

HIENLTH



Chủ đề 3 (tt): Phân tích Phần mềm

HIENLTH



Phân tích Hướng đối tượng

Nội dung



- **Sơ đồ lớp ở mức phân tích**
 - 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 lớp và Khả năng tiến hóa của hệ thống**
- **Sơ đồ trạng thái**
 - Khái niệm và các ký hiệu
 - Trạng thái
 - Biến cố, điều kiện
 - Trạng thái đầu, trạng thái cuối
 - Superstate
 - Áp dụng

Một số khái niệm mở đầu



- Vấn đề: Mô tả lại bằng một ngôn ngữ nào đó (thường là các sơ đồ) nhằm diễn tả trực quan về vấn đề
- Phân tích: xây dựng mô hình Thế giới thực
- Phân tích theo hướng đối tượng: xây dựng các mô hình về các đối tượng của Thế giới thực
- Một số loại Sơ đồ:
 - Sơ đồ lớp đối tượng: Mô tả hệ thống các lớp đối tượng (thuộc tính, hành động) cùng với các quan hệ giữa chúng
 - Sơ đồ trạng thái: Mô tả chu trình sống của đối tượng
 - ...



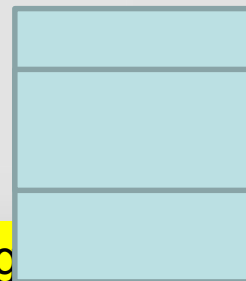
Lớp (class) là gì?

- Đối tượng là cái gì đó tồn tại trong thế giới thực
- Lớp là mô tả thuộc tính, hành vi, ngữ nghĩa của một nhóm đối tượng
 - Lớp xác định thông tin nào được lưu trữ trong đối tượng và hành vi nào đối tượng có
- Thí dụ về lớp: Lớp **NhanVien**
 - Đối tượng của lớp có các attribute: **HoTen**, **DiaChi**, **Luong**
 - Các hành vi: Thuê mượn, Đuổi việc và Đề bạt nhân viên?

Sơ đồ Lớp Class Diagram



- Là biểu đồ quan trọng nhất.
- Mô tả các đối tượng và mối quan hệ của chúng trong hệ thống.
- Mô tả các thuộc tính và các hành vi (Behavior) của đối tượng.
- Có biểu đồ lớp mức phân tích và mức cài đặt.
- Ký pháp đồ họa của lớp trong biểu đồ
 - Tên lớp
 - Thuộc tính
 - Thao tác



+ : public
- : private
: protected

Nhắc lại về hướng đối tượng



Một số ký hiệu

Tên class

Tên class

(Các) thuộc tính

(Các) phương thức

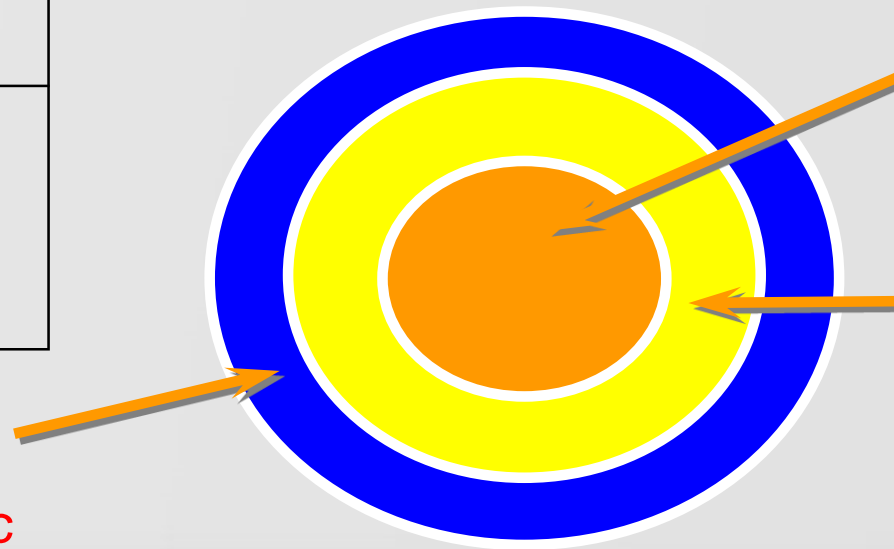


Public/Protected/Private

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

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

Phương
thức **Public**



Phương thức
Private

Phương
thức
Protected

Tầm vực



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

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



Ví dụ

CStudent

- 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
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



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

Tìm kiếm lớp như thế nào?



