



Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin
Bộ môn Công nghệ Phần mềm

Phân tích và thiết kế phần mềm

Phân tích

Giảng viên:
TS. Trần Minh Triết – ThS. Đặng Bình Phương

“Mastering Object-Oriented Analysis and Design with UML 2.0”

IBM Software Group



Nhắc lại về hướng đối tượng



Một số ký hiệu

Tên class

Tên class

(Các) thuộc tính

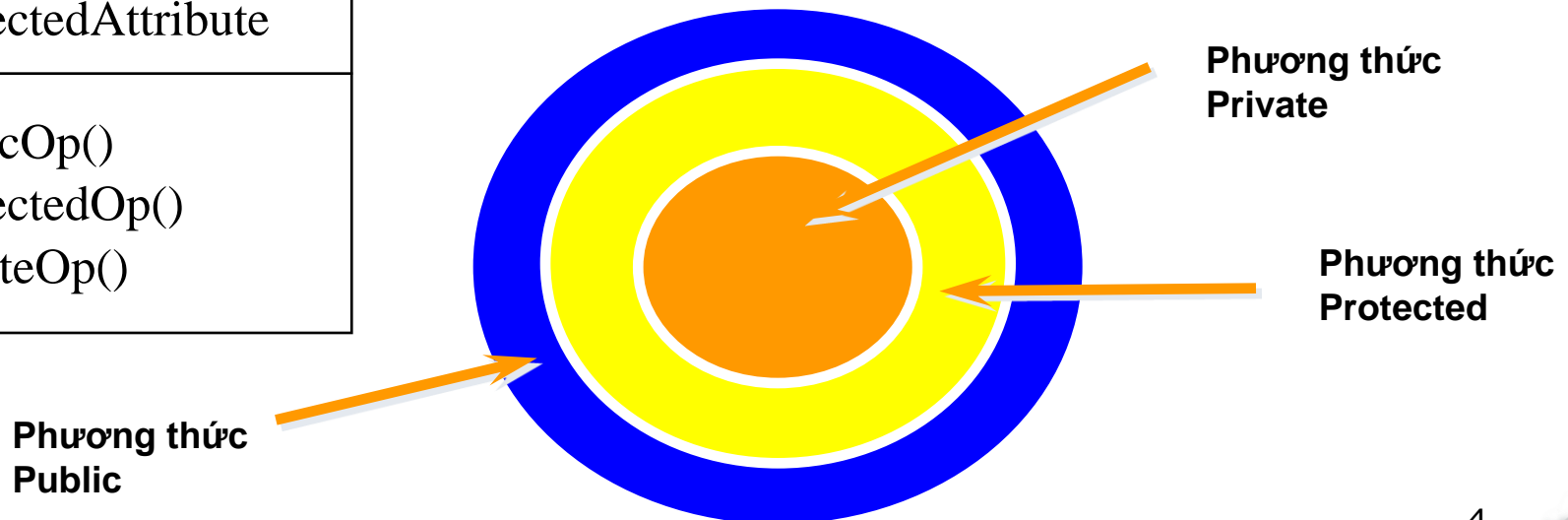
(Các) phương thức



Public/Protected/Private

- + Thuộc tính/Phương thức **public**
- # Thuộc tính/Phương thức **protected**
- Thuộc tính/Phương thức **private**

Class
- privateAttribute # protectedAttribute
+publicOp() # protectedOp() - privateOp()



- Xác định số lượng thể hiện của thuộc tính / phương thức

Class
- <u>classifierScopeAttribute</u> - instanceScopeAttribute
<u>classifierScopeOperation()</u> instanceScopeOperation()

Student

- name
- address
- studentID
- nextAvailID : int

- + addSchedule(theSchedule : Schedule, forSemester : Semester)
- + getSchedule(forSemester : Semester) : Schedule
- + hasPrerequisites(forCourseOffering : CourseOffering) : boolean
- # passed(theCourseOffering : CourseOffering) : boolean
- + getNextAvailID() : int

Nhận xét



Tên class
(Các) thuộc tính
(Các) phương thức

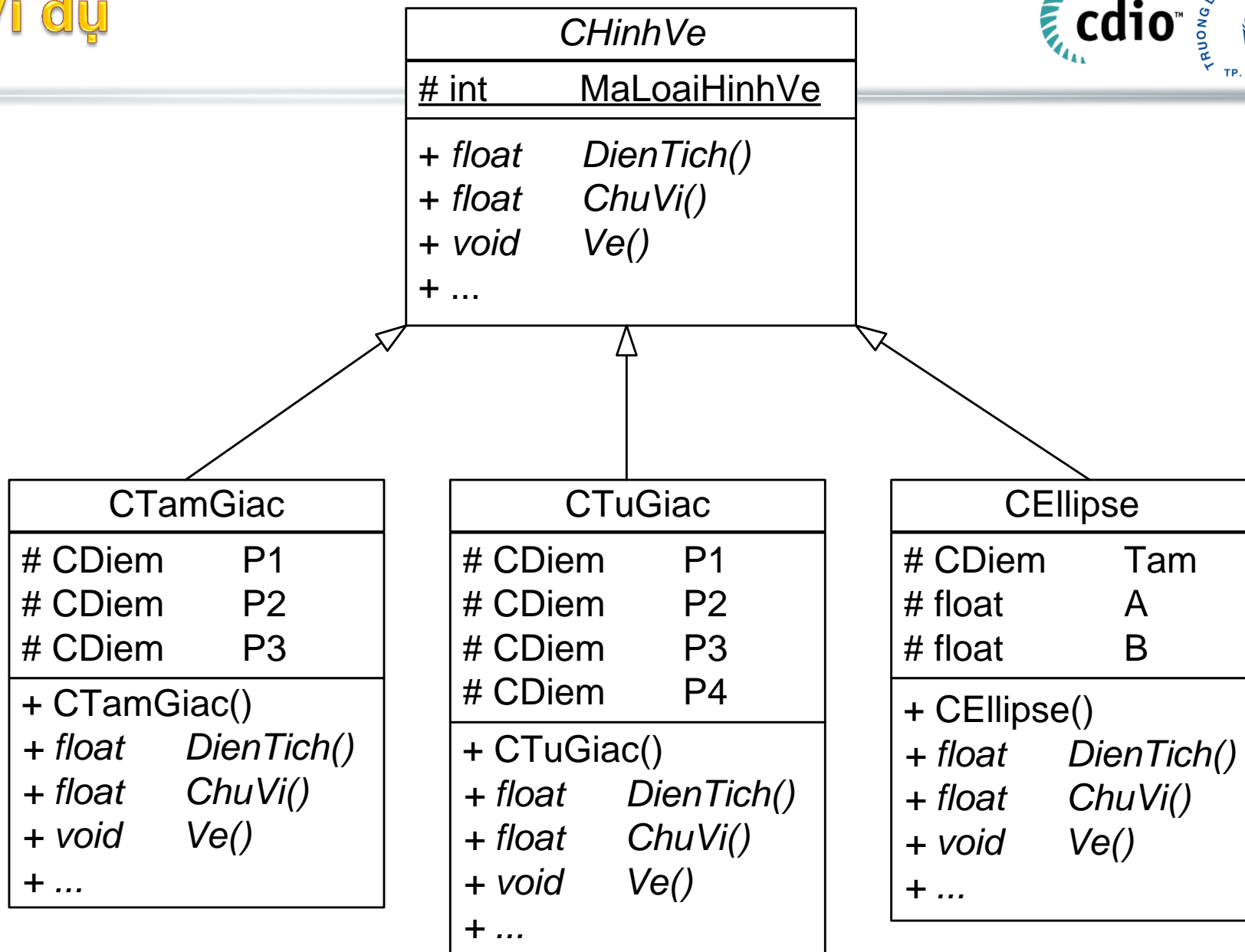
Bình thường: Class bình thường/Interface
In nghiêng: Class thuần ảo
Gạch dưới: Object (không phải class)

