une chaîne de caractères.

- Ecrivez les contraintes en utilisant let et result pour écrire la postcondition de mention

```
context Etudiant :: mention() : String    post:
let moyenne : Real =(note1+note2+note3)/3 in
 if (moyenne >= 16)
           then result='très bien'
           else if  (moyenne >= 14)
                          then result='bien'
                         else result='moins bien'
                  endif
endif
```

Exercice

- Ecrivez, dans le contexte de la classe Collection, l'opération size à l'aide de l'opération iterate

```
context Collection :: size()
post : result = self->iterate(elem; acc:integer=0| acc
+1)
```

- Ecrivez, dans le contexte de la classe Collection, l'opération forAll à l'aide de l'opération iterate.

```
context Collection :: forAll(expr)
post : result = self->iterate(elem; acc : Boolean=true |
                                    acc and expr)
```

Exercice

- **Quelle est la signification de cette expression ?**

```
context Personne inv:

Personne.allInstances()->forAll(p1, p2 |

          p.1 <> p2 implies p1.nom <> p2.nom )
```

- **Comment l'écrire avec isUnique ?**

```
context Personne inv:
Personne.allInstances()->isUnique(nom)
```