Excercise from last class

Inform yourself about unary – binary – ternary relationships

Discussion / new examples next class!

Design criteria

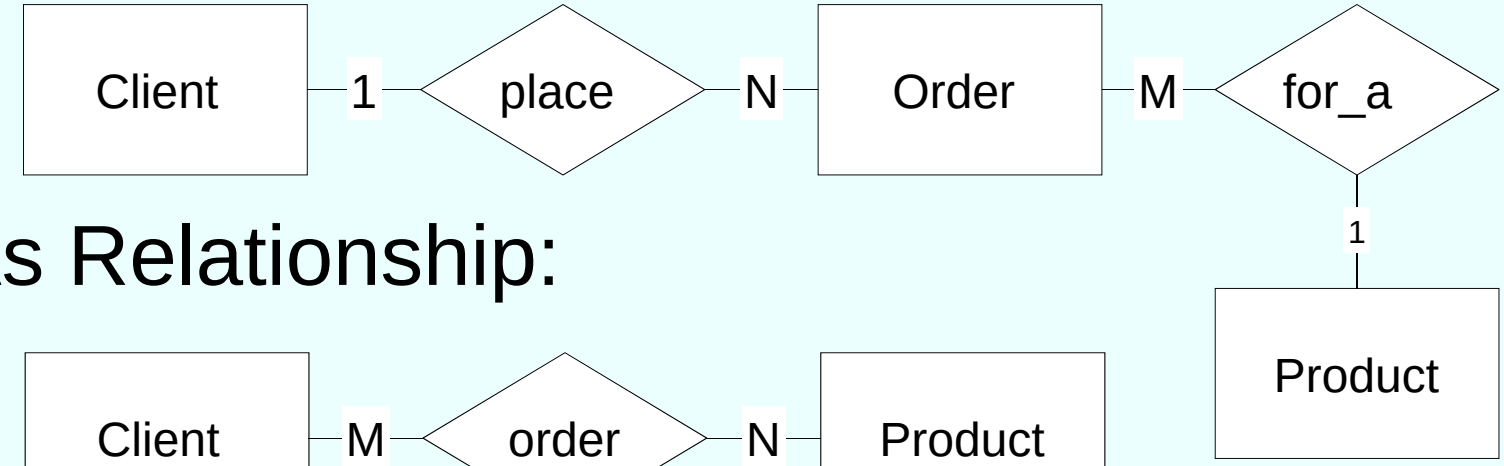
Rules for Classification of Entities and Attributes:

- Entities should contain descriptive information
- Multi valued attributes should be classified as Entities
- Attribute should be assigned to that Entity which desribes it most directly
- Redundant relationships should be avoided

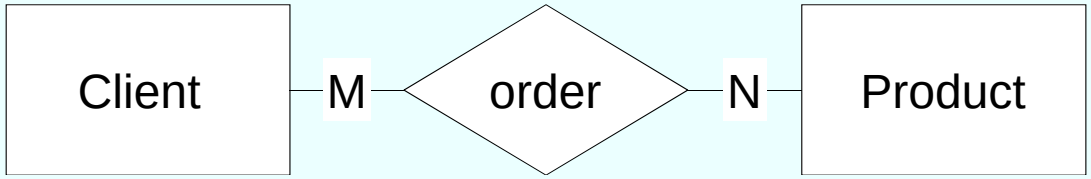
It is *dependent on the application* how to represent an information

Example:Order

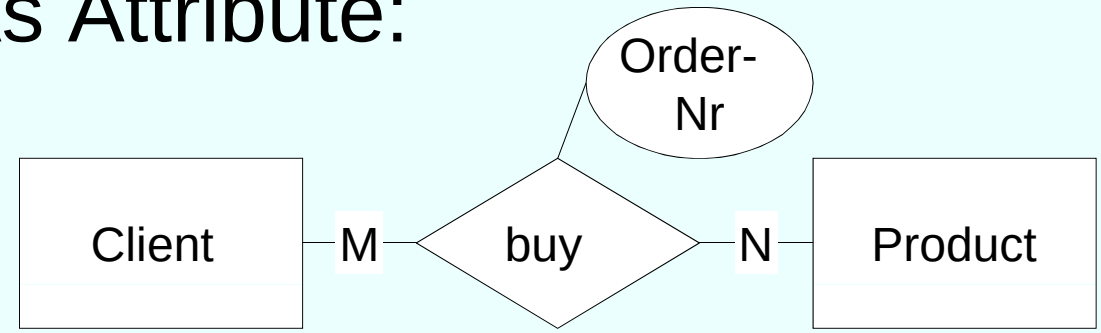
As Entity:



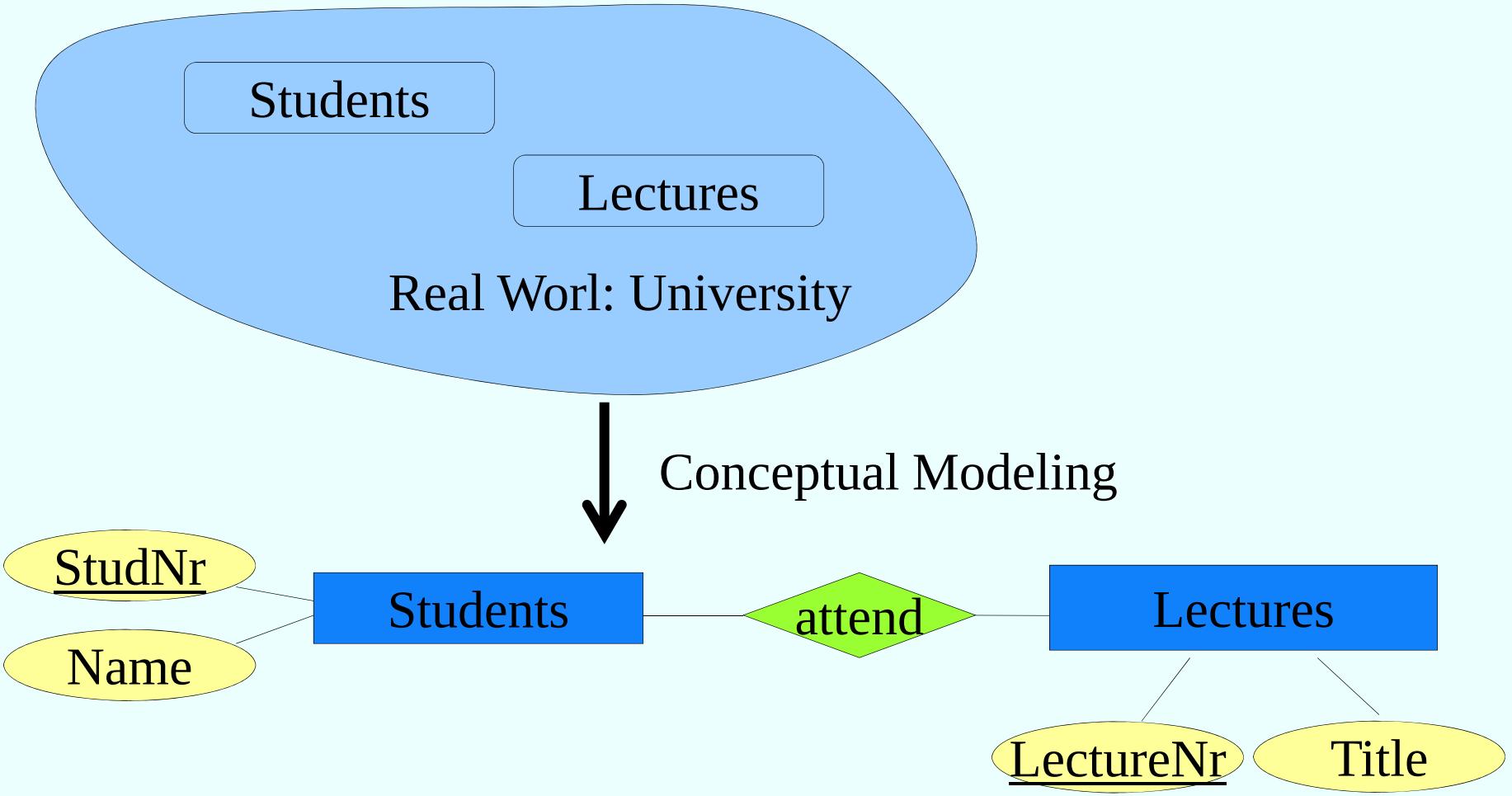
As Relationship:



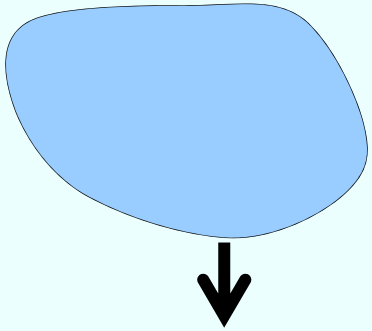
As Attribute:

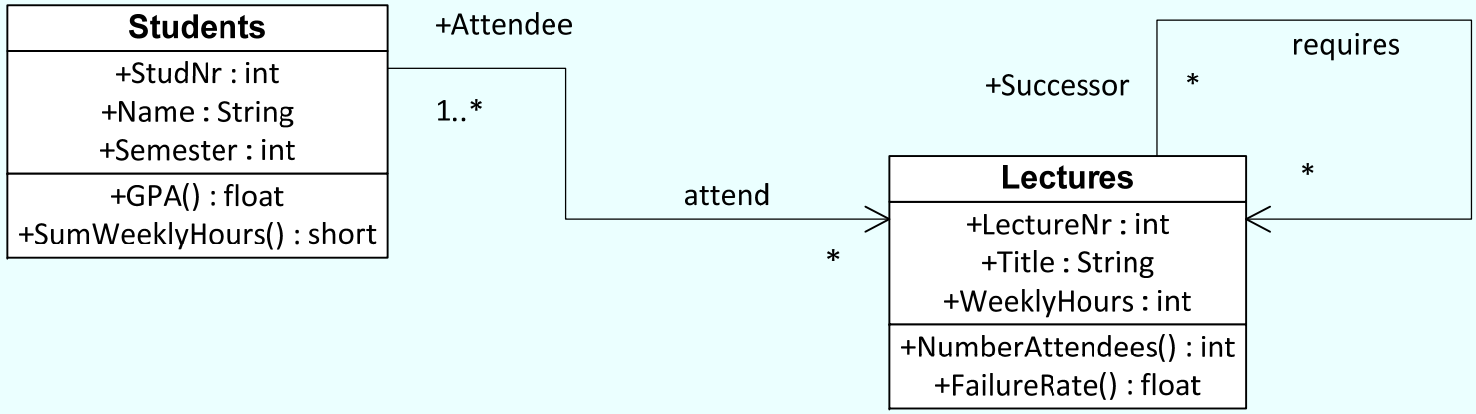


Modeling a small example application: E/R



Modeling a small example application: UML





Data modelling with UML

UML: Unified Modelling Language

De facto standard for object orientierted software design

Centrales construct: class,
models similar objects according to

- Structur (~Attributes)
- Behavior (~Operations/Methods)

Associations between classes correspond to relationships

Generalisation

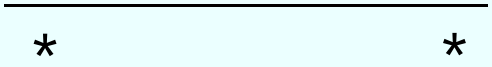
Aggregation

Cheat sheet Class Diagram:

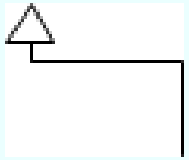
http://www.code-meets-design.de/wp-content/uploads/2013/07/uml-classdiagram-cheat-sheet.pdf

UML Notation

Association:

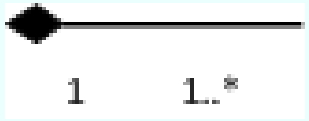


Generalisation:

