

Xác định mối quan hệ giữa các lớp thực thể

- ❖ Kế thừa (Inheritance): Kế thừa là mối quan hệ giữa một lớp có các đặc trưng mang tính khái quát cao hơn và một lớp có các tính chất đặc biệt hơn
- ❖ Liên kết (Association): Một liên kết (association) là một sự nối kết giữa các lớp, cũng có nghĩa là sự nối kết giữa các đối tượng của các lớp này
- ❖ Tụ hợp (Aggregation): Là dạng quan hệ mô tả một lớp A là một bộ phận của lớp B và lớp A có thể tồn tại độc lập
- ❖ Hợp thành (Composition): Lớp A có quan hệ hợp thành với lớp B nếu lớp A là một phần của lớp B và sự tồn tại của đối tượng lớp B điều khiển sự tồn tại của đối tượng lớp A
- ❖ Người phát triển phải tìm ra các quan hệ giữa các lớp và xác định cụ thể quan hệ đó thuộc dạng nào

Xác định mối quan hệ giữa các lớp thực thể

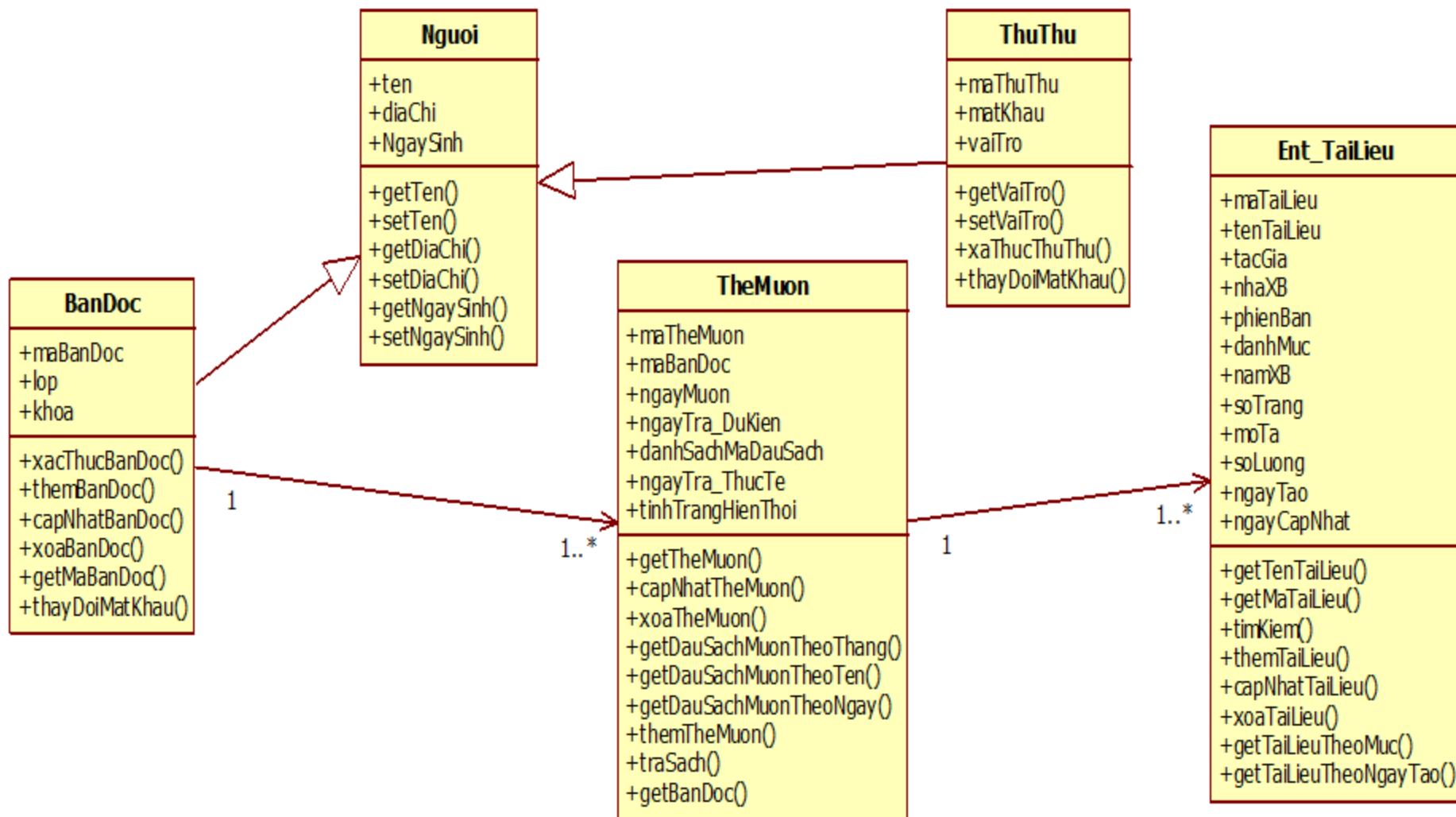
- ❖ Các bước thực hiện
 - ❖ Xác định cụ thể dạng của quan hệ giữa các lớp
 - ❖ Xác định số lượng trong mỗi mối quan hệ
- ❖ Các kiểu biểu diễn số lượng trong biểu đồ lớp

