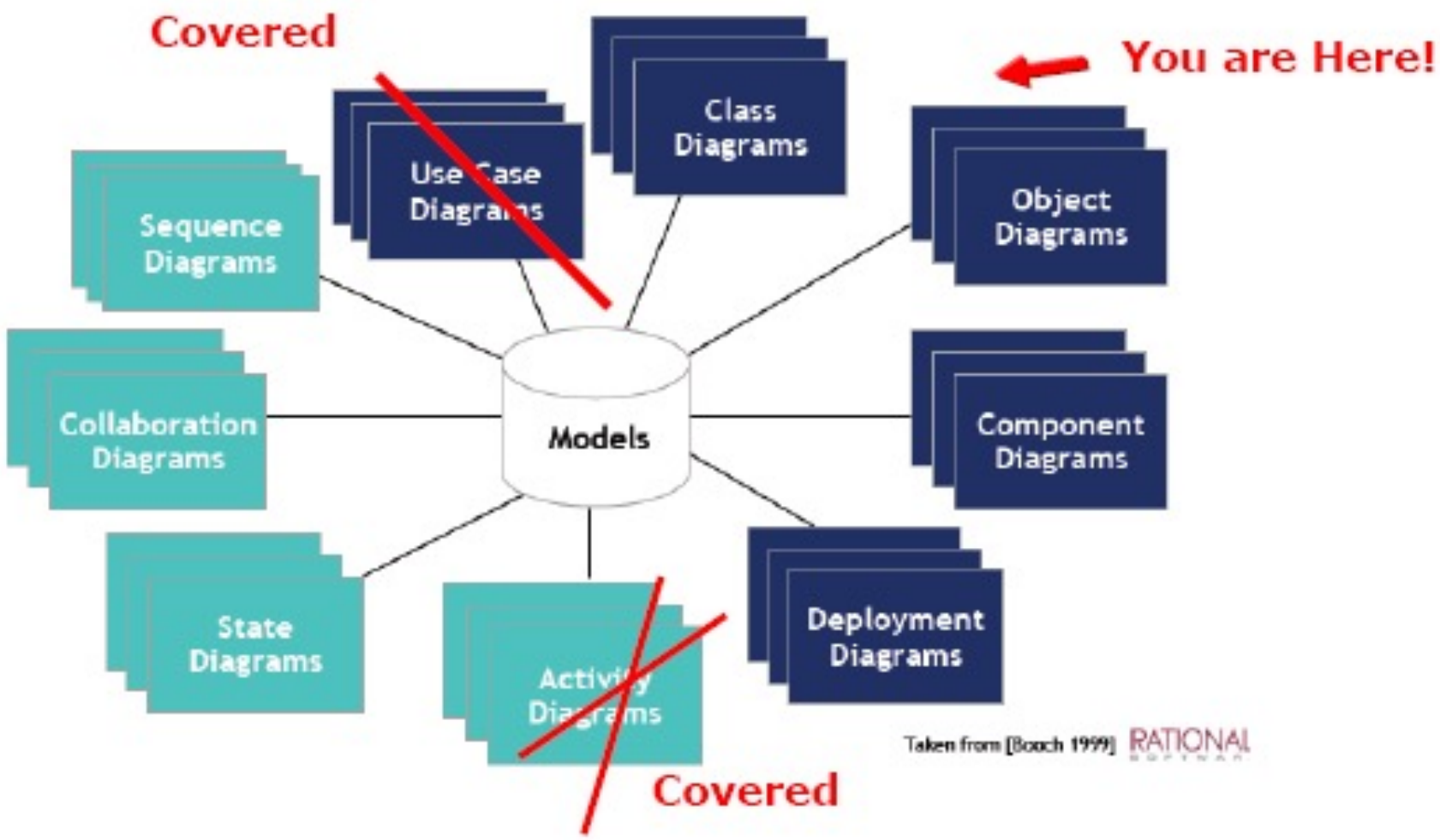
# UML Diagrams



# Class diagrams

Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.

An object class can be thought of as a general definition of one kind of system object.

An association is a link between classes that indicates that there is some relationship between these classes.

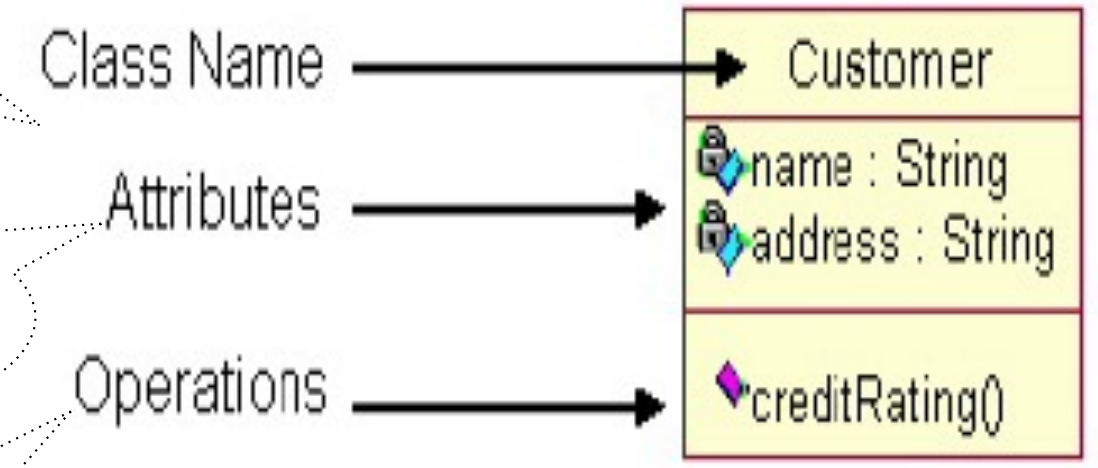
When you are developing models during the early stages of the software engineering process, objects represent something in the real world, such as a patient, a prescription, doctor, etc.

# Simple Class Diagram

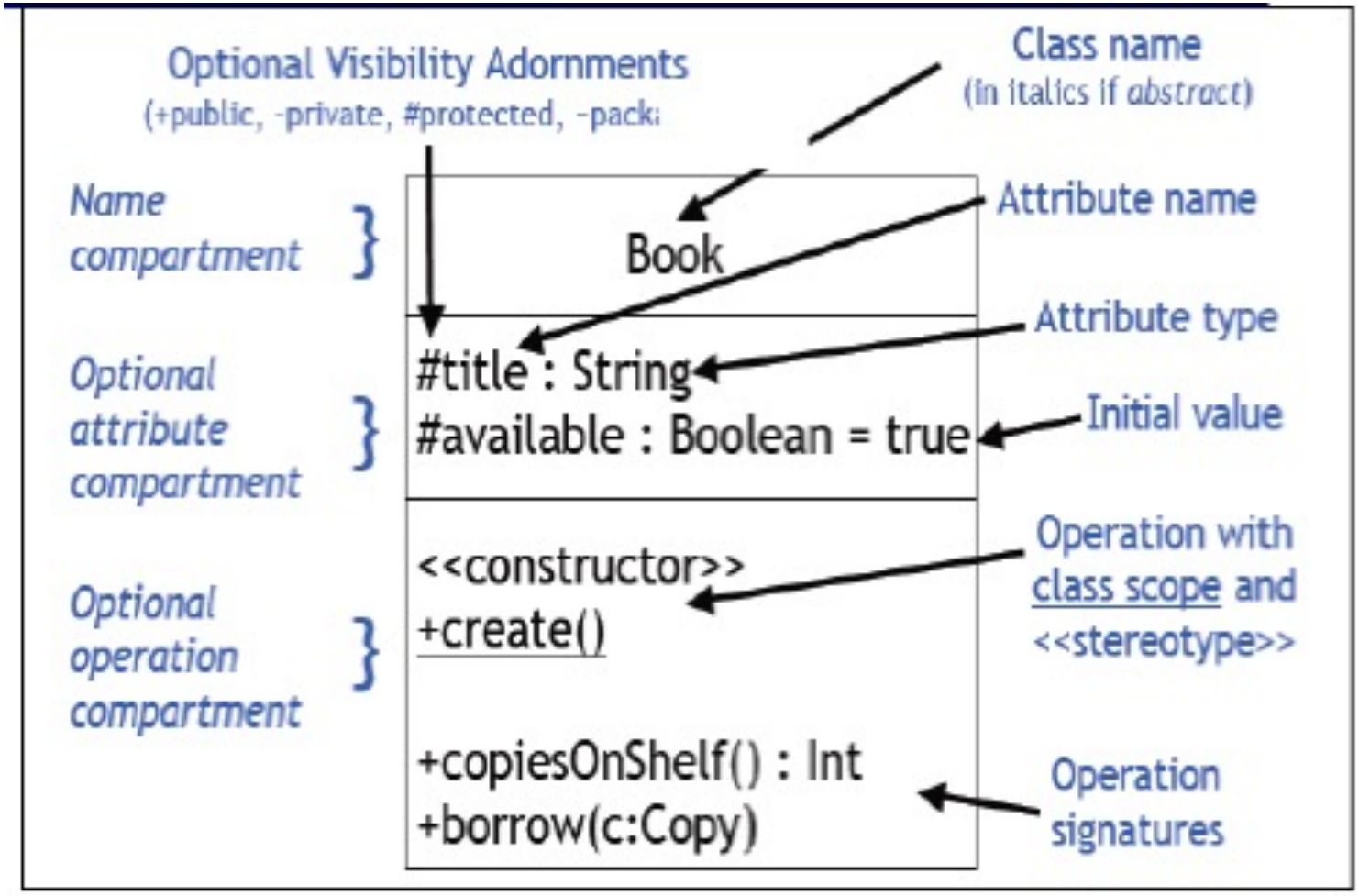
This represents a system object, which may refer to an Actor or entity of the system