- Tìm đầy đủ lớp rất khó khăn.
- Khuyến cáo
 - Tìm lớp từ các danh từ trong luồng sự kiện
 - Chú ý rằng danh từ có thể là tác nhân, lớp, (thuộc tính và biểu thức không phải loại trên
 - Tìm lớp từ biểu đồ tương tác
 - Những cái chung của đối tượng tạo thành lớp
 - Tìm lớp ở các nơi khác
 - Các báo cáo tìm ra trong pha phân tích yêu cầu hình thành lớp giao diện
 - Các thiết bị phần cứng được biểu diễn bởi lớp khác nhau

Tìm kiếm lớp như thế nào?



- Cùng với chuyên gia lĩnh vực vấn đề trả lời các câu hỏi sau đây để tìm ra lớp
 - Có **thông tin nào cần lưu trữ** hay phân tích? Nếu có, nó là lớp
 - Có **hệ thống ngoài** không? Nếu có thì nó được xem như những lớp chứa trong hệ thống của ta hay hệ thống của ta tương tác với chúng
 - Có **mẫu, thư viện lớp, thành phần...**? Nếu có, thông thường chúng chứa các ứng viên lớp
 - Hệ thống cần quản lý các thiết bị ngoại vi nào? Mọi thiết bị kỹ thuật nối với hệ thống đều là ứng viên lớp.
 - Tác nhân đóng vai trò tác nghiệp nào? Các nhiệm vụ này có thể là lớp; thí dụ người sử dụng, thao tác viên hệ thống, khách hàng...

Mối quan hệ giữa các class

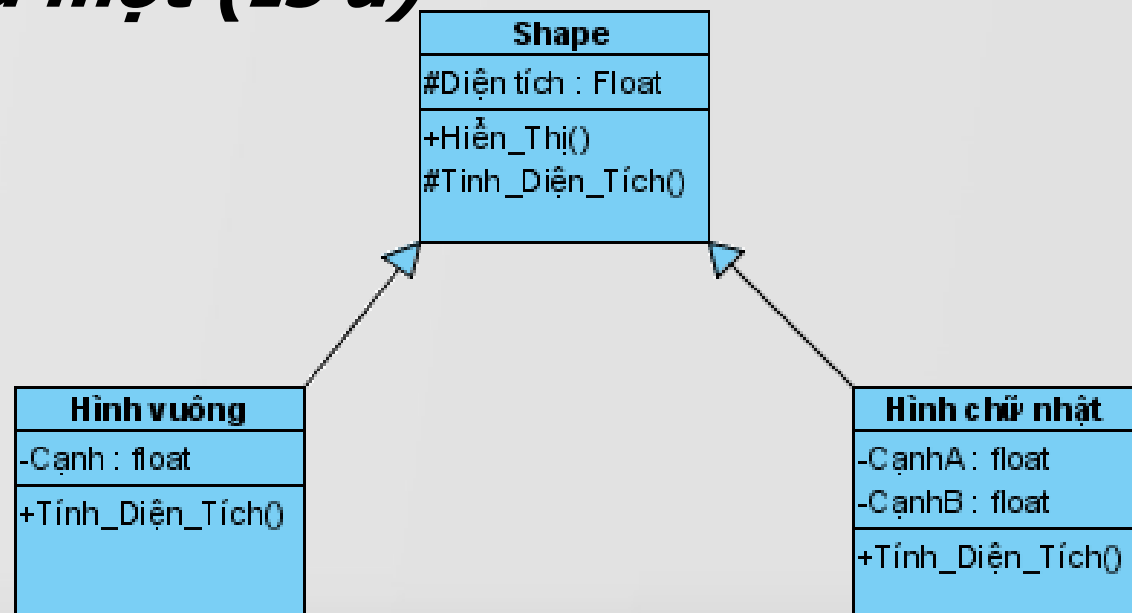


- Generalization: tổng quát hóa
- Association:
 - dependency
 - aggregation
 - composition