Ký hiệu	Ý nghĩa
0..1	Không có hoặc có 1 thể hiện. Tương tự n..m sẽ thể hiện có từ n đến m thể hiện
0..* hoặc *	Không giới hạn số thể hiện của lớp (gồm cả giá trị 0)
1	Có chính xác 1 thể hiện
1..n	Có ít nhất một thể hiện

Vẽ biểu đồ lớp thực thể

- ❖ Ví dụ: Biểu đồ lớp thực thể của hệ quản lý thư viện
 - ❖ Các lớp Bạn đọc và Thủ thư được kế thừa từ một lớp chung tên là Người
 - ❖ Tại một thời điểm, một bạn đọc có tương ứng một Thẻ mượn, có nhiều thẻ trong những lần mượn khác nhau
 - ❖ Một thẻ mượn có thể cho mượn cùng một lúc một hoặc nhiều cuốn sách

Vẽ biểu đồ lớp thực thể



Biểu đồ trạng thái

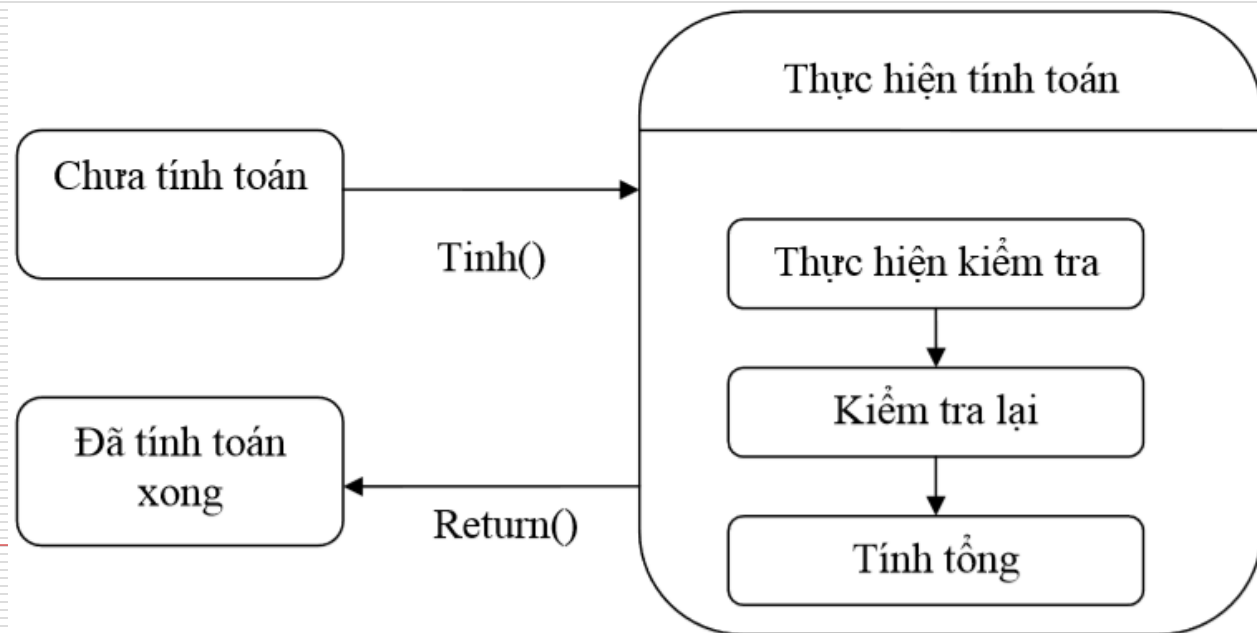
- ❖ Khái niệm biểu đồ trạng thái
- ❖ Tập ký hiệu UML cho biểu đồ trạng thái
- ❖ Ví dụ biểu đồ trạng thái
- ❖ Xây dựng biểu đồ trạng thái

