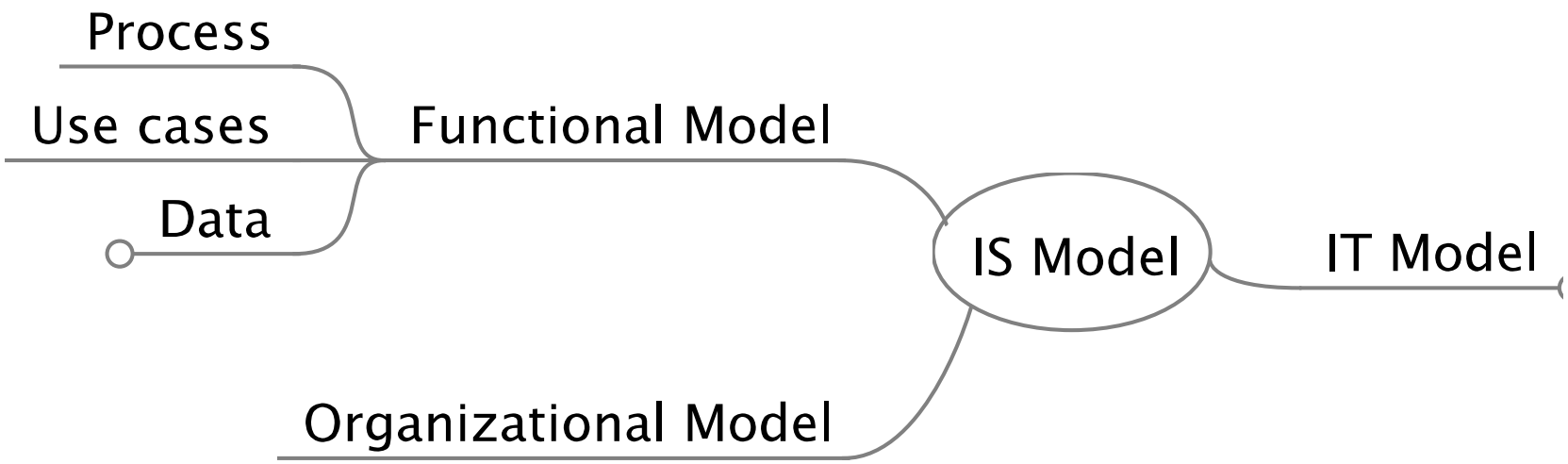
Business Process Modeling

Information modeling, UML Class diagrams

Licensing Note

Functional model



Functional model – submodels

- Process flow
  - Process modeling
    - UML Activity Diagrams
    - BPMN
- Information
  - Conceptual modeling
    - UML Class diagrams
    - Entity-Relationships
- Interaction
  - Interaction modeling
    - Use cases

UML

- Unified Modeling Language
- Standardized by OMG
- Several diagrams
  - Class diagrams
  - Activity diagrams
  - Use Case diagrams
  - (Sequence diagrams)
  - (Statecharts)

Conceptual modeling

Process modeling

Functional modeling

Conceptual Modeling

CLASS DIAGRAM

Goal

- Capture
  - Main (abstract) concepts
  - Characteristics of the concepts
    - Attributes associated to the concepts
  - Relationships between concepts

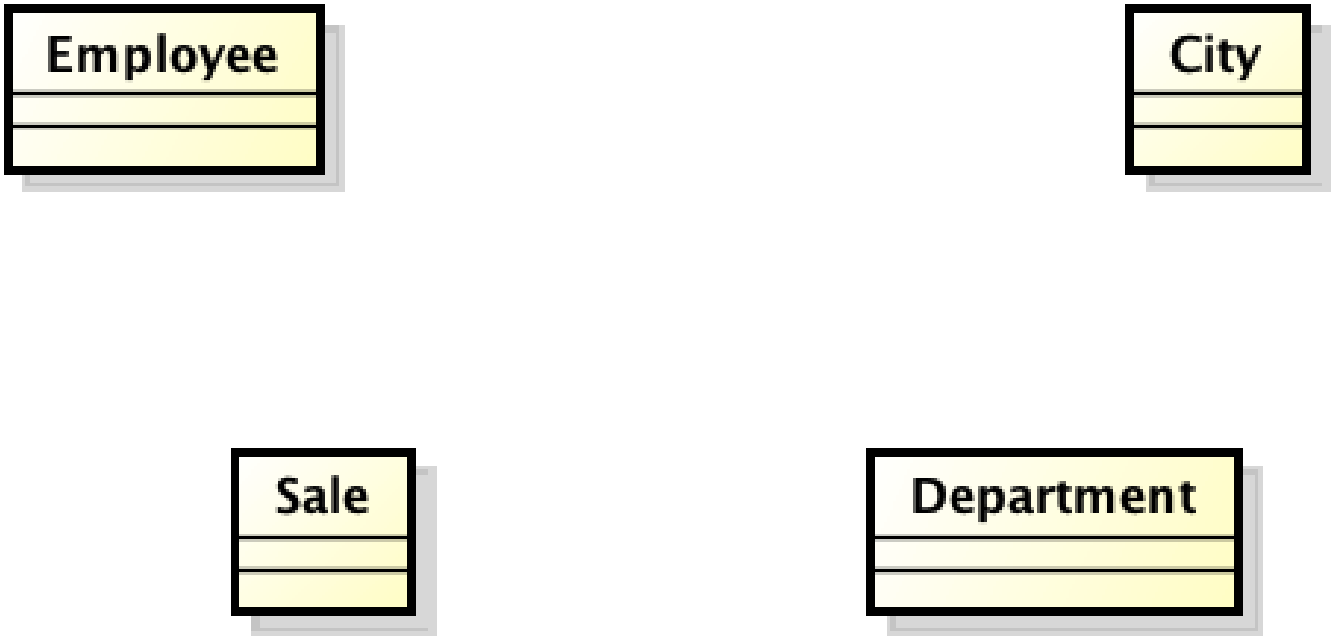
Object

- Model of item (physical or within the software system)
  - ex.: a student, an exam, a window
- Characterized by
  - identity
  - attributes (or data or properties)
  - operations it can perform (behavior)
  - messages it can receive

Class

- They describe set of objects
  - Common properties (attributes, behaviours)
  - Autonomous existence
  - E.g. facts, things, people
- An instance of a class is an object of the type that the class represents.
  - In an application for a commercial organization CITY, DEPARTMENT, EMPLOYEE, PURCHASE and SALE are typical classes.