These relate to the data that system must store for this object

These relate to the use cases provided by this object



# UML Class Icons




Reference: D. Rosenblum, UCL

# +, #, -

+ means public: public members can be accessed by any client of the class
# means protected: protected members can be accessed by members of the class or any subclass
- means private: private members can only be accessed by members of the same class

Additional:
~ package visibility
/ derived classes visibility

# General Design Approaches

- Top-Bottom Approach

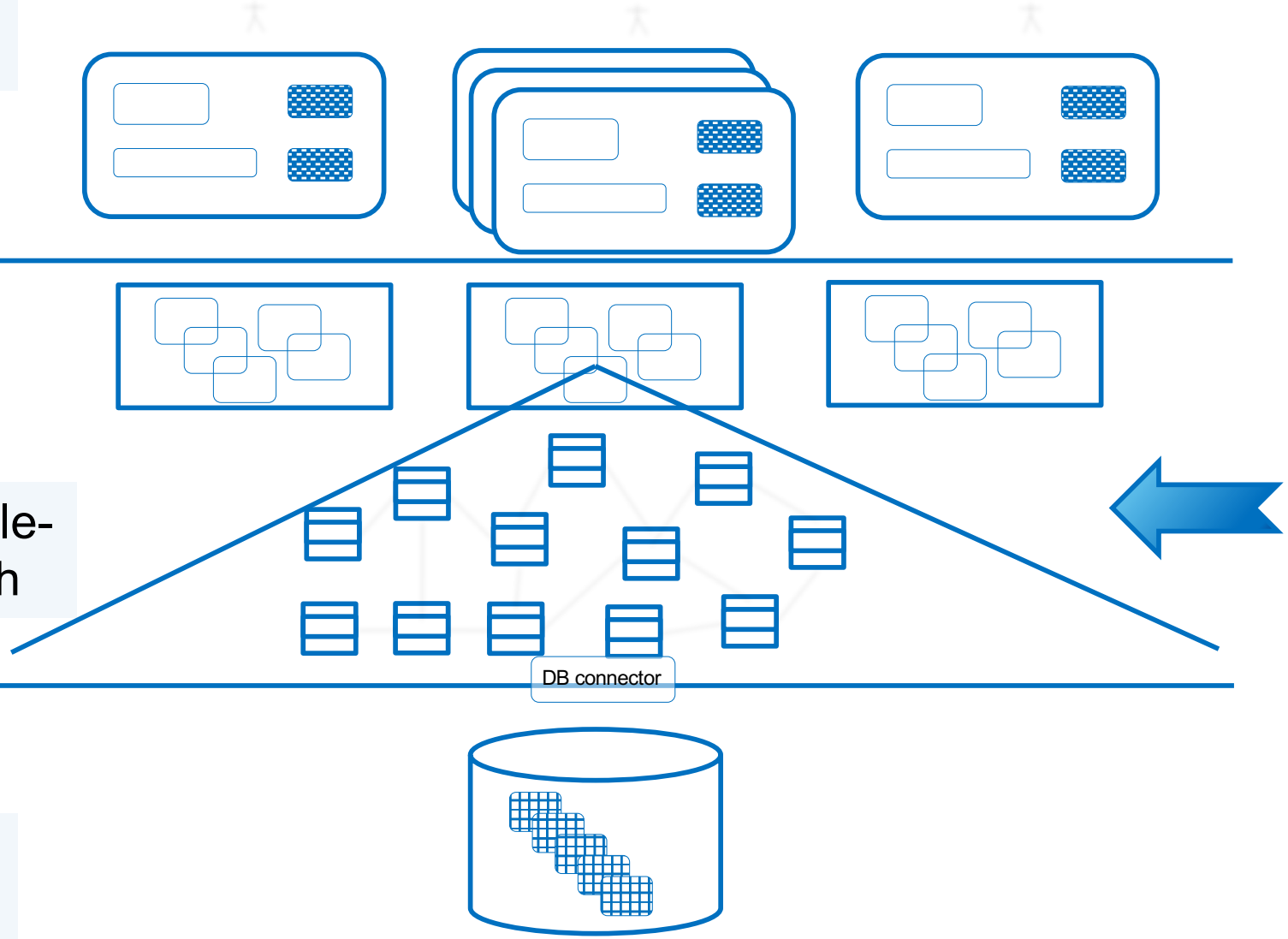
Top

Middle

- Middle-top-Middle-bottom Approach

Bottom

- Bottom-Top Approach



# Analysis Class

An analysis class abstracts one or more classes
and/or
subsystems in the system's design
- Focuses on handling functional requirements
- Defines responsibilities (cohesive subsets of behaviour defined by the class, e.g. use cases or services it provides to other classes)
- Defines attributes
- Expresses relationships the class is involved in

# Approach: Data-Driven Design

Identify all the data in the system
Divide into classes before considering responsibilities
Common approach: **noun identification**
- Identify candidate classes by selecting all the nouns and nouns phrases in the requirements document
- Discard inappropriate candidates
  - Redundant or omnipotent entities
  - Vague entities
  - Events or operations
  - Meta-language
  - Entities outside system scope
  - Attributes

Verbs and verb phrases highlight candidate operations!

# Data-Driven Design Approach

Some heuristics/hints of what kind of things are classes [Shlaer and Mellor; Booch]:

**Tangible** or "**real-world**" things – e.g. book, copy, course;

**Roles**- e.g. library member, student, director of studies,

**Events**- e.g. arrival, leaving, request;

**Interactions**- e.g. meeting, intersection

# Exercise

Perform **noun-verb** analysis of a requirements document (example text from next slide);
Underline all the noun and noun phrases,
Create a list of candidate classes (in examining the discard criteria, you may also identify some candidate attributes)

Identify all verb and verb phrases
Create a list of candidate operations and assign them to classes