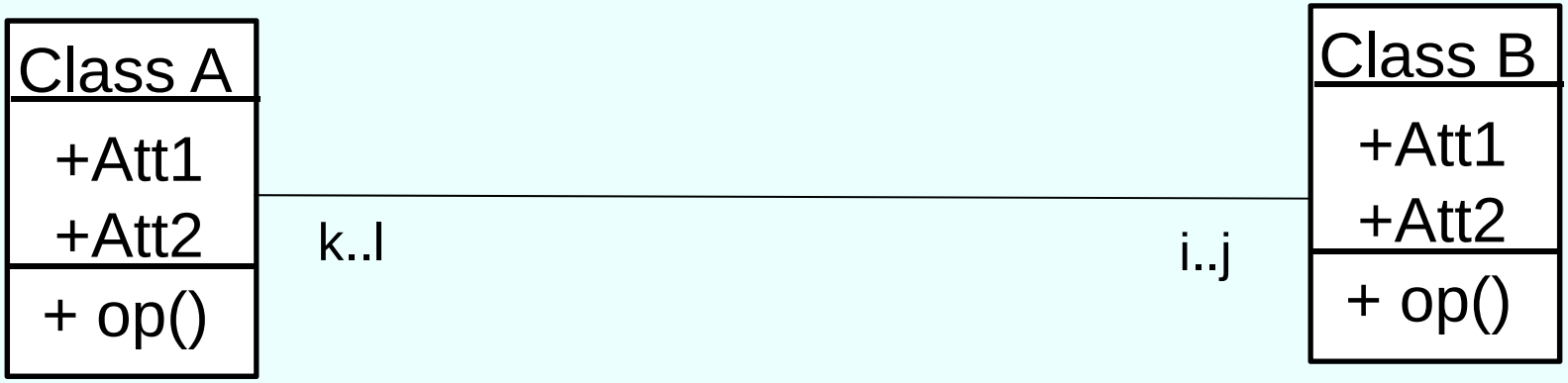


Aggregation:
(Part-of)

Composition:
(Special case of Aggregation)



Multiplicity



Every element of ClassA is associated with at least i elements of ClassB

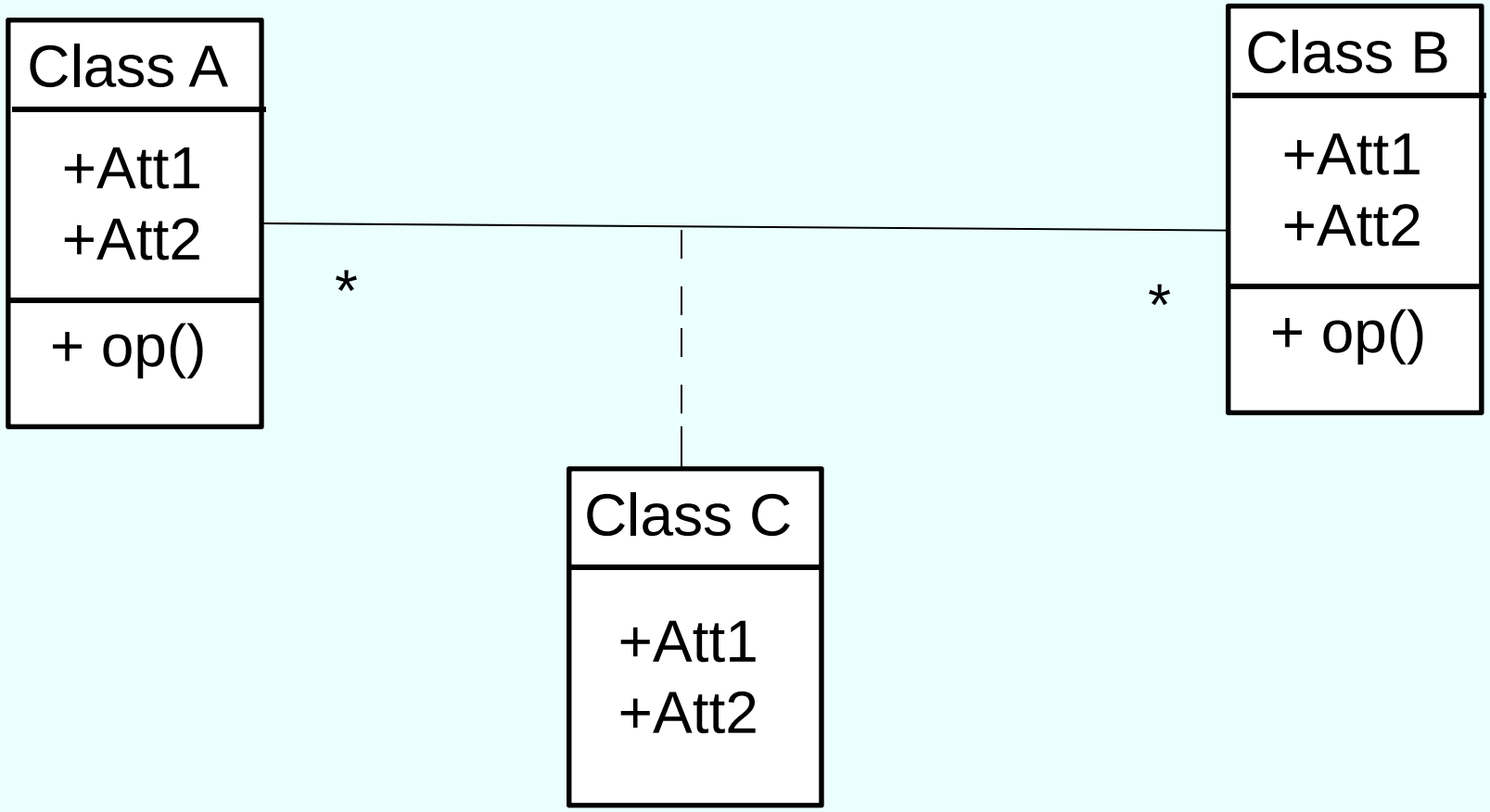
... and with at most j elements of ClassB

