

MÔ HÌNH HÓA YÊU CẦU

Class-based Methods



NỘI DUNG

- Các phương pháp mô hình hóa
- Mô hình hóa cấu trúc
 - Thẻ CRC
 - Sơ đồ lớp
 - Sơ đồ đối tượng



MÔ HÌNH HÓA YÊU CẦU

Mô hình hóa chức năng

- Use Case diagram
- Activity diagram

Mô hình hóa cấu trúc

- Class diagram
- Object diagram
- CRC Card

Mô hình hóa hành vi

- Sequence diagram
- State diagram
- Communication diagram



Mô hình hóa cấu trúc

- Mô tả cấu trúc dữ liệu sử dụng trong hệ thống phần mềm
- Rút ngắn khoảng cách giữa thế giới thực và thế giới phần mềm
- Xây dựng thuật ngữ chung cho người sử dụng và người phân tích hệ thống
- Biểu diễn sự vật, ý tưởng và khái niệm quan trọng trong hệ thống
- Còn được gọi là phương pháp mô hình hóa dựa trên class (Class-based methods)

Class
diagram

Object
diagram

CRC
Card

Lớp (class) và đối tượng (object) trong hệ thống thật

Tôi là 1 **chiếc xe hơi**.
Thông tin của tôi gồm màu, hãng sản xuất, model, giá, chủ xe, ...



Tôi có khả năng:
• Dừng
• Chạy
• Tăng tốc ...

Tôi là **1 nhân viên**.
Thông tin của tôi gồm họ tên, số CMND, nghề nghiệp, địa chỉ...



Tôi có khả năng:
• Làm việc
• Lãnh lương
• Đi nhậu...

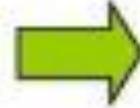
Tôi là **1 con cá**.
Thông tin của tôi gồm loài, cân nặng, đơn giá, ngày hết hạn ...



Tôi có khả năng:
• Bơi
• Được nấu cháo
• Được chiên xù ...

Lớp (class) và đối tượng (object) trong hệ thống thật

NhanVien
- Mã nhân viên
- Họ và tên
- Năm sinh
- Nghề nghiệp
- Địa chỉ
- Hệ số lương
- Chức vụ
+ Làm việc ()
+ Nghỉ phép ()
+ Lãnh lương ()



Ông NVA

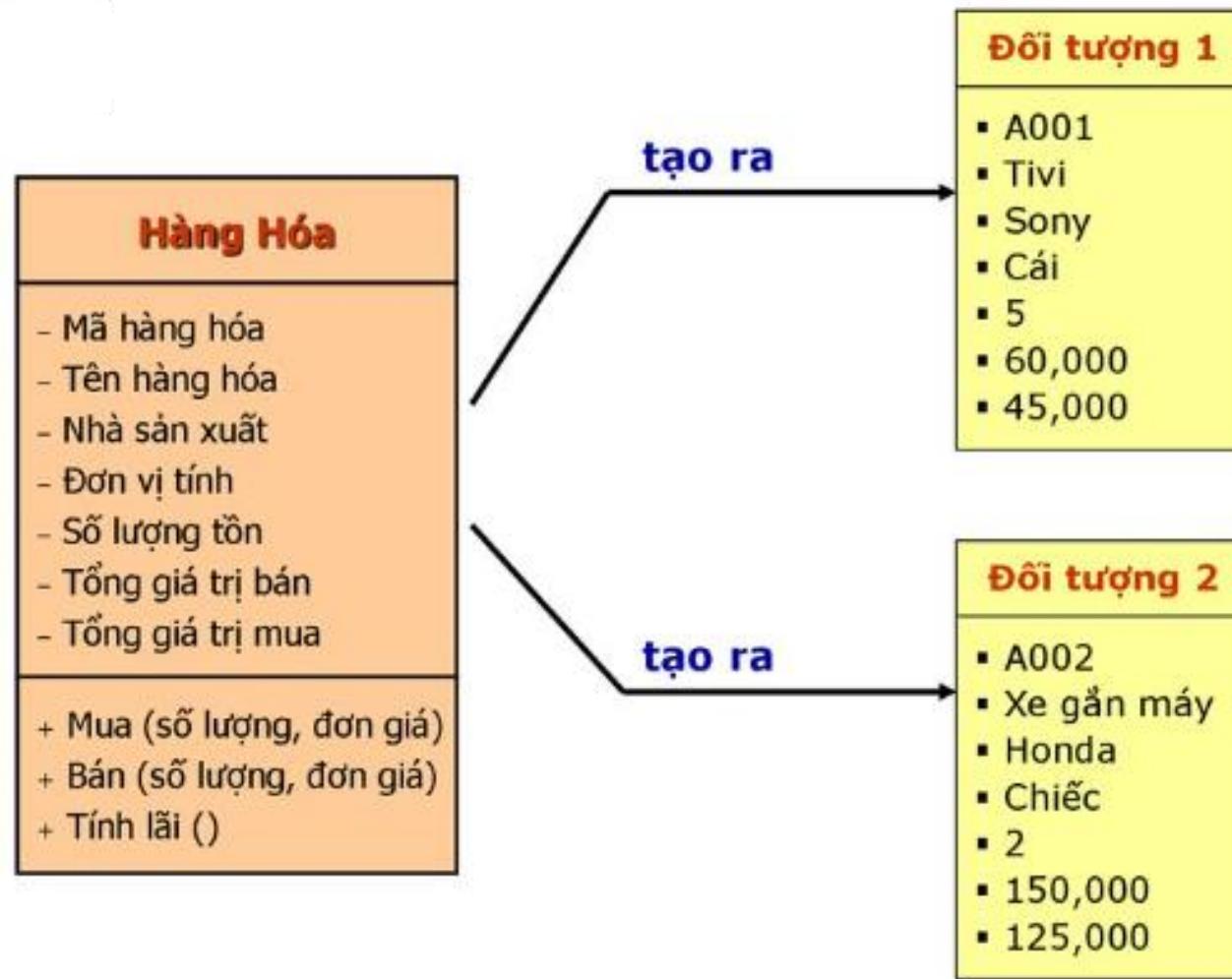


Bà TTB



Ông LVC

Lớp (class) và đối tượng (object) trong hệ thống thật



Thẻ CRC (Class Responsibility Collaboration card)

Trách nhiệm

- Biết (knowing)
 - Mô tả thông tin, đặc điểm, tính chất, ... của lớp
 - Tương tự “thuộc tính” (attribute) của lớp
- Làm (Doing)
 - Các hành vi, hành động, chức năng của lớp
 - Tương tự “hành vi” (behavior) của lớp

Sự cộng tác

- Mối quan hệ với các lớp đối tượng khác

Cấu trúc thẻ CRC

- Class Name: tên lớp đối tượng
- ID: mã số của lớp đối tượng
- Description: mô tả ngắn gọn về lớp đối tượng
- (Doing) Responsibilities: những công việc mà lớp đối tượng đảm nhận
- Collaborators: các lớp đối tượng có liên quan
- Attributes: các thuộc tính của lớp đối tượng
- Relationships: mô tả cụ thể mối quan hệ với các lớp đối tượng khác
 - Generalization
 - Aggregation (Composition)
 - Other Association

Cấu trúc thẻ CRC

- Mặt trước:

Front:		
Class Name: Patient	ID: 3	Type: Concrete, Domain
Description: An Individual that needs to receive or has received medical attention		Associated Use Cases: 2
Responsibilities		Collaborators
Make appointment		Appointment
Calculate last visit		
Change status		
Provide medical history		Medical history

Cấu trúc thẻ CRC

- Mặt sau:

Attributes:

Amount (double)

Insurance carrier (text)

Relationships:

Generalization (a-kind-of): Person

Aggregation (has-parts):

Medical History

Other Associations:

Appointment

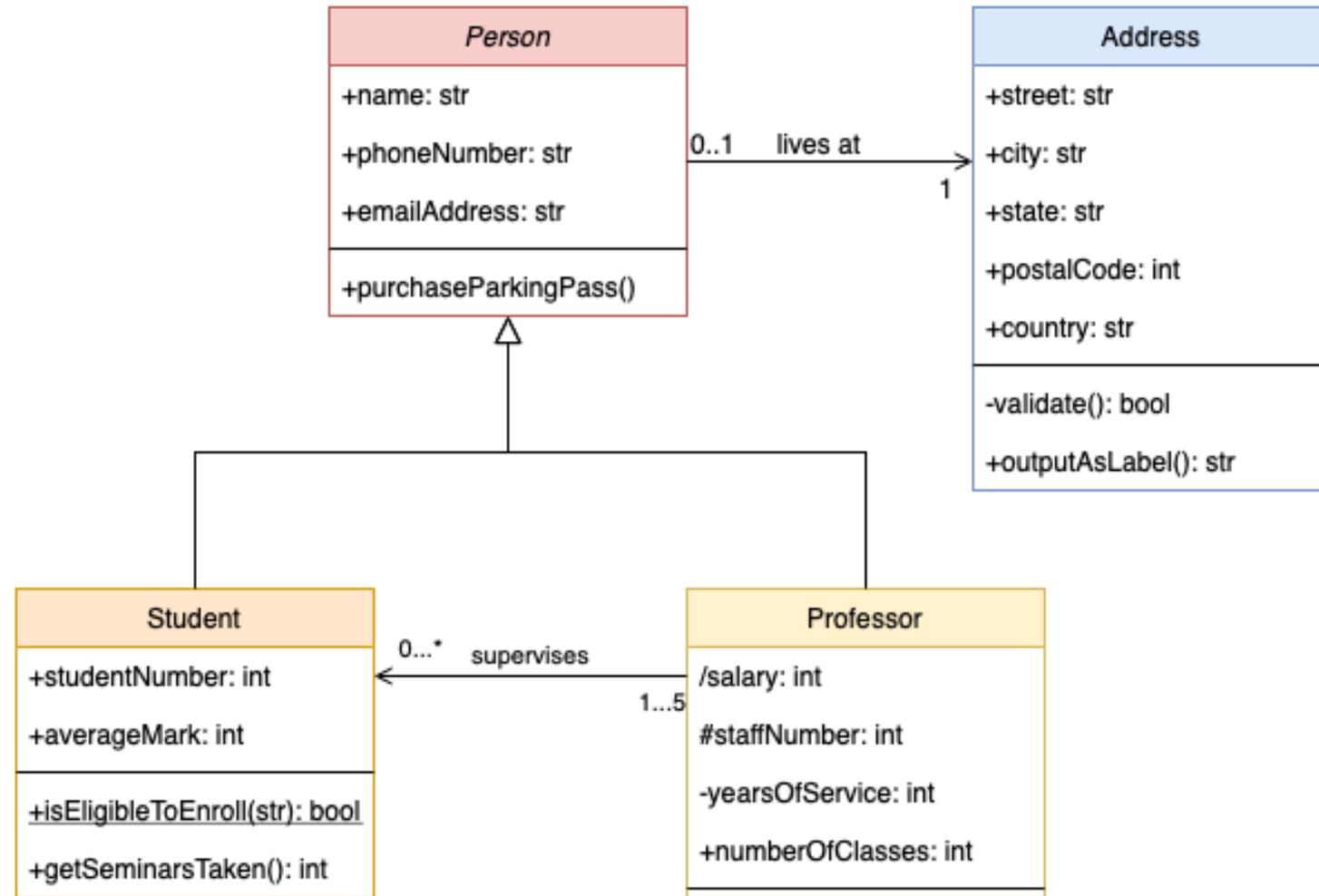


ĐỐI TƯỢNG (object) và LỚP (class)

- Đối tượng (object-instance): sự vật, sự việc cụ thể trong thực tế. Ví dụ:
 - Sinh viên Nguyễn Văn A, sinh viên Nguyễn Văn B, ...
 - Môn Toán, môn Văn, môn Tiếng Anh, ...
- Lớp (class): nhóm các đối tượng có cùng đặc điểm và hành vi. Ví dụ:
 - Sinh Viên: có đặc điểm: mã số, họ tên, lớp, điểm số, ...; hành vi: đi học, thi, xem điểm, ...
 - Môn học: đặc điểm: mã môn, tên môn, số tiết; hành vi: cập nhật, tìm kiếm, xóa...

Sơ đồ lớp (Class diagram)

-



Biểu diễn lớp (class)

- **Tên lớp (Class Name):** là tên của lớp.
- **Thuộc tính (Attribute):** mô tả tính chất của các đối tượng. Ví dụ như khách hàng có Mã khách hàng, Tên khách hàng, Địa chỉ, Ngày sinh v.v...
- **Thao tác (Operation):** chỉ các hành động mà đối tượng này có thể thực hiện trong hệ thống. Nó thể hiện hành vi của các đối tượng do lớp này tạo ra.

Class_Name
+Attribute1
+Attribute2
+Operation1()
+Operation2()

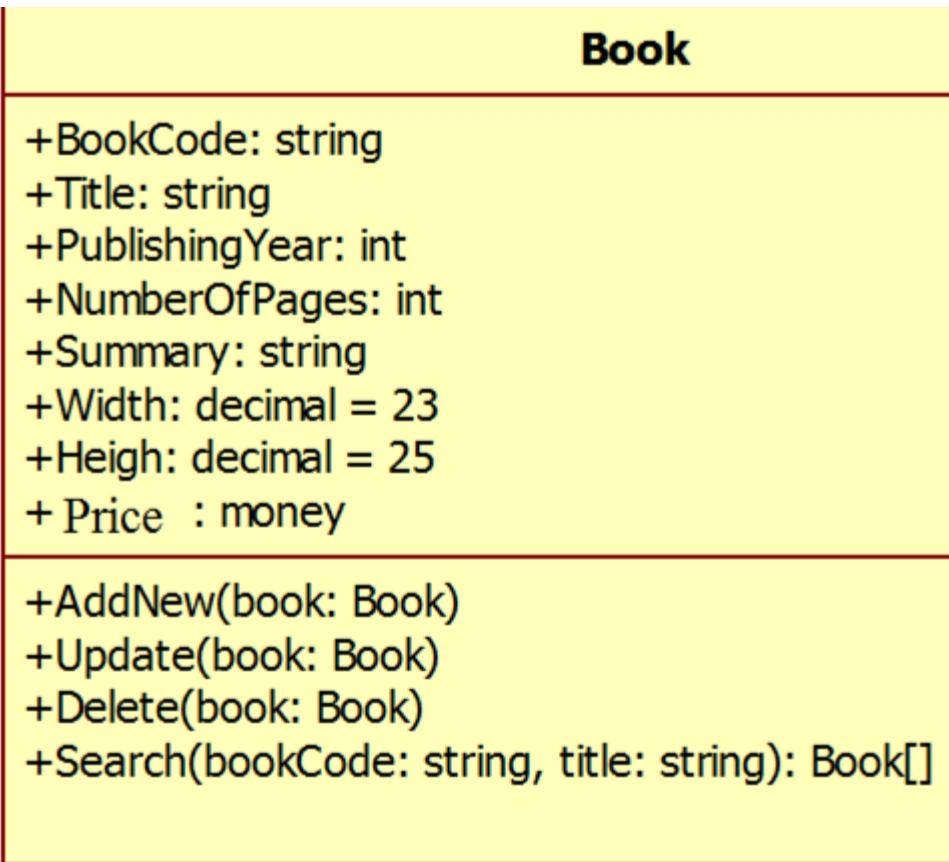
Biểu diễn lớp (class)

Thuộc tính (attribute):

- Biểu diễn thông tin/đặc điểm của lớp
- Thường có giá trị đơn
- Được mô tả bằng: tên, kiểu dữ liệu, phạm vi nhìn thấy (visibility) theo cú pháp:

Visibility: + : public - : private # : protected

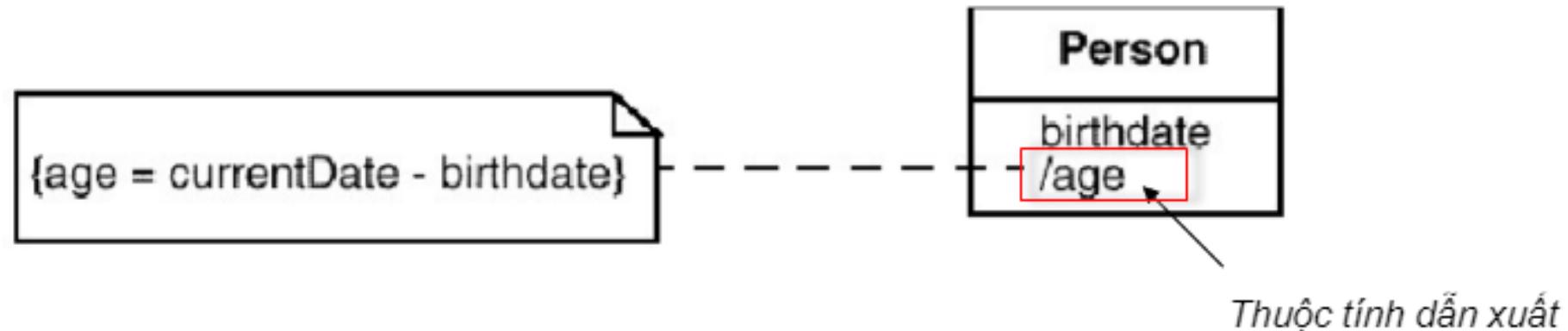
visibility name: type = defaultValue



Biểu diễn lớp (class)