Class – Examples



Usage of class diagram

- Model of concepts (glossary)
- Model of system (hw + sw) == system design
- Model of software classes (software design)

- Class in conceptual model (UML class diagram)
  - Ex Employee class
- Corresponding entities in software application
  - Data layer: Employee table in RDB
  - Business logic layer: Employee class in Java / C++, C#
  - Presentation layer: form to enter employee data, form to show employee data, and more

- Before doing a class diagram,

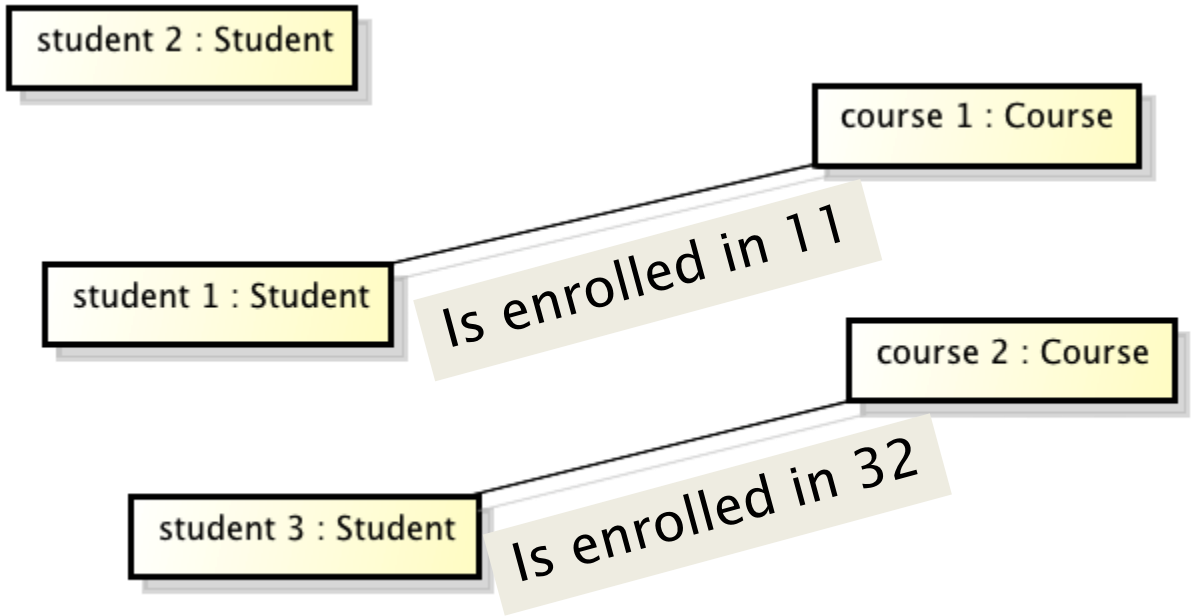
  DECIDE WHAT YOU WANT TO MODEL

Classes in conceptual diagram

- Where to look for
  - Physical entities: Person, Car,
  - Roles: Employee, Director, Doctor,
  - Social / legal / organizational entities: University, Company
  - Events: Sale, Order, Request, Claim, Call
  - Time intervals: Car rental, Booking, Course, Meeting
  - Geographical entities: City, Road, Nation
  - Reports, summaries: weather report, bank account statement

Link

- Model of property between objects
  - A property that cannot be represented on one object only



Association

- Represent set of links between objects of different classes.

  {Is enrolled in 11,

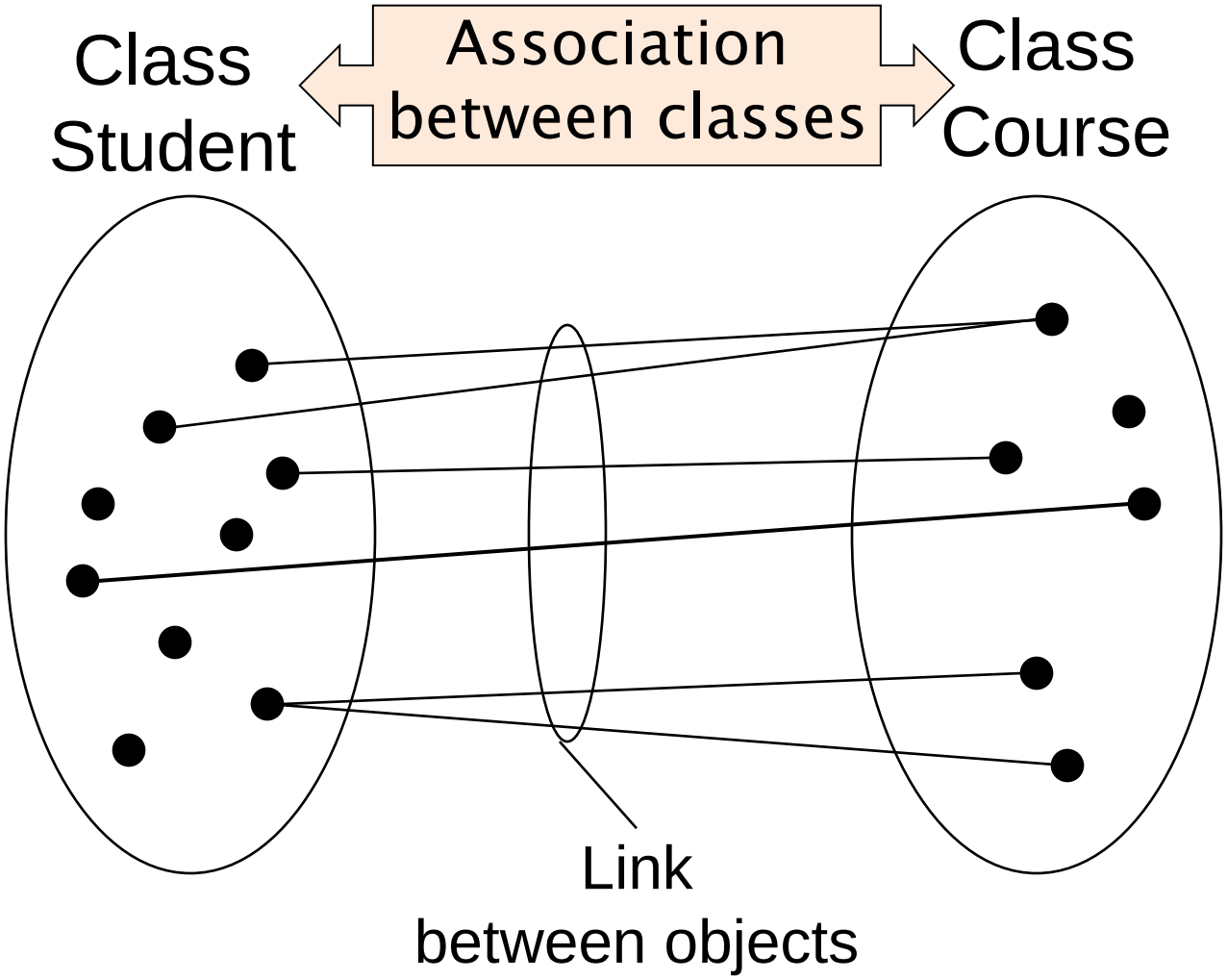
  Is enrolled in 32}

- Or pairs of objects (one per class):