Analogously for the intervall k..l

Multiplicity is analogously to the functionalities in the ER-Model

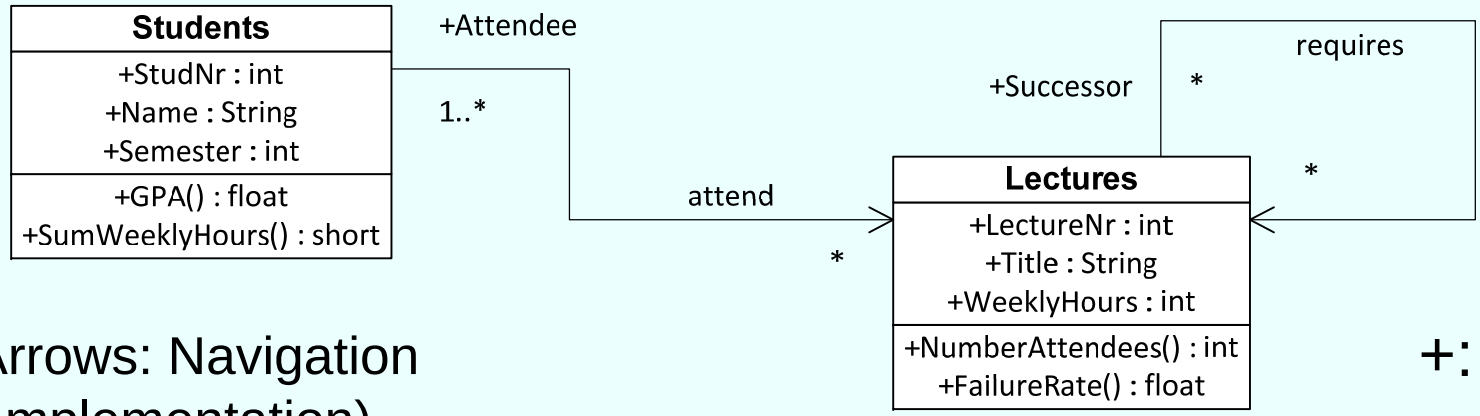
Not to the (min,max)-Notation: **Watch out!**

Association class



… for attributes of the association

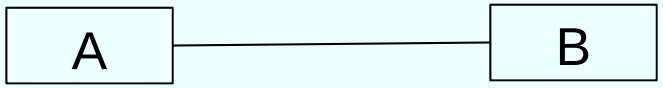
Klassen und Assoziationen



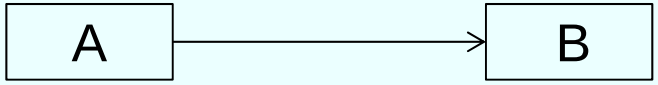
Arrows: Navigation (Implementation)