Bình thường: Thuộc tính bình thường
In nghiêng: không sử dụng
Gạch dưới: Thuộc tính static

Bình thường: Phương thức bình thường
In nghiêng: Phương thức virtual
Gạch dưới: Phương thức static

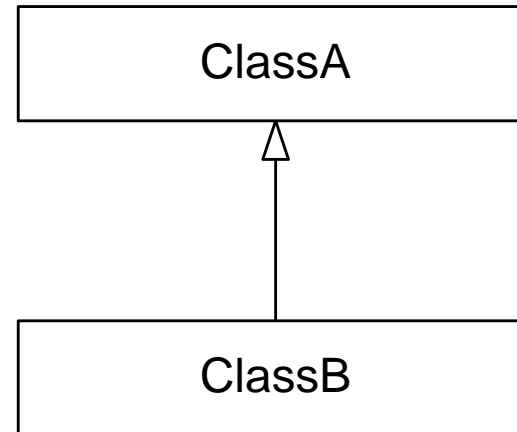


Ví dụ



Quan hệ giữa các lớp đối tượng

- Quan hệ kế thừa



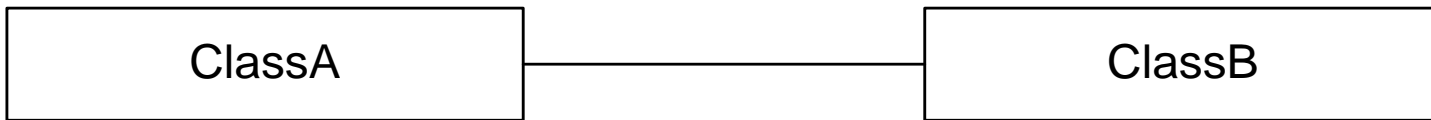
- ClassB kế thừa từ ClassA
- ClassB là một trường hợp đặc biệt của ClassA
- ClassA là trường hợp tổng quát của ClassB



Quan hệ giữa các lớp đối tượng



- Quan hệ Association

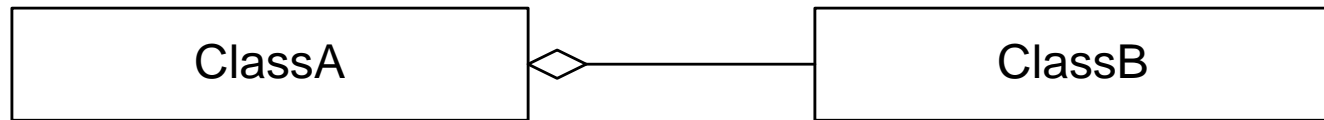


- Hoặc
 - Trong **ClassA** có thuộc tính có kiểu là **ClassB**
- Hoặc
 - Trong **ClassB** có thuộc tính có kiểu là **ClassA**
- Nhận xét: Về mặt lập trình, thuộc tính có thể được lưu trữ dạng **biến đơn**, **biến mảng**, hay **biến con trỏ**
- Ví dụ:?



Quan hệ giữa các lớp đối tượng

- Quan hệ Aggregation

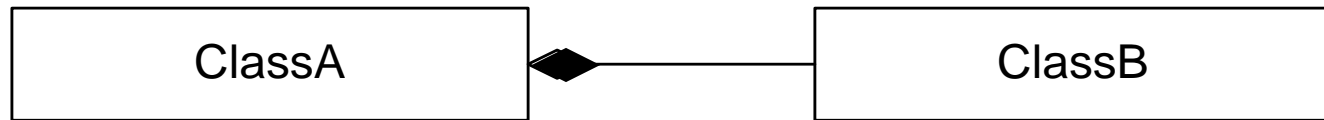


- Đã xác định được **ClassA** và **ClassB** có quan hệ Association với nhau
- Xác định rõ hơn:
 - Trong object của **ClassA** có chứa (trong phần thuộc tính) object của **ClassB**
 - **ObjectX** của **ClassA** bị hủy thì **ObjectY** của **ClassB** (bên trong **ObjectX**) vẫn có thể còn tồn tại
- Ví dụ:?