Khái niệm biểu đồ trạng thái

- ❖ Biểu đồ trạng thái được sử dụng để biểu diễn các trạng thái và sự chuyển tiếp giữa các trạng thái của các đối tượng trong một lớp xác định
- ❖ Biểu đồ trạng thái được biểu diễn dưới dạng máy trạng thái hữu hạn với các trạng thái và sự chuyển tiếp giữa các trạng thái đó

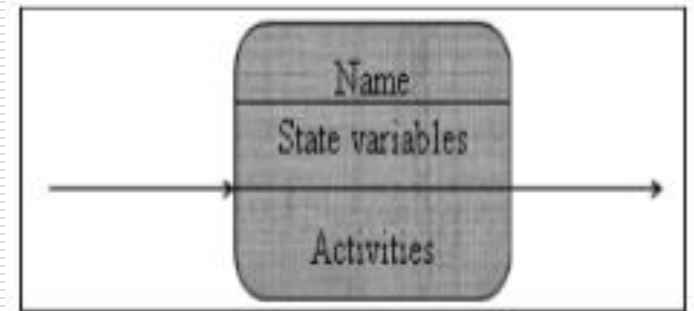
Tập ký hiệu UML cho biểu đồ trạng thái

- ❖ Các thành phần trong biểu đồ trạng thái;
- ❖ Trạng thái (state): Bên trong các trạng thái có thể miêu tả các biến trạng thái hoặc các hành động (action) tương ứng với trạng thái đó
- ❖ Trạng thái con (substate): là một trạng thái chứa bên trong một trạng thái khác, trạng thái có nhiều trạng thái con gọi là trạng thái tổ hợp



Tập ký hiệu UML cho biểu đồ trạng thái

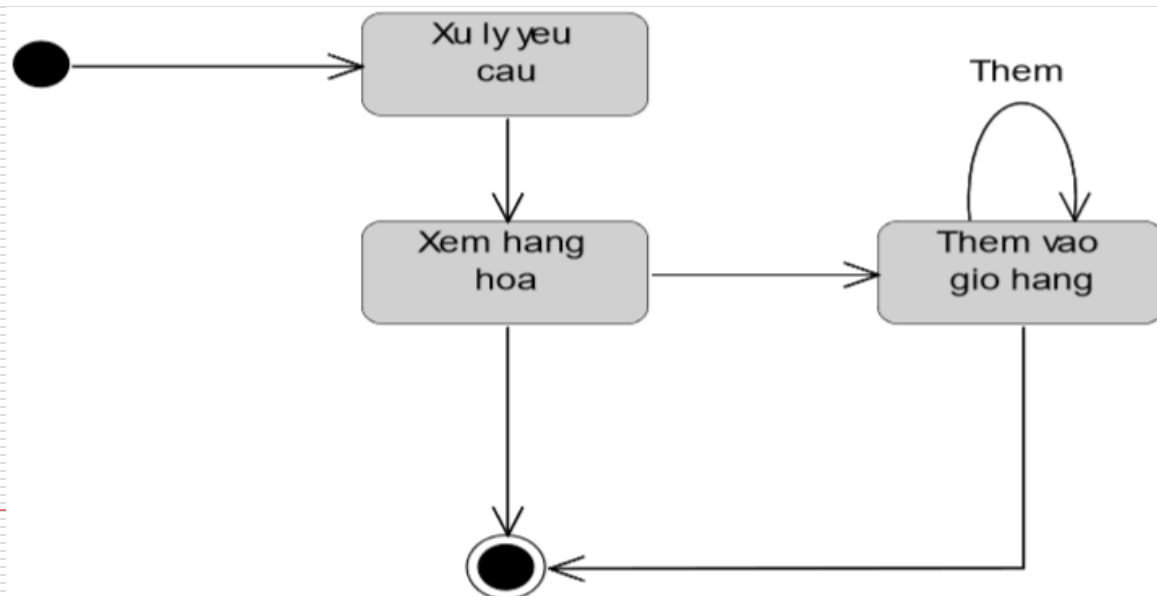
- ❖ Trạng thái khởi đầu (initial state): trạng thái đầu tiên khi kích hoạt đối tượng, hình tròn đặc
- ❖ Trạng thái kết thúc (final state): kết thúc vòng đời đối tượng, Hai hình tròn lồng nhau
- ❖ Các chuyển tiếp (transition): biểu diễn các chuyển đổi giữa các trạng thái
- ❖ Sự kiện (event): sự kiện tác động gây ra sự chuyển đổi trạng thái. Mỗi sự kiện được đi kèm với các điều kiện (guard) và các hành động (action)