+: public

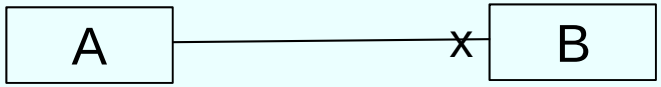
No statement on navigation



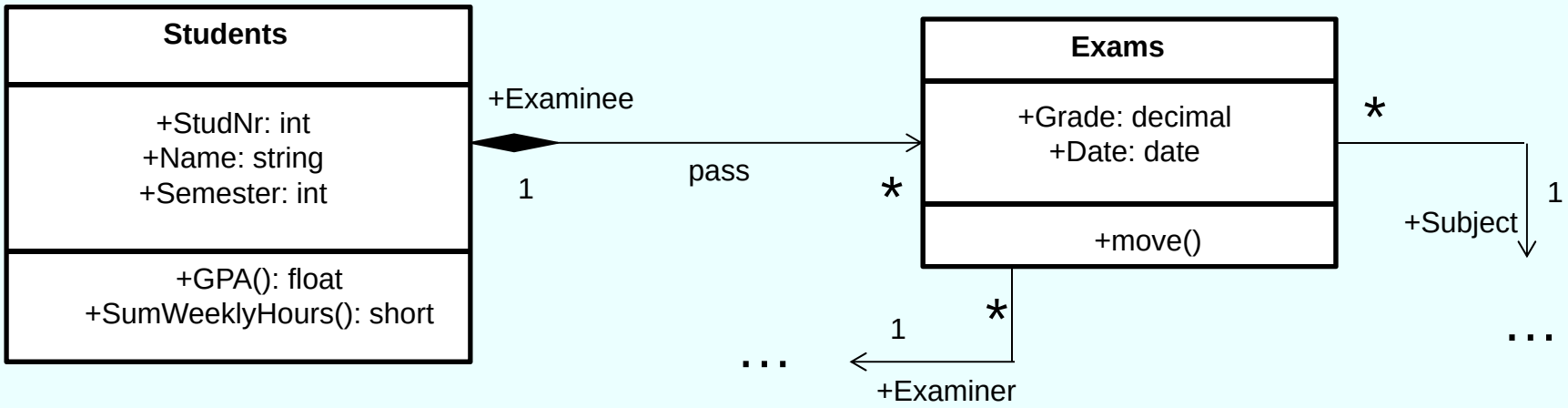
Navigation from A to B allowed

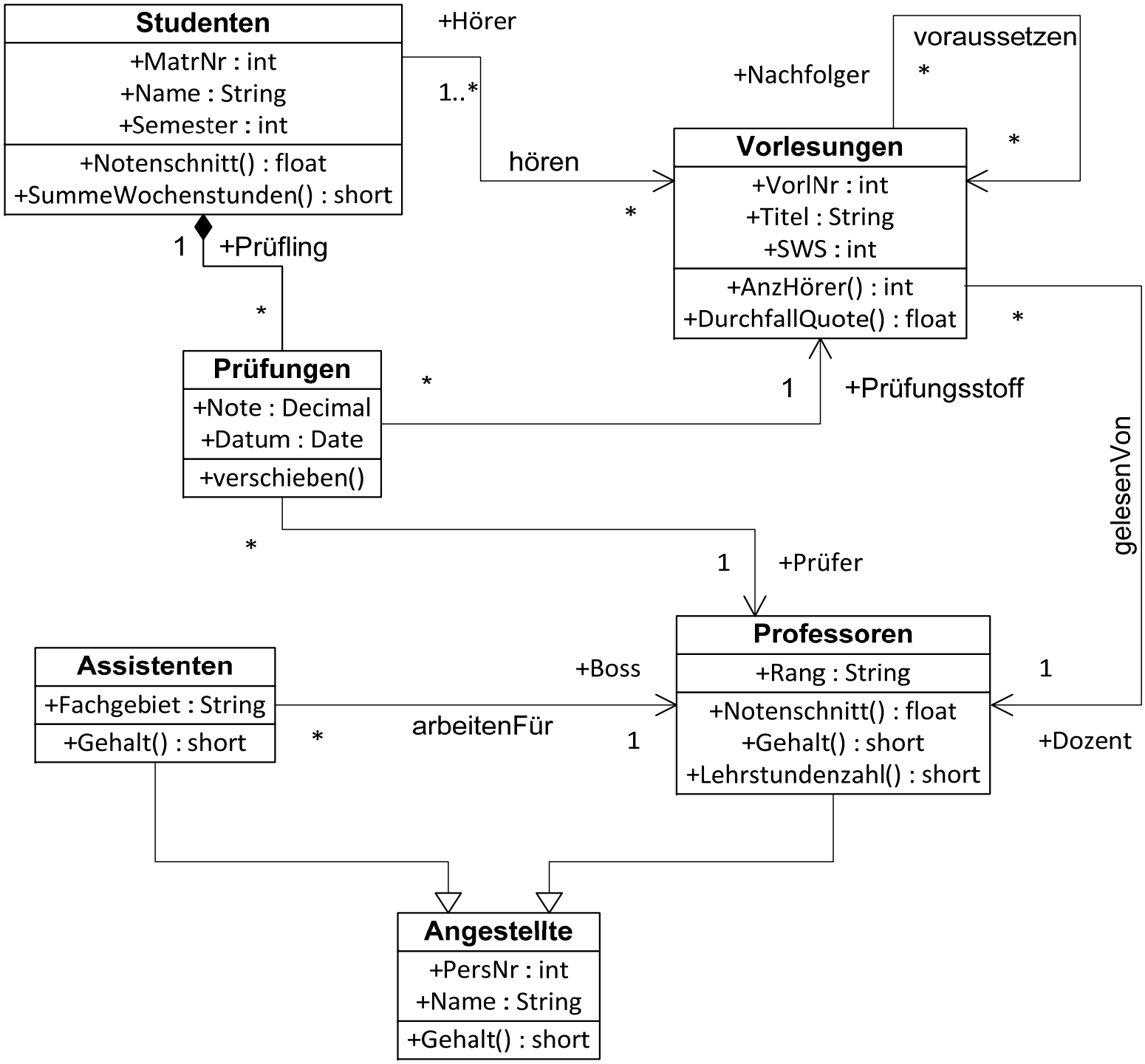


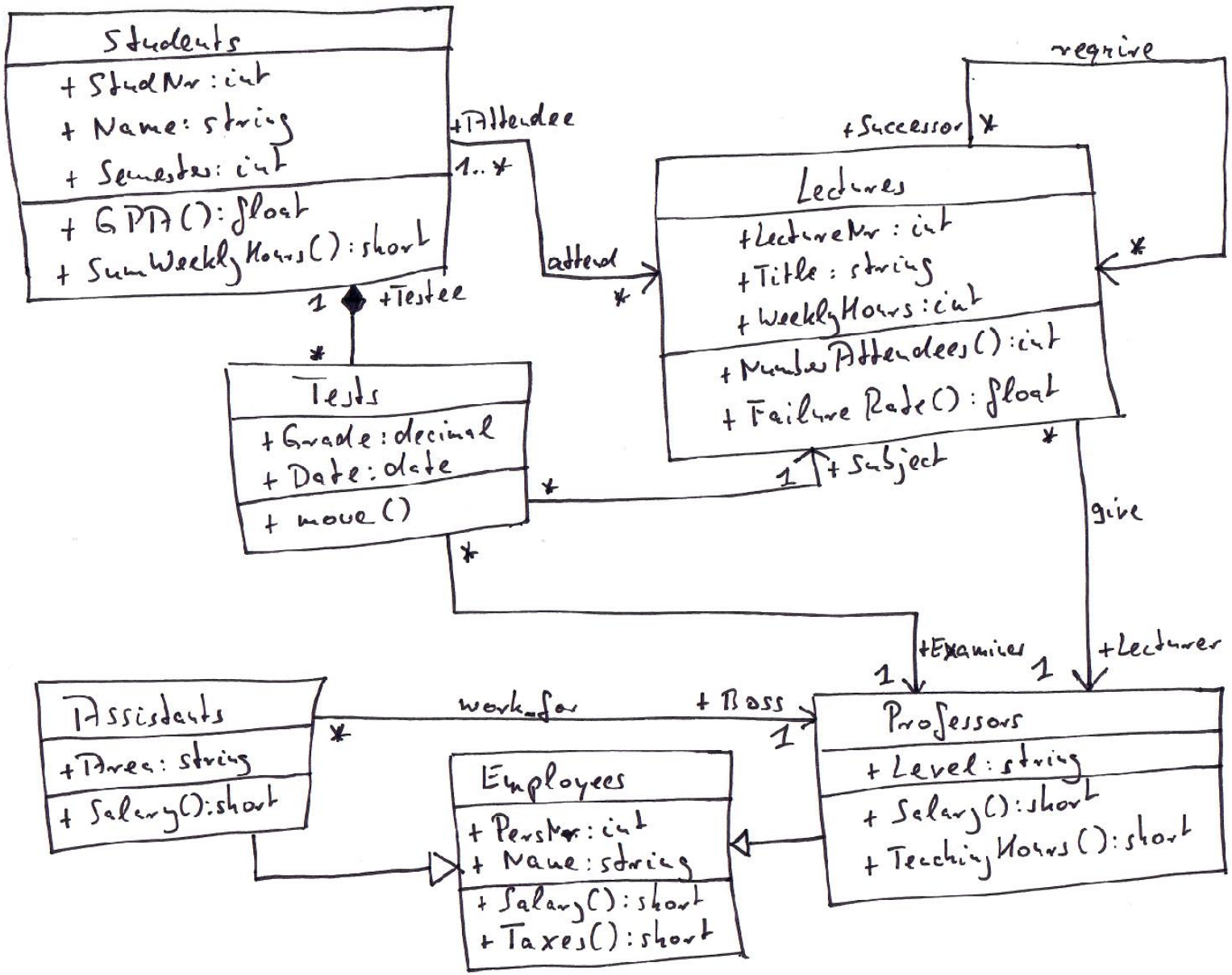
Navigation from A to B forbidden



Komposition




Studenten
+MatrNr : int
+Name : String
+Semester : int
+Notenschnitt() : float
+SummeWochenstunden() : short
+Hörer
1..*
hören
*
Vorlesungen
+VorlNr : int
+Titel : String
+SWS : int
+AnzHörer() : int
+DurchfallQuote() : float
+Nachfolger
voraussetzen
*
*
1
+Prüfling
*
Prüfungen
+Note : Decimal
