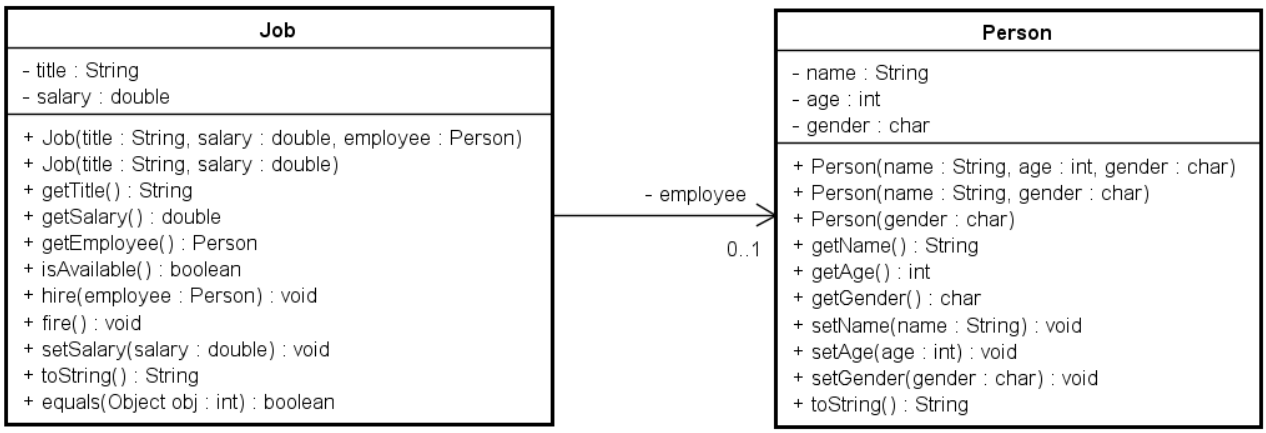
Exercise: Job (Association)

Implement the class `Job` exactly as shown in the UML class diagram below (Note: class `Person` can be taken directly from one of your exercises no matter if it is exactly as shown – could be the version with association to `Name` and `MyDate`):



Create the class `Job` such that you implement:

a) three instance variables, a job title, a monthly salary and an employee, name them `title`, `salary` and `employee` where `title` should be of type `String`, `salary` of type `double` and `employee` of type `Person` (use the `Person` class from the previous exercise and copy this (and possible classes `Name` and `MyDate`) into your Eclipse-project for this exercise).

b) a 3-argument constructor setting all three instance variables. DON'T make a copy of the `Person` object. Why not?

c) a 2-argument constructor with job title and salary as argument. Set the employee to `null`.

d) getters for all instance variables

e) A method `setSalary` returning the salary

f) A method `isAvailable` returning `true` if no employees are set (`employee` is `null`) otherwise returning `false`.

g) A method `hire` (a set method for `employee`) setting the employee given as argument to the employee of the job – but only if the job is available, if not nothing is set.

h) A method `fire` removing the employee of the job (setting it to `null`).

i) A method `toString` returning all information in a `String`

j) A method `equals` returning `true` if the argument to the method is a `Job` object with the same title, salary and employee – otherwise returns `false`.

Note: the relation between `Job` and `Person` is association (weaker than composition) and thus, methods taking a `Person` object and methods returning a `Person` object needs to use the object reference and NOT a copy.

Create a test class (`JobTest`) with a main method and test the class `Job`

k) Create at least two `Job`-objects (note you also need `Person` and maybe `Name` and `MyDate` objects)

l) Call all the methods you made in class `Job`, i.e. both constructors and all methods