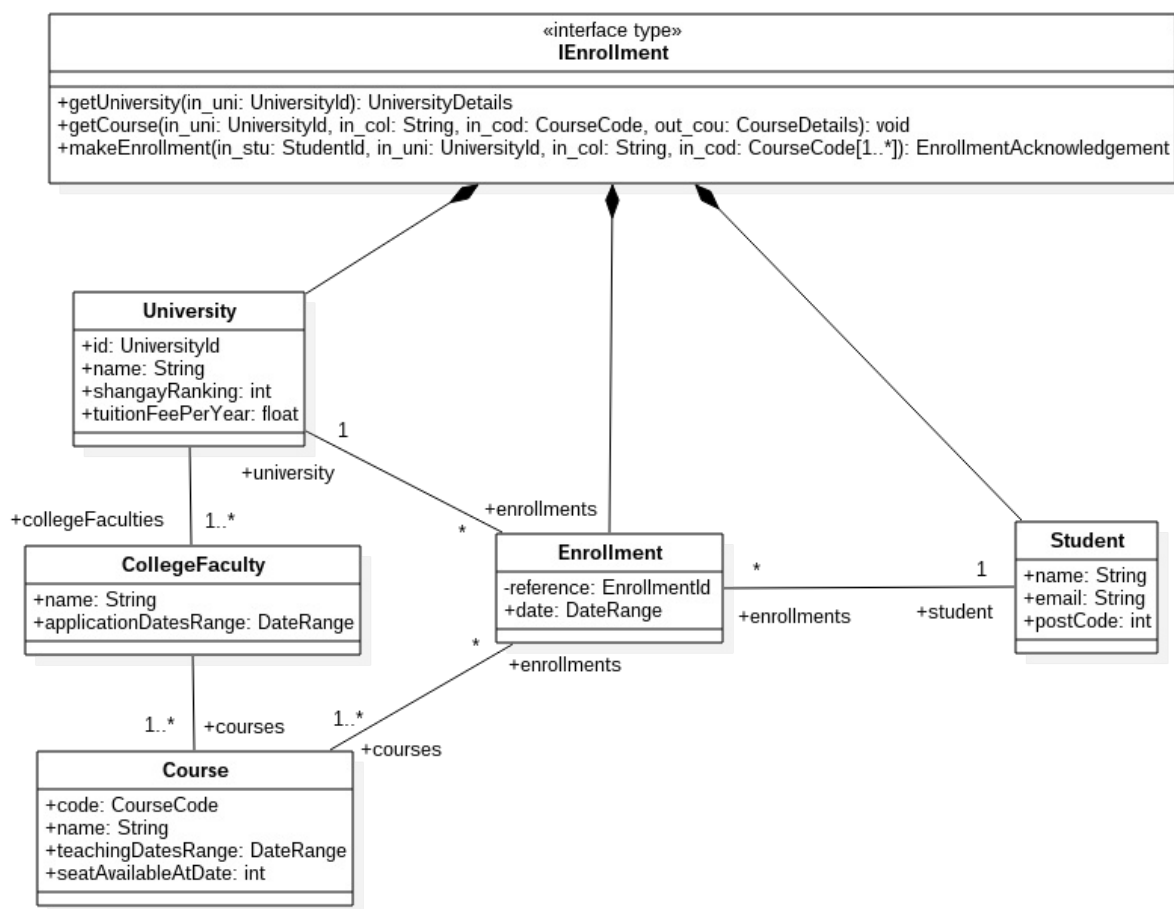


DSS - Ejercicio especificación de interfaz en UML y OCL

Francisco Javier Bolívar Lupiáñez

I. Ejercicio de especificación de interfaz de un sistema de matriculación en una universidad



Invariantes

- Todas las universidades deben tener un id distinto:

```
context University inv:
University.allInstances()->forAll(u1, u2 | u1 <> u2 implies u1.id <> u2.id)
```

- En una universidad no hay dos facultades con el mismo nombre:

```
context University inv:
  collegeFaculties->forAll(cf1, cf2 | cf1 <> cf2 implies cf1.name <> cf2.name)
```

- Todos los estudiantes tienen un email distinto:

```
context Student inv:
  Student.allInstances()->forAll(s1, s2 | s1 <> s2 implies s1.email <> s2.email)
```

- Un estudiante solo puede tener una matriculación para un mismo rango de fechas:

```
context Student inv:
  self.enrollments->forAll(e1, e2 | e1 <> e2 implies e1.date <> e2.date)
```

- Los rangos de fechas de los cursos deben coincidir con los de la matriculación:

```
context Enrollment inv:
  self.courses->forAll(c | c.teachingDatesRange->includes(self.date))
```

- Los rangos de fecha de las facultades deben coincidir con el de sus cursos:

```
context CollegeFaculty inv:
  self.courses->forAll(c | c.teachingDatesRange->includes(self.applicationDatesRange))
```

- Las asignaturas de una matriculación tienen que ser de la misma universidad que la matriculación:

```
context Enrollment inv:
  self.courses->forAll(c | university.collegeFaculties->forAll(cf)->
    exists(cf | cf.courses->includes(c)))
```

- Los asientos disponibles en un curso no pueden ser negativos:

```
context Course inv:  
self.seatAvailableAtDate >= 0
```

II. Ejercicio de especificación utilizando OCL de la operación getStudentDetails(...)

```
context: IStudentManagement::getStudentDetails(in stu: StudentIdentifier): StudentDetails  
pre:  
  -- 'stu' es un identificador valido de estudiante  
  student->exists(s | s.id = stu)  
post:  
  -- los detalles devueltos tras la ejecucion coinciden con los del estudiante  
  -- cuyo identificador es 'stu'  
  Let elEstudiante = student->select(s | s.id = stu) in  
    -- especificar el resultado  
    result.name = stu.name  
    result.email = stu.email  
    result.postCode = stu.postCode
```