Quan hệ giữa các lớp đối tượng

- Quan hệ Composition

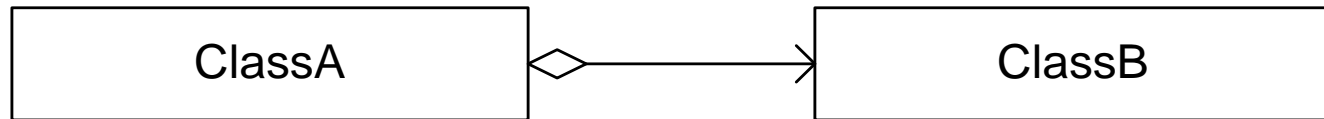


- Đã xác định được **ClassA** và **ClassB** có quan hệ Association với nhau
- Xác định rõ hơn:
 - Trong object của **ClassA** có chứa (trong phần thuộc tính) object của **ClassB**
 - **ObjectX** của **ClassA** bị hủy thì **ObjectY** của **ClassB** (bên trong **ObjectX**) không thể còn tồn tại
- Ví dụ:?



Quan hệ giữa các lớp đối tượng

- Chiều của quan hệ (Association, Aggregation, Composition)

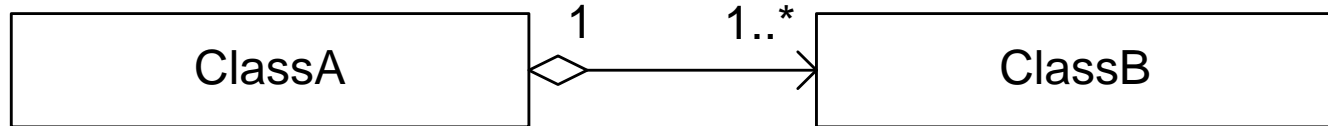


- Nếu quan hệ là 1 chiều: đa số các lời gọi hàm được gọi theo đúng chiều của quan hệ
- Nếu quan hệ là 2 chiều: không vẽ mũi tên



Quan hệ giữa các lớp đối tượng

- Bản số - Multiplicity (Association, Aggregation, Composition)

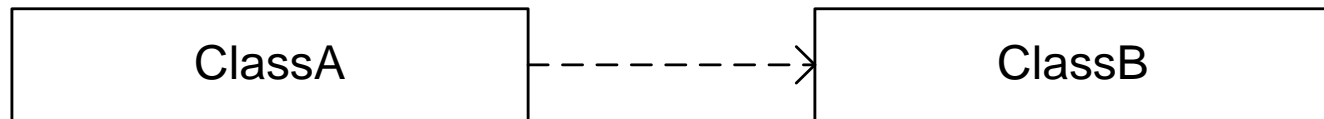


- Ý nghĩa
- Ví dụ:
 - 1
 - 2
 - 1..*
 - 0..*
 - *
 - 1, 3, 5..9



Quan hệ giữa các lớp đối tượng

- Quan hệ Dependency



- ClassA và ClassB không có quan hệ Association
- ClassA “phụ thuộc” vào ClassB

Tham số truyền vào

```
class A
{
    void F(B x)
    {
        ...
    }
};
```

Kết quả trả ra

```
class A
{
    B F()
    {
        ...
    }
};
```

Biến cục bộ

```
class A
{
    void F()
    {
        B x;
    }
};
```

Trong ClassA có sử dụng biến toàn cục (kiểu B), hoặc sử dụng phương thức/thuộc tính static của ClassB





Xây dựng sơ đồ lớp ở mức phân tích

Lập danh sách các đối tượng

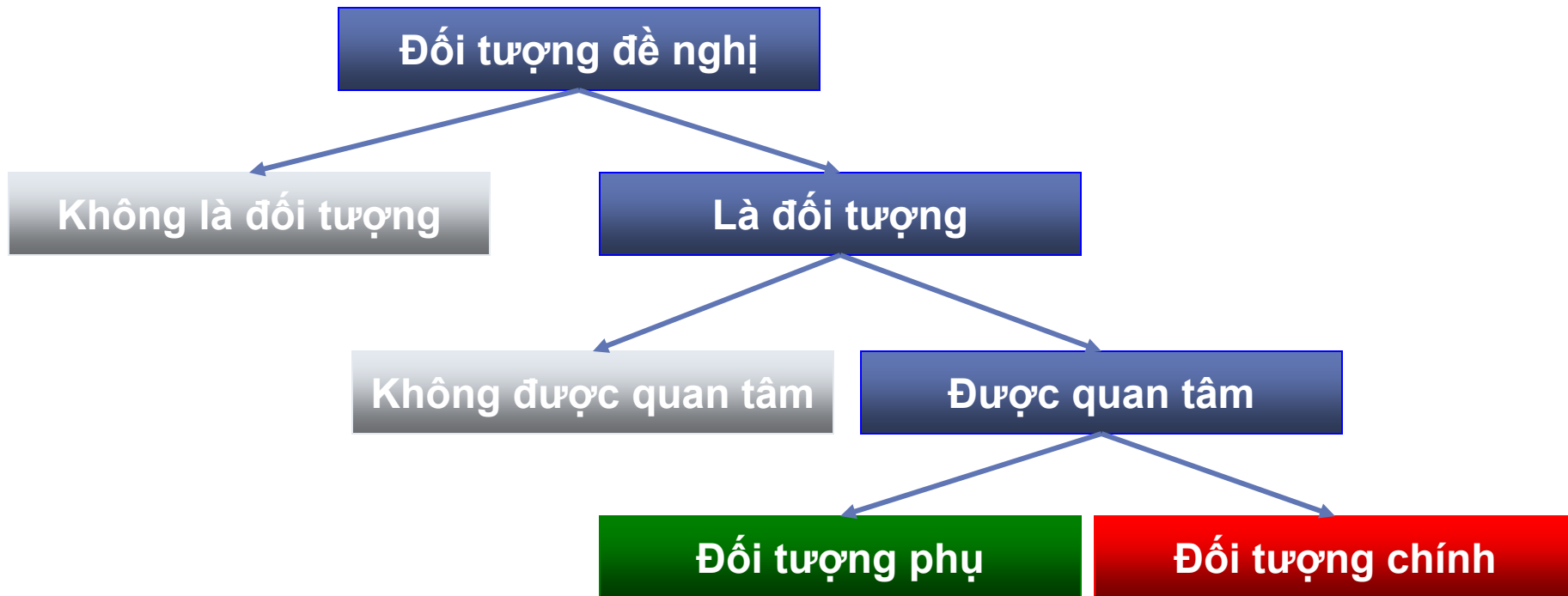


- Tiêu chuẩn nhận dạng đối tượng
 - Định danh: Đối tượng phải có tên (thường là danh từ/ngữ danh từ)
 - Chu trình sống: có thời điểm sinh ra, có khoảng thời gian hoạt động, có thời điểm chấm dứt
 - Sự độc lập tương đối với các đối tượng khác
 - ...
- Đề nghị:
 - Con người
 - Vật thể
 - Tổ chức
 - Vật lý
 - Không gian
 - Thời gian...