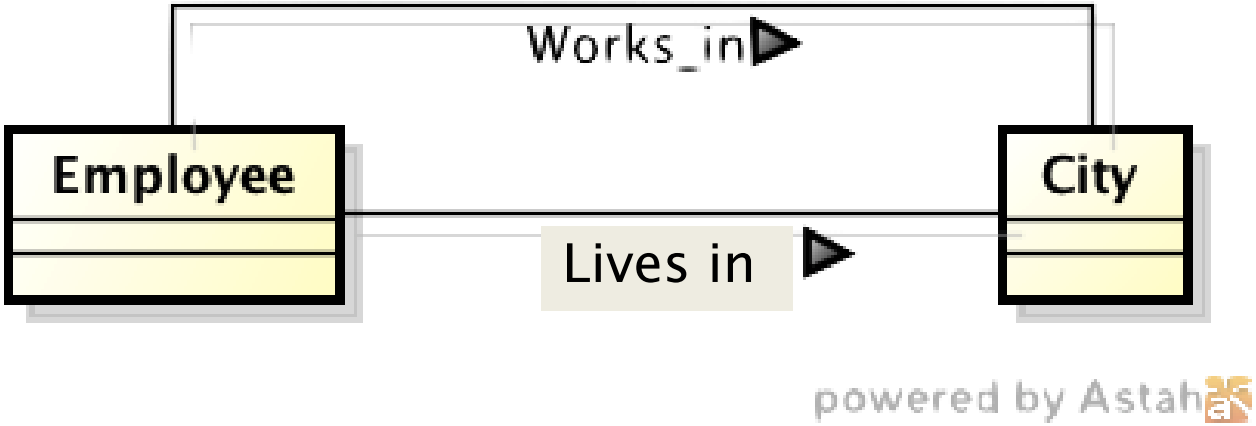
  {student1 – course1,

  student3 – course2 }

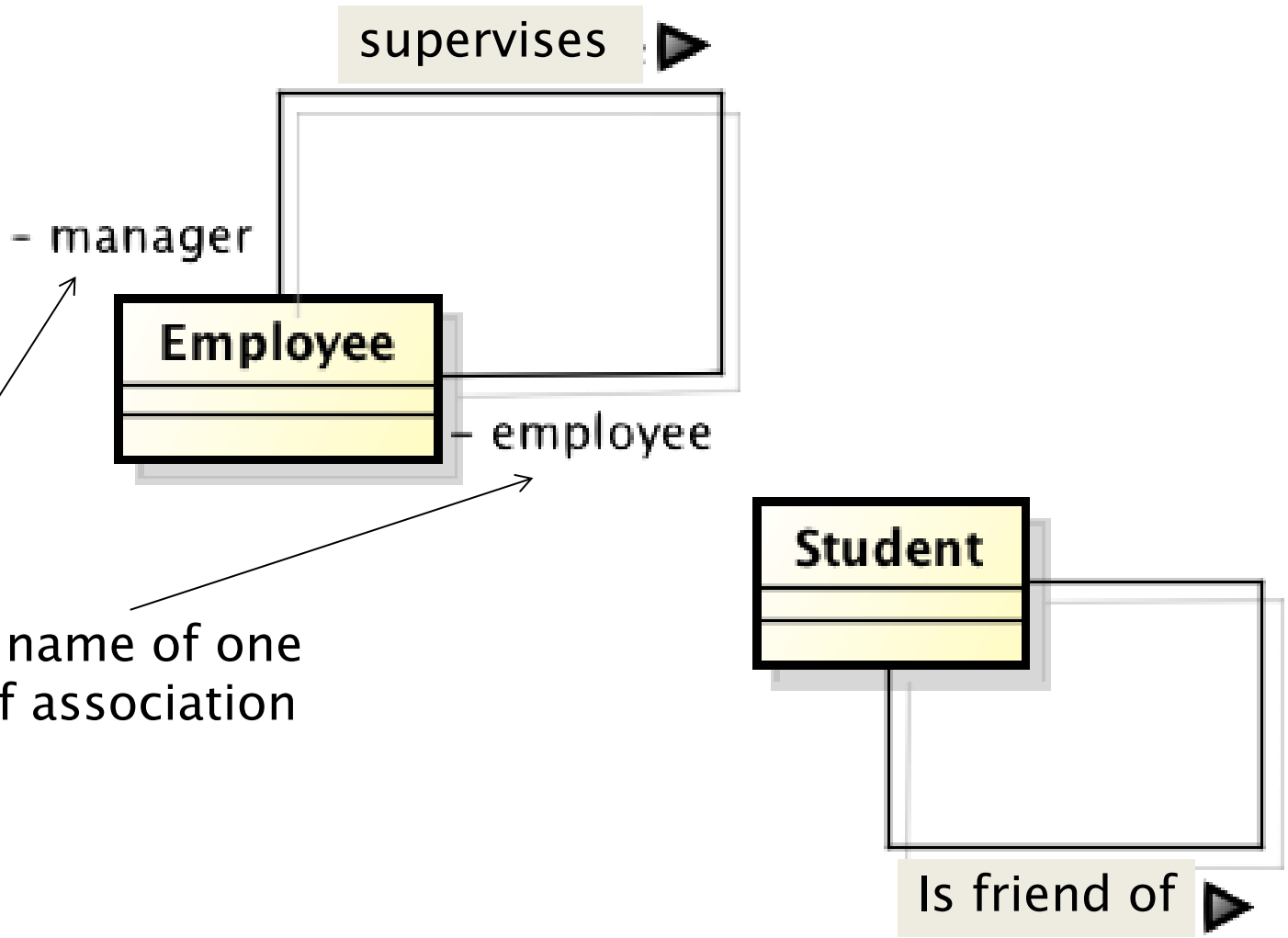
Associations



Association – Examples



Recursive associations

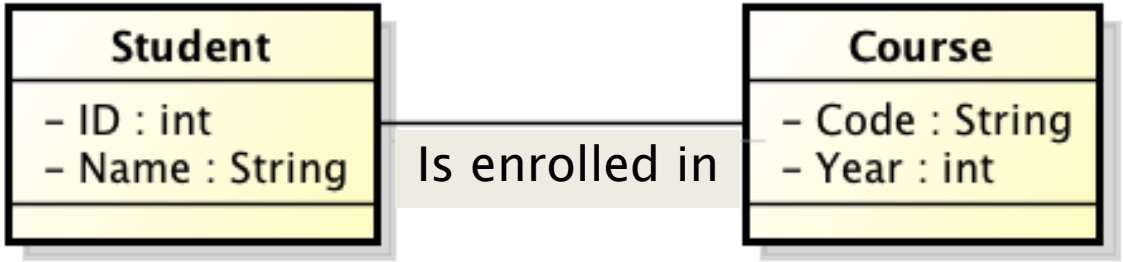


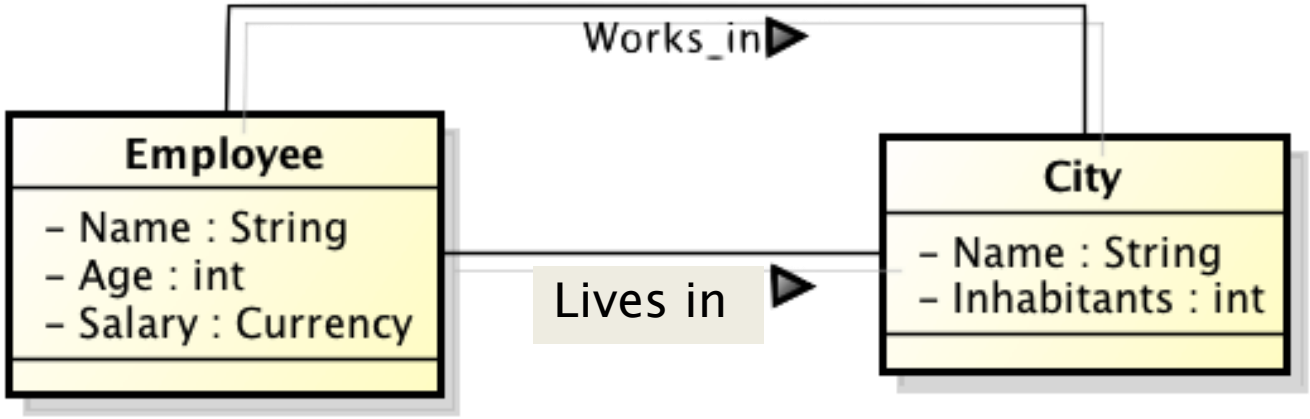
Role = name of one side of association

Attribute

- Elementary property of classes
  - Name
  - Type
- An attribute associates to each object a value of the corresponding type
  - Name: String
  - ID: Numeric
  - Salary: Currency

Attribute – Example





Style suggestions

- Class names
  - Singular noun
- Association name
  - Verb
- Attributes
  - Type of attribute not needed in conceptual model

