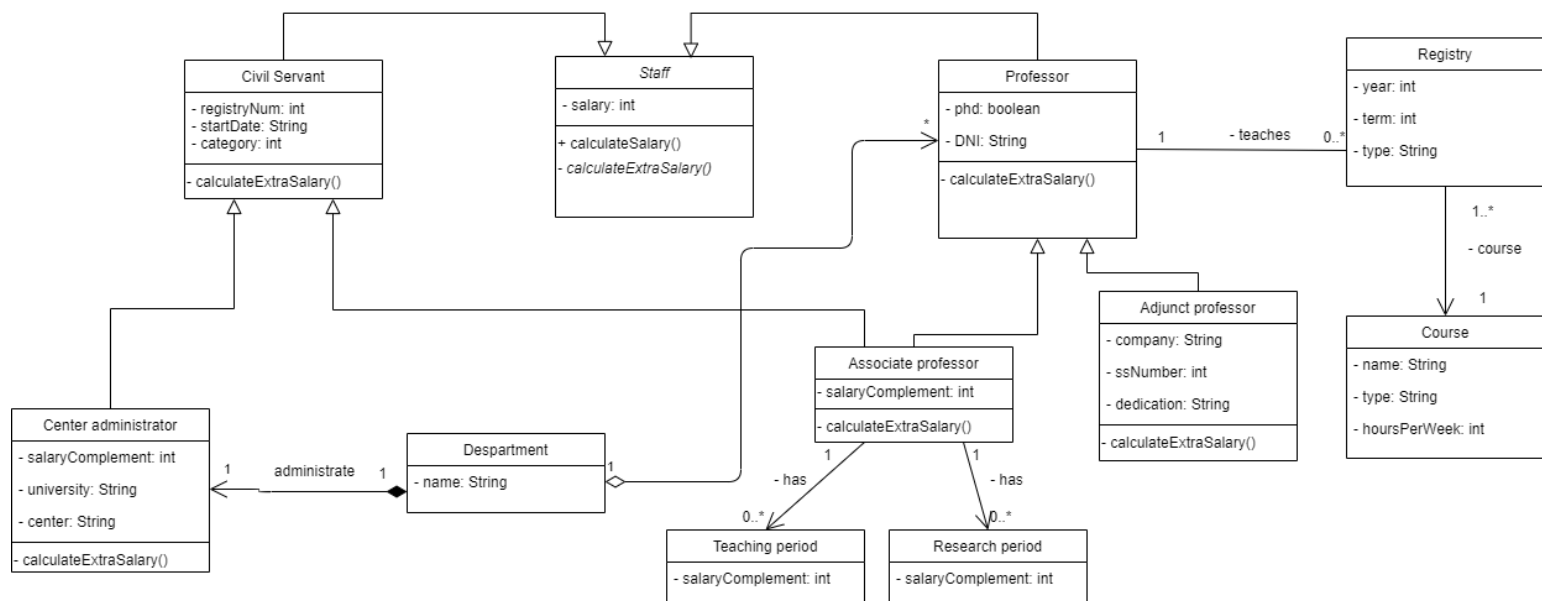


EXERCISE 3

A)



In this exercise, we need to design an application for the managing of the staff of public universities. For implementing it, we created the abstract class *Staff* where two classes called *Civil Servant* and *Professor* are inherited. *Staff* contains one attribute and two methods use by its subclasses.

Then, from *Civil Servant*, there are inherited normal subclasses as *Center administrator* which administrates a department created in another class called *Department*, and also it is inherited the class *Associate professor* which is also inherited from *Professor* (and that will be a problem in implementation because of multiple inheritance). *Associate professor* will have the registry of its teaching and research periods, divided in two subclasses for adding the unique methods for calculating the salary complement of each of them.

The *Professor* class will be connected to another class called *Registry* that has all the courses in which that professor has given class, and all the information about them will be given in the *Registry* class by its attributes.

The method `calculateExtraSalary` is useful to calculate supplements from different kinds of Staff.

Juan Luis Sanz - juanluis.sanzc@estudiante.uam.es

Rodrigo Juez - rodrigo.juezh@estudiante.uam.es

B)

For implementing in Java, we will need the setters and getters, and many more functions in order to make it work efficiently. Then, as java doesn't accept classes belonging to more than one class (multiple inheritance) we will have to divide in some way the class *Associate professor* as it belongs to *Civil servant* and *Professor*. This following design could be a good implementation, where the *Associate professor* class inherits from a class called *Civil Servant Professor* which is a copy of *Civil Servant* and *Professor* classes together and inherits directly from the *Staff* class, it's also abstract as we don't want to be able to create an object to just be that and the objects will be created by Associate Professor.