# Noun/Verb Analysis

**Books and journals:**
The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

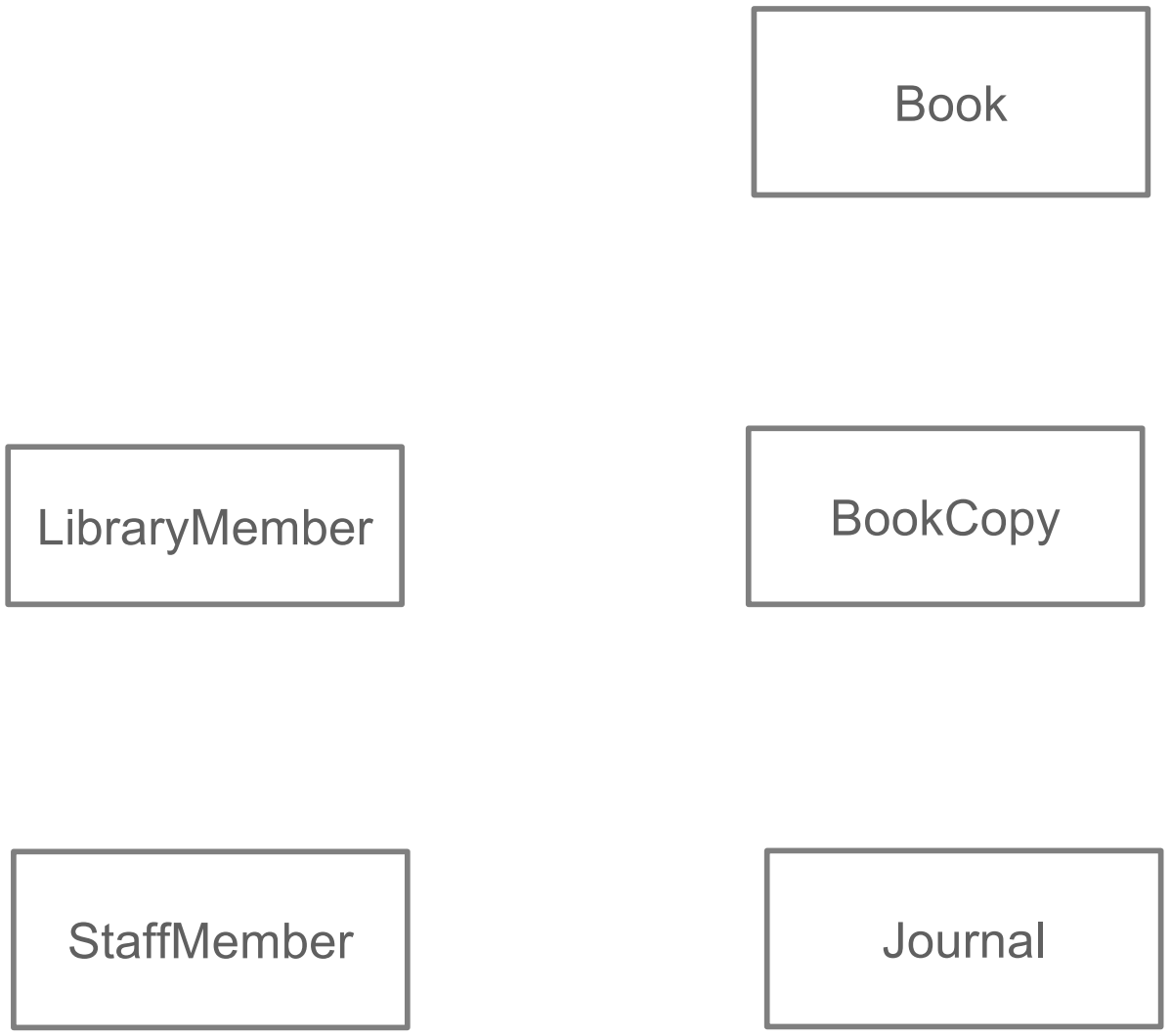
**Borrowing**:
The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

# 1. Noun Analysis

**Books and journals:**
The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

**Borrowing**:
The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

# First-Cut Class Diagram: Class Model (Analysis Classes)

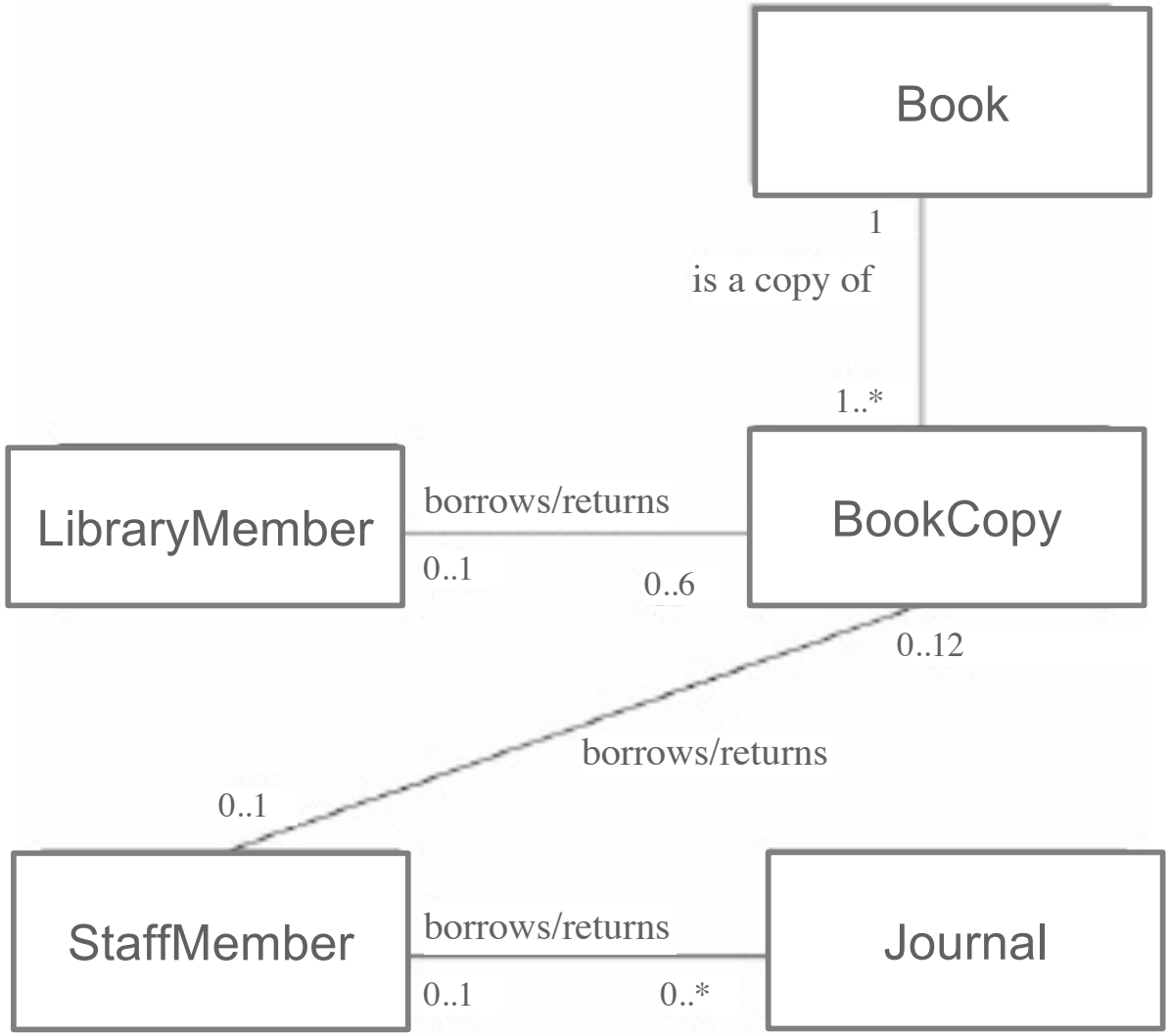


# 2. Verb Analysis

**Books and journals:**
The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

**Borrowing**:
The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

# First-Cut Class Diagram: Class Model



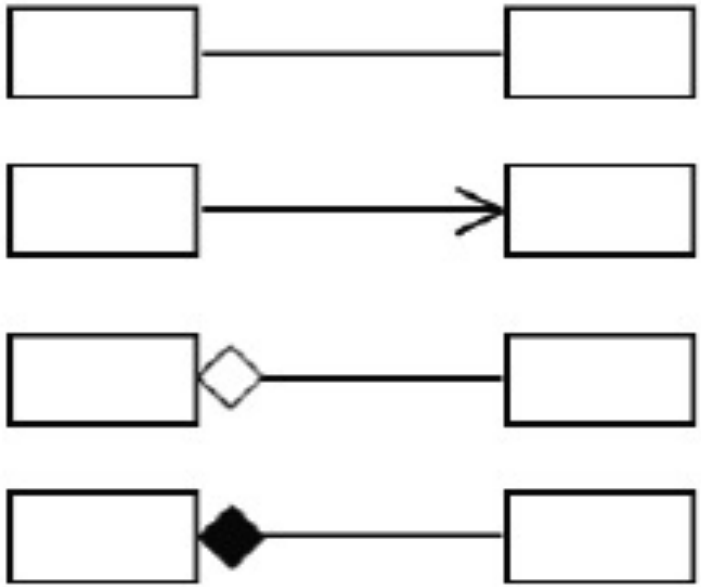
# Relationships/Associations

Relationships are connections between modelling elements
Improve understanding of the domain, describing how objects work together
Act as a sanity check for good modelling

Associations are relationships between classes
Examples
- Object of class A sends a message to object of class B
- Object of class A creates an object of class B
- Object of class A has attribute whose values are objects of class B
- Object of class A receives a message with argument of class B

Links are relationships between objects
- Links can be instances of associations (as in UML 1.4)
- Allow one object to invoke operations on another object