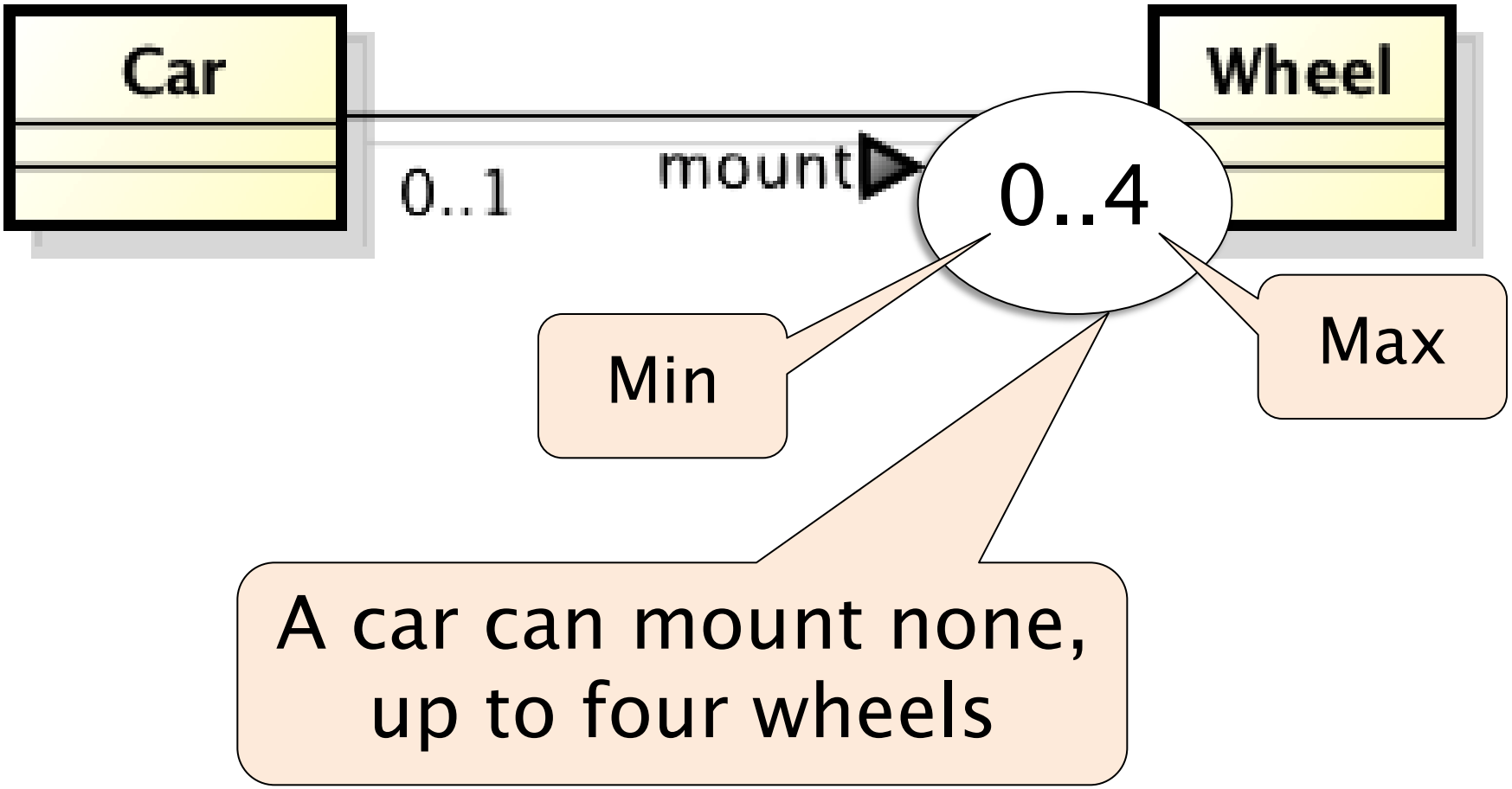
Multiplicity

多样性

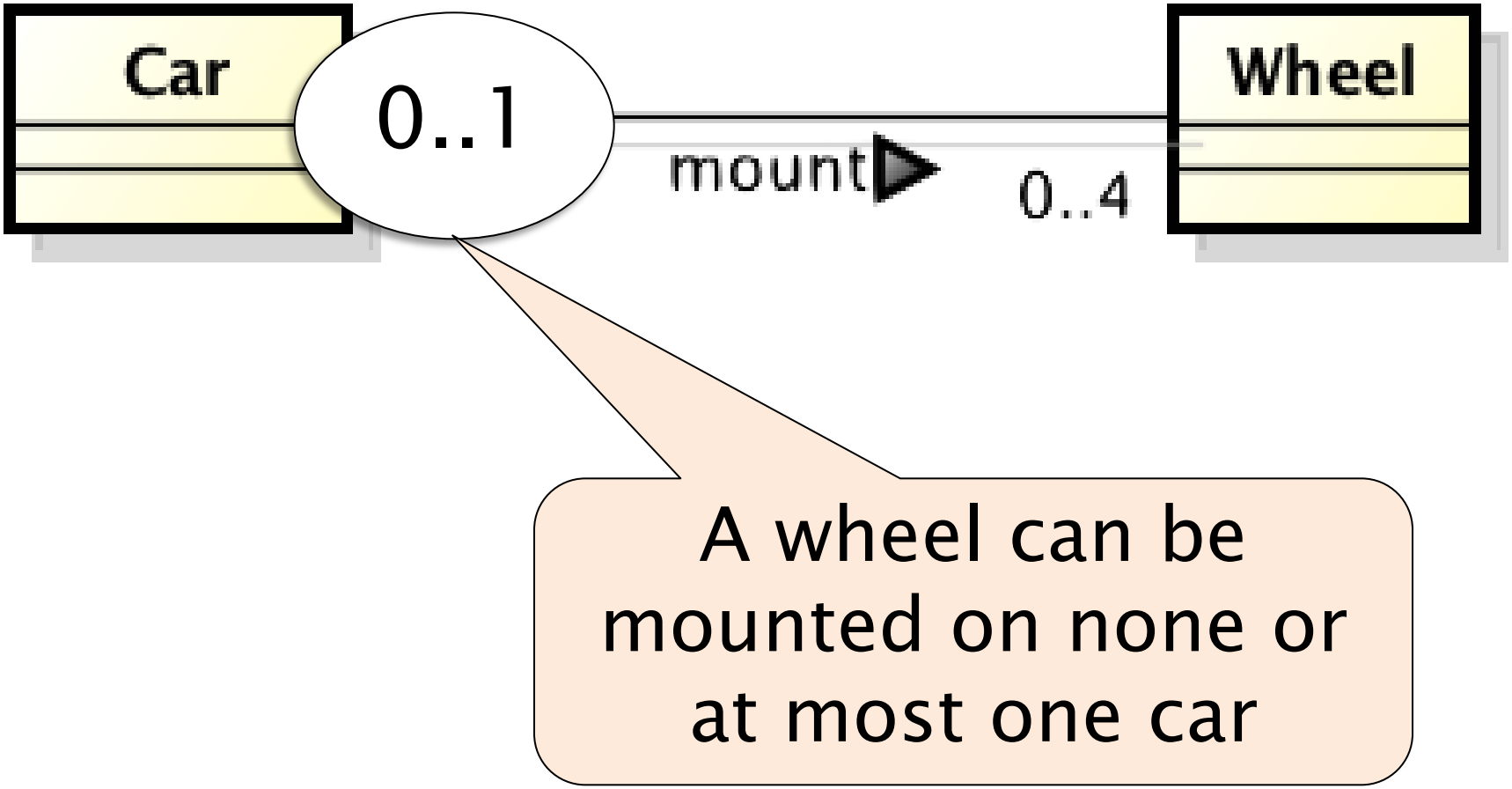
- Describe the maximum and minimum number of links in which an object of a class can participate

- Should be specified for each class participating in an association

Multiplicity – Example



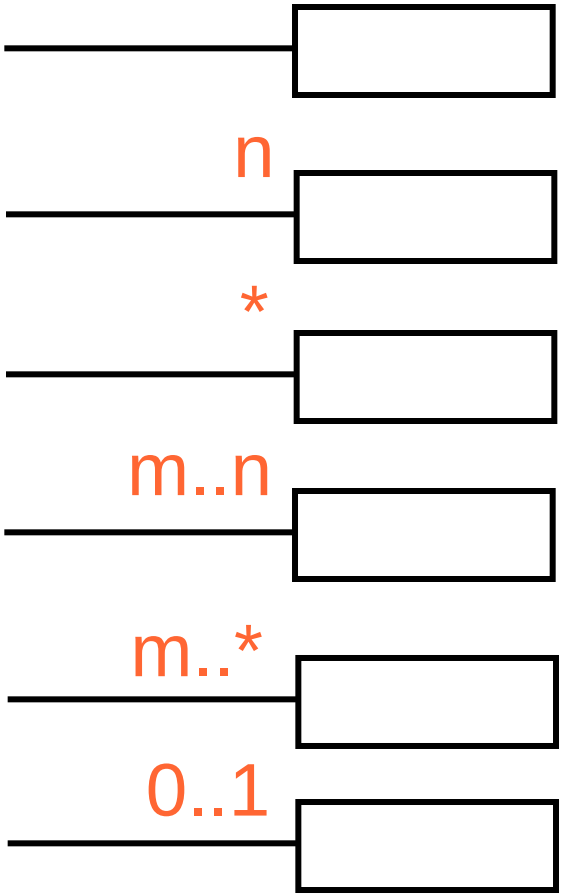
Multiplicity – Example



Multiplicity

- Typically, only three values are used: 0, 1 and the symbol * (many)