Thuộc tính dẫn xuất (derived attribute):

- Có giá trị được tính toán/xử lý từ các thuộc tính khác



Biểu diễn lớp (class)

Thao tác (operation):

- Là chức năng/hành vi/phương thức của lớp
- Được mô tả bằng: tên, kiểu dữ liệu, phạm vi nhìn thấy (visibility) theo cú pháp:

Visibility: + : public - : private # : protected

visibility name (parameter-list) : return-type

Book

+BookCode: string
+Title: string
+PublishingYear: int
+NumberOfPages: int
+Summary: string
+Width: decimal = 23
+Heigh: decimal = 25
+ Price : money

+AddNew(book: Book)
+Update(book: Book)
+Delete(book: Book)
+Search(bookCode: string, title: string): Book[]

Biểu diễn lớp (class)

KhachHang

KhachHang

- hoten: String
- ngaysinh: Date
- socmnd: Number
- # diachi: String
- # tk: TaiKhoan

KhachHang

- hoten: String
- ngaysinh: Date
- socmnd: Number
- # diachi: String
- # tk: TaiKhoan

- + DangKyTK(tk)
- + ThayDoiDiaChi(dc)

- : private
- +: public
- #: protected

Thuộc tính
(attribute)

Phương thức
(method)

Quan hệ của các lớp (Relationship)

- Thể hiện mối quan hệ giữa các lớp với nhau
- Các loại quan hệ:
 - Kết hợp (Association)
 - Kết tập (Aggregation)
 - Hợp thành (Composition)
 - Tổng quát (Generalization)

Các mức độ kết hợp (association)

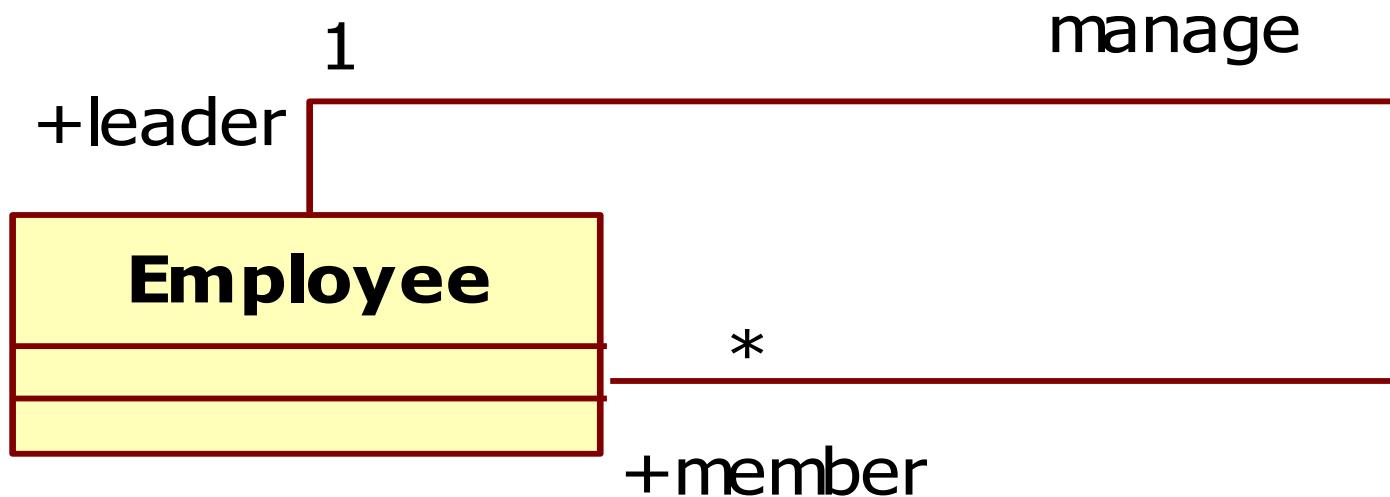
3 loại:

- Quan hệ 1 ngôi (Unary association):
- Quan hệ 2 ngôi (Binary association): hai lớp tham gia vào 1 mối quan hệ
- Quan hệ n-ngôi (n-ary association): nhiều lớp tham gia vào 1 mối quan hệ kết hợp

Các mức độ kết hợp (association)

Quan hệ 1 ngôi:

- Còn gọi là quan hệ tự thân/đệ quy
- Các đối tượng tham gia vào mối quan hệ thuộc cùng một lớp



Unary
association

Các mức độ kết hợp (association)

Quan hệ 2 ngôi:

- Các đối tượng tham gia vào mối quan hệ thuộc 2 lớp khác nhau

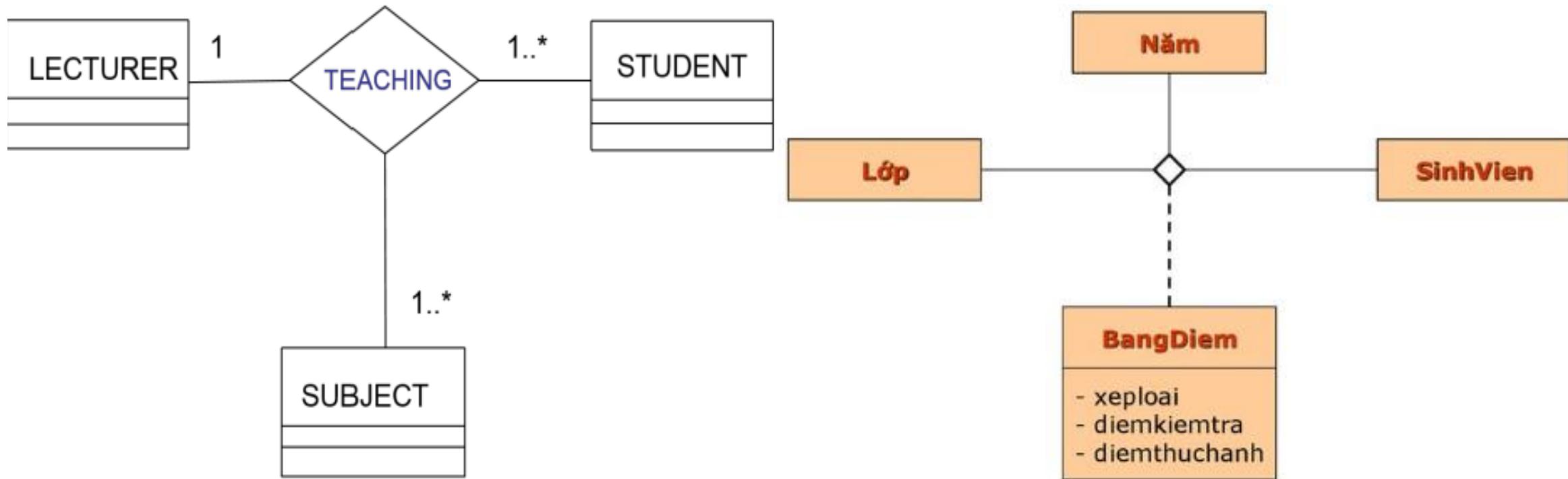


Binary association

Các mức độ kết hợp (association)