# UML Relationships Notations



bidirectional / binary

unidirectional

aggregation

composition

Generalization

Inheritance

Composition

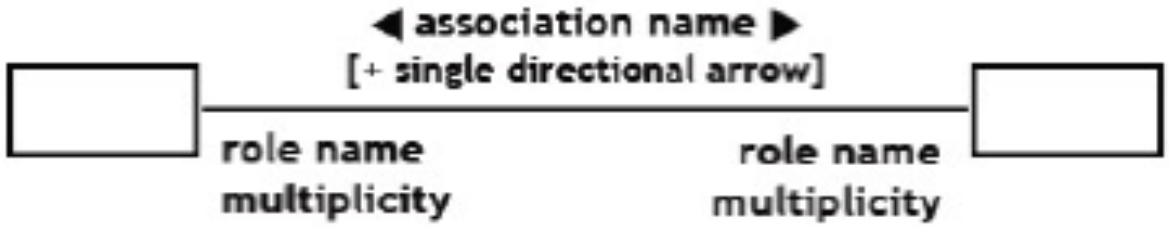
Aggregation

Dependencies

<< >>

Properties

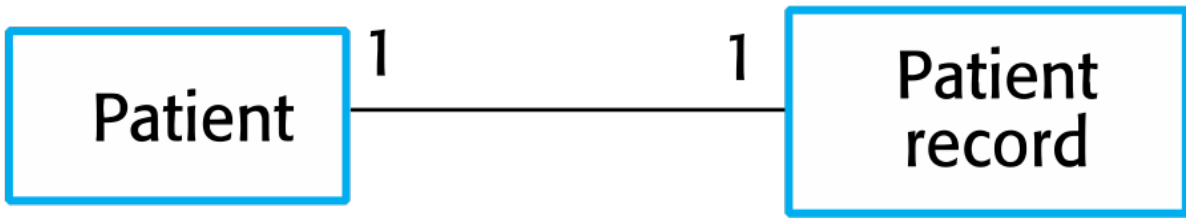

Multiplicity



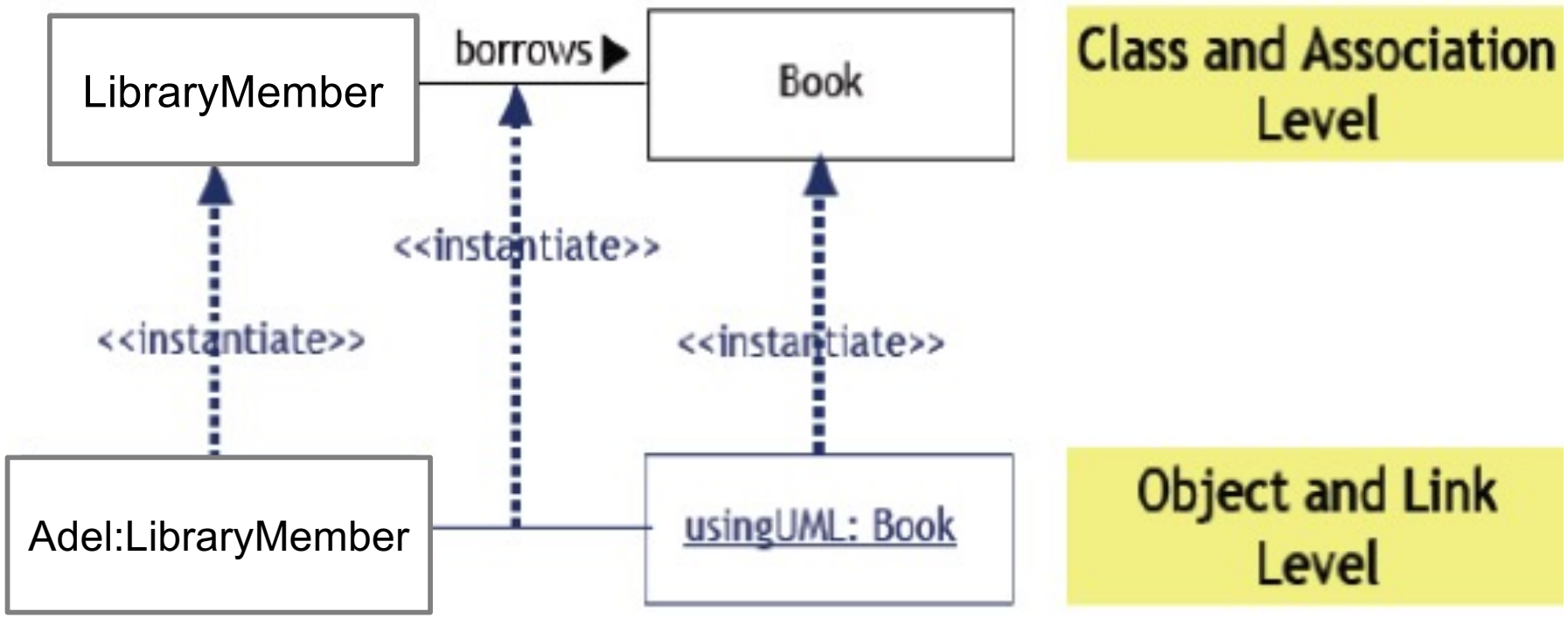

supplementary characteristics

Reference: D. Rosenblum, UCL

# UML classes and association



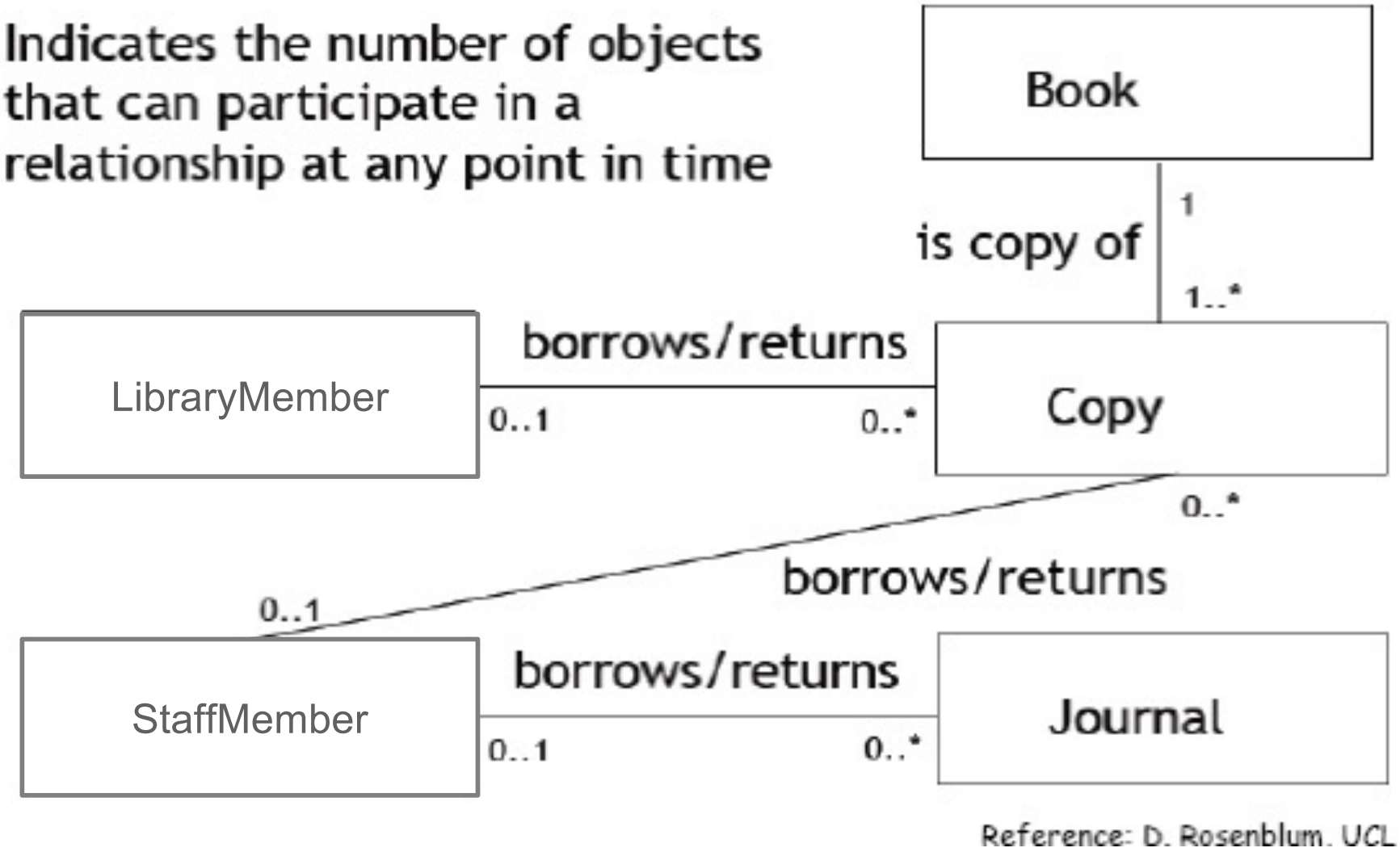