- Minimum: 0 or 1
  - 0 means the participation is *optional,*
  - 1 means the participation is *mandatory;*
- Maximum: 1 or *
  - 1: each object is involved in at most one link
  - *: each object is involved in many links

Multiplicity



Exactly 1

Exactly n

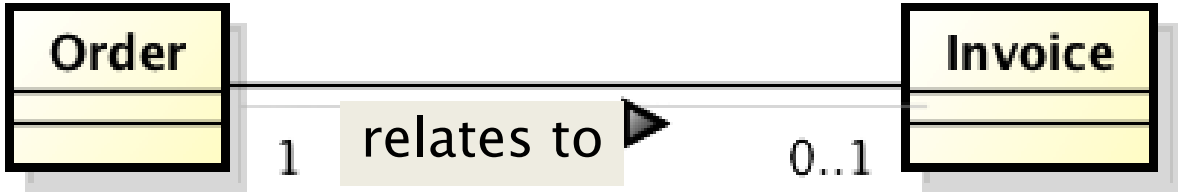
Zero or more

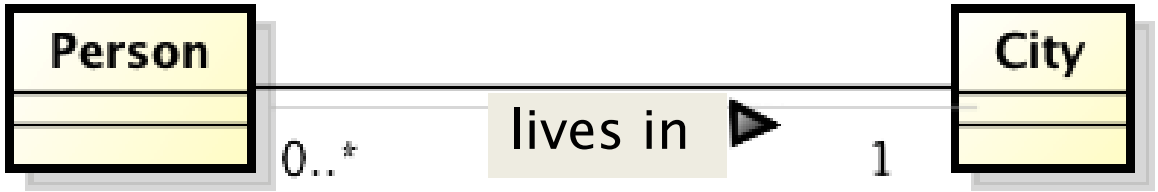
Between m and n (m,n included)

From m up

Zero or one (optional)