Tên chuyển tiếp →

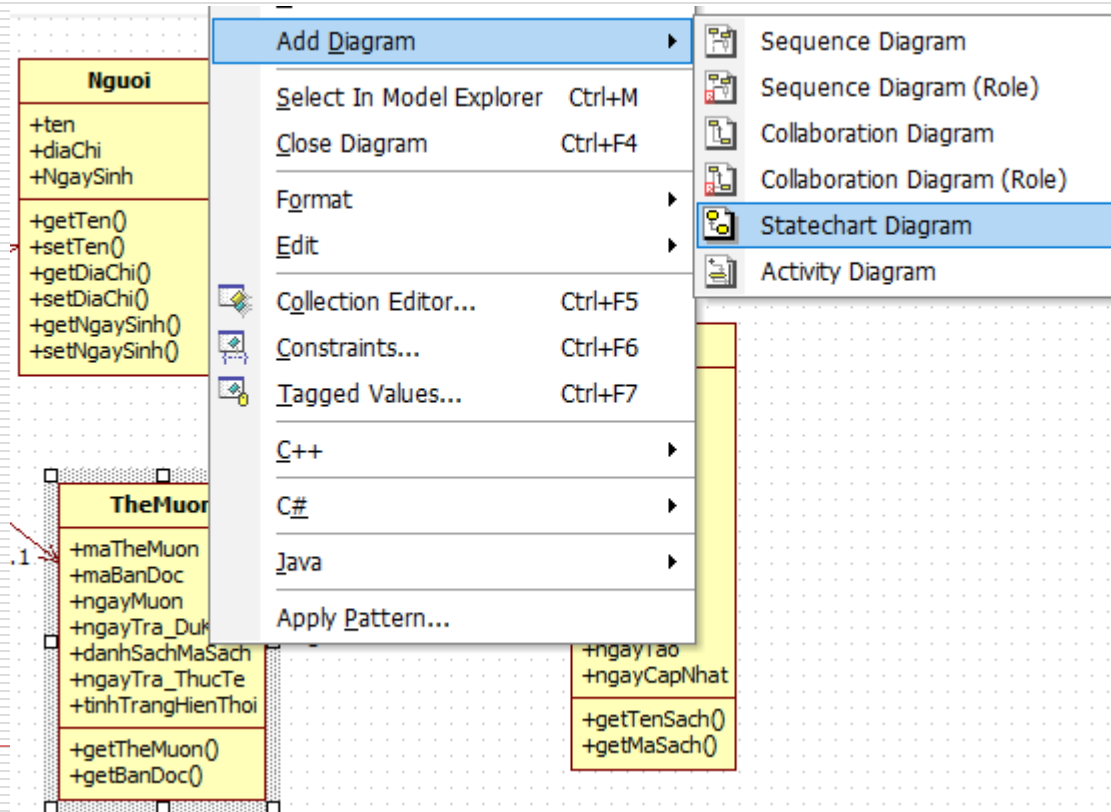
Ví dụ biểu đồ trạng thái

- ❖ Ví dụ biểu đồ trạng thái: xét hệ thống mua hàng đơn giản
 - ❖ Một khách hàng gửi yêu cầu cần tìm mua một số hàng đến hệ thống
 - ❖ Đối tượng hệ thống được tạo ra và chuyển sang trạng thái xử lý yêu cầu
 - ❖ Tùy thuộc vào yêu cầu, đối tượng này sẽ chuyển sang trạng thái xem hàng hóa
 - ❖ Với mỗi hàng hóa được lựa chọn, đối tượng sẽ chuyển sang trạng thái thêm vào giỏ hàng cho đến khi kết thúc quá trình lựa chọn



Xây dựng biểu đồ trạng thái

- ❖ Khởi tạo biểu đồ trạng thái đối tượng trên StarUML
 - ❖ Nháy chuột phải vào model
 - ❖ Chọn Add Diagram, chọn Statechart Diagram
- ❖ Các bước xây dựng biểu đồ trạng thái
 - ❖ Bước 1: Nhận biết các trạng thái và sự kiện
 - ❖ Bước 2: Xây dựng biểu đồ
 - ❖ Bước 3: Hiệu chỉnh biểu đồ



