

Lab 02: Implementin Class Relations

[lab02.pdf](#)

UPF-POO22-G202-21

Ariadna Prat / Chris Matienzo

5. Documentation

5.1 El programa consiste en diseñar un sistema de gestión para una universidad, esta organiza los diferentes alumnos, profesores, clases, asignaturas, entre otros, leyendo un archivo .xml donde se almacenan dichos datos, mediante el uso de diferentes clases y sus métodos podemos agregar los datos disponibles.

Las clases definidas son:

Clases principales

- `public class Student`
 - `public Student (String name, int age)` // Es el método constructor de la clase Student, esta inicializa
 - `public void addEnrollment (Enrollment enrollment)` // Agrega a la LinkedList de enrollment los datos leídos del archivo enrollments.xml
 - `public String toString ()` // Método para pasar a cadena de texto dado un objeto
 - `public String getCourses ()` //
- `public class Teacher`
 - `public Teacher (String name)` // Método constructor que inicializa el nombre del profesor

- `public void addAssignment (Assignment assignment)` // Agrega a la LinkedList de assignment los datos leídos del archivo assignments.xml
- `public String toString ()` // Método para pasar a cadena de texto dado un objeto
- `public class Classroom`
 - `public Classroom (String code)` // Método constructor que inicializa el código del aula
 - `public void addLecture (Lecture lecture)` // Agrega a la LinkedList de lecture los datos leídos del archivo lectures.xml
 - `public String toString ()` // Método para pasar a cadena de texto dado un objeto
- `public class Course`
 - `public Course (String name)` // Método constructor de la clase Course que inicializa
 - `public void addLecture (Lecture lecture)` // Constructor que inicializa lecture
 - `public void addAssignment (Assignment assignment)` // Agrega a la LinkedList de assignment los datos leídos del archivo assignments.xml
 - `public void addEnrollment (Enrollment enrollment)` // Agrega a la LinkedList de enrollment los datos leídos del archivo enrollments.xml
 - `public String toString ()` // Método para pasar a cadena de texto dado un objeto

Clases secundarias

- `public class Assignment`
 - `public Assignment (LinkedList<String> groups)` // Método constructor que recibe una LinkedList de grupos y se inicializa recorriendo dicha lista
 - `public void addTeacher (Teacher teacher)` // Agrega a teacher los datos leídos del archivo teachers.xml
 - `public void addCourse (Course course)` // Agrega a course los datos leídos del archivo courses.xml
- `public class Enrollment`

- `public Enrollment (String seminarGroup) // Método onstructor de la clase Enrollment que inicializa los seminarGroup`
- `public void addCourse (Course course) // Agrega a course los datos leídos del archivo courses.xml`
- `public void addStudent (Student student) // Agrega a student los datos leídos del archivo students.xml`
- `public class Lecture`
 - `public Lecture (String group, int timeSlot, int type) //`
 - `public void addClassroom (Classroom classroom) // Agrega a classroom los datos leídos del archivo classrooms.xml`
 - `public void addCourse (Course course) //`

Clase universtity

- `public class University`
 - `public Univesity () // Método constructor donde hacemos la lectura de los diferentes archivos .xml recorriendo y almacenandolos en un array para después agregarlos a sus respectivas listas (p. ej. studentList.add(student))`
 - `public LinkedList<String> coursesOfStudent (String student) // Método de consulta para obtener los cursos de un estudiante`
 - `public LinkedList<String> TeachersOfCourse (String teacher) // Método de consulta para obtener los cursos de un profesor`
 - `public LinkedList<String> getStudents () //`
 - `public LinkedList<String> getCourses () //`
 - `public LinkedList<String> getClassrooms () //`
 - `public LinkedList<String> getTeachers () //`

5.2 Describe una posible solución y una descripción de la solución escogida y porqué?

Nos encontramos con que el tipo de dato de los archivos .xml, en los datos de Classroom presentaban un tipo de dato incompatible con el que tratábamos en el código. En el archivo se presentaba con el siguiente formato de Code: 10.100,

10.101, ... mientras que en el código se nos pedía que se casteara a tipo integer, por lo que decidimos pasarlo a formato String para tratarlo. Otra alternativa podría haber sido usando el siguiente código para “eliminar” el punto.

```
// remove "." parse to integer
String[] parts = array[0].split(".");
String part1 = parts[0];
String part2 = parts[1];
String join = part1 + part2;
```

5.3 Una conclusión de lo bien que la solución funciona. Alguna dificultad?

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Classroom.addLecture(Lecture)" because "classroom" is null
    at University.<init>(University.java:73)
    at TestUniversity.main(TestUniversity.java:4)
```

Constantemente al ejecutar nos retornaba que la clase University devolvía un classroom null al no ingresar ningún Lecture, lo que producía este error era la falta de declaración del método toString() en las clases principales.

```
public String toString(){
    return this.code;
}
```