UML Data Modelling

Carla Teixeira Lopes

Bases de Dados
Mestrado Integrado em Engenharia Informática e Computação, FEUP

Based on Jennifer Widom slides

Key concepts

~~Classes~~

~~Associations~~

Association Classes

Generalizations

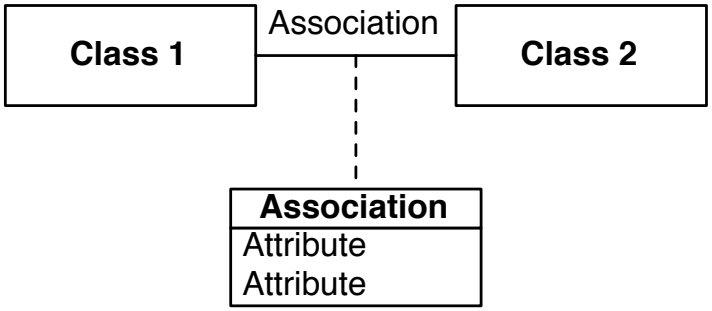
Composition & Aggregation

Constraints

Derived Elements

Association Classes

Relationships between objects of two classes, *with attributes on relationships*

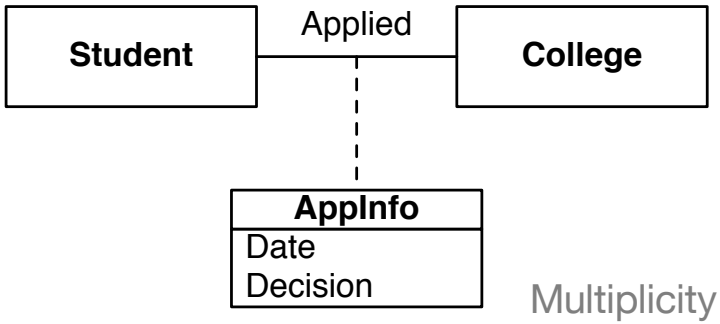


The name may be placed in the association, in the class or in both

It only captures **one** relationship between the two specific objects across the two classes

Association Classes

Suppose that, in the previous example, we also want to have the date the students applied to college and the decision.



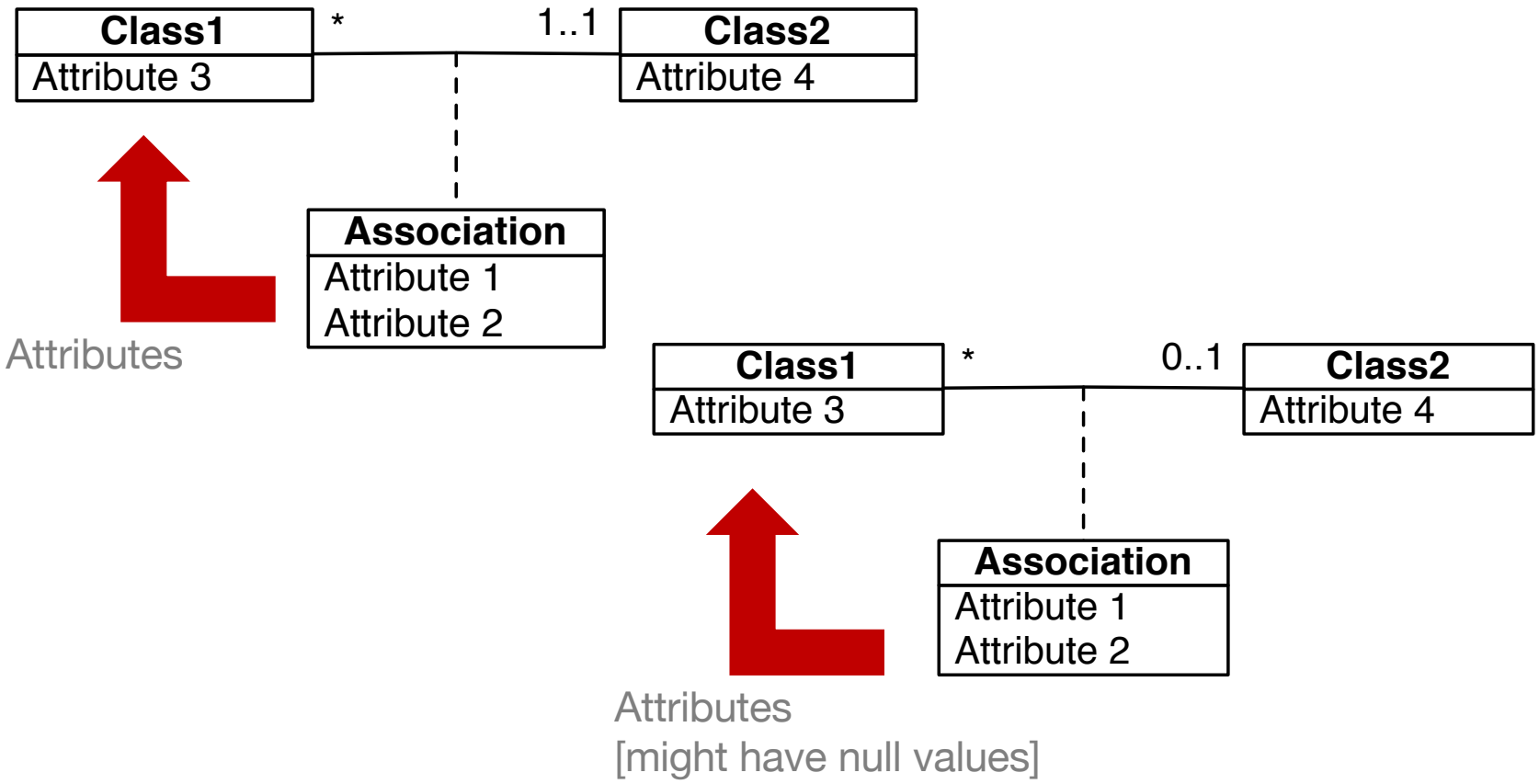
Multiplicity omitted

Doesn't allow students to apply multiple times to the same college.