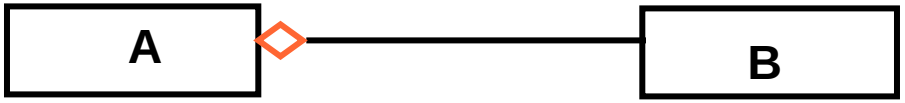
Multiplicity



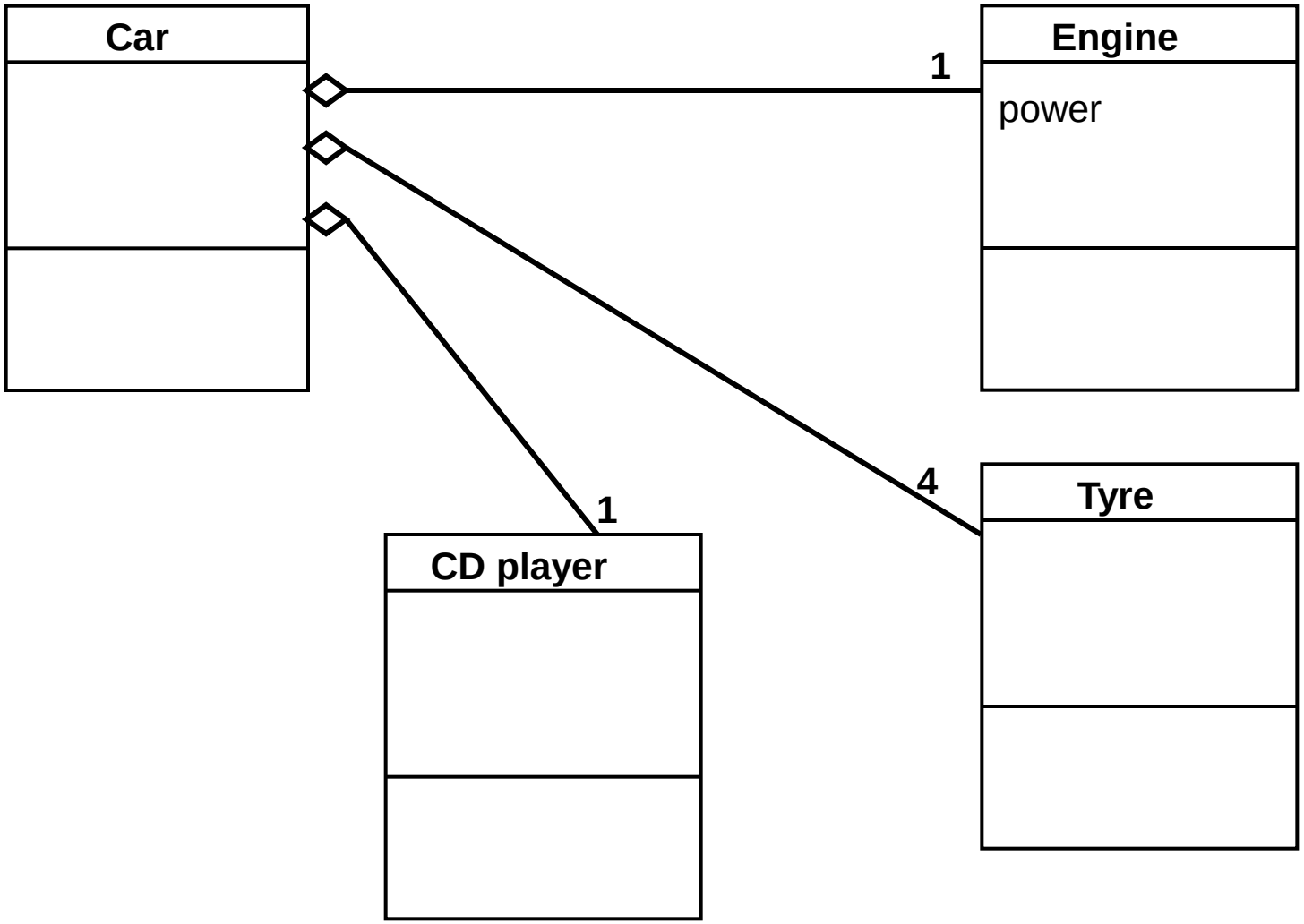


Aggregation

- Association with specific meaning
- B *is-part-of* A means that objects described by class B are part of (are components of) objects described by A
- Has
- Part-of

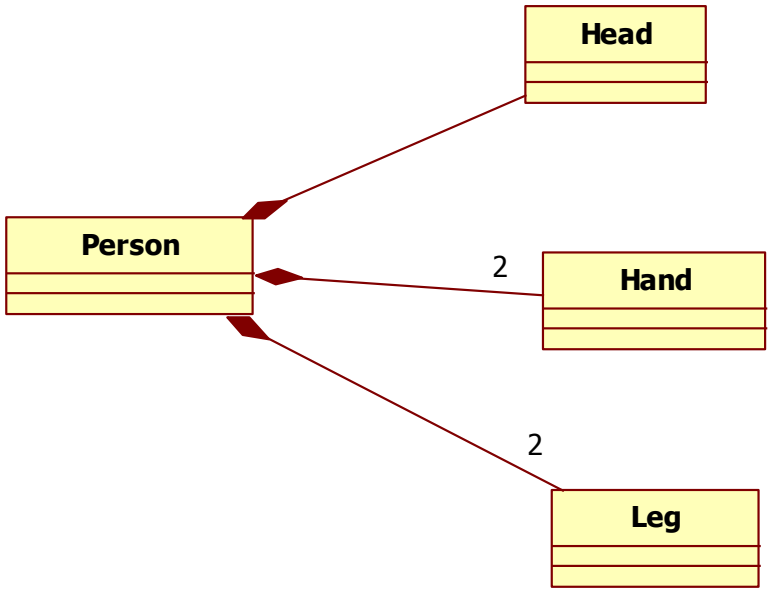


Example



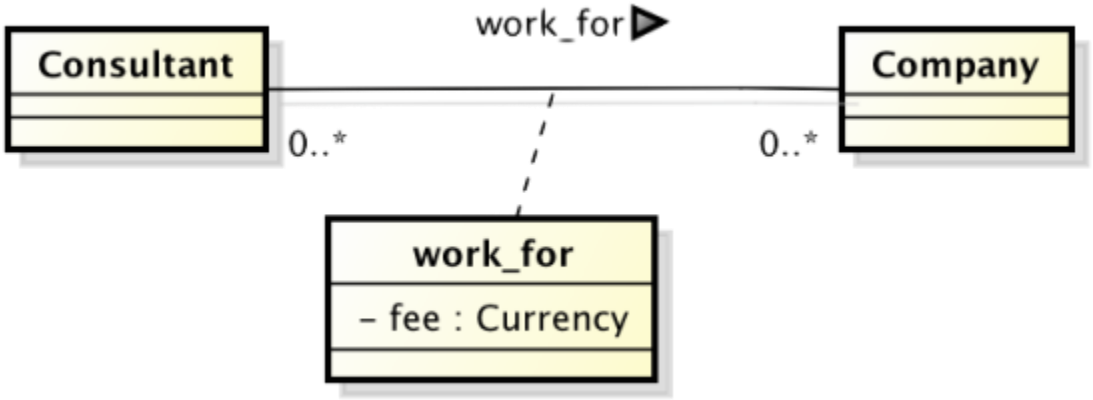
Composition

- An aggregation where the link part / whole is more strict: lifecycle of both classes is the same
  - if object Person disappears, so the corresponding 2 objects Leg, Hand



Association Class

- The association class allows to attach attributes to the association
- A link between two object includes
  - The two linked objects
  - The attributes of the link