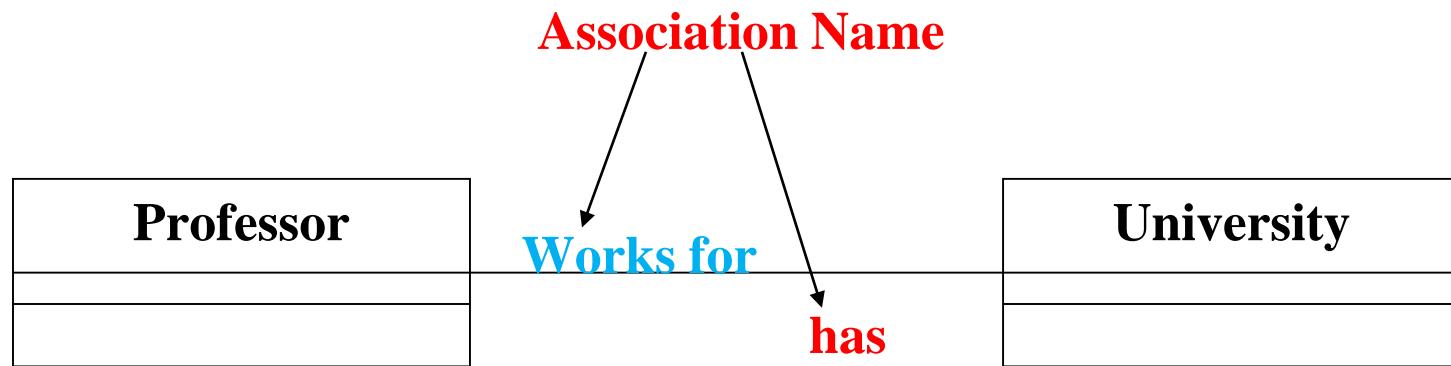
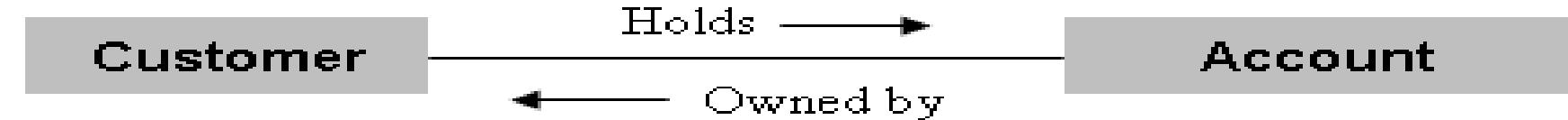
Quan hệ n-ngôi:

- Các đối tượng tham gia vào mối quan hệ thuộc n lớp khác nhau



Quan hệ của các lớp (Relationship)

- **Association** (liên quan): 2 lớp có quan hệ liên quan với nhau. Ví dụ: có, thuộc, sở hữu, ...



Quan hệ của các lớp (Relationship)

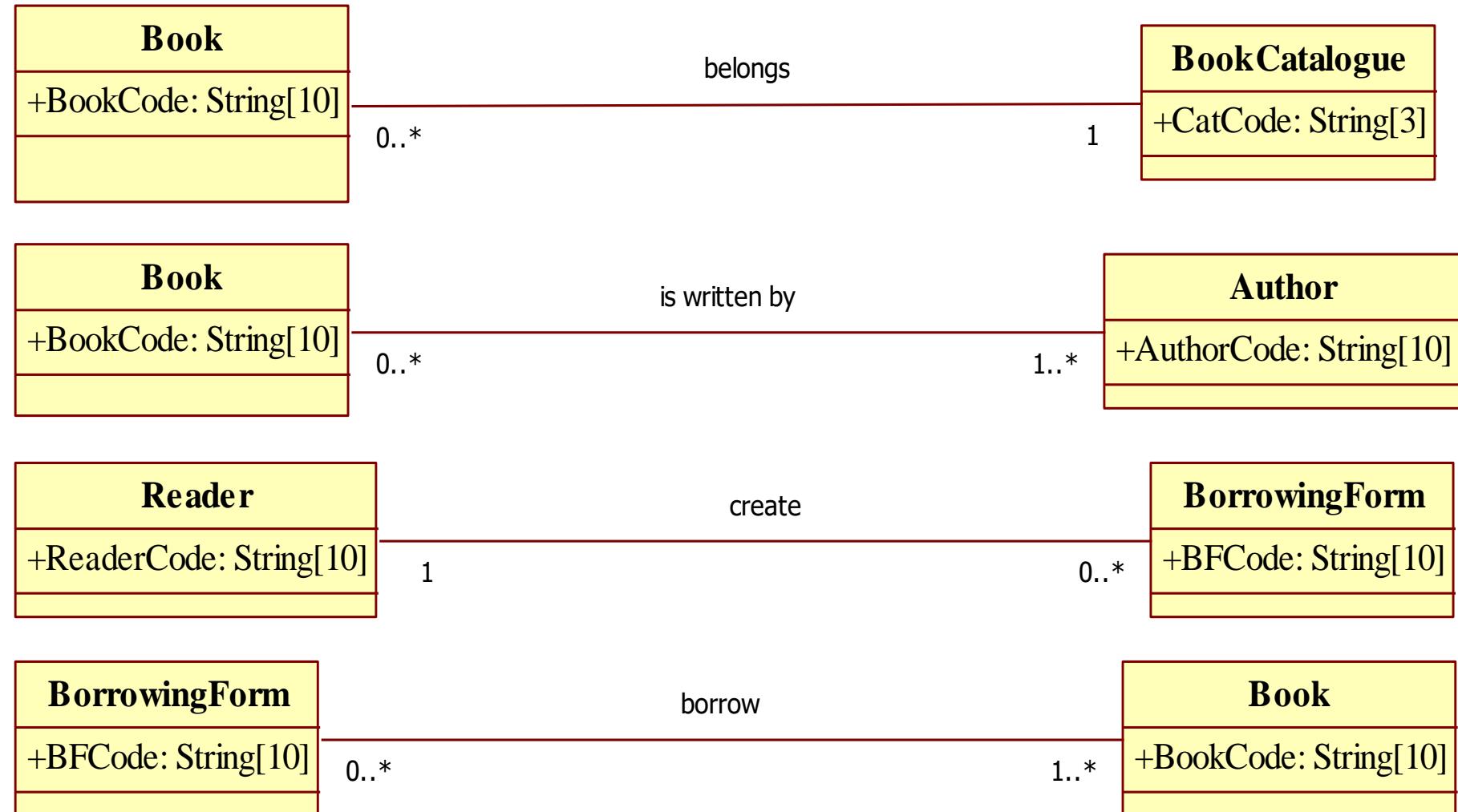
- Số lượng tham gia mỗi quan hệ: có bao nhiêu đối tượng tham gia mỗi quan hệ.

Exactly one	1
Zero or one	0, 1
Zero or more	0 ... *
One or more	1 ... *
Specified range	m ... n



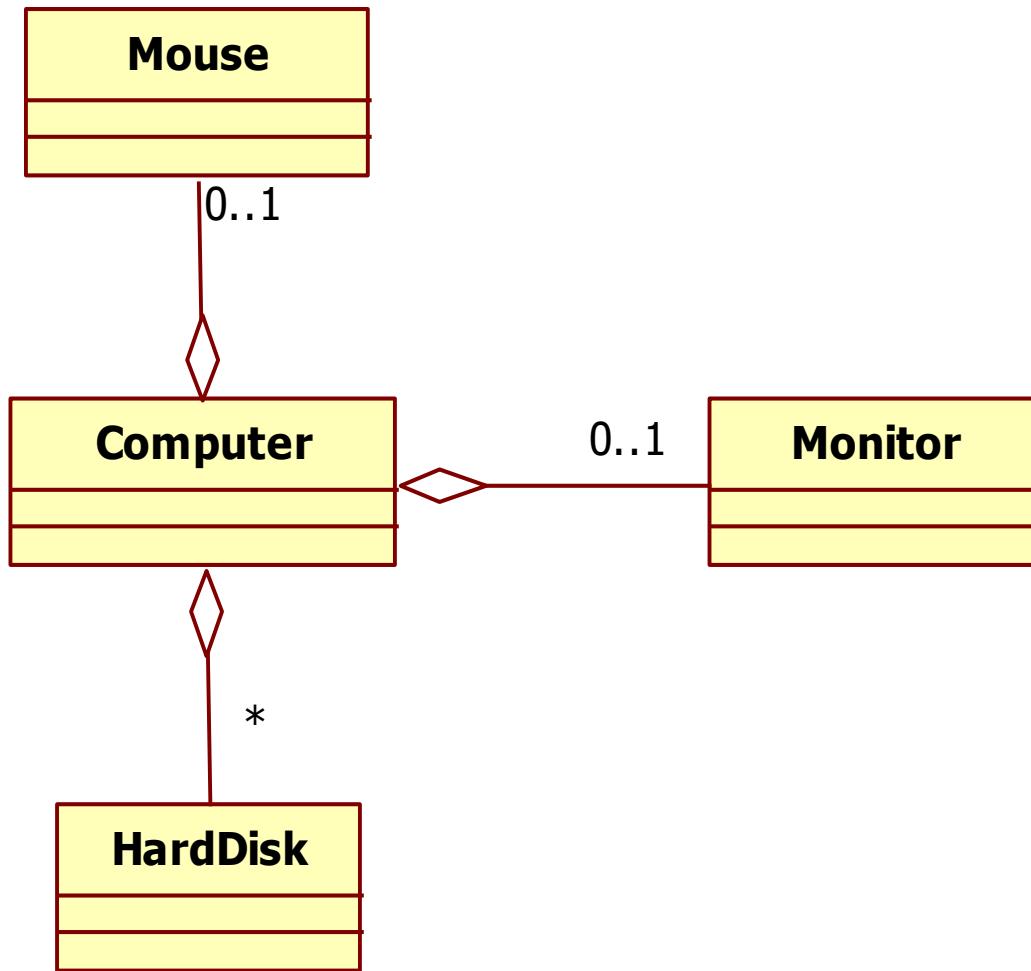
Quan hệ của các lớp (Relationship)