How could we model this situation?

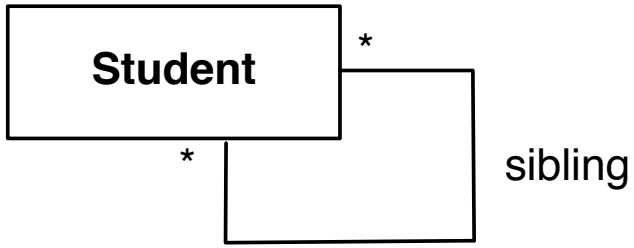
Eliminating Association Classes

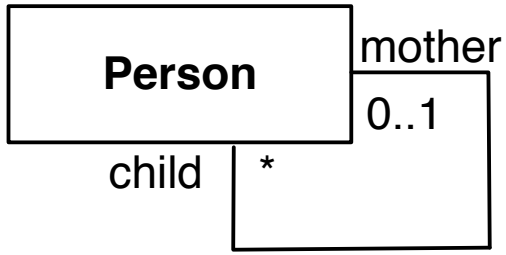
Unnecessary if 0..1 or 1..1 multiplicity



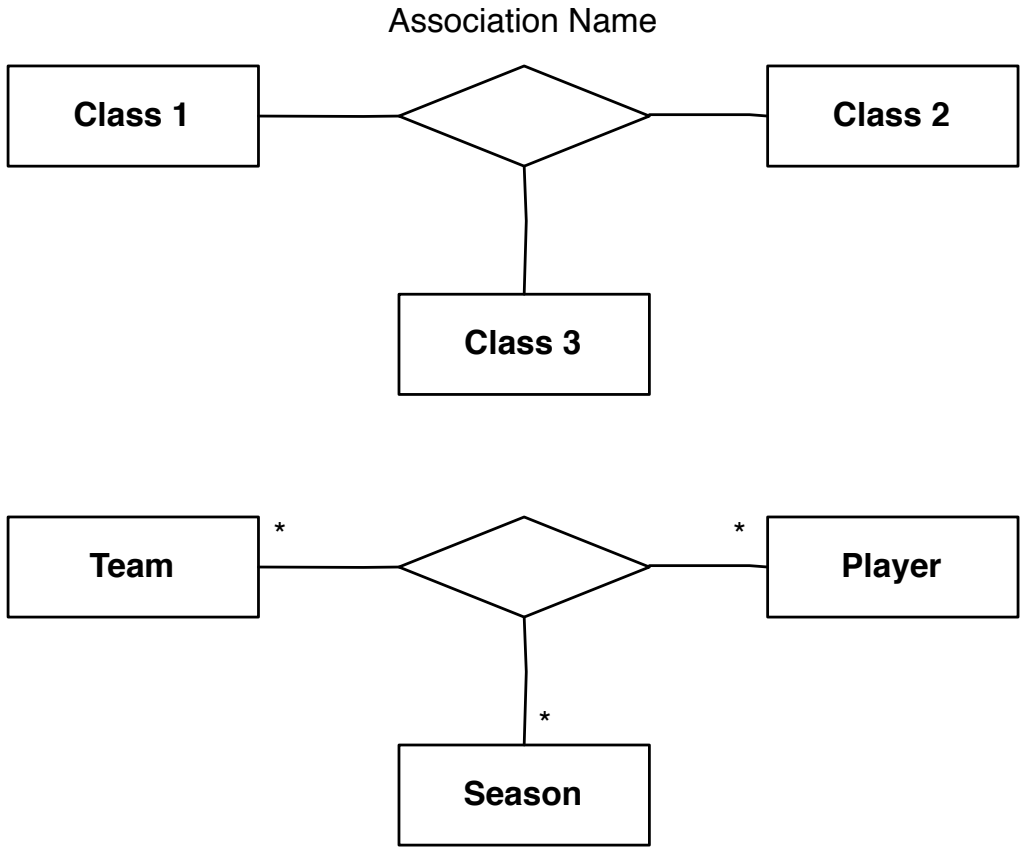
Self-Associations

Associations between a class and itself



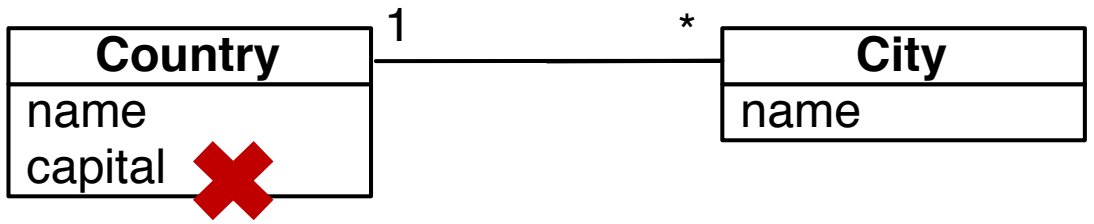


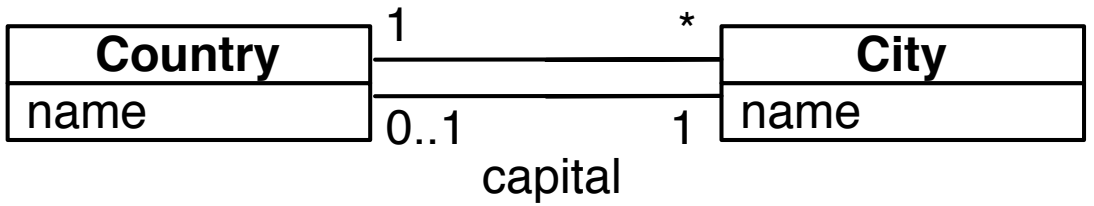
Associations n-arys



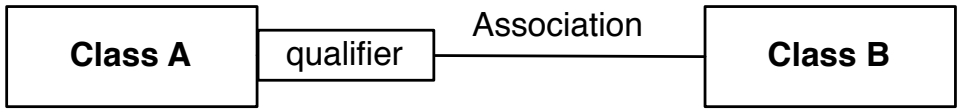
Association versus attribute

An attribute should never be a reference to a class





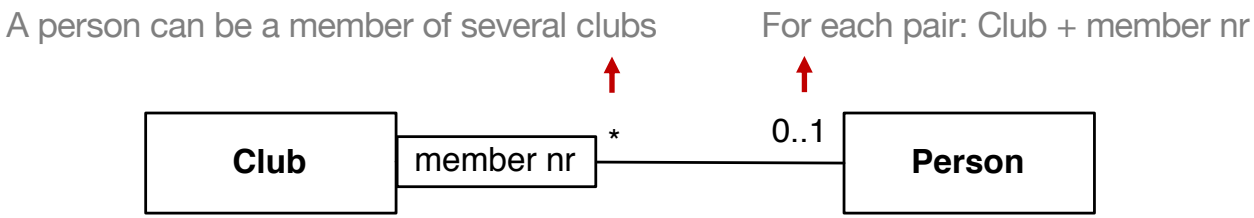
Qualified associations



Qualifier