Consultant Company fee

---

consultant1 – company2 – 300

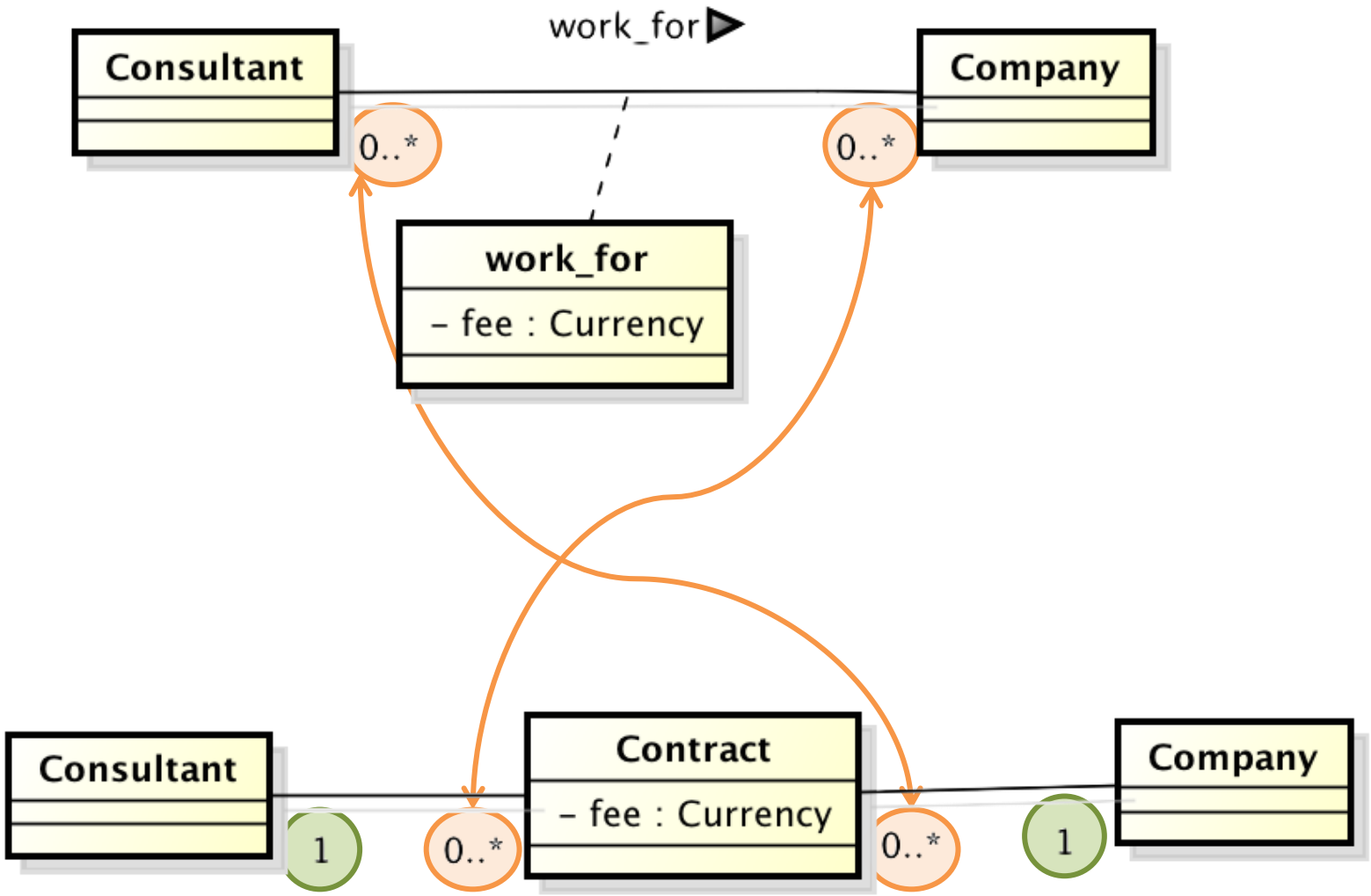
consultant1 – company3 – 200

~~consultant1 – company3 – 250~~

consultant2 – company2 – 100

consultant3 – company2 – 300

Instead of Association class

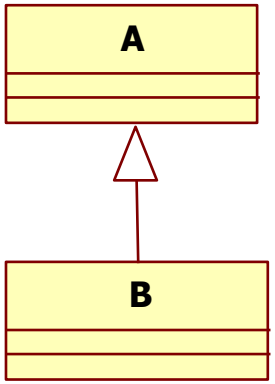


Consultant	Company	Contract
consultant1	company2	300
consultant1	company3	200
consultant1	company3	250
consultant2	company2	100
consultant3	company2	300

- The two options are equivalent, except

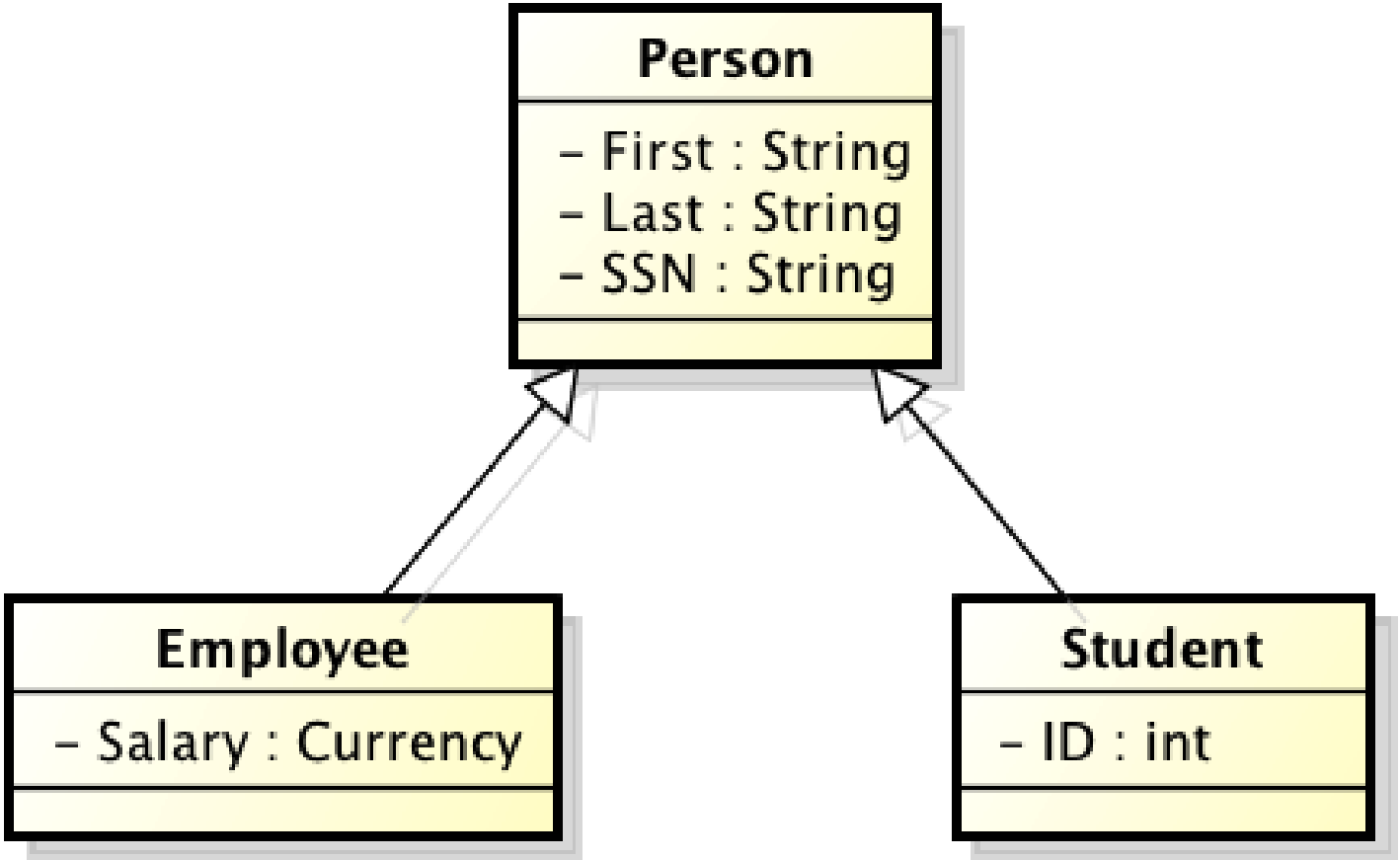
- Intermediate class:
  - More than one value for a link
- Association class:
  - Only one value for a link