Lập danh sách các đối tượng

- Lập danh sách các đối tượng liên quan đến hệ thống



Tiêu chuẩn nhận dạng đối tượng: có rất nhiều trường phái



- Ví dụ: Xét ngữ cảnh là 1 trường PTTH với phần mềm quản lý trường cấp 3:
- Danh sách đề nghị:
 - Học sinh Tổ Bộ môn Số tiết
 - Giáo viênBGH TKB
 - Môn học Khối Bảng điểm
 - Lớp Phụ huynh Phòng
 - Học kỳ ĐTB Học phí
 - Năm học Diện HS ...
- Đối tượng/Không phải đối tượng?



- Được quan tâm?
 - Phần mềm quản lý học sinh:
 - Học sinh, Giáo viên, Môn học, Lớp, Khối, Phụ huynh, Học kỳ, Năm học...
 - Phần mềm quản lý giáo viên:
 - Giáo viên, Tổ bộ môn, Môn học, Khối, Lớp, Học kỳ, Năm học...
 - Phần mềm xếp thời khóa biểu:
 - Giáo viên, Môn học, Lớp, Phòng, Học kỳ, Năm học...



- **Đối tượng chính? Đối tượng phụ**
 - **Phần mềm quản lý học sinh:**
 - Học sinh, Giáo viên, Môn học, Lớp, Khối, Phụ huynh, Học kỳ, Năm học...
 - **Phần mềm quản lý giáo viên:**
 - Giáo viên, Tổ bộ môn, Môn học, Khối, Lớp, Học kỳ, Năm học...
 - **Phần mềm xếp thời khóa biểu:**
 - Giáo viên, Môn học, Lớp, Phòng, Học kỳ, Năm học...



Lập danh sách các quan hệ



- Tiêu chí đánh giá:
 - Động từ
 - Sự phụ thuộc giữa các đối tượng (chủ yếu xét các đối tượng chính)
- Đề nghị:
 - Quan hệ theo thời gian
 - **Ít biến động**: sau 1 thời gian dài mới thay đổi (thường làm về mặt tổ chức)
 - **Biến động**: quan hệ xảy ra vào lúc nào, trong thông tin có thuộc tính về thời gian, thay đổi theo thời gian (**thường quan tâm nhiều đến loại quan hệ này**)
 - Quan hệ về tổ chức (thường liên quan đến đối tượng phụ)
 - Quan hệ về không gian (thường liên quan đến đối tượng phụ)
 - Quan hệ theo vai trò: Chủ động/Bị động
- Ví dụ:?



Nhận dạng thuộc tính



- Sự phụ thuộc (không có ý nghĩa rõ ràng khi đứng độc lập)
 - Phụ thuộc một đối tượng → Thuộc tính của đối tượng
 - Phụ thuộc nhiều đối tượng → Thuộc tính của quan hệ
- Các loại thuộc tính
 - Định danh (thường của đối tượng)
 - Phân loại
 - Thời gian
 - Không gian
 - Định lượng
 - ...
- Ví dụ:?



Các bước xây dựng sơ đồ lớp ở mức phân tích



- Bước 1: Xác định các lớp đối tượng, quan hệ và thuộc tính và phương thức (trách nhiệm) trực tiếp từ yêu cầu của hệ thống
 - Xét lần lượt từng biểu mẫu và quy định
 - Nếu trong sơ đồ lớp hiện tại chưa có thể lưu trữ được thông tin cần thiết:
 - Cần bổ sung thuộc tính vào lớp đối tượng đã có?
 - Cần bổ sung thuộc tính vào quan hệ đã có?
 - Cần bổ sung thêm quan hệ giữa các lớp đối tượng đã có?
 - Cần bổ sung thêm lớp đối tượng mới?



Các bước xây dựng sơ đồ lớp ở mức phân tích



- Bước 2:
 - Nếu một lớp đối tượng có thuộc tính có cấu trúc phức tạp hoặc có các thuộc tính có liên hệ chặt chẽ với nhau và có ngữ nghĩa cụ thể thì nên tách ra thành lớp đối tượng phụ



Các bước xây dựng sơ đồ lớp ở mức phân tích



- Bước 3:
 - 3.1. **Nhiều** lớp đối tượng có **nhiều đặc điểm chung**
→ **Xây dựng** lớp đối tượng tổng quát chung cho các lớp đối tượng cụ thể này
 - 3.2. **Một** lớp đối tượng có **thuộc tính phân loại** và **cách xử lý** trong các **phương thức** của đối tượng thuộc lớp này **phụ thuộc** vào **giá trị của thuộc tính phân loại**
→ **Tách** lớp đối tượng này thành nhiều lớp đối tượng con **tương ứng** với **mỗi (nhóm) giá trị của thuộc tính phân loại**



Các bước xây dựng sơ đồ lớp ở mức phân tích



- Bước 4:
 - Hiệu chỉnh các quan hệ đã có để phù hợp với các lớp đối tượng vừa được điều chỉnh
- Bước 5:
 - Kiểm tra lại sơ đồ lớp và hiệu chỉnh (theo kinh nghiệm)



Kết quả: Sơ đồ lớp ở mức phân tích



- Sơ đồ lớp
- Danh sách các lớp đối tượng và quan hệ

STT	Tên lớp/quan hệ	Loại	Ý nghĩa/ghi chú
...

- Mô tả chi tiết từng lớp đối tượng và quan hệ
 - Với mỗi lớp đối tượng:
 - Mô tả các thuộc tính

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
...

- Danh sách các trách nhiệm chính
 - Với mỗi quan hệ:

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
...