One or more attributes of an association used to navigate from A to B

"Access key" to B from an object of A

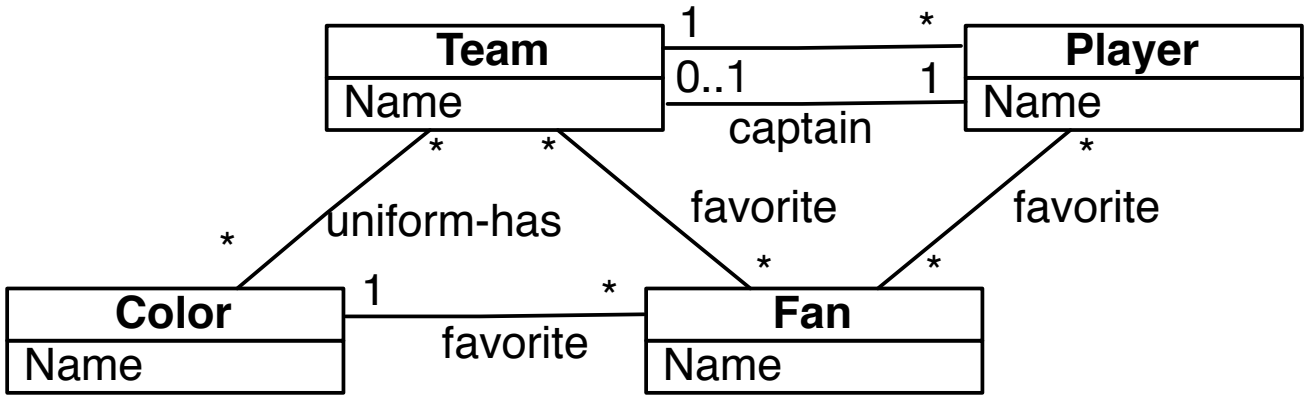


Exercise

Draw a UML diagram for a database recording information about teams, players, and their fans, including:

- For each team, its name, its players, its team captain (one of its players), and the colors of its uniform
- For each player, his/her name
- For each fan, his/her name, favorite teams, favorite players, and favorite color.

Exercise


Team
Name
Player
Name
Color
Name
Fan
Name
1
*
0..1
1
captain
*
*
uniform-has
favorite
favorite
*
*
*
*
1
*
favorite