# Links Instantiate Associations




Reference: D. Rosenblum, UCL

# Multiplicity of an Association

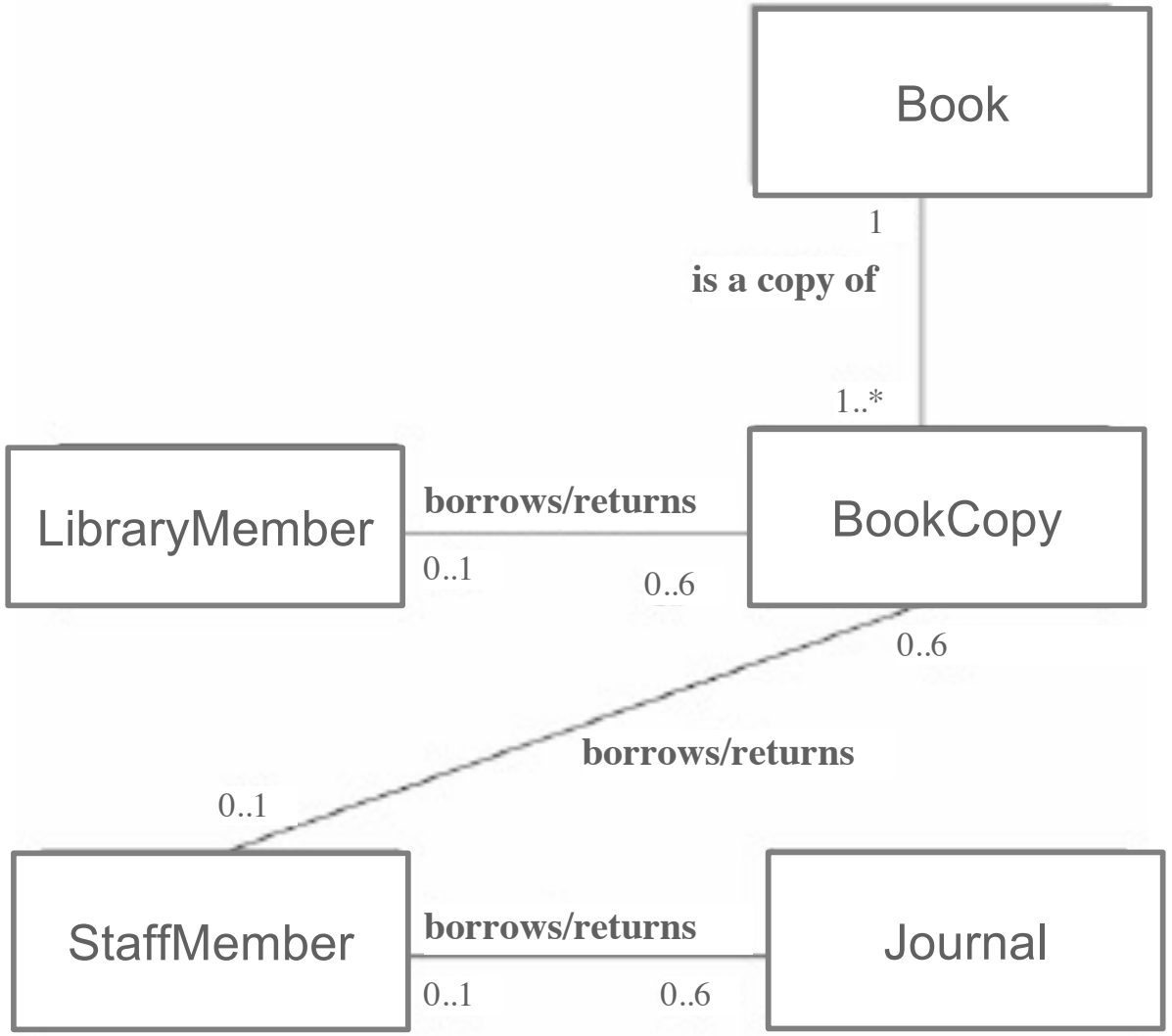
- Indicates the number of objects that can participate in a relationship at any point in time



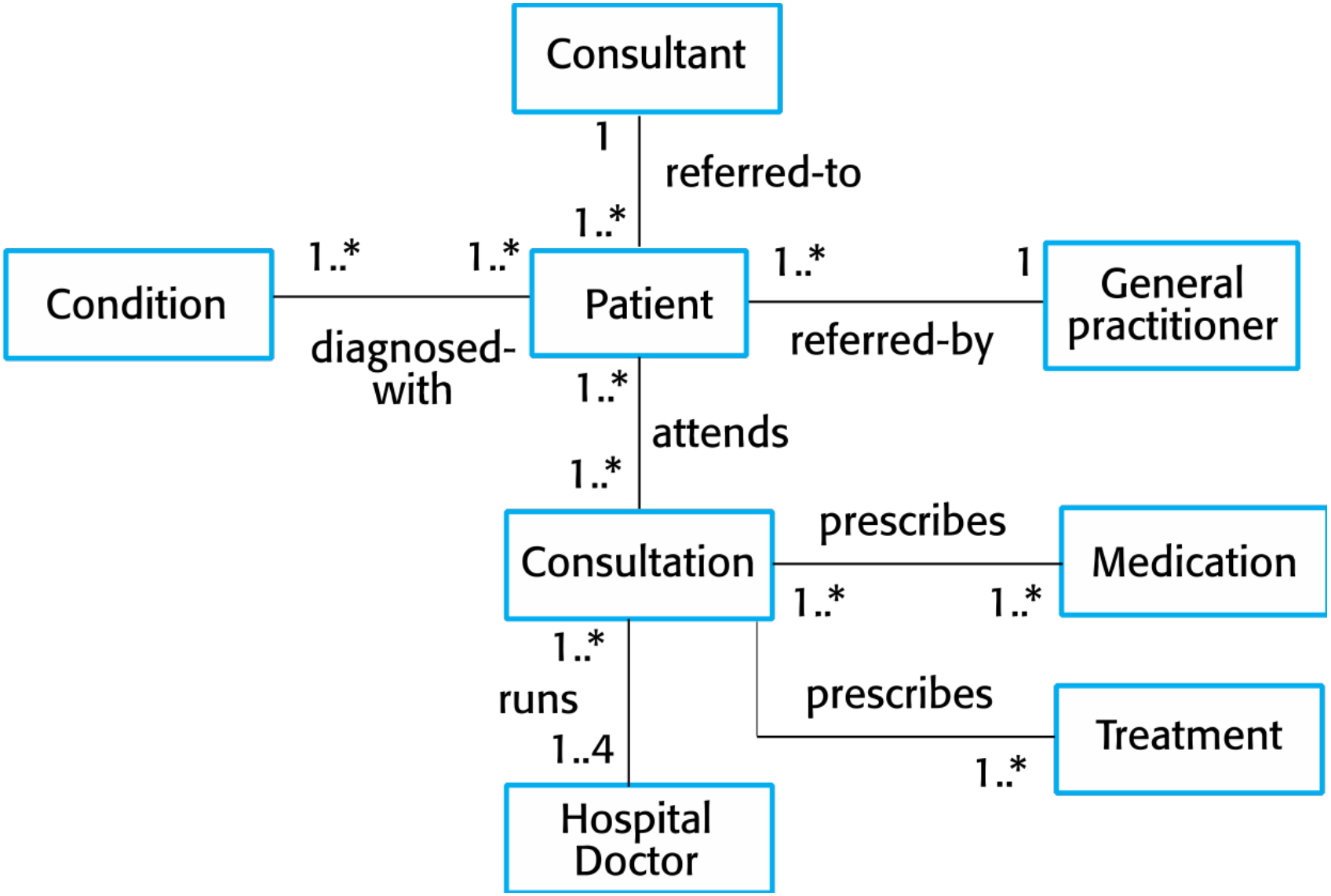

Reference: D. Rosenblum, UCL

# Multiplicity of an Association

- Indicates the number of objects that can participate in a relationship at (any point in) a time