+Datum : Date
+verschieben()
*
1
+Prüfungsstoff
*
gelesenVon
*
1
+Prüfer
Assistenten
+Fachgebiet : String
+Gehalt() : short
*
arbeitenFür
+Boss
1
Professoren
+Rang : String
+Notenschnitt() : float
+Gehalt() : short
+Lehrstundenzahl() : short
1
+Dozent
Angestellte
+PersNr : int
+Name : String
+Gehalt() : short


Students
+ StudNr : int
+ Name : string
+ Semester : int
+ GPA() : float
+ SumWeeklyHours() : short
+Attendee
1..*
attend
*
regrive
+Successor
*
Lectures
+LectureNr : int
+Title : string
+WeeklyHours : int
+NumberAttendees() : int
+FailureRate() : float
*
1
+Subject
1
+Testee
*
Tests
+Grade : decimal
+Date : date
+ move()
*
*
give
+Examiner
1
+Lecturer
1
Assistants
+Area : string
+ Salary() : short
*
works_for
+Boss
1
Professors
+ Level : string
+ Salary() : short
+ TeachingHours() : short
Employees
+ PersNr : int
+ Name : string
+ Salary() : short
+ Taxes() : short

More useful Diagrams

- Use Case Diagram
- Interaction Diagram
- Sequence Diagram

Search for examples in the internet

Quiz UML

From the Stanford MOOC:

https://lagunita.stanford.edu/courses/DB/UML/SelfPaced/courseware/ch-unified_modeling_language/seq-quiz-uml/

Quiz Q2 + Q5 – Q7