Key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

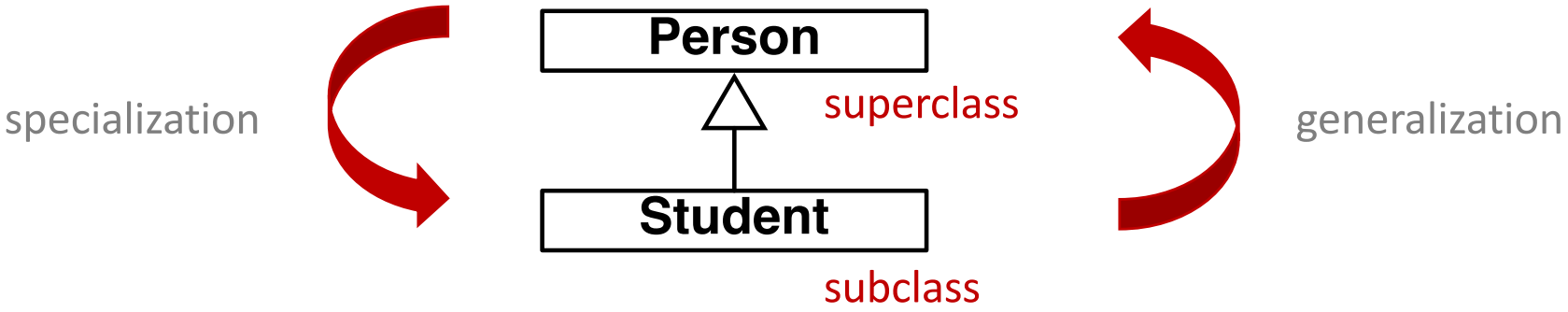
Generalizations

Composition & Aggregation

Constraints

Derived Elements

Generalizations

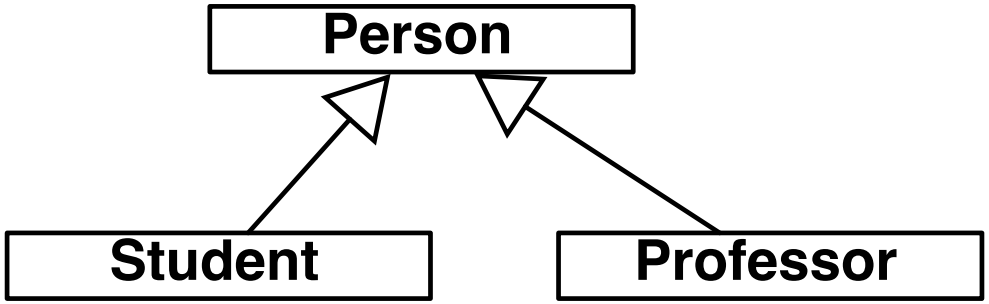


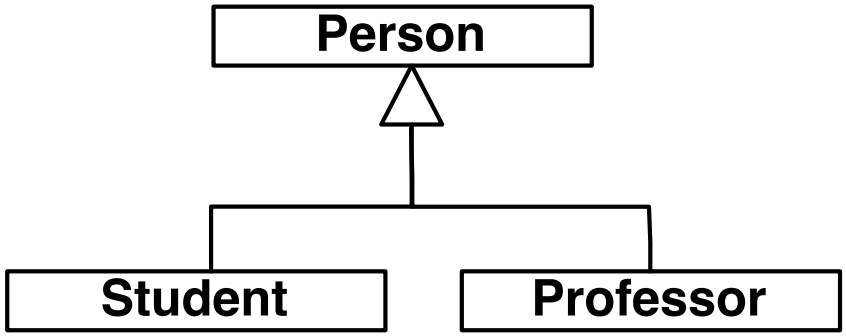
"is a" semantic relationship

A student "is a" person

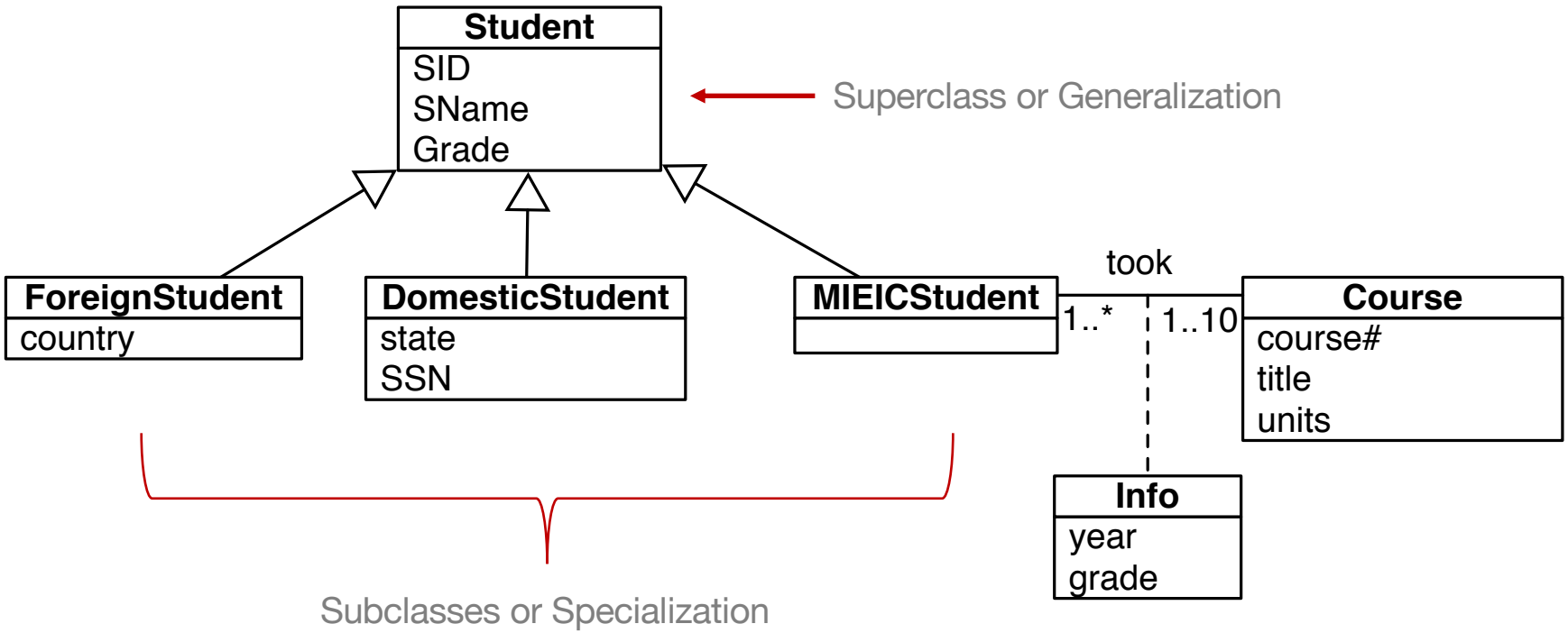
The subclass inherits the properties (attributes and relationships) of the superclass and may add other