- Áp dụng thực tế vào các bài tập
 - Xác định các lớp đối tượng chính
 - Xác định các thông tin và hành động/trách nhiệm của mỗi lớp đối tượng chính
 - Xác định các quan hệ chính
 - Xác định các lớp đối tượng phụ, các danh mục



Sơ đồ trạng thái

- **Sơ đồ trạng thái**
 - Mô tả chu trình sống của các đối tượng chính từ khi sinh ra, hoạt động & mất đi
 - Mỗi đối tượng có thể có nhiều sơ đồ trạng thái theo các góc nhìn khác nhau
- **Trạng thái & biến cố**
 - **Trạng thái** của đối tượng diễn đạt tình trạng hiện có của đối tượng (có ý nghĩa trong một đoạn thời gian)
 - **Biến cố** là các sự kiện xảy ra làm cho đối tượng chuyển trạng thái.



Một số ký hiệu



Tên trạng thái

Trạng thái

Tên trạng thái

stateVar : type = value

entry/ entry action
do/ activity
exit/ exit action

Tên biến cố (tham số)

Tên hành động (tham số)

→
[Điều kiện]

Chuyển đổi trạng thái



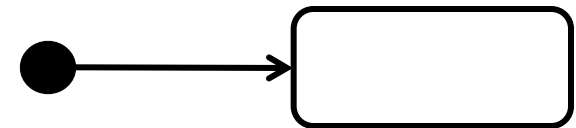
Các trạng thái đặc biệt



- **Trạng thái bắt đầu**

- Là trạng thái khi mới được khởi tạo của object
- Bắt buộc phải có
- Chỉ có thể có 1 trạng thái bắt đầu

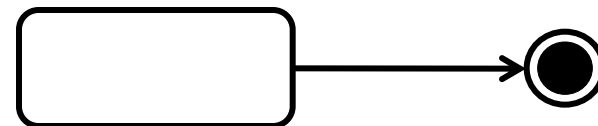
Tr/thái bắt đầu



- **Trạng thái kết thúc**

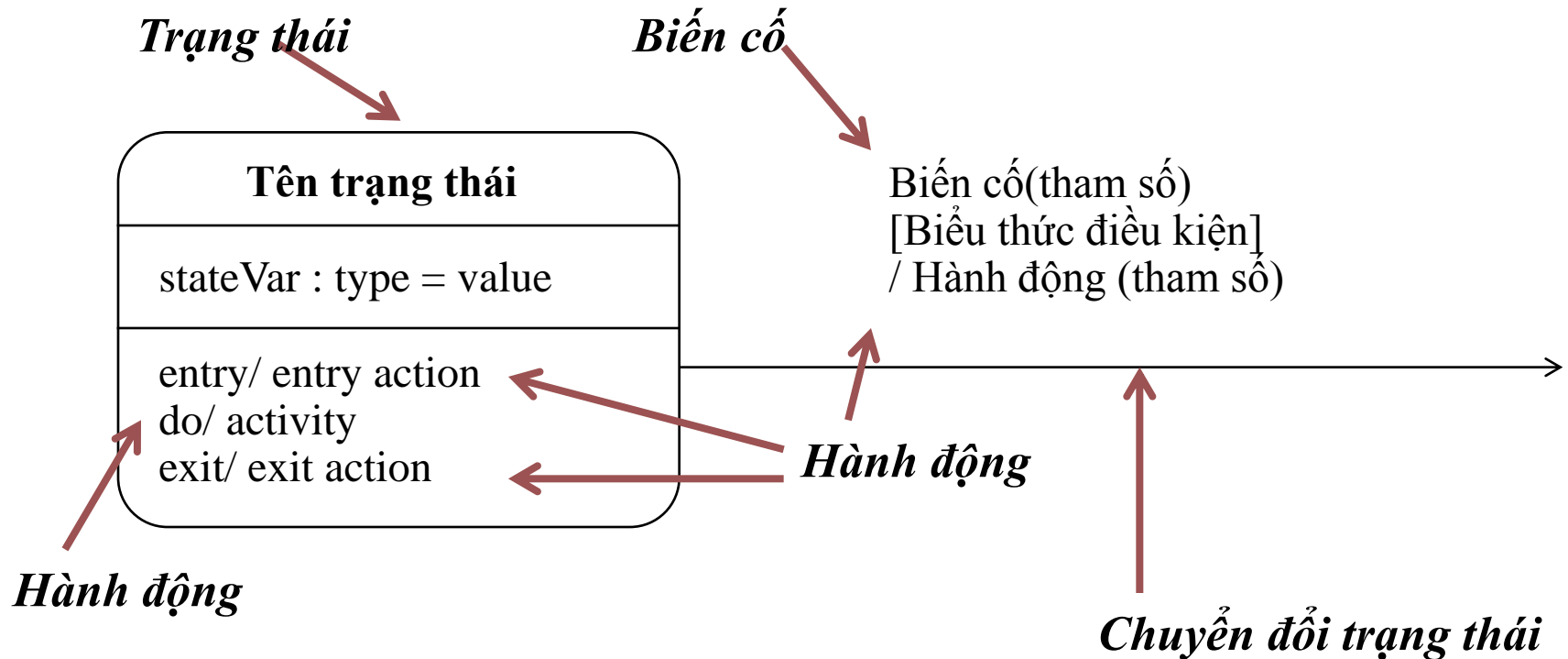
- Chỉ vị trí kết thúc đời sống của object
- Không nhất thiết phải thể hiện
- Có thể có nhiều