Các quan hệ trong biểu đồ lớp

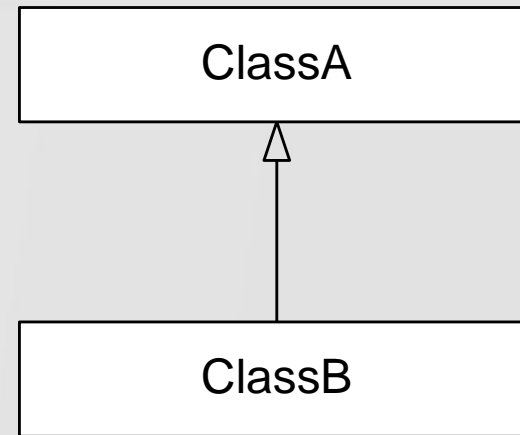
- Quan hệ **Generalization**: Thể hiện rằng một lớp A kế thừa từ một lớp B (Hay A là trường hợp riêng của B; B là tổng quát của A)
- Gọi là quan hệ **Là một (Is a)**
- Thể hiện:



Quan hệ giữa các lớp đối tượng



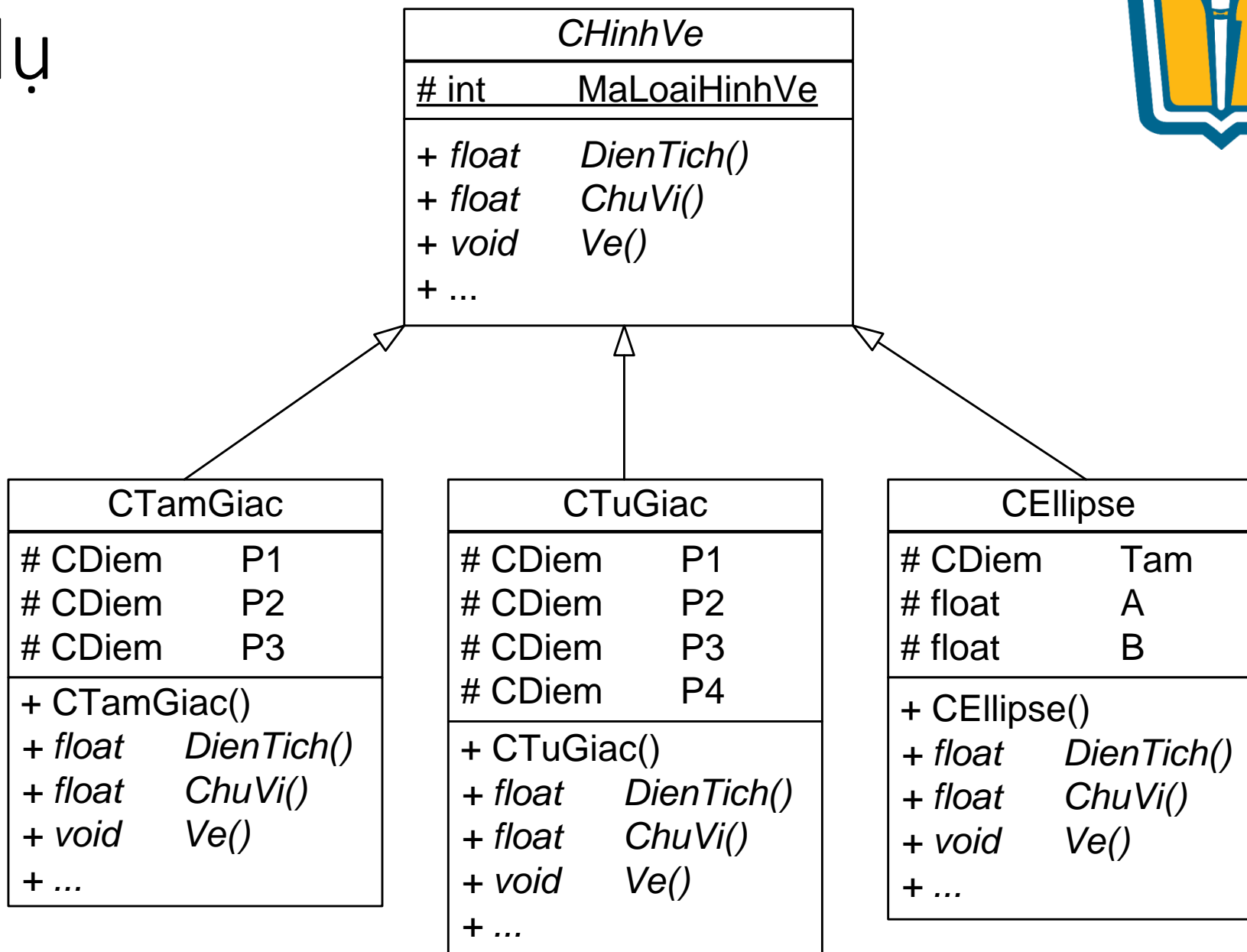
- Quan hệ kế thừa



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