• Association



Quan hệ của các lớp (Relationship)

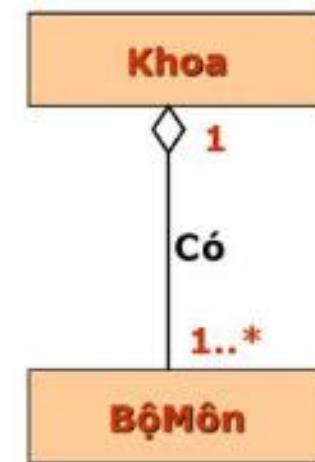
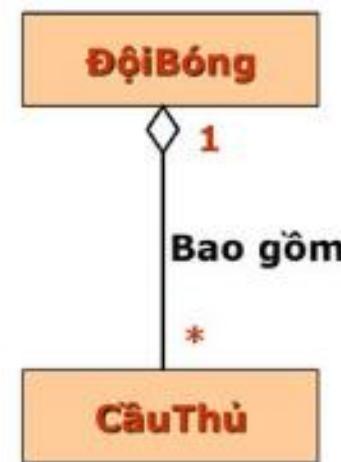
- **Kết tập (Aggregation):** mối quan hệ thành phần (whole – parts) giữa các lớp.
 - Lớp chính: có thành phần bao gồm các lớp khác.
 - Các thành phần này có thể tồn tại độc lập với lớp chính
 - Xóa lớp chính → Các lớp thành phần vẫn có thể tồn tại



Quan hệ của các lớp (Relationship)

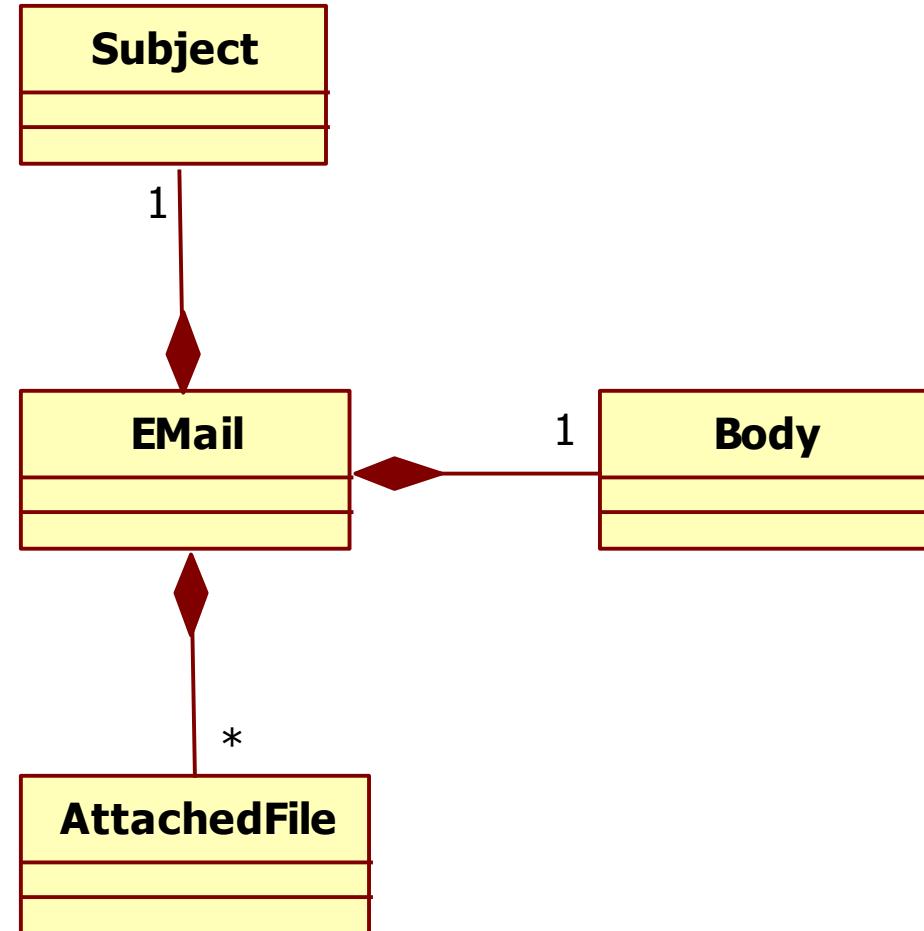
- **Kết tập (Aggregation):** mối quan hệ thành phần (whole – parts) giữa các lớp.

- Lớp chính: có thành phần bao gồm các lớp khác.
- Các thành phần này có thể tồn tại độc lập với lớp chính
- Xóa lớp chính → Các lớp thành phần vẫn có thể tồn tại



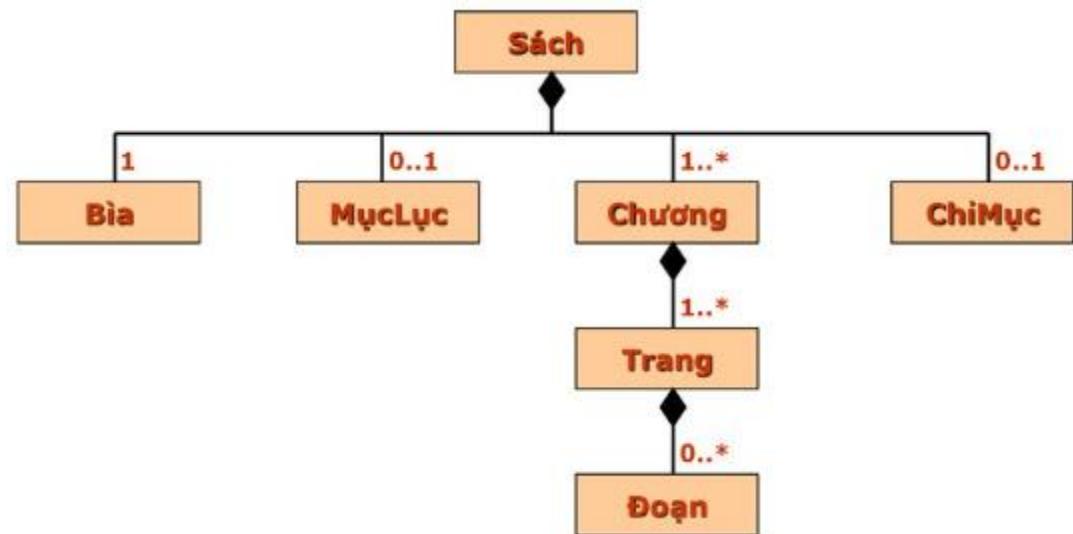
Quan hệ của các lớp (Relationship)

- **Hợp thành (Composition):** mối quan hệ thành phần giữa các lớp.
 - Lớp chính: có thành phần bao gồm các lớp khác.
 - Các thành phần này phụ thuộc hoàn toàn vào lớp chính, không thể tồn tại mà không có lớp chính, có cùng thời gian sinh và mất đi với lớp chính
 - Xóa lớp chính → Các lớp thành phần sẽ bị xóa theo
 - Composition mạnh hơn Aggregation



Quan hệ của các lớp (Relationship)