Tr/thái kết thúc



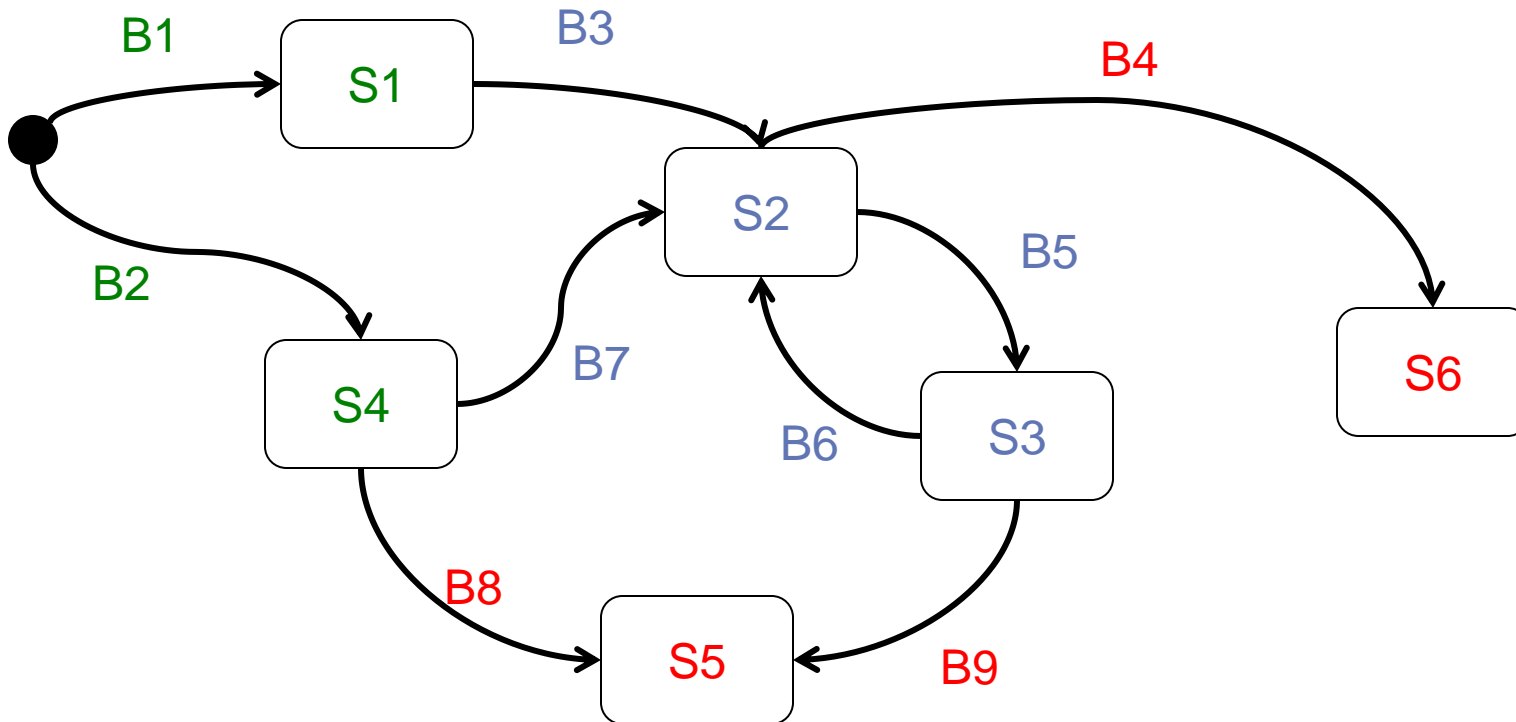
Sơ đồ trạng thái

- Là đồ thị có hướng với các node là các trạng thái nối với nhau bởi các cung mô tả việc chuyển đổi trạng thái



Sơ đồ trạng thái

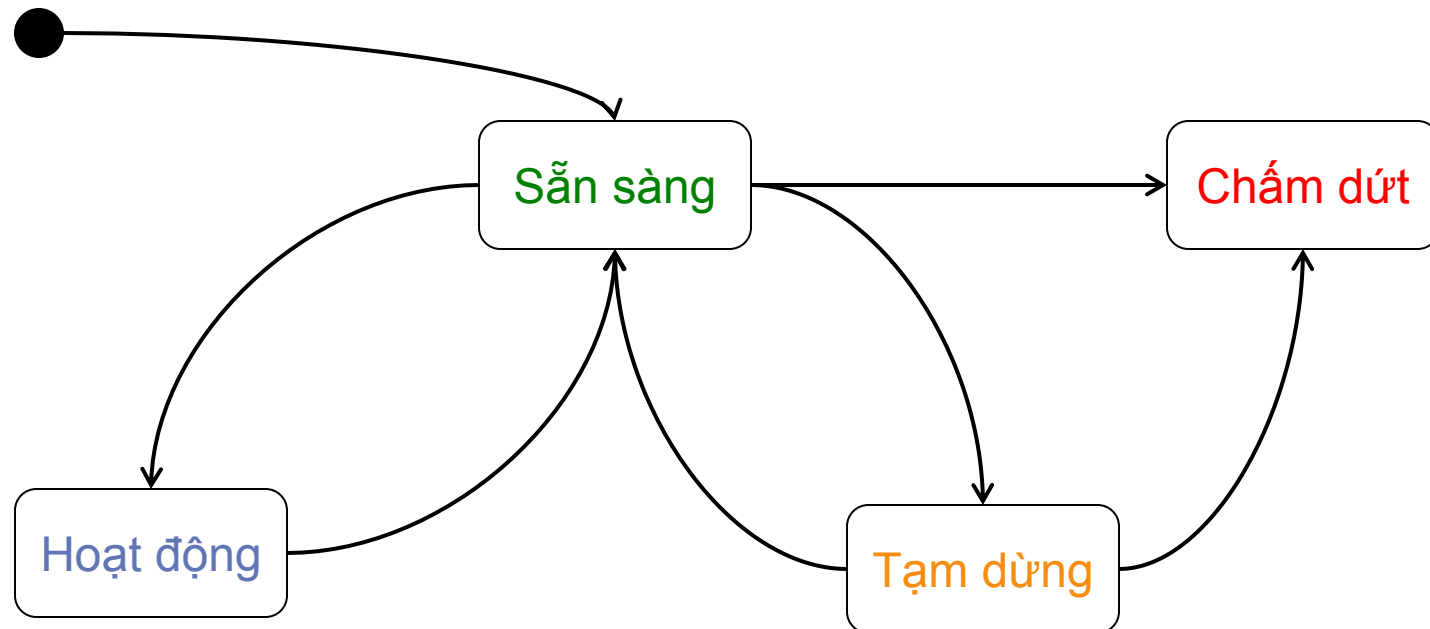
B1, B2: biến cố sinh
B4, B8, B9: biến cố mất
B3, B5, B6, B7: biến cố hoạt động