Generalizations – Alternative Notations





Generalizations Example



Generalizations Properties

Complete

If every object in the super class is in at least one subclass

Incomplete or partial

If it's not complete

Disjoint or Exclusive

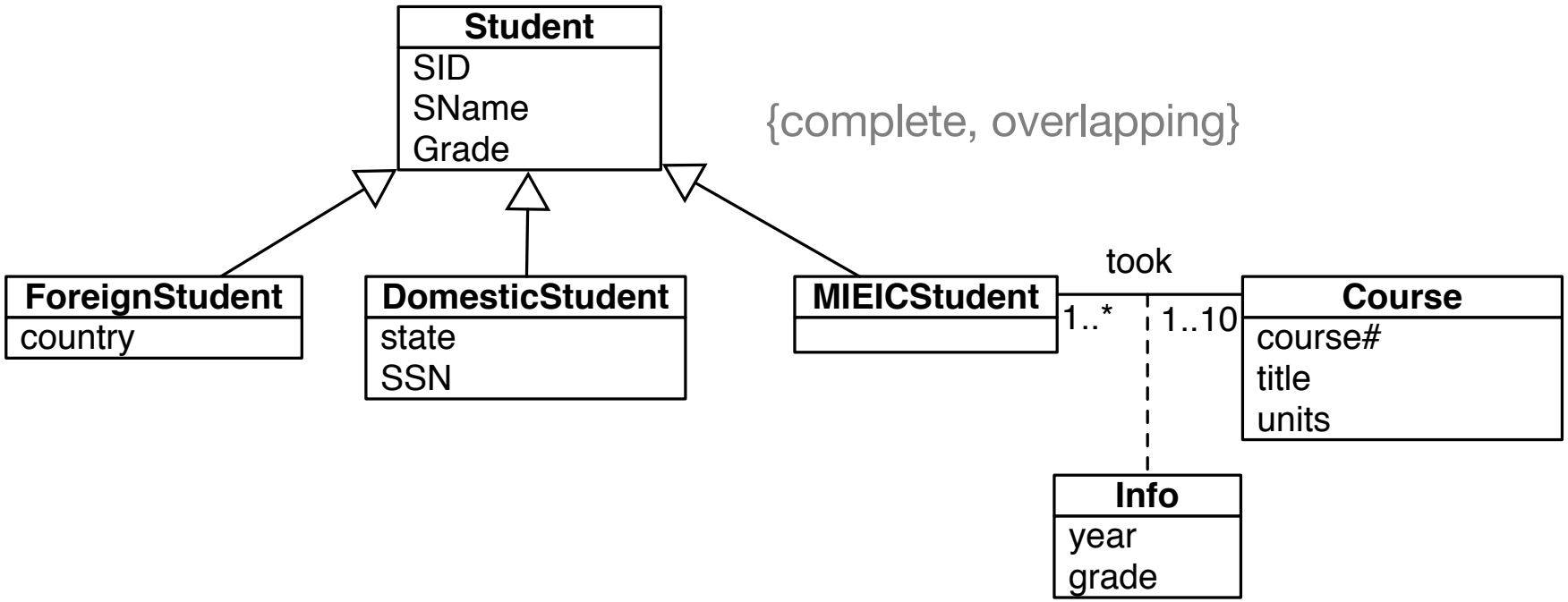
If every object is on at most one subclass

Overlapping

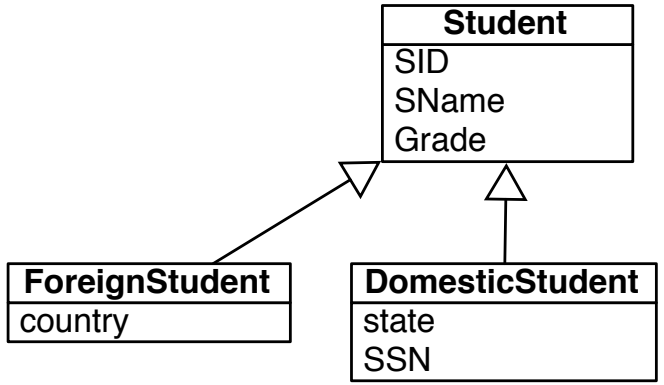
If it's not disjoint

We can have any combination of the first two with the second two.