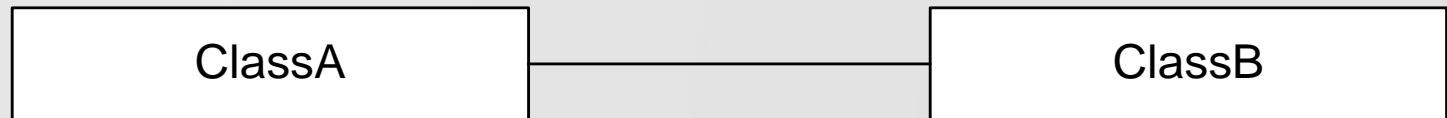
Ví dụ



Quan hệ giữa các lớp đối tượng



- Quan hệ Association



- Hoặc

- Trong **ClassA** có thuộc tính có kiểu là **ClassB**

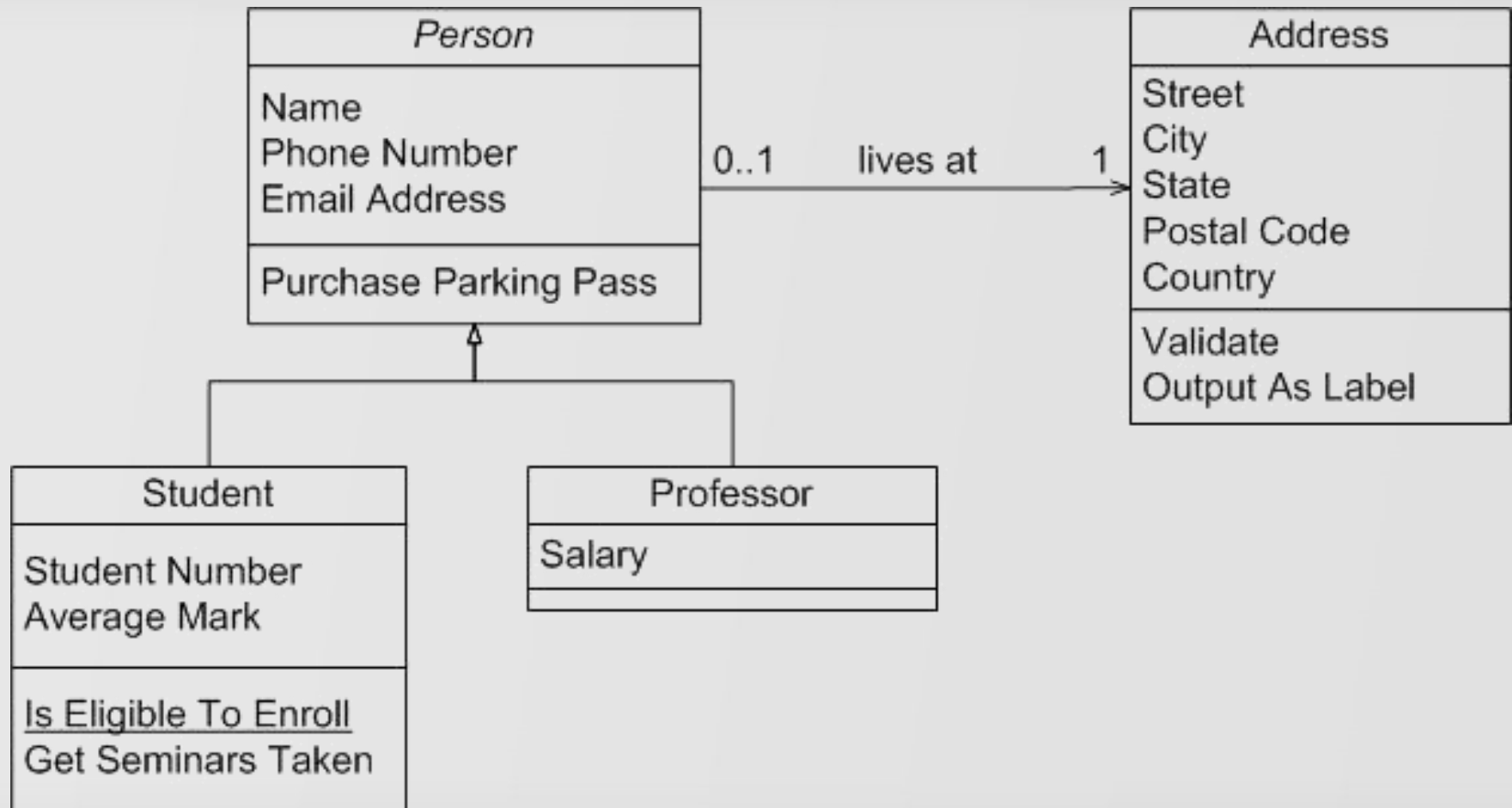
- Hoặc

- 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: ?

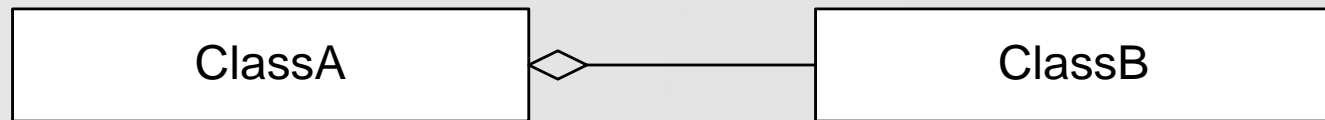
Ví dụ



Quan hệ giữa các lớp đối tượng



- Quan hệ Aggregation

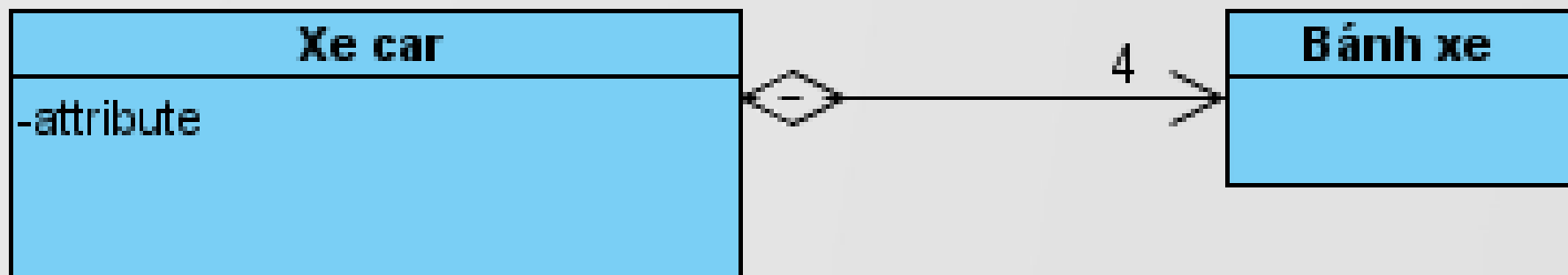


- Đã xác định được **ClassA** và **ClassB** có quan hệ Association với nhau
 - Xác định rõ hơn:
 - Trong object của **ClassA** có chứa (trong phần thuộc tính) object của **ClassB**
 - **ObjectX** của **ClassA** bị hủy thì **ObjectY** của **ClassB** (bên trong **ObjectX**) vẫn có thể còn tồn tại
- Ví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



Quan hệ Aggregation

- Còn gọi là mối quan hệ: ***Có một (Has a)***
- Ví dụ:



Quan hệ giữa các lớp đối tượng



- Quan hệ Composition



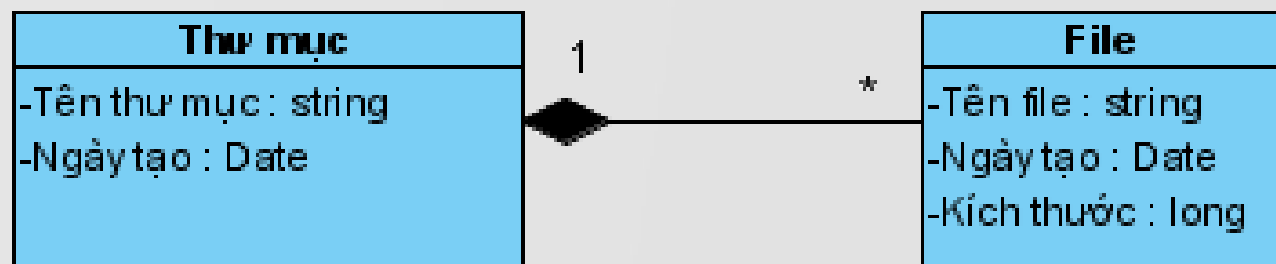
- Đã xác định được **ClassA** và **ClassB** có quan hệ Association với nhau
 - Xác định rõ hơn:
 - Trong object của **ClassA** có chứa (trong phần thuộc tính) object của **ClassB**
 - **ObjectX** của **ClassA** bị hủy thì **ObjectY** của **ClassB** (bên trong **ObjectX**) không thể còn tồn tại
 - Ví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



Quan hệ Composition

- Thể hiện rằng một lớp A bao hàm lớp B. Nhưng lớp B không thể tồn tại độc lập (Tức không thuộc lớp nào). Tức là, nếu có B thì phải suy ra được A.

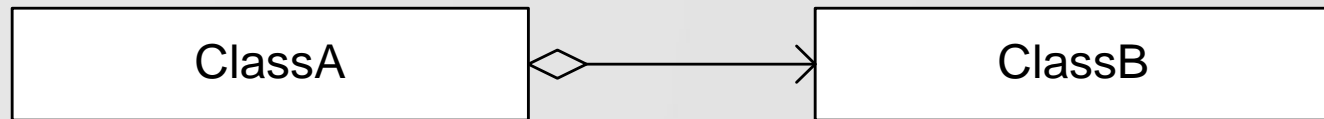
- Thể hiện:



Quan hệ giữa các lớp đối tượng



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

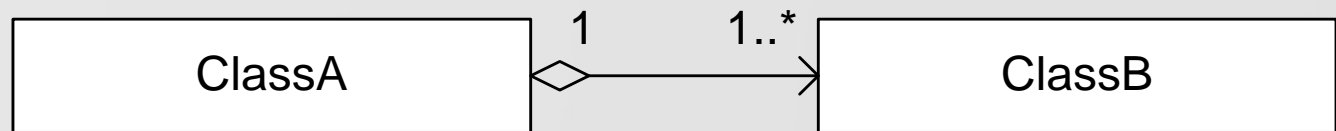


- Nếu quan hệ là 1 chiều: đa số các lời gọi hàm được gọi theo đúng chiều của quan hệ
- Nếu quan hệ là 2 chiều: không vẽ mũi tên

Quan hệ giữa các lớp đối tượng



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



– Ý nghĩa

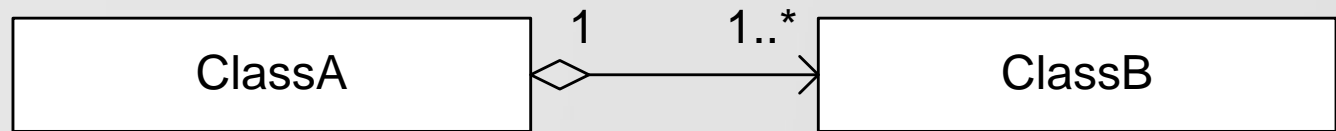
– Ví dụ:

- 1
- 2
- 1..*
- 0..*
- *
- 1, 3, 5..9

Quan hệ giữa các lớp đối tượng



- 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

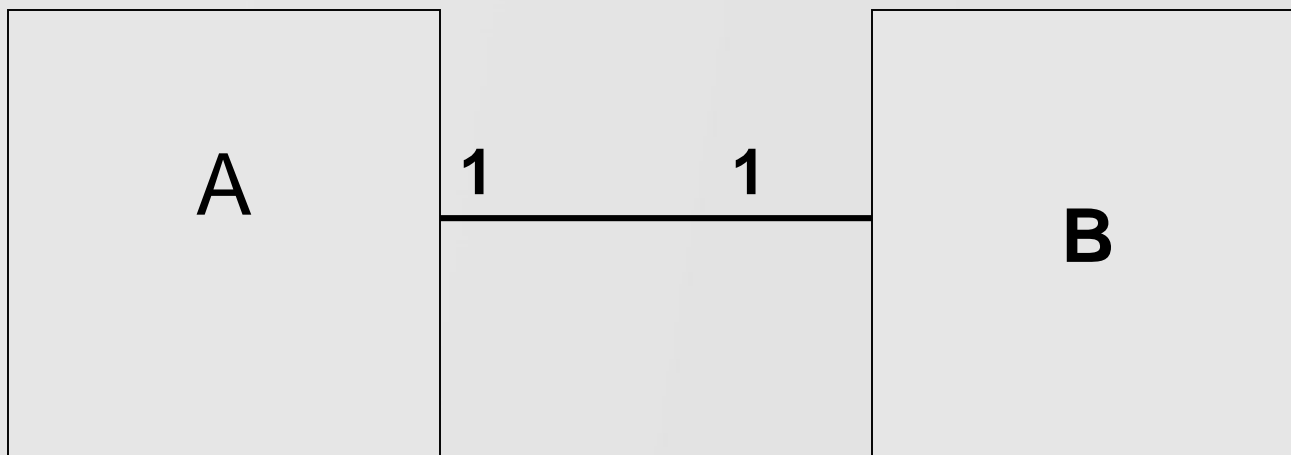
```
class A
{
    void F()
    {
        B x;
    }
};
```

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

Bản số (Multiplicity)



- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?

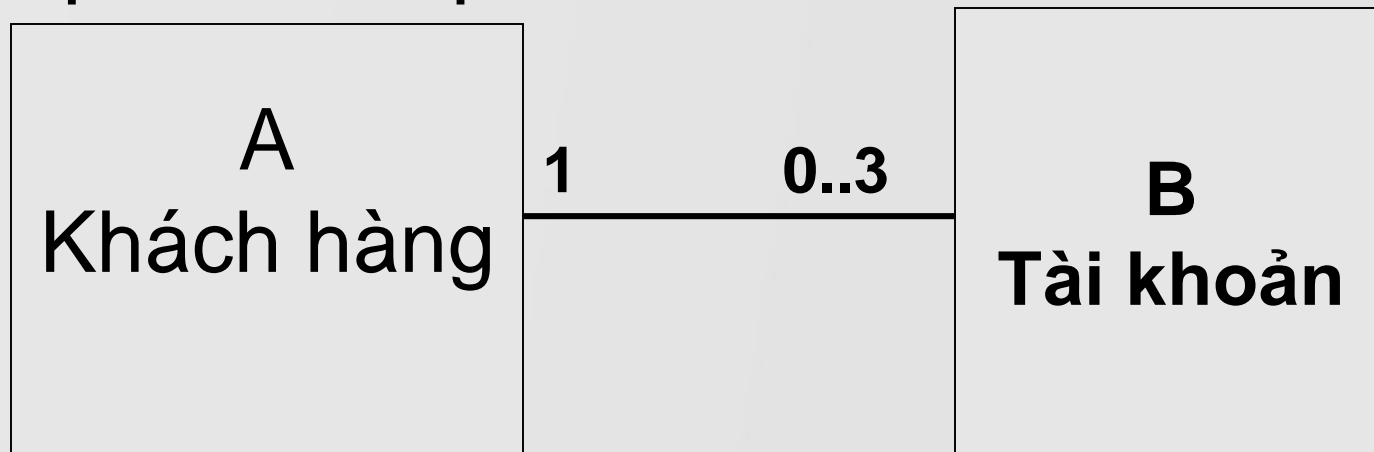


Một phần tử lớp A có 1 phần tử lớp B

Bản số (Multiplicity)



- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