B1: Nhận biết các trạng thái và sự kiện

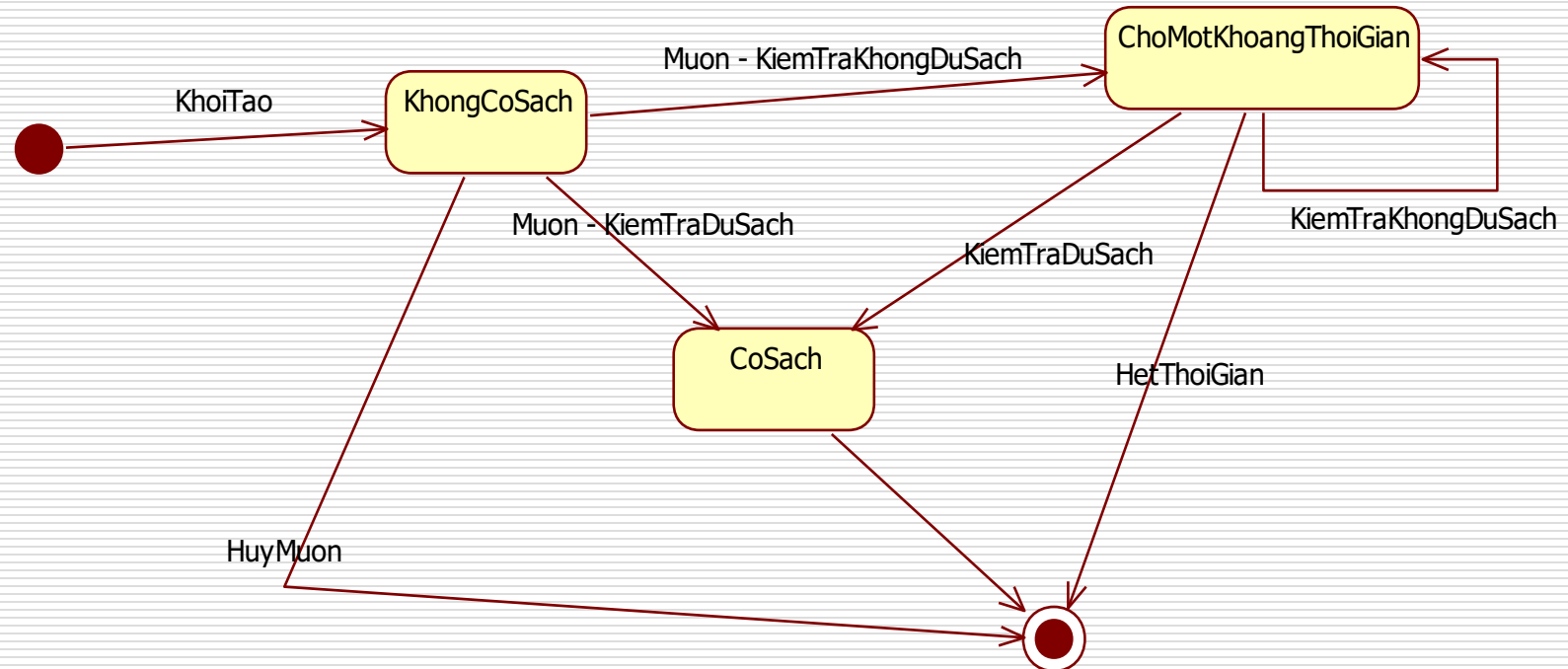
- ❖ Để phát hiện sự kiện và trạng thái của một đối tượng được thực hiện bằng việc trả lời các câu hỏi sau
 - ❖ Một đối tượng có thể có những trạng thái nào? Gồm tất cả những trạng thái mà một đối tượng có thể có trong vòng đời của nó
 - ❖ Những sự kiện nào có thể xảy ra?
 - ❖ Trạng thái mới sẽ là gì?
 - ❖ Có những thủ tục ảnh hưởng đến trạng thái của một đối tượng?
 - ❖ Cái gì khiến cho một đối tượng được tạo ra? => Xác định chuyển tiếp đầu tiên trong biểu đồ trạng thái
 - ❖ Cái gì khiến cho một đối tượng bị hủy? => Tìm ra các chuyển tiếp cuối cùng đến trạng thái kết thúc

B2: Xây dựng biểu đồ trạng thái

- ❖ Người phát triển tiếp tục sắp xếp các trạng thái và sự kiện tìm được vào trong một biểu đồ
- ❖ Xuất phát từ trạng thái khởi đầu, người thiết kế sẽ xác định các trạng thái tiếp theo và biểu diễn các chuyển tiếp giữa các trạng thái đó
- ❖ Gắn với mỗi chuyển tiếp là một sự kiện
- ❖ Các sự kiện sẽ được biểu diễn theo cấu trúc chung như sau: Sự kiện [điều kiện] hoạt động
- ❖ Trạng thái có thể được chuyển tiếp từ một hoặc nhiều trạng thái trung gian
- ❖ Dựa trên quá trình chuyển tiếp trạng thái, người phát triển sẽ phải xác định chuyển tiếp nào có thể dẫn tới trạng thái kết thúc trong vòng đời đối tượng