Generalizations Properties

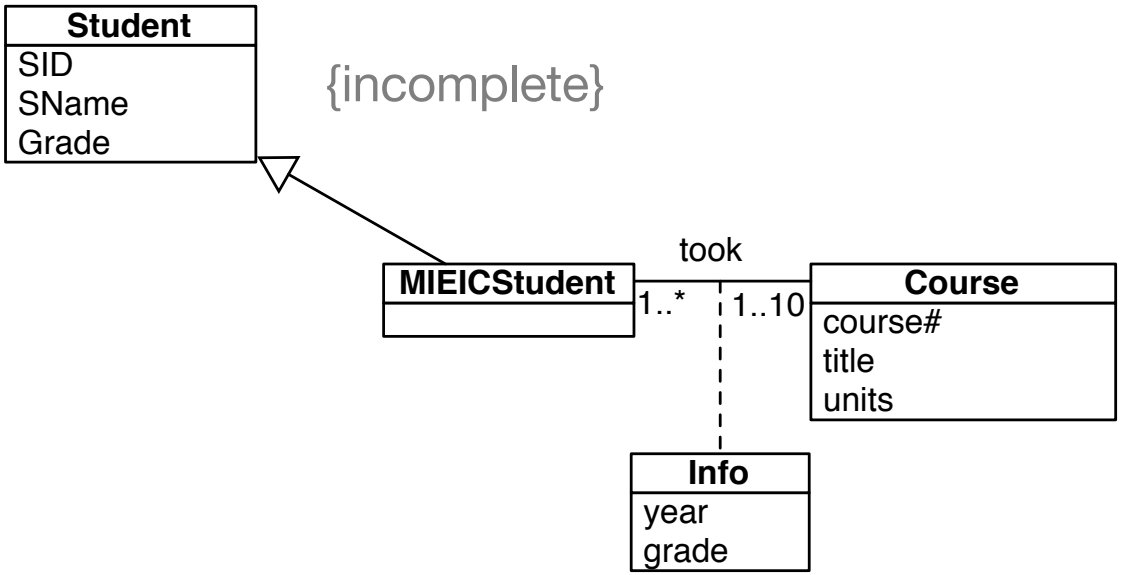


Generalizations Properties



{complete, disjoint}

Generalizations Properties



Key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

~~Generalizations~~

Composition & Aggregation

Constraints

Derived Elements

Aggregation

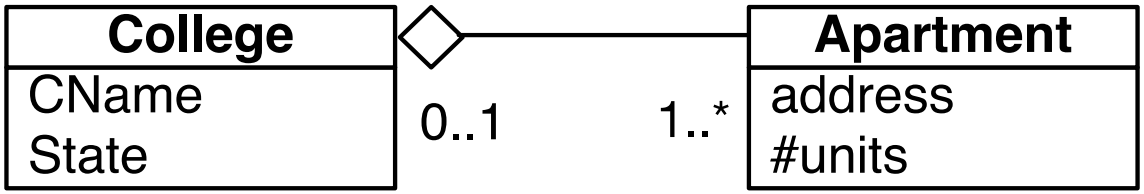
Special type of association

Models a “whole/part” relationship

Represents a “has-a” relationship

Does not link the lifetimes of the whole and its parts

0..1 is implicit



Aggregation

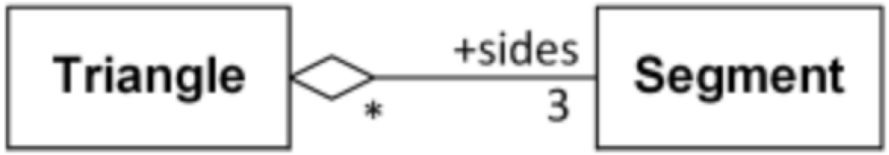
Binary association

Asymmetric

- Only one end of association can be an aggregation

Shared part can be included in several composites

If some or all of the composites are deleted, shared part may still exist



Triangle has ‘sides’ collection of three line Segments.
Each line Segment could be part of none, one, or several triangles.

Composition

Strongest form of aggregation

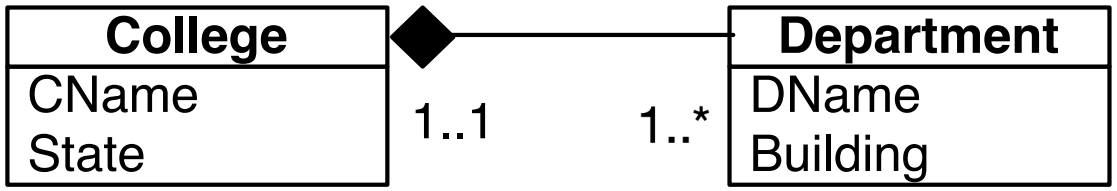
Strong ownership and coincident lifetime as part of the whole

An object may be a part of only one composite at a time

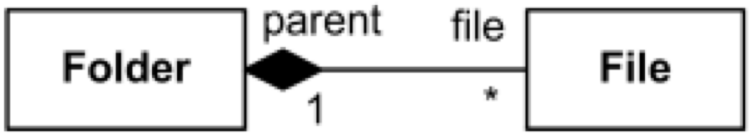
The whole is responsible for managing the creation and destruction of its parts

1..1 is implicit

Composition



College has 1 or more Departments and each Department belongs to exactly one College.
If College is closed, so are all of its Departments.



Folder can contain many files, while each File has exactly one Folder parent.
If Folder is deleted, all contained Files are deleted as well.

Key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

~~Generalizations~~

~~Composition & Aggregation~~

Constraints

Derived Elements

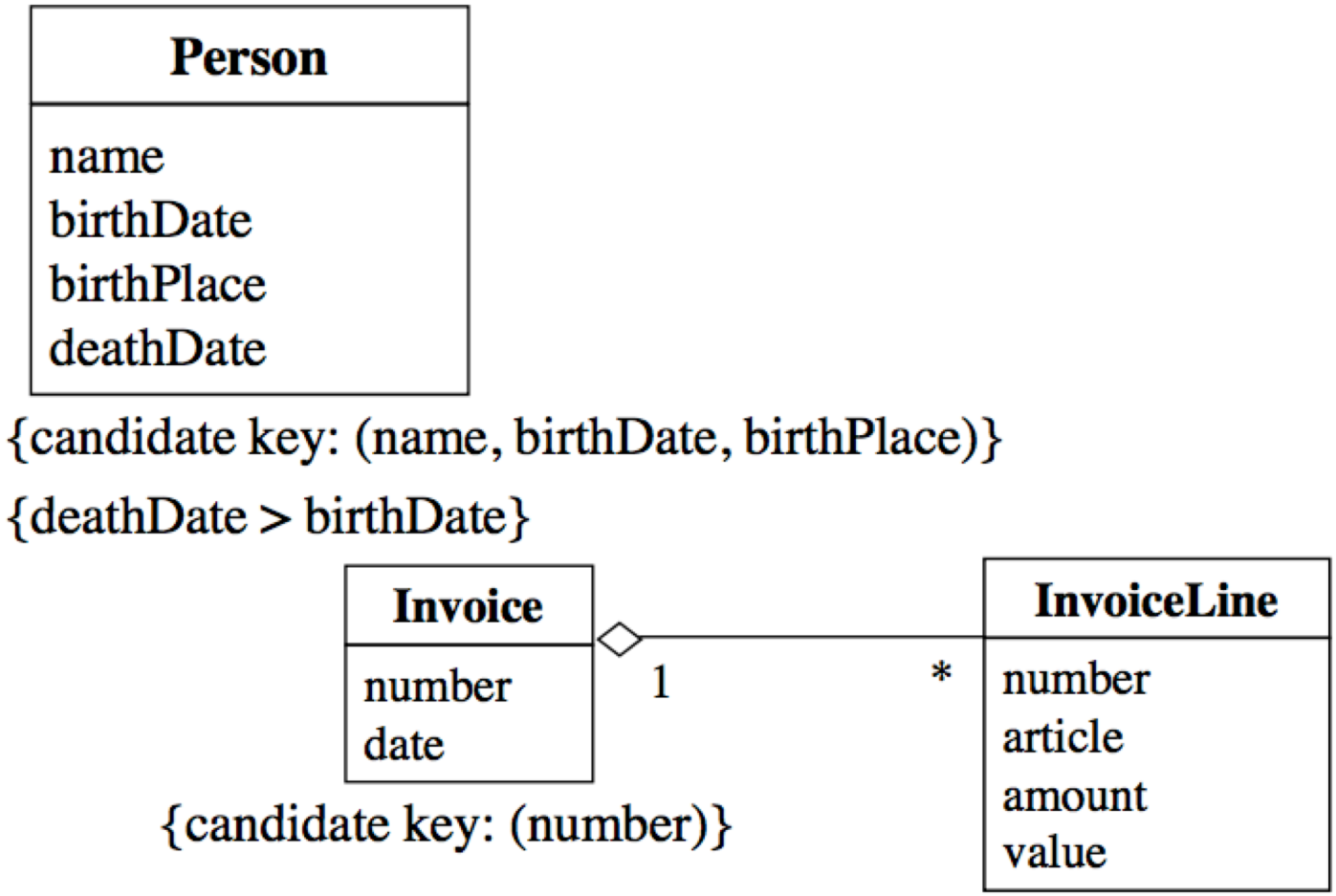
Constraints

Specifies a condition that has to be present in the system

It is indicated by

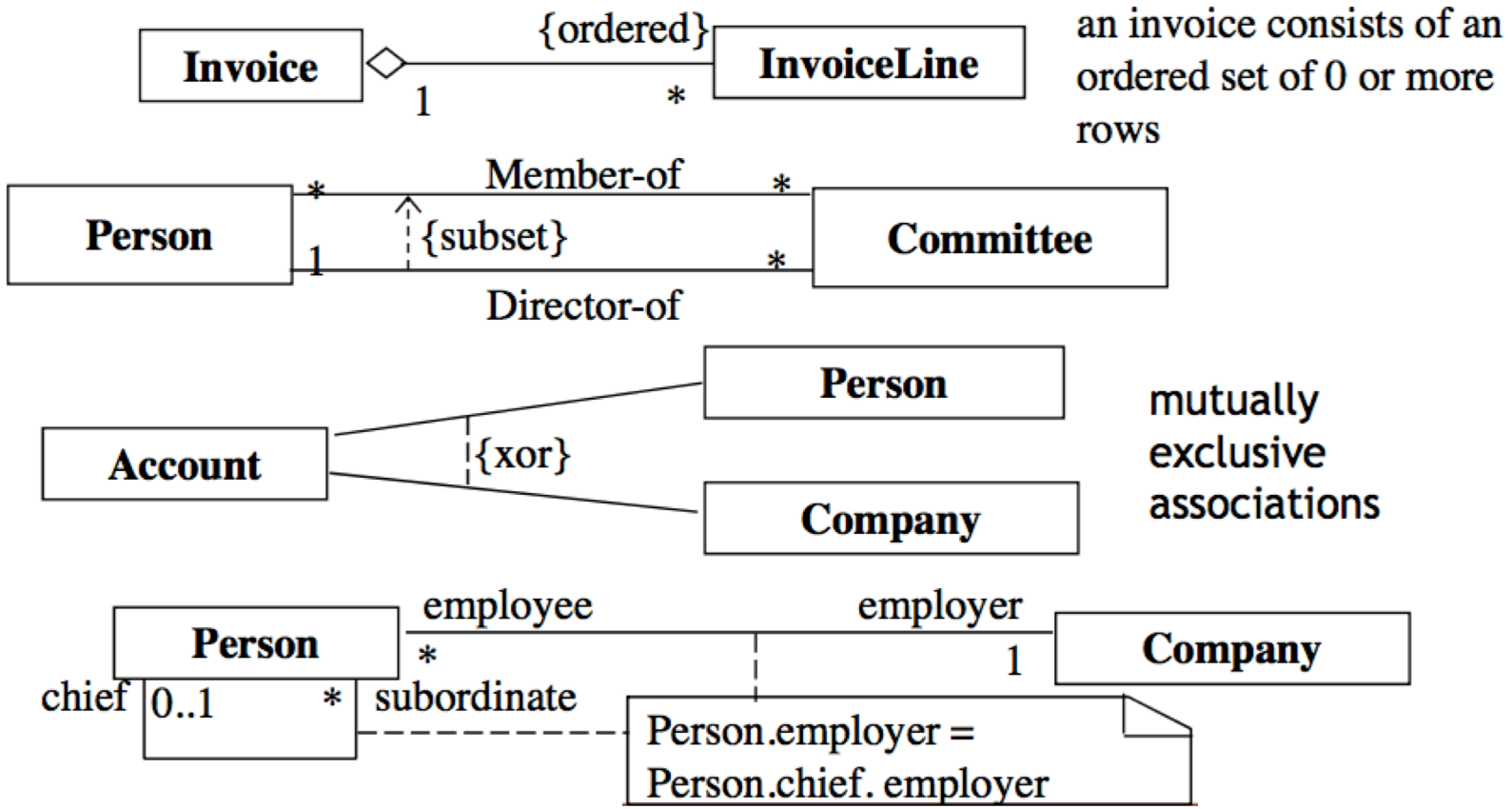
- an expression or text between brackets
- note placed near (or connected by dotted lines to) the elements to which it relates

Constraints in classes



Credits: João Moreira

Constraints in associations



Credits: João Moreira

Key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

~~Generalizations~~

~~Composition & Aggregation~~

~~Constraints~~

Derived Elements

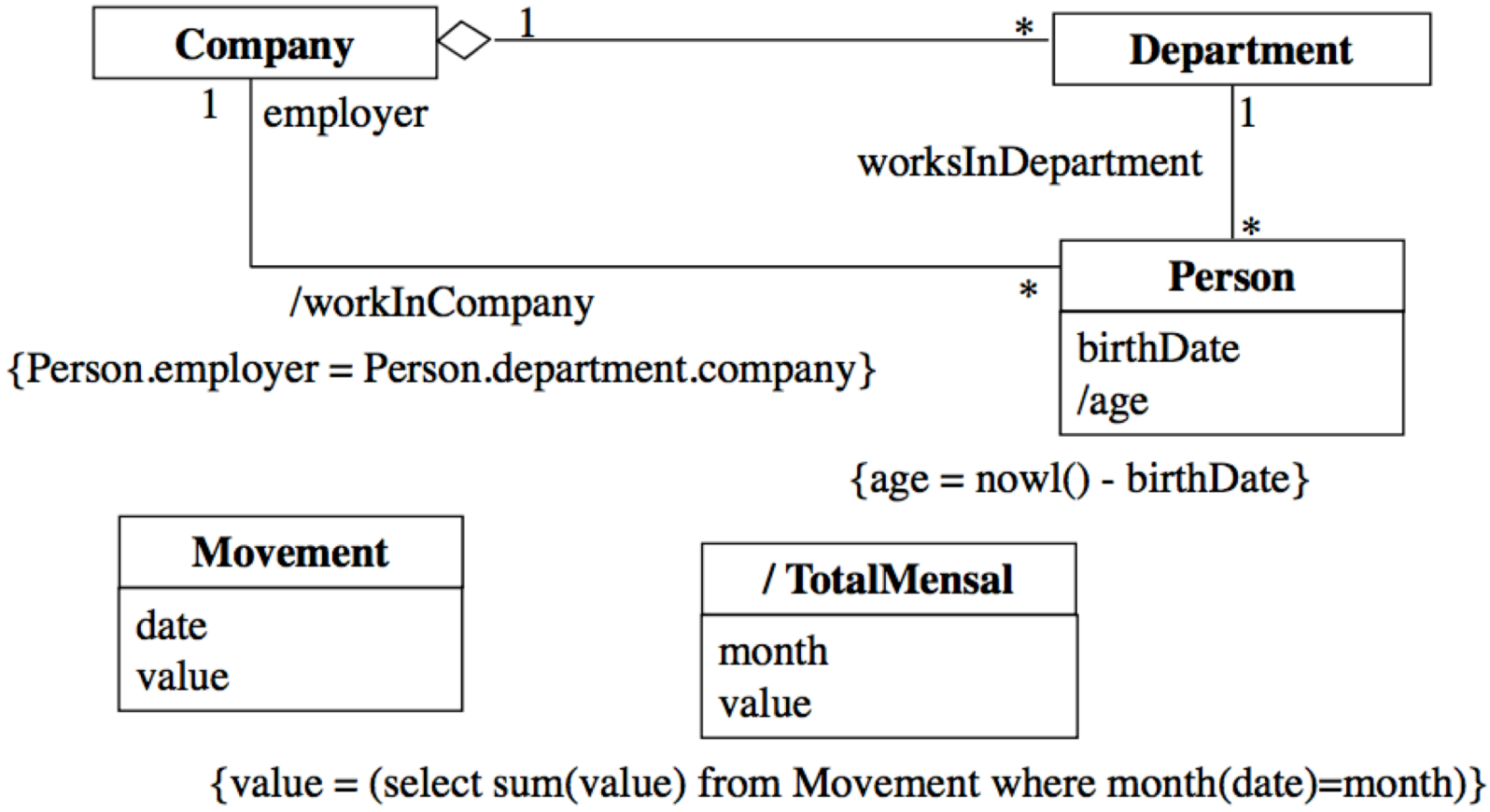
Derived Elements

Element (class, attribute or association) computed using other elements in the model

Notation: ‘/’ before the name of the derived element

Usually have an associated constraint that relates them with other elements

Derived Elements



Credits: João Moreira

Higher-Level Database Design

- Unified Modeling Language (UML)
  - Data modeling subset
- Graphical
- Key concepts
  - Classes
  - Associations
  - Association Classes
  - Generalizations
  - Composition & Aggregation
  - Constraints
  - Derived Elements
- Can be translated to relations automatically

Kahoot time!

Any doubts?

Readings

Jeffrey Ullman, Jennifer Widom, A first course in Database Systems 3rd Edition

Section 4.7 - Unified Modeling Language