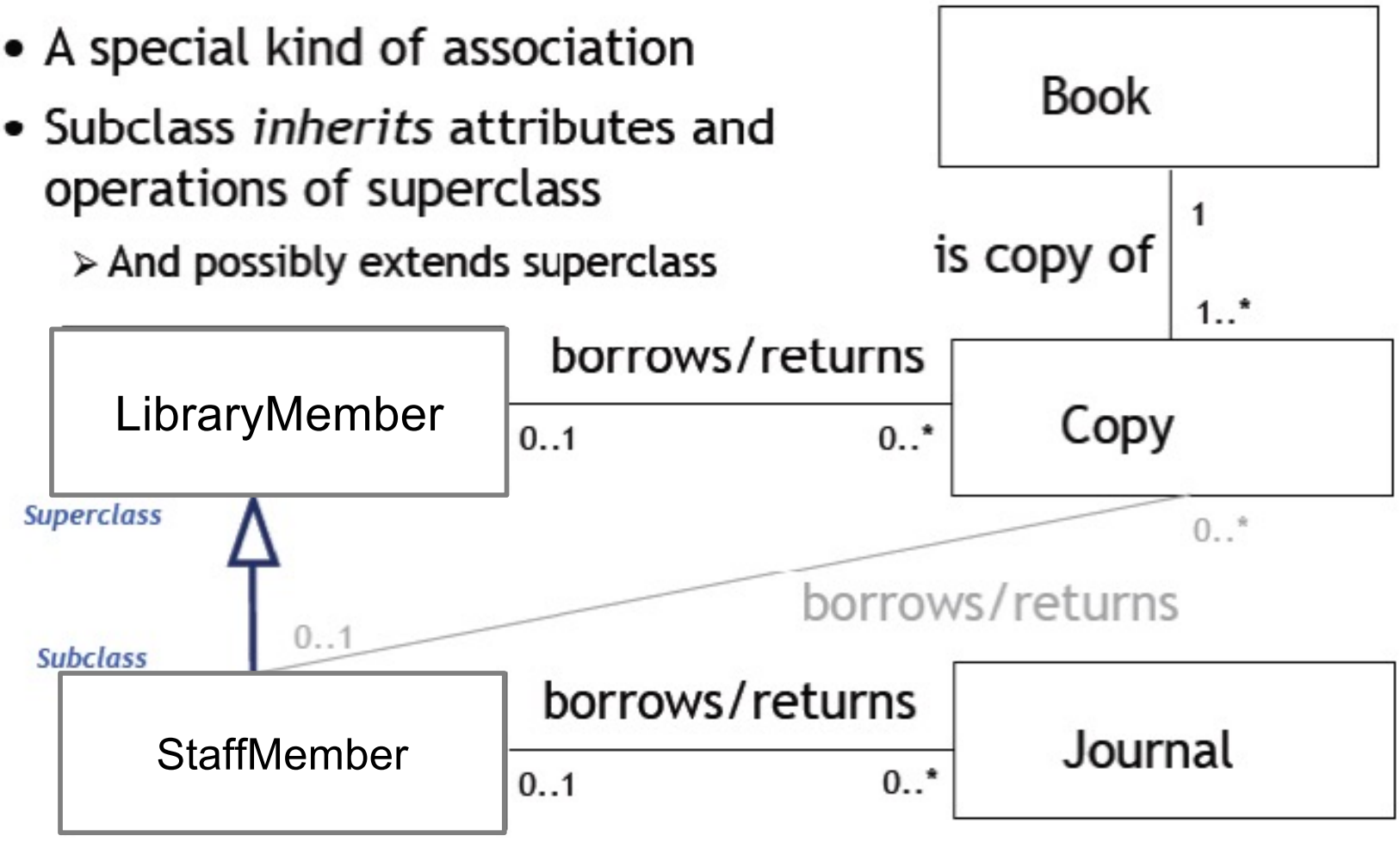


# Class diagram/Model of the MHC-PMS



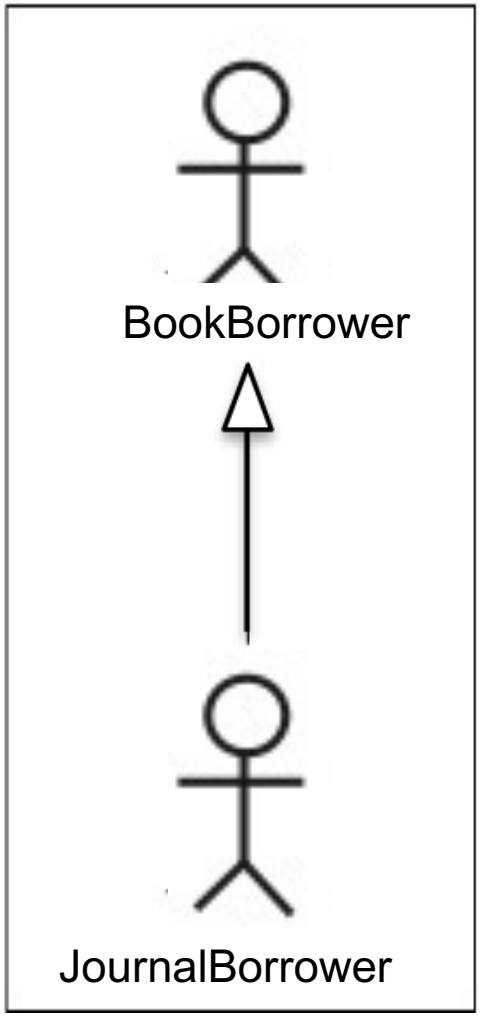
# Generalisation (Inheritance)

- A special kind of association
- Subclass *inherits* attributes and operations of superclass
  - And possibly extends superclass

