

## EISTI 2009-2010 – Méthodologies d'Analyse

### TD4 : OCL – Object Constraint Language

*NB : il faudra bien expliquer aux étudiants que ces solutions n'utilisent que les éléments d'OCL qu'on a vu en cours cette semaine (i.e. jusqu'à la page 18-19 des transparents en français, ou page 21 de la traduction en anglais) mais que d'autres seront possibles dès qu'on aura vu les opérateurs sur les listes (isEmpty(), sum(), forAll() etc.). Il faut aussi rappeler que  $A \Rightarrow B \Leftrightarrow \text{NOT } A \text{ OR } B$ .*

#### Person & Company

Etudiez le diagramme de classe UML qui se trouve au verso de cette feuille et écrivez les contraintes suivantes :

1. Initialement, toutes les personnes doivent être des femmes.

```
context Person
init : self.gender = Gender::female
```

2. Initialement le salaire d'un employé est de 1000 euros.

```
context Job
init : self.salary = 1000
```

3. Le directeur d'une société doit avoir plus de 40 ans.

```
context Company::manager : Person
inv : self.age >= 40
```

-- la notation '::' sert à sélectionner comme contexte un attribut ou une méthode de la classe Company

4. L'âge d'un individu est fonction du moment présent et de sa date de naissance.

```
context Person
derive : self.age = Date.now() - self.birthdate -- léger abus de notation : on considère que la
                                                -- différence entre deux dates est un nombre entier
```

5. Si le chiffre d'affaires (turnover) d'une société est supérieur à 1 million d'euros, elle doit avoir plus que 10 employés.

```
context Company
inv : NOT self.turnOver() > 1.000.000 OR self.numberOfEmployees > 10
```

6. Tous les employés d'une société doivent avoir plus de 18 ans.

```
context Job
inv : self.employee.age >= 18
```

7. On ne peut pas commencer à travailler avant sa date de naissance.

```
context Job
inv : self.startDate > self.employee.birthDate
```

8. Un homme est marié avec une femme et une femme avec un homme.

```
context Person
inv : not self.isMarried OR
      (self.gender = Gender::male AND self.wife.gender = Gender::female) OR
      (self.gender = Gender::female AND self.husband.gender = Gender::male)
```

Mais aussi (peut être mieux)

```
context Marriage
inv : self.husband.gender=Gender::male AND self.wife.gender=Gender::female
```

9. Pour être mariés, il faut avoir plus que 18 ans.

```
context Person
```

```
inv : NOT self.isMarried OR self.age >= 18
```

10. On ne peut pas commencer à travailler le jour de son mariage.

```
context Job
```

```
inv : NOT self.startDate = self.employee.marriage.date
```