B3: Hiệu chỉnh biểu đồ trạng thái

- ❖ Người phát triển tiến hành xem xét lại toàn bộ các biểu đồ trạng thái cho từng lớp và sửa đổi lại biểu đồ trạng thái nếu cần thiết
- ❖ Các biểu đồ trạng thái sẽ được sử dụng để xác định đầy đủ các thuộc tính cho biểu đồ lớp
- ❖ Ví dụ: Biểu đồ trạng thái đối tượng thẻ mượn

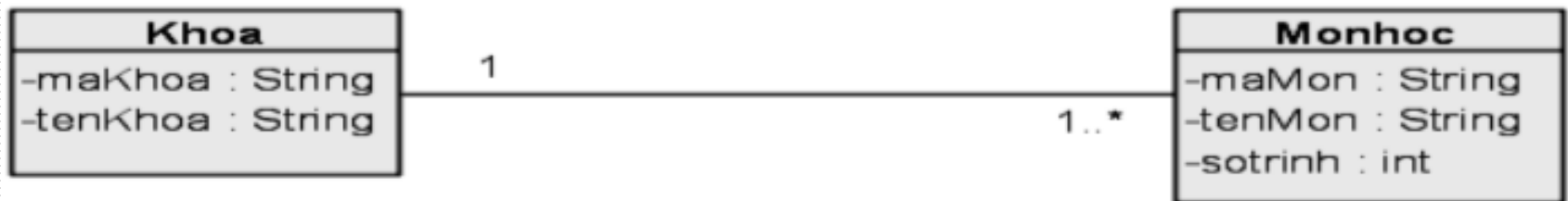


Thiết kế cơ sở dữ liệu

- ❖ Nhiệm vụ:
 - ❖ Ánh xạ mô hình đối tượng thành mô hình quan hệ
 - ❖ Phân phối hành vi của lớp vào csdl
- ❖ Ánh xạ các lớp thực thể:
 - ❖ Tạo bảng có cùng tên như lớp thực thể, mỗi dòng biểu diễn một đối tượng duy nhất của miền nghiệp vụ
 - ❖ Nếu thuộc tính có kiểu đơn giản thì tên trường của bảng là tên thuộc tính thực thể
 - ❖ Nếu thuộc tính có kiểu đối tượng thì xem xét dưới dạng ánh xạ liên kết
- ❖ Ánh xạ liên kết 1-1
 - ❖ Cách 1: Thêm một khóa ngoại vào một bảng thực thể, tham chiếu đến khóa chính của bảng kia
 - ❖ Cách 2: Gộp 2 bảng thành 1 bảng

Ánh xạ thành mô hình quan hệ

- ❖ Ánh xạ liên kết 1-n
 - ❖ Thêm một khóa ngoại vào bảng “nhiều” tham chiếu đến khóa chính của bảng 1