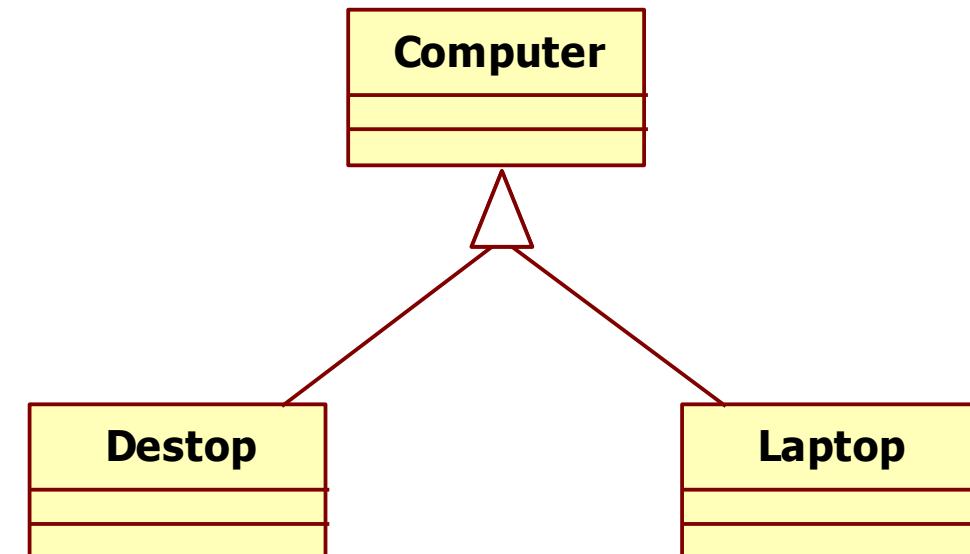
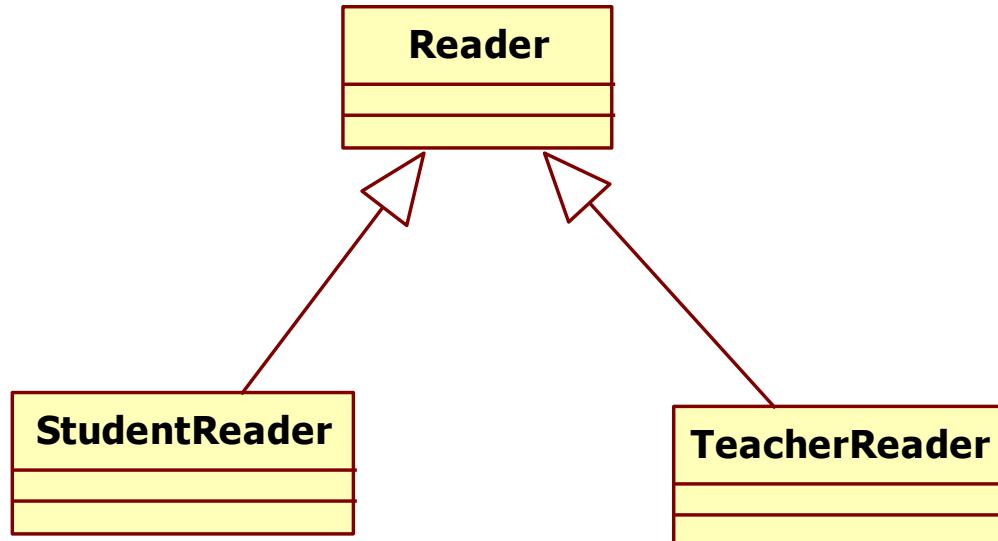
- **Hợp thành (Composition):** mối quan hệ thành phần giữa các lớp.
 - Lớp chính: có thành phần bao gồm các lớp khác.
 - Các thành phần này phụ thuộc hoàn toàn vào lớp chính, không thể tồn tại mà không có lớp chính, có cùng thời gian sinh và mất đi với lớp chính
 - Xóa lớp chính → Các lớp thành phần sẽ bị xóa theo
 - Composition mạnh hơn Aggregation



Sơ đồ lớp (Class diagram)

Tổng quát (Generalization):

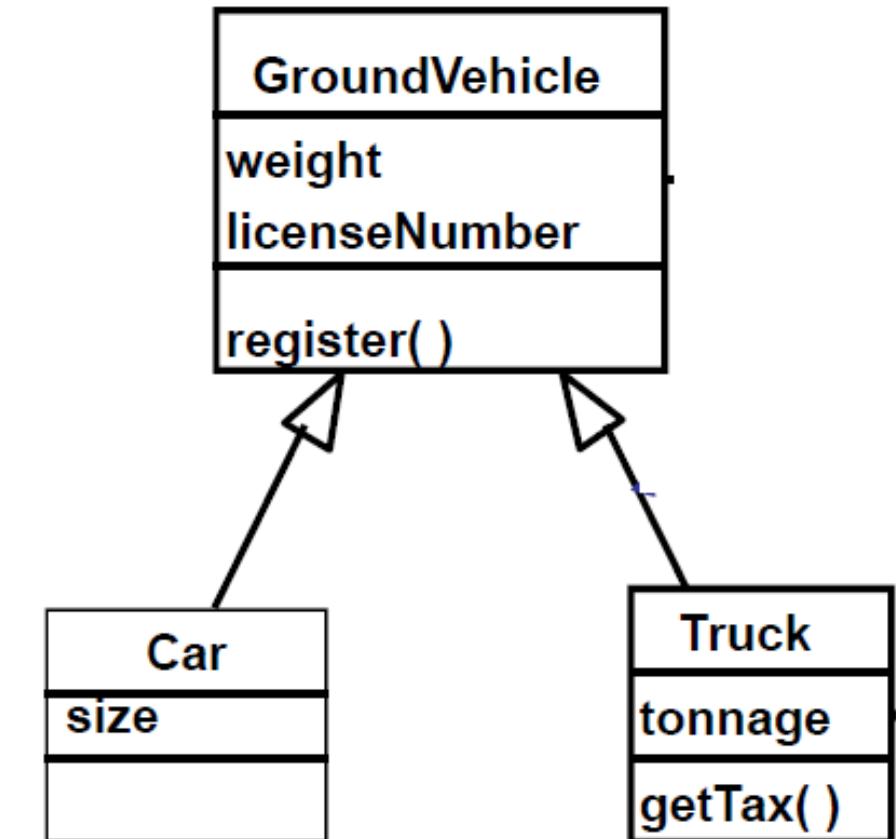
- Còn gọi là quan hệ chuyên biệt hóa/phân loại/kế thừa



Sơ đồ lớp (Class diagram)

Generalization:

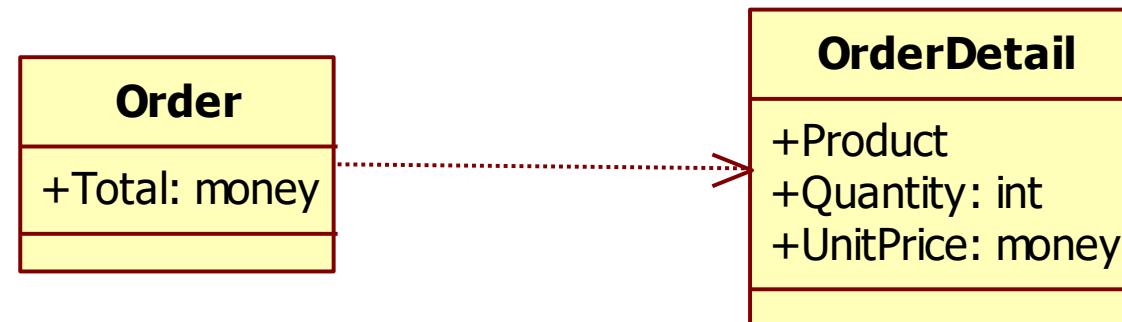
- Còn gọi là quan hệ chuyên biệt hóa/tổng quát hóa/phân loại/kế thừa



Sơ đồ lớp (Class diagram)

Dependency:

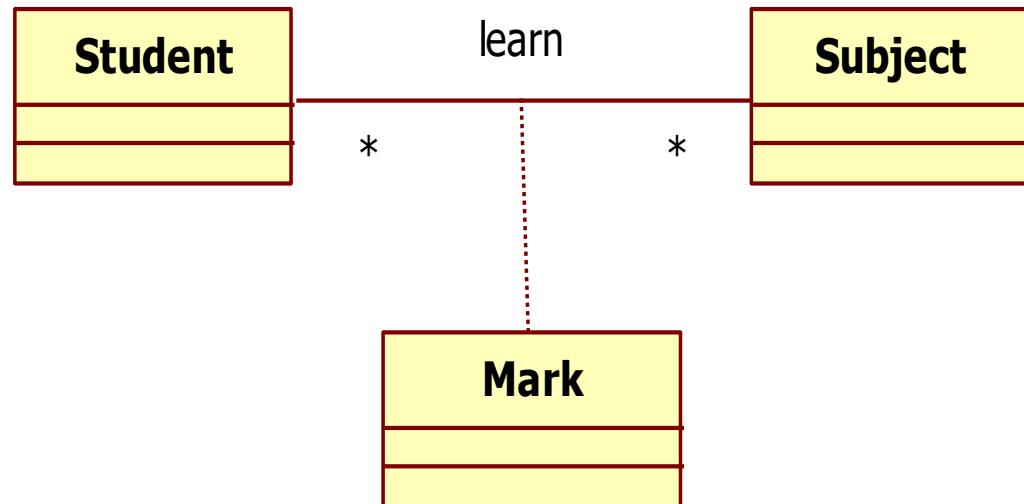
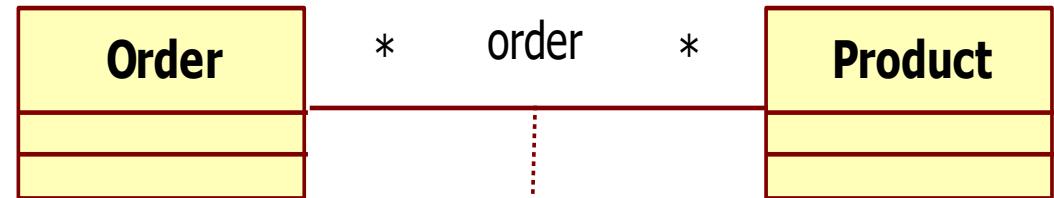
- Là mối quan hệ yếu
- Khi một lớp được sửa → các lớp phụ thuộc cũng phải được chỉnh sửa