- Mỗi sơ đồ trạng thái sẽ ánh xạ thành **một thuộc tính** của đối tượng
- Các trạng thái của đối tượng trong 1 sơ đồ trạng thái **không giao nhau** và giá trị của thuộc tính tương ứng chỉ mang 1 trong các giá trị tương ứng
- Sơ đồ trạng thái càng chi tiết sẽ phục vụ:
 - Có những xử lý thích hợp trong kiểm tra ràng buộc
 - Có những xử lý thích hợp trong xử lý biến cố
 - Phục vụ tra cứu, tìm kiếm
- Khi đã quan tâm đến sơ đồ trạng thái của 1 đối tượng
 - Thường trong ứng dụng **không còn chức năng xóa** (thật sự) đối tượng này
 - Các đối tượng sẽ **tồn tại ở trạng thái mất** thay vì bị xóa thật sự



- Thông thường, sơ đồ trạng thái của 1 đối tượng có dạng



Cách xây dựng sơ đồ trạng thái

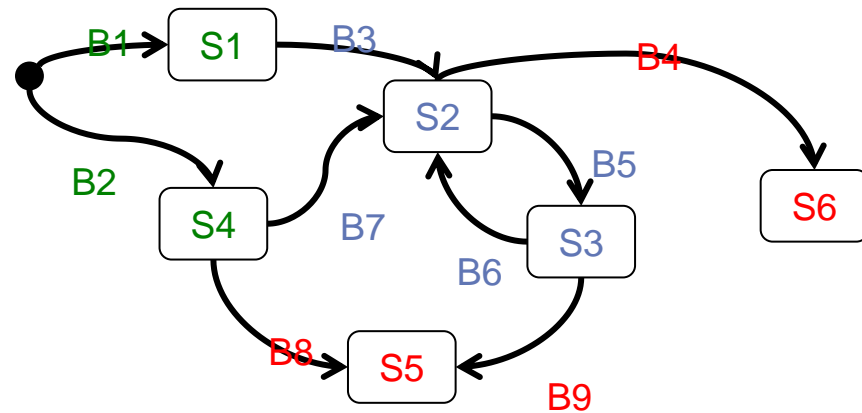


- Cách 1:
 - Xác định quá trình hoạt động bình thường của đối tượng từ khi sinh ra đến khi mất đi
 - Bổ sung các trạng thái và biến cố liên quan đến
 - Sinh ra
 - Hoạt động
 - Tạm ngừng
 - Mất đi
- Cách 2:
 - Xác định các **trạng thái sinh**
 - Xác định các **trạng thái cuối**
 - Xác định các **trạng thái trung gian** và các biến cố/hành động làm chuyển trạng thái



Kết quả: Sơ đồ trạng thái

Sơ đồ trạng thái



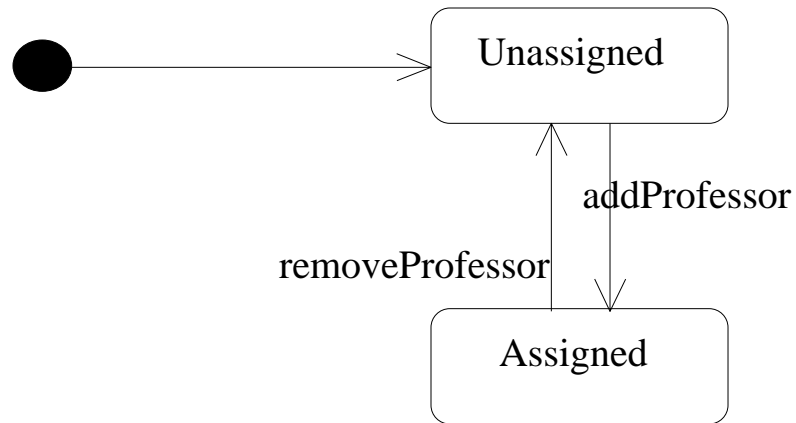
Bảng mô tả các trạng thái

STT	Trạng thái	Ý nghĩa	Xử lý liên quan	Ghi chú
...