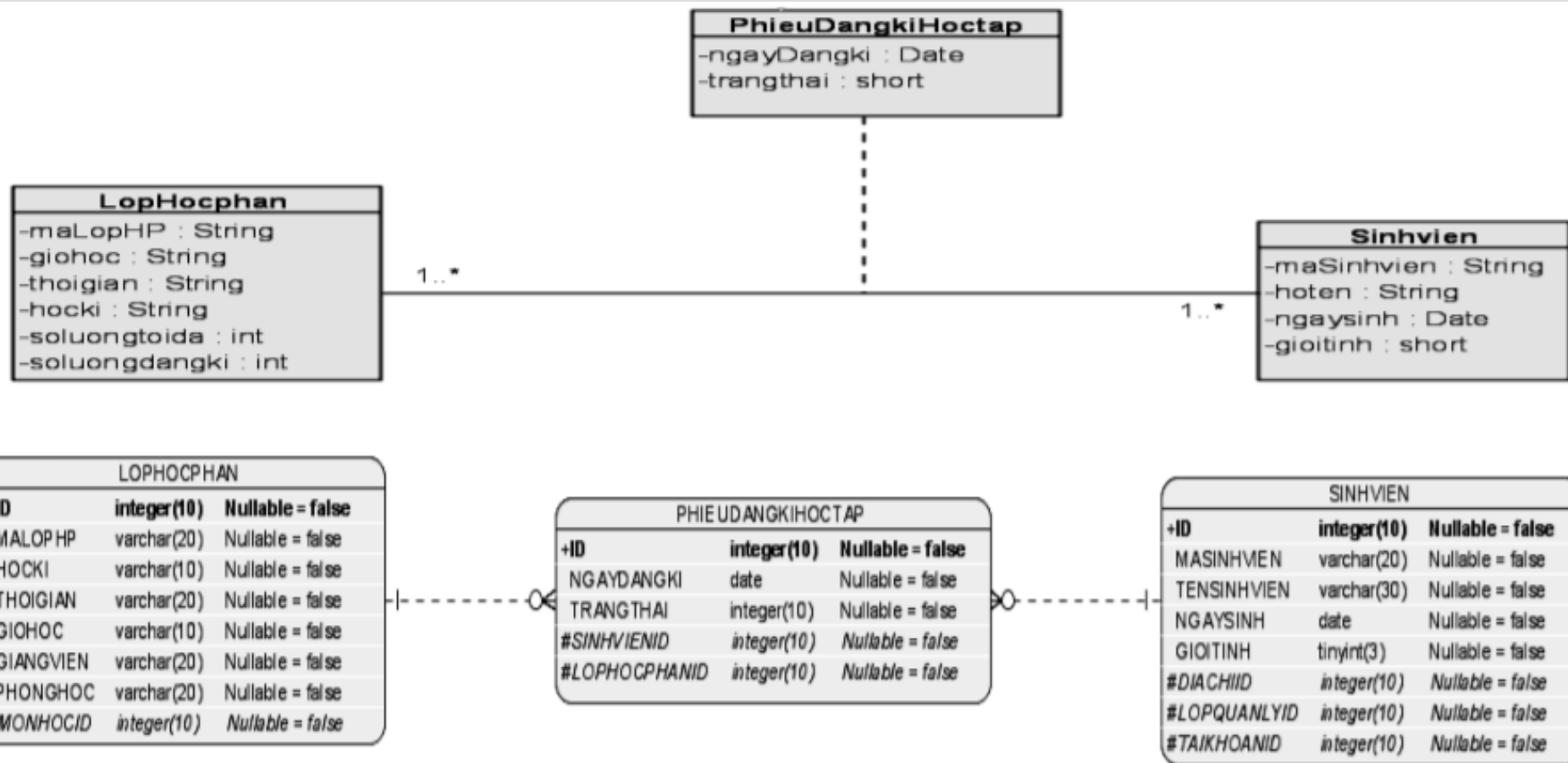
Ánh xạ thành mô hình quan hệ

❖ Liên kết n-m

- ❖ Một bảng kết nối (link table), trong đó mỗi dòng trong bảng liên kết biểu diễn một liên kết giữa một thực thể trong một bảng và một thực thể trong bảng khác
- ❖ Bảng liên kết có khoá chính là hợp của hai khoá ngoại ánh xạ đến đến bảng
- ❖ Có thể sử dụng bảng liên kết để lưu trữ quan hệ một - một, một - nhiều
- ❖ Ví dụ: lớp liên kết PhieuDangkiHocTap giữa 2 lớp LopHocPhan và lớp SinhVien