Sơ đồ lớp (Class diagram)

Lớp kết hợp (Association Classes):

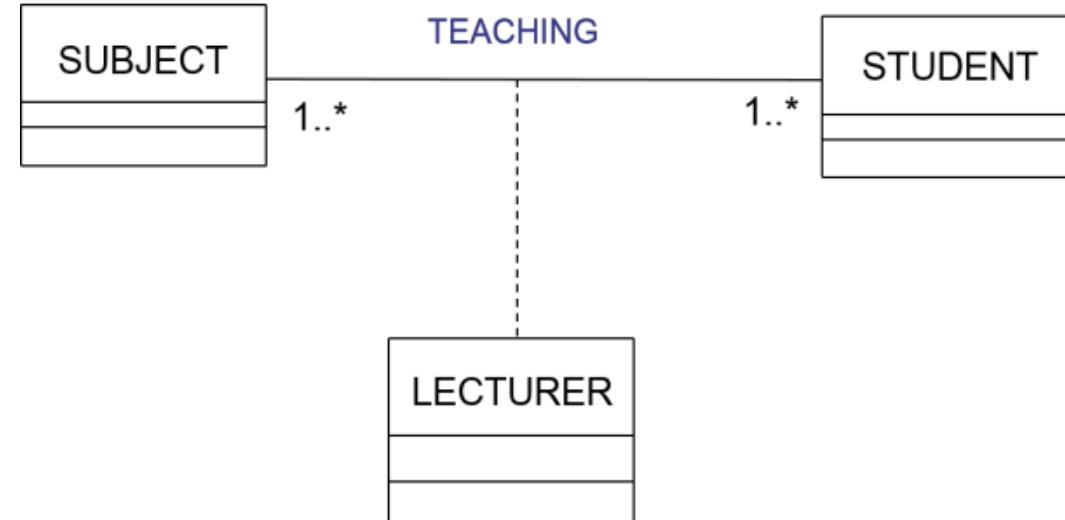
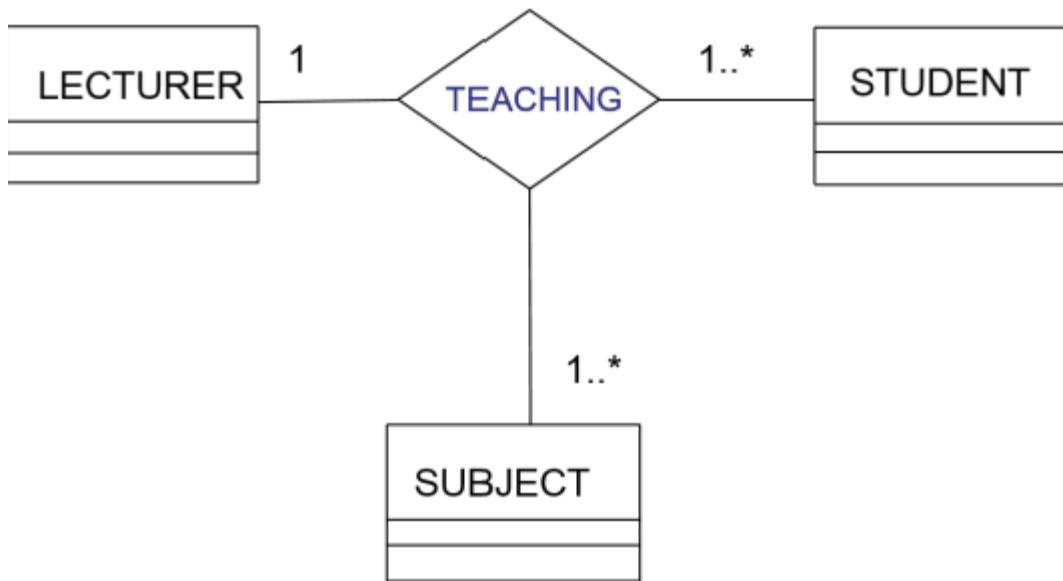
- Được sinh ra để lưu các thuộc tính của mỗi quan hệ association



Sơ đồ lớp (Class diagram)

Lớp kết hợp (Association Classes):

- Thay thế mỗi quan hệ kết hợp n-ngôi



Hai dạng lớp: phân tích và thiết kế

Analysis

Order
Placement Date
Delivery Date
Order Number
Calculate Total
Calculate Taxes

Bỏ qua các chi tiết
không cần thiết

Design

Order

- deliveryDate: Date
- orderNumber: int
- placementDate: Date
- taxes: Currency
- total: Currency

```
# calculateTaxes(Country, State): Currency  
# calculateTotal(): Currency  
getTaxEngine() {visibility=implementation}
```

Phải đầy đủ & chi tiết các thành phần



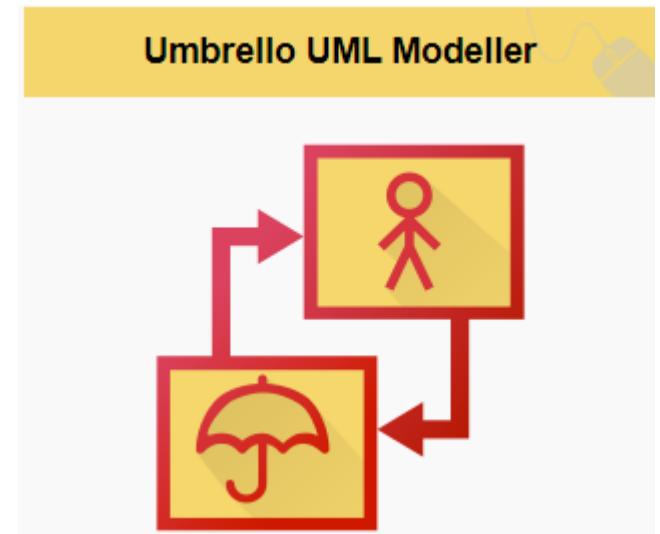
Chú ý khi vẽ sơ đồ lớp

- Không cố gắng sử dụng tất cả các ký hiệu khác nhau
- Không vẽ mô hình cho mọi thứ, tập trung vào các thông tin quan trọng

Phương pháp xây dựng sơ đồ lớp

- Xác định lớp (class): các danh từ/cụm danh từ chỉ sự vật, sự việc
- Xác định các thuộc tính (attribute) của lớp: những danh từ xác định các thông tin, đặc điểm, tính chất, ... của lớp
- Xác định các thao tác (operation) của lớp: các động từ mô tả chức năng, hành động, hành vi, công việc, ... của lớp
- Xác định các mối quan hệ giữa các lớp

Công cụ vẽ sơ đồ lớp



Sơ đồ đối tượng (Object diagram)

- Là trường hợp đặc biệt của Class diagram.
- Được dùng để:
 - nhấn mạnh mối quan hệ giữa các đối tượng của class.
 - mô tả chi tiết về class diagrams **tại một thời điểm**

Sơ đồ đối tượng (Object diagram)