Specialization / Generalization



- B *specializes* A means that objects described by B have the same properties of objects described by A
- Objects described by B may have additional properties
- B is a special case of A
- A is a generalization of B (and possibly of other classes)

Generalization

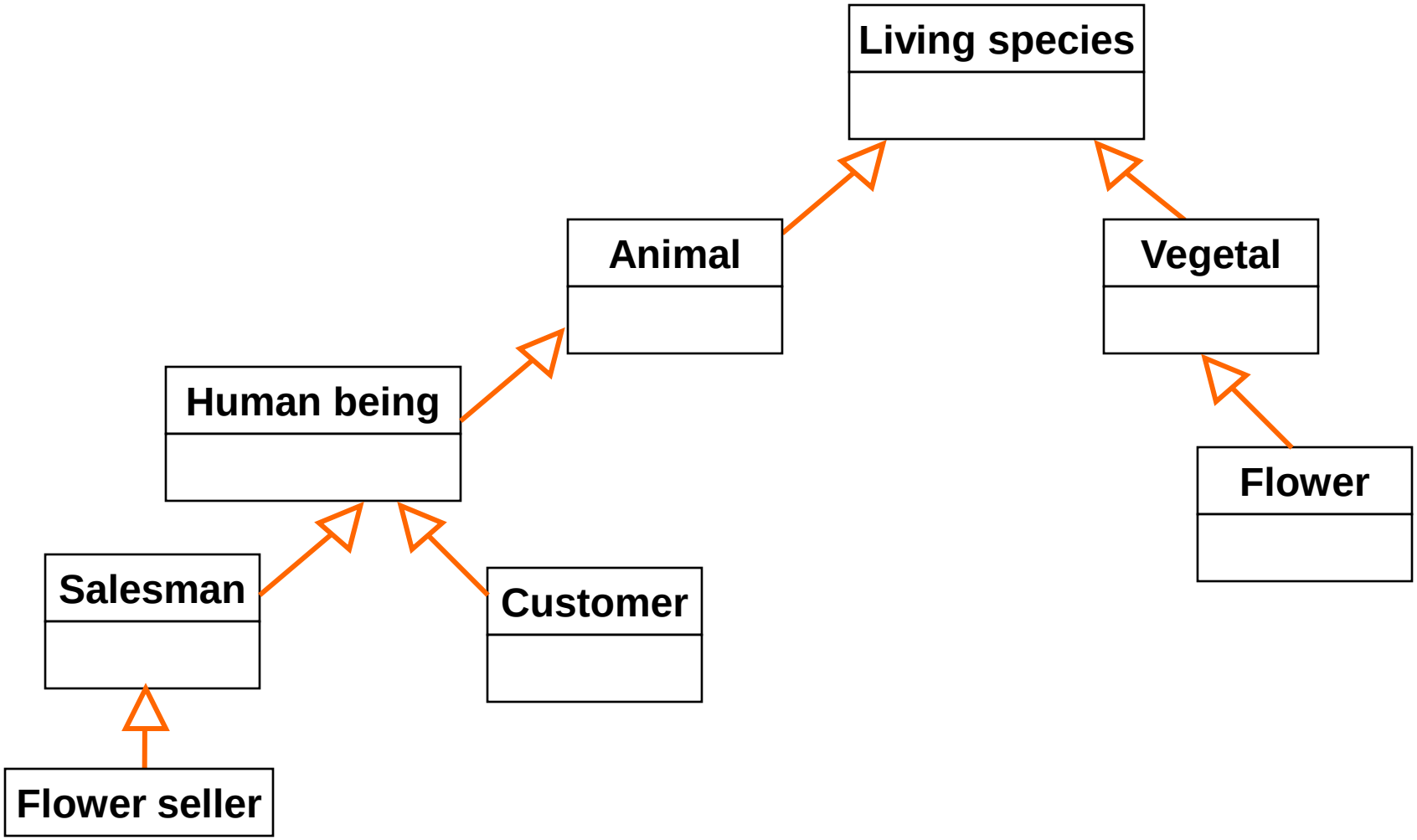


- Specialization can be used only if it is possible to state
  - B is-a A
- Employee is-a Person – yes
- Student is-a Person – yes
- Head is-a Person – no
- Person has-a Head – yes

Inheritance terminology

- Class one above
  - Parent class
- Class one below
  - Child class
- Class one or more above
  - Superclass, Ancestor class, Base class
- Class one or more below
  - Subclass, Descendent class, Derived class

Example of inheritance tree



DOs in Class Diagram

- Decide goal of model
  - In context of this course, conceptual model

Dos – consider:

- Physical entities: Person, Car,
- Roles: Employee, Director, Doctor,
- Social / legal / organizational entities: University, Company, Department
- Events: Sale, Order, Request, Claim, Call
- Time intervals: Car rental, Booking, Course, Meeting
- Geographical entities: City, Road, Nation
- Reports, summaries, paper documents: weather report, bank account statement, travel request

DO NOT in class diagrams

- Use plurals for classes
  - Person yes, PersonS no
- Use transient (dynamic) relationships
  - (they will be modeled in scenarios, sequence diagrams)
- Forget multiplicities
- Forget roles / association classes, when needed
- Use class as an attribute
- Use attribute that represents many objects

DO NOT in class diagrams

- Repeat as an attribute of a class a relationship starting from the class
- Confound system design, software design, glossary
  - DO decide goal of diagram

BE CAREFUL in class diagrams

- Loops in relationships (normally avoid them)