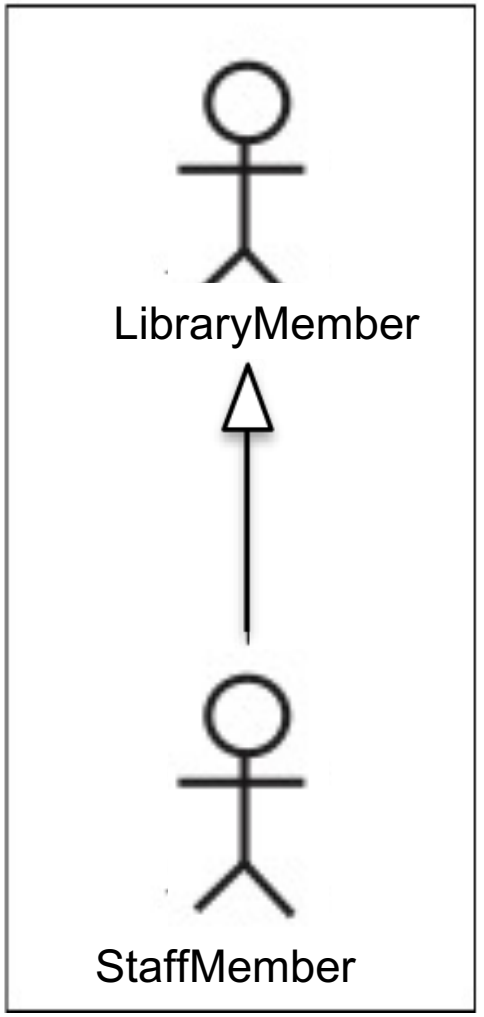


# Generalization

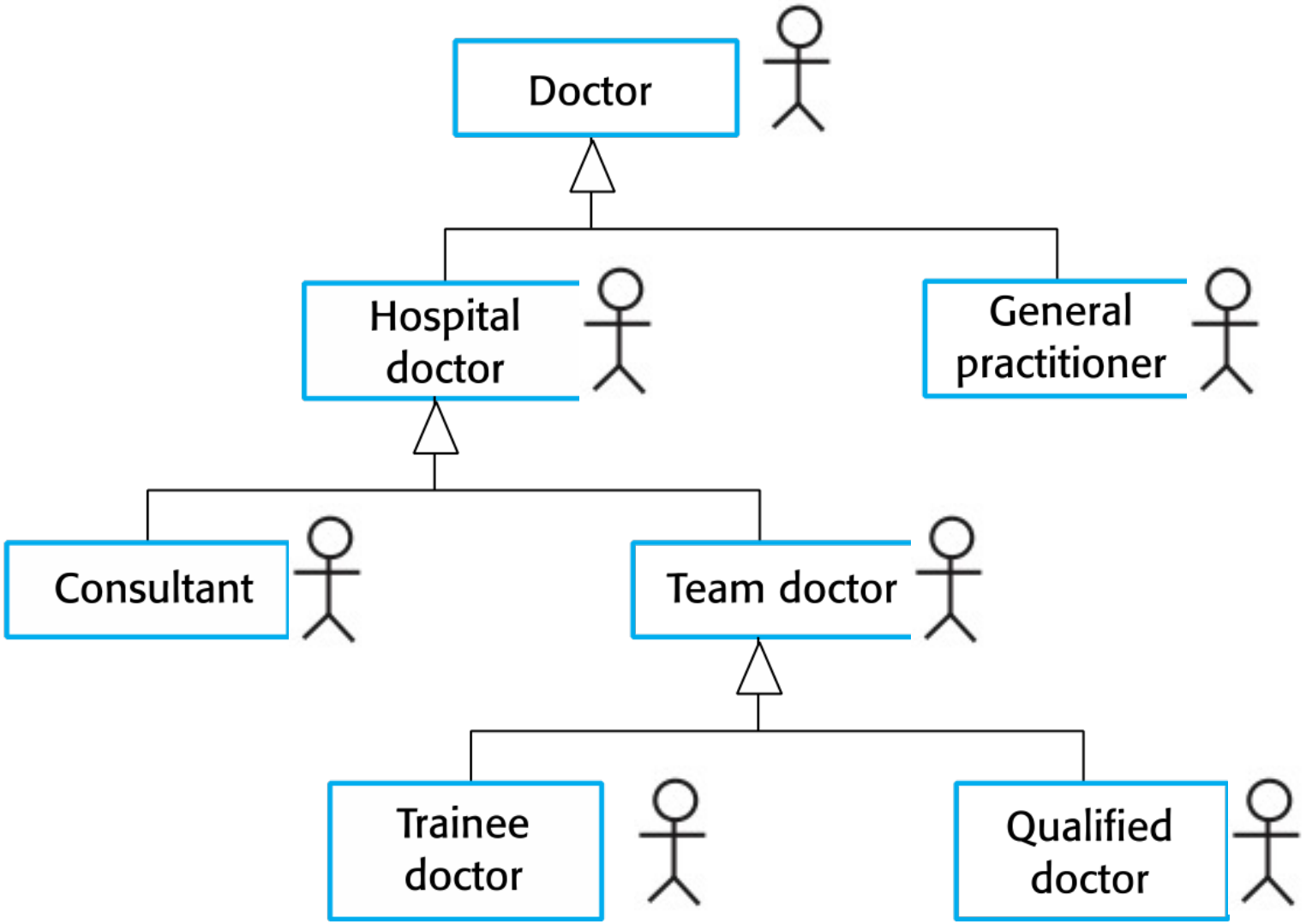
Journal Borrower is a book borrower



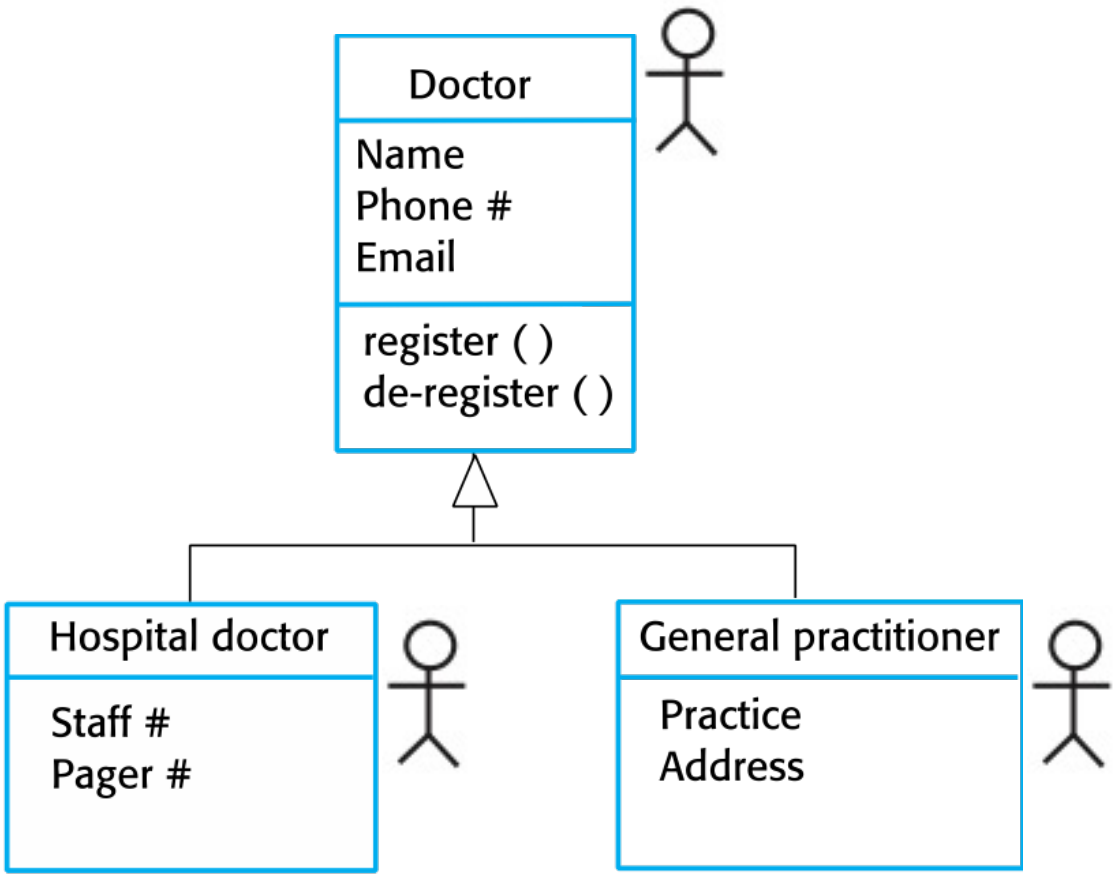

Member of Staff is a member of library



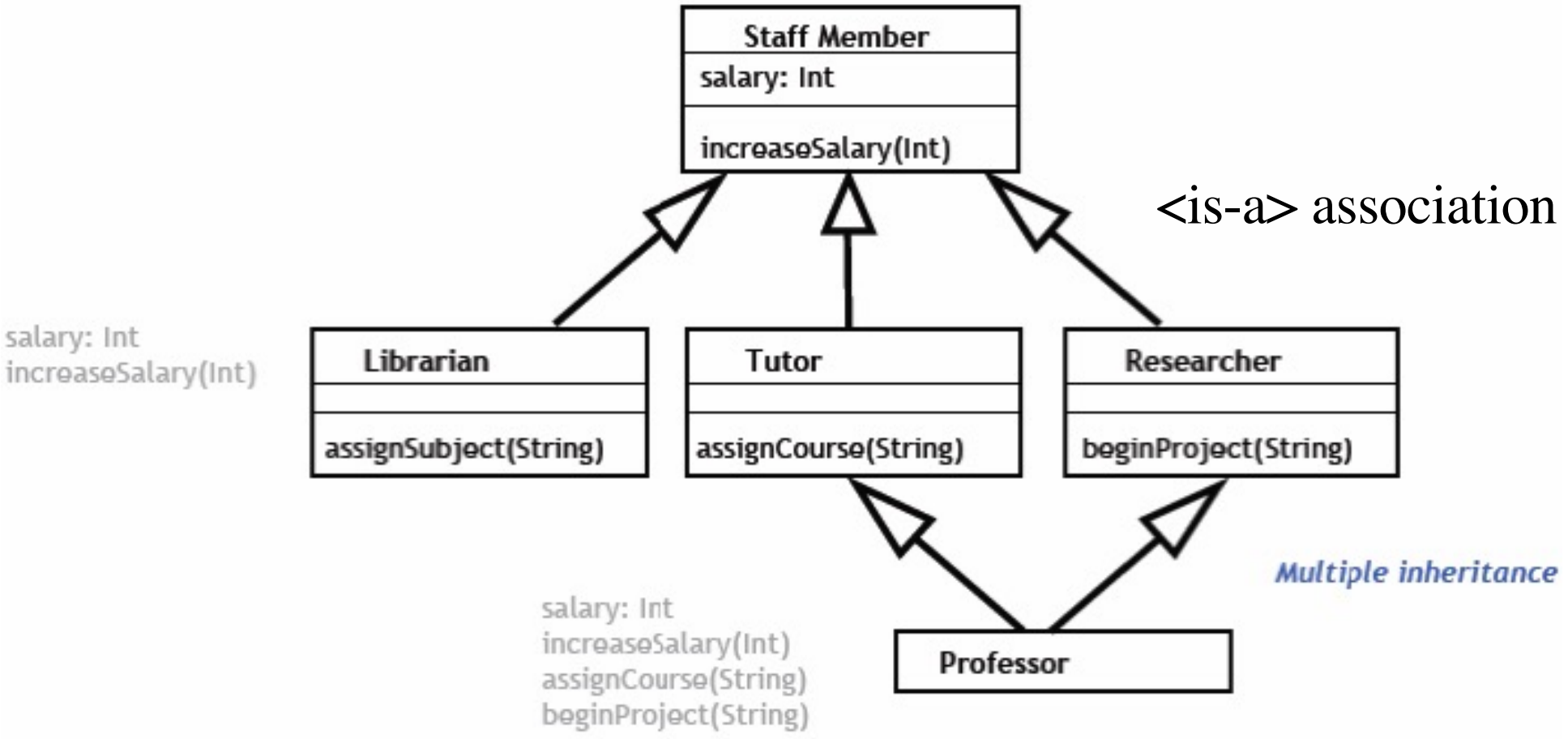
# A generalization hierarchy



# A generalization hierarchy: Details



# Another Generalisation Example



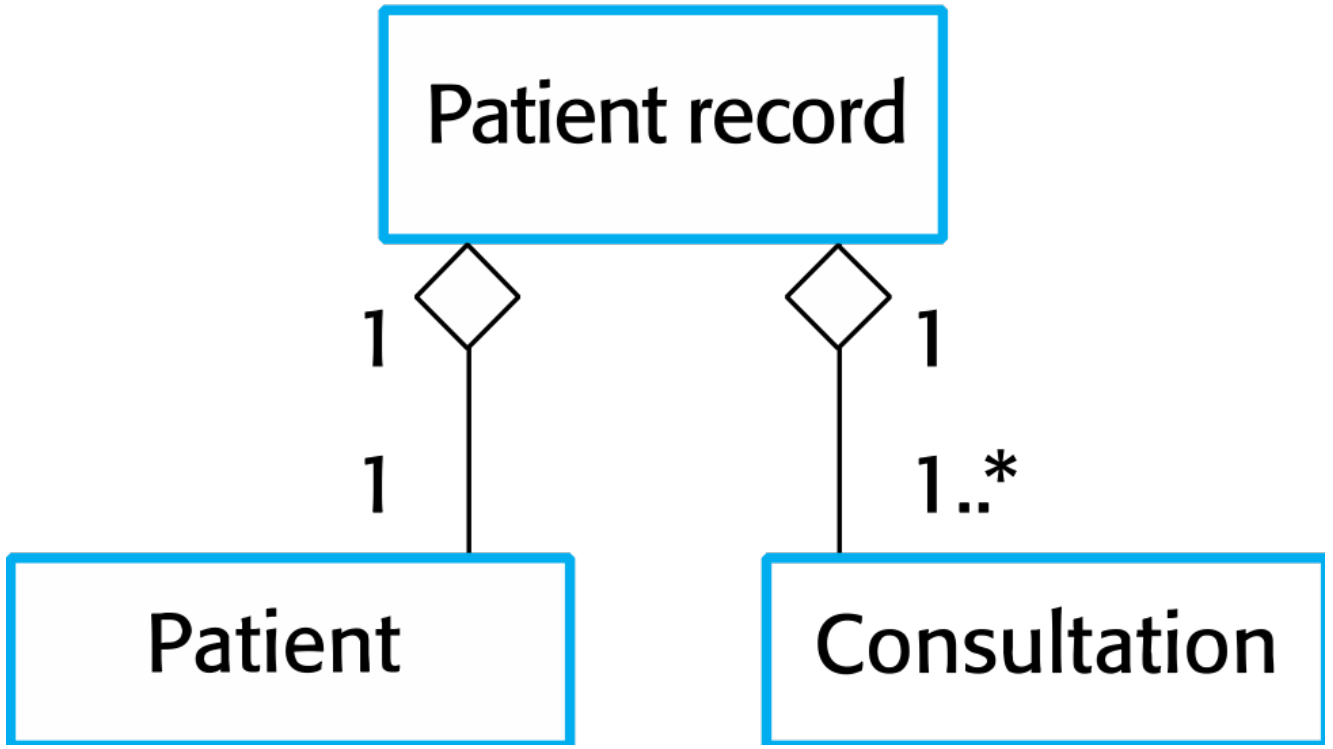
# Part/Whole Associations (Aggregation)

- **Aggregation:** Weak Ownership
  - The part objects can feature simultaneously in any number of other whole objects


<made-up-of> association
<consist-of> association

a Course is part of a Programme
In fact,
5 or more courses are part of one or more programmes

# aggregation association: Example



# Part/Whole Associations: Example

Composed of 64 squares

- **Composition:** Strong Ownership
  - The whole strongly owns its parts, so the parts cannot feature elsewhere

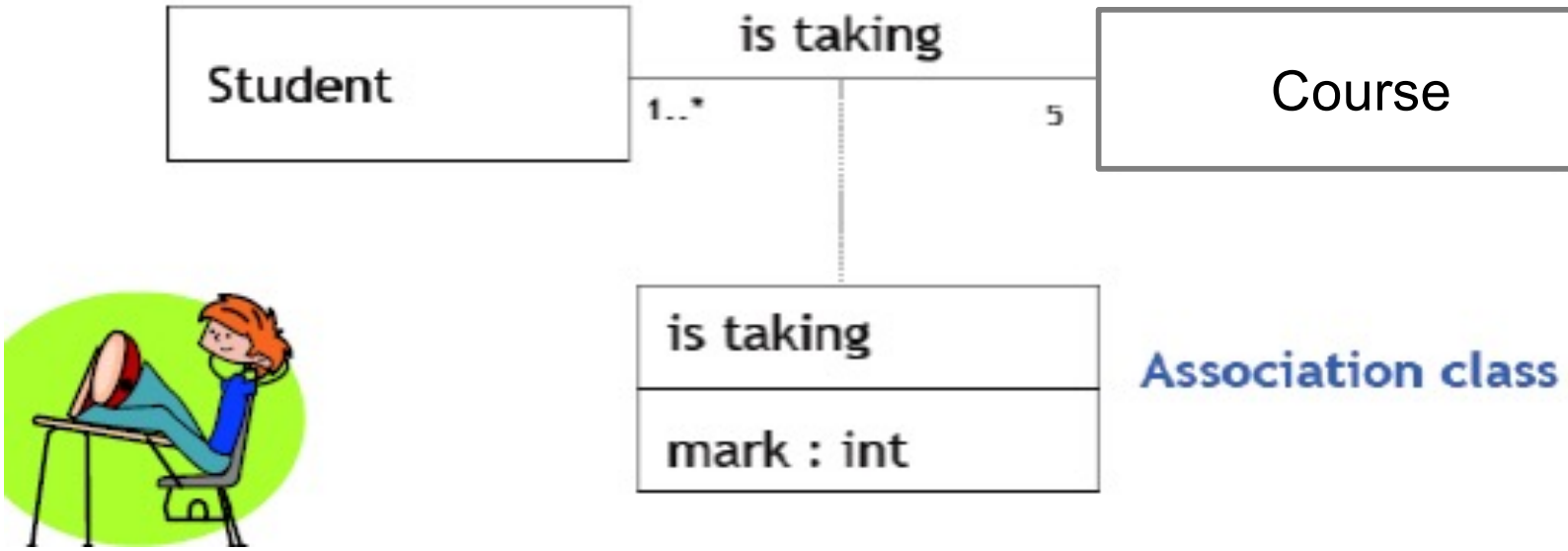

**[CheckBoard] is <made-up-of> 64 [Square]**

- NOTE: Not all 1-to-* relationships imply ownership

# Association Classes

Used to attach attributes to an association itself rather than the classes themselves

Class association line must have the same name!



# What Makes a 'Good' Analysis Class..

Its name reflects its intent

It is a crisp abstraction that models one specific element of the problem domain

It has a small but defined set of responsibilities

It has high *cohesion*

It has low *coupling* with other classes

# Exercise: Class Model

Students take courses as part of their degree. Some lecturers can teach as many courses as they wish, other can choose not to teach any course. Director of studies is one of the lecturers, who directs students' studies and help them in their course selection. Students can be graduates or non-graduates. Graduate student can graduate with an honours degree, or a non-honour degree for their graduation year. Students with honours should pass at least 6 courses, in their final graduating year in their speciality, with a mark of "very good (or first class)" and above to gain an honour degree.