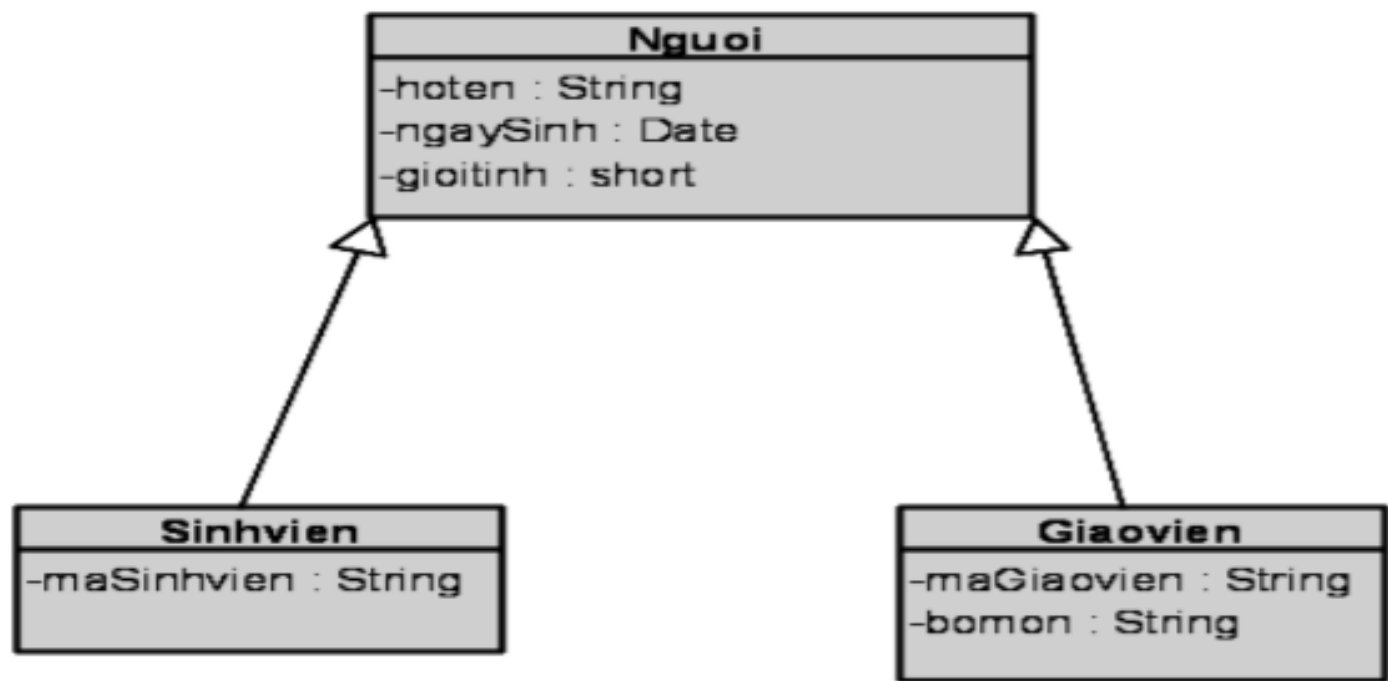
Ánh xạ thành mô hình quan hệ

❖ Liên kết n-m



Ánh xạ thành mô hình quan hệ

❖ Ánh xạ kết thừa:



❖ Cách 1: Sử dụng một bảng cho một lớp con cụ thể, mỗi bảng chứa cả thuộc tính của lớp cha và con => Bảng Giaovien, Sinhvien

Ánh xạ thành mô hình quan hệ

- ❖ Cách 2: Sử dụng mỗi bảng cho mỗi lớp, bảng con chứa khóa ngoại tham chiếu khóa chính của bảng cha

NGUOI		
+ID	integer(10)	Nullable = false
HOTEN	varchar(30)	Nullable = false
NGAYSINH	date	Nullable = false
GIOITINH	tinyint(3)	Nullable = false
#DIACHIID	integer(10)	Nullable = false

GIAOVIEN		
+ID	integer(10)	Nullable = false
MAGIAOVIEN	varchar(20)	Nullable = false
BOMON	varchar(20)	Nullable = false
#TAIKHOANID	integer(10)	Nullable = false
#KHOAID	integer(10)	Nullable = false

SINHVIEN		
+ID	integer(10)	Nullable = false
MASINHVIEN	varchar(20)	Nullable = false
#LOPQUANLYID	integer(10)	Nullable = false
#TAIKHOANID	integer(10)	Nullable = false

Phân phối hành vi của lớp vào CSDL

- ❑ Phân phối hành vi của lớp thành thủ tục lưu trong CSDL
- ❑ Một thủ tục lưu là mã có thể thực thi được chạy bên dưới RDBMS
- ❑ Có hai loại thủ tục lưu
 - Procedures: Được thực thi một cách tường minh bởi một ứng dụng
 - Triggers: Được gọi một cách ngầm định khi xảy ra một số sự kiện cơ sở dữ liệu

Phân phối hành vi của lớp vào CSDL

- ❑ Xác định xem có hoạt động (operation) nào có thể được cài đặt như một thủ tục lưu
- ❑ Các hoạt động có thể cân nhắc:
 - Các hoạt động làm việc với dữ liệu bền vững
 - Các hoạt động trong đó một truy vấn được gọi trong một tính toán
 - Các hoạt động cần thiết cho việc truy cập csdl để phê duyệt (validate) dữ liệu

Phân phối hành vi của lớp vào CSDL

Class

Student.

```
+ getTuition()  
+ addSchedule()  
+ getSchedule()  
+ deleteSchedule()  
+ hasPrerequisites()  
# passed()  
+ getNextAvailID()  
+ getStudentID()  
+ getName()  
+ getAddress()
```

Candidate Operations

- getTuition
- addSchedule
- getSchedule
- deleteSchedule
- getStudentID
- getName
- getAddress