- **Đối tượng (Object)**

Object Name : Class Name

Object Name : Class Name

Attributes

Methods

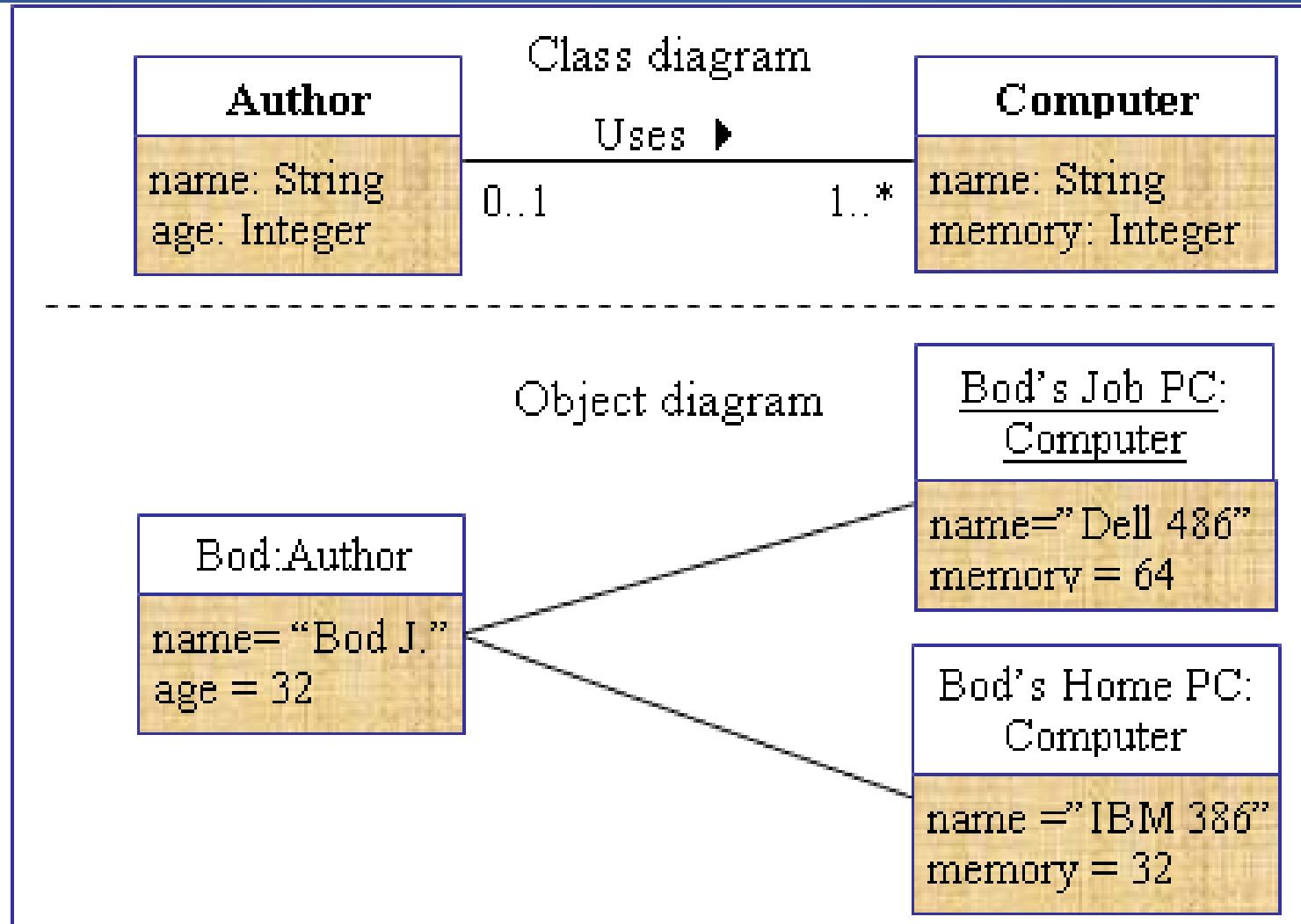
Ví dụ:

IntroUML : Book

John : Reader

Sơ đồ đối tượng (Object diagram)

.



Sơ đồ đối tượng (Object diagram)

- Không quan trọng bằng biểu đồ lớp
- Được sử dụng để ví dụ hóa một biểu đồ lớp phức tạp, chỉ ra với những thực thể cụ thể và những mối quan hệ như thế thì bức tranh toàn cảnh sẽ ra sao.
- Được sử dụng làm một thành phần của một biểu đồ cộng tác (collaboration), chỉ ra lối ứng xử động giữa một loạt các đối tượng.

BT: Sơ đồ lớp cho hệ thống Quản lý thư viện

- Tại thư viện của thành phố ABC, việc quản lý vẫn được thực hiện thủ công. Trong thư viện, thủ thư chịu trách nhiệm quản lý sách và các tài nguyên, quản lý đọc giả và việc mượn trả sách của đọc giả.
- Sách được quản lý theo các danh mục sách. Mỗi danh mục là một loại sách, như sách văn học, hóa học, ngoại ngữ, tin họcMỗi danh mục sách được đánh một mã để dễ quản lý. Mỗi quyển sách được xếp vào một loại danh mục sách. Mỗi quyển sách được đánh một mã sách dùng để phân biệt với các quyển sách khác (giả sử nếu một tác phẩm có nhiều bản giống nhau hoặc có nhiều tập thì cũng xem là có mã sách khác nhau). Ngoài ra, mỗi quyển sách còn lưu trữ các thông tin khác như : tên sách, năm xuất bản. Mỗi quyển sách cũng cần được biết do tác giả nào viết. Thông tin tác giả bao gồm mã tác giả, tên tác giả, năm sinh, quốc tịch. Một tác giả có thể viết nhiều sách. Một quyển sách có thể do nhiều tác giả viết.
- Thủ thư chịu trách nhiệm quản lý danh mục sách và các quyển sách. Khi có một quyển sách được nhập về, thủ thư sẽ xem xét quyển sách đó thuộc danh mục nào. Nếu quyển sách thuộc một danh mục mới, thủ thư có thể tạo mới danh mục đó. Việc nhập thông tin quyển sách phải được thực hiện trước khi quyển sách đó được đưa vào sử dụng.
- Đọc giả muốn thực hiện việc mượn sách phải là đọc giả của thư viện. Khi đăng ký làm đọc giả của thư viện, mỗi đọc giả được thư viện cấp cho một mã đọc giả. Ngoài ra, thư viện còn lưu trữ họ tên, ngày sinh, địa chỉ, giới tính, nghề nghiệp, ảnh cá nhân Khi không sử dụng các dịch vụ của thư viện nữa, đọc giả có thể yêu cầu hủy tư cách đọc giả của mình.

BT: Sơ đồ lốp cho hệ thống Quản lý thư viện

- Cứ mỗi lượt mượn sách, đọc giả phải đăng ký các quyền sách cần mượn vào một phiếu mượn, mỗi phiếu mượn có một số phiếu mượn khác nhau, mỗi phiếu mượn xác định các thông tin như: ngày mượn, đọc giả mượn và các quyền sách được mượn. Các các quyền sách trong cùng một phiếu mượn không nhất thiết phải trả trong một lần. Mỗi quyền sách có thể thuộc nhiều phiếu mượn khác nhau (tất nhiên là tại các thời điểm khác nhau). Mỗi phiếu mượn chỉ được mượn tối đa là 5 quyền sách và khoảng thời gian mượn tối đa là 1 tuần. Sau khoảng thời gian quy định, nếu đọc giả có nhu cầu mượn tiếp tục sẽ phải làm phiếu gia hạn cho quyền sách mình mượn. Thời gian gia hạn cũng không quá 1 tuần. Một quyền sách được mượn chỉ được gia hạn tối đa 3 lần. Đọc giả không được mượn một quyền sách ngay sau khi trả quyền sách đó. Thời gian có thể mượn lại quyền sách đó tối thiểu phải là 24 giờ.
- Khi nhận được phiếu mượn, thủ thư sẽ quyết định xem phiếu mượn có được chấp nhận hay không. Việc chấp nhận phụ thuộc vào việc quyền sách đó có thể cho mượn hay không. Khi quyền sách đọc giả cần mượn đã cho người khác mượn rồi hoặc nó đang bị hư hỏng gì đó ... thì không thể cho đọc giả đó mượn.
- Khi trả sách, thủ thư phải ghi nhận lại tình trạng của sách là bình thường hay có hư hỏng hoặc mất mát gì hay không. Nếu quyền sách bị hư hoặc mất mát, tùy theo trường hợp mà thủ thư sẽ đưa ra các hình thức xử lý khác nhau. Việc trả sách quá thời hạn quy định cũng phải chịu phạt. Mức phạt được áp dụng tùy theo thời gian quá hạn.
- Cuối mỗi tháng, thủ thư phải lập và gửi báo cáo về tình trạng cho mượn sách lên cấp quản lý của thư viện. Quản lý của thư viện cũng có thể yêu cầu thủ thư lập các bảng thống kê tình hình quản lý sách ở thư viện như: thống kê các quyền sách theo danh mục, thống kê các quyền sách được mượn nhiều nhất hoặc ít nhất, thống kê các quyền sách hư hỏng không sử dụng được ...

Câu hỏi

- Nêu khái niệm và cho ví dụ về đối tượng và lớp?
- Mục tiêu của sơ đồ lớp?
- Nêu các thành phần và ký hiệu trong sơ đồ lớp?
- Nêu các kiểu quan hệ giữa các lớp? Cho ví dụ?
- Sơ đồ đối tượng là gì? Mục tiêu của sơ đồ đối tượng?
- Sự khác nhau giữa sơ đồ đối tượng và sơ đồ lớp?