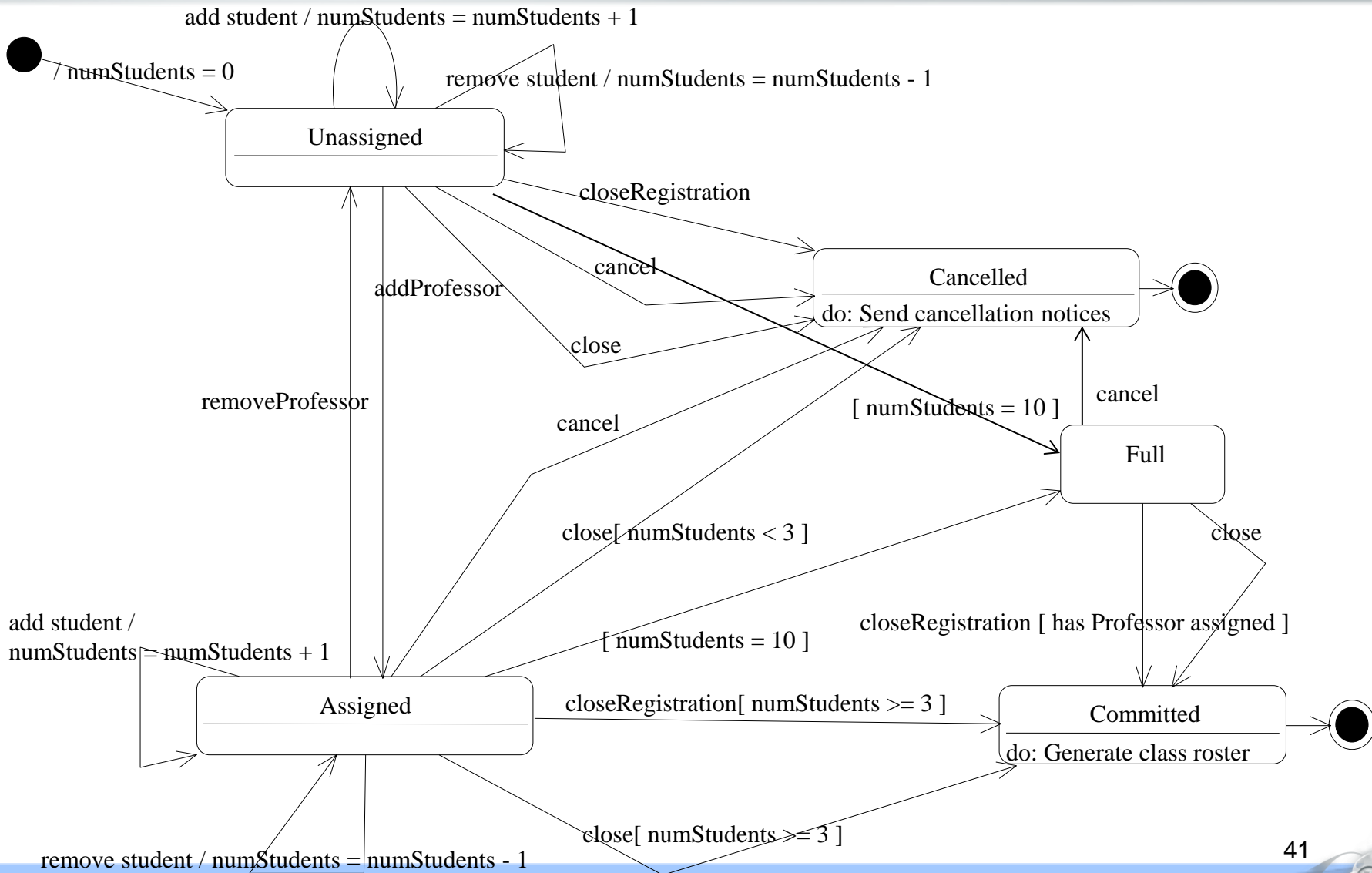
Bảng mô tả các biến cố

STT	Biến cố	Ý nghĩa	Xử lý liên quan	Ghi chú
...

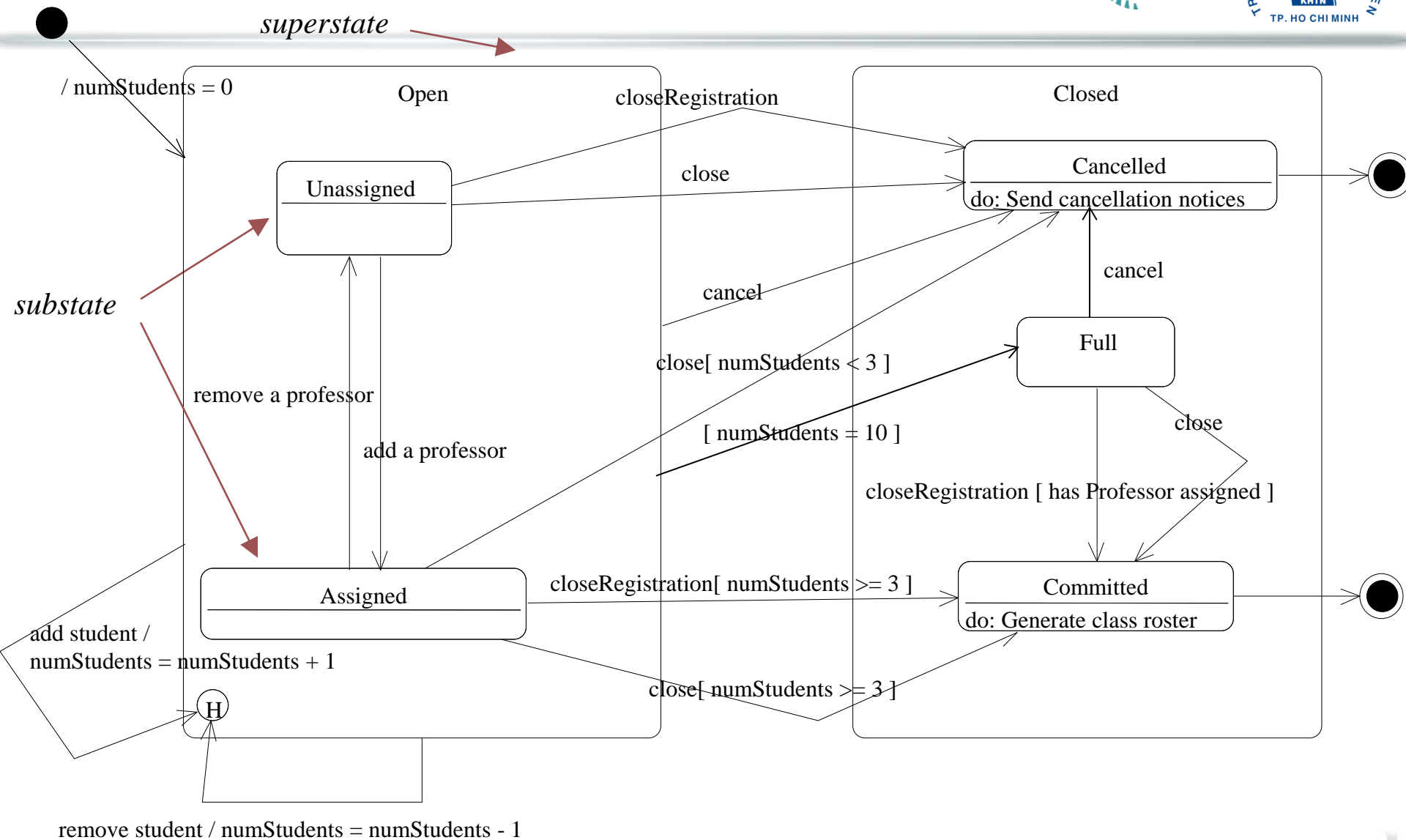




Ví dụ



Ví dụ: trạng thái lồng nhau



- Vẽ sơ đồ trạng thái của thang máy
- Vẽ sơ đồ trạng thái của 1 đèn giao thông
- Vẽ sơ đồ trạng thái của 1 cặp đèn giao thông tại ngã tư
- Vẽ sơ đồ trạng thái của 1 độc giả
- Vẽ sơ đồ trạng thái của 1 cuốn sách
- Vẽ sơ đồ trạng thái của 1 phòng trong khách sạn
- Vẽ sơ đồ trạng thái của 1 quân cờ
- ...

