Interfaces and Inheritance

In the following text, we describe the university domain. Model this domain using classes, abstract classes, interfaces. In this exercise, we do not want an implementation but just a chart that represents the classes, interfaces, etc. Provide the relationship between the different components, and if possible, provide the signature of the methods that would be needed.

We want to implement an application for managing university staff and the different training programs. We will have a `Student` class. For now, a student will have a name, an id, and a training program. A training program has a list of courses. Each course belongs to a domain (Management, Finance, Economy, Computer Science, Mathematics), so we will implement a `Domain` class and a `TrainingProgram` class.

A teaching staff is modelled by a name, an office number, and a domain. There are different levels of teaching staff. Adjuncts only do teaching. Assistant professors, associate professors, and full professors do research and teaching. Adjuncts cannot have a PhD students, the others can. Associate professors and professors can have administrative duties. Professors can be training program chair.

All teaching staff have the following duties

- they give lectures
- they evaluate the students

A training program chair have four duties

- they manage the admission of students in the program
- they sign the diploma of the students
- they manage the list of students (we will use an array of size `max_Num_students`).
- they manage the list of courses (we will use an array of size `max_Num_courses`).

If a staf has an administrative duty, she has meetings.