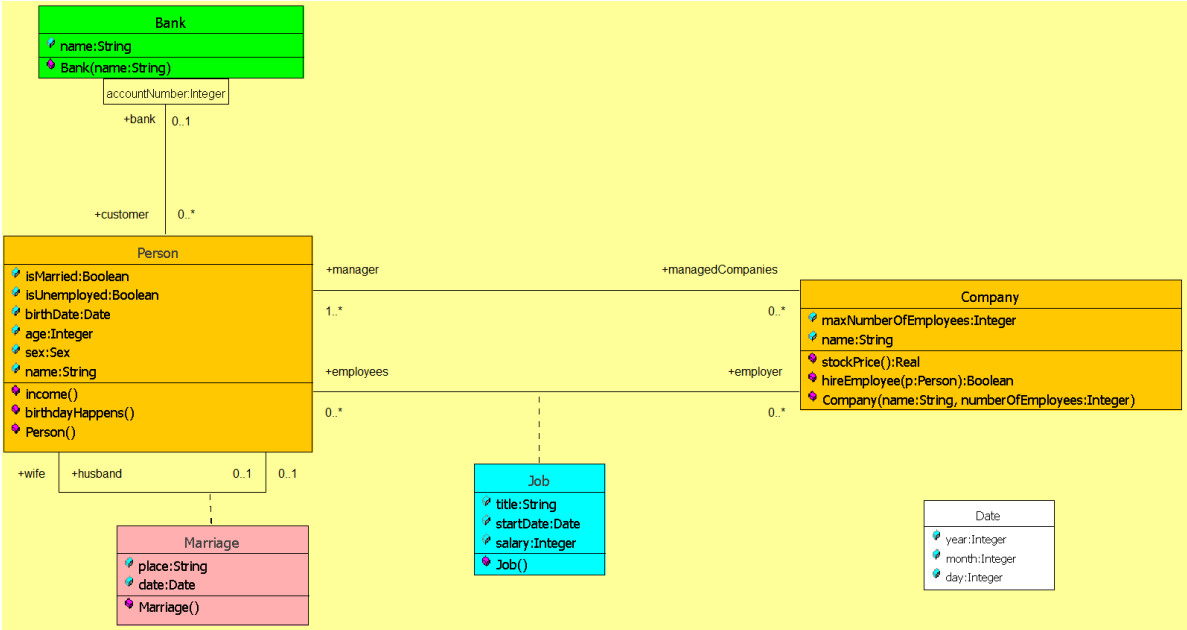
ISS 12th of July 2023

I. A company, referred as employer has a set of persons, which in the relationship with the company are named employees. The company is characterized by its name and the maximal number of employees. Each person is featured by: name, birthDate, sex, and two derived attributes: isMarried and isUnemployed. In the company, each employee has a job featured by a title, a startigDate and a salary. A person may be employed by many companies (employers). On the other hand, each company has a manager. The manager may also manages other companies. Between persons there can be marriage relationships. In each marriage, a person play the role of wife and the other of husband. For simplicity, let's consider that sex may have 2 values: female ans male, and in a marriage, the wife is a female and the husband is a male. A person may be customer of a bank. In this quality, the customer may have accounts. Each account has a unique corresponding accountNumber which is an Integer.

a. By usig UML, please specify a class diagram complying with the above mentioned requirements. 3p



b. Please use OCL to specify the derived attributes: isMarried and isUnemployed.1.5p

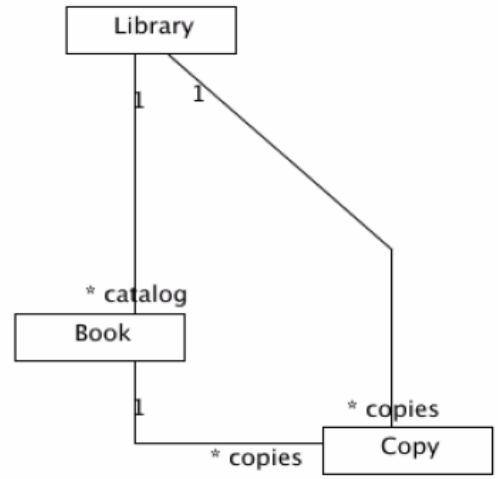
```
context Person

    def isMarried_and_isUnemployed:
          let  isMarried:Boolean = if sex=Sex::female then
                                                  if self.husband.isDefined and self.husband.sex=Sex::male
                                                      then true
                                                      else false
                                                  endif
                                                  else if sex=Sex::male then
                                                                  if self.wife.isDefined and self.wife.sex=Sex::female
                                                                      then true
                                                                      else false
                                                                  endif
                                                  else false endif
                                  endif

          let isUnemployed:Boolean = if self.employer->isEmpty then true else false endif
```

c. Please explain what do you mean by derived attributes and by derrived association end. 1p

In UML (but not only), a derived element is an element that can be computed from another one. Derived elements are used for clarity or for increasing the runnig speed even though these add no semantic information.
In diagrams, a derived element is shown by placing a slash (/) in front of the name of the derived element, such as an attribute or a rolename (associationEnd name).
In UML derived attributes can be specified as observers, by using OCL. The specifications of isMarried and isUnemployed are examples in this sense. As concerning derived associations, usually these are constrained by invariants requiring that the result of navigating a derived association is the same with the result obtained by navigating the chained associations by navigating the chained associations.



In the class disgram above, the direct association between Library and Copy is a derived association, the corresponding invariant constraining the derivative being:

```
context Library
inv derivateAssociation:
self.copies = self.catalog.copies->asSet
```

d. What do you mean by qualified association? What role plays a qualified association in modeling. 1p

By **qualified association,** we understand an association in which one end is indexed by an attribute. For example, the association between Banc and Person is a qualified association indexed by an integer *account* on the *Banc* end. The attribute used to index the qualified end of a qualified association is name **qualifier**.

II. In the context of direct engineering, what do you mean by the target language profile. Example: in case of Java code generation for a UML model, what do you mean by Java profile. Please give a concret example. 1.5p

In UML, by profile we mean a collection of new stereotypes, new interfaces, or new constraints, thus providing new concepts specialized to an application domain or a solution domain. Direct engineering is the automate transformation of UML models in

a language programming code. In our case, it is about the transformation of UML models in Java. In this context, in order to be compilable, the UML models have to comply with some constraints concerning only the Java language and not the UML. Java does not support multiple inheritance. If the UML model include classes inheriting for more than another class, then, before code generation UML the model must be transformed into an equivalent one in which only one parent will be directly inherited, the other parents will be transformed into interfaces implemented by different classes. Each programming language includes a list of reserved words which cannot be attributed to different elements in UML. In the course, a UML example containing an associationEnd name (role name) abstract was analized. As in Java abstract is a reserved word, the cod generated for a such model will not be compilable. This problem must be identified before code generation (direct engineering) by means of constraints corresponding to the Java language, constraints specified in WFRs. So, by target language profile, we mean a set of constraints meant to support the generation of a compilable code, constraints specified at the metamodel level.

III. Please explain the difference between the requirements model and the analysis model. 1p

Reuirements models and analysis models differ only by the specification language. In case of requirements models, these are specified in a natural/spoken language. Analysis models are specified in a formal or semi-formal language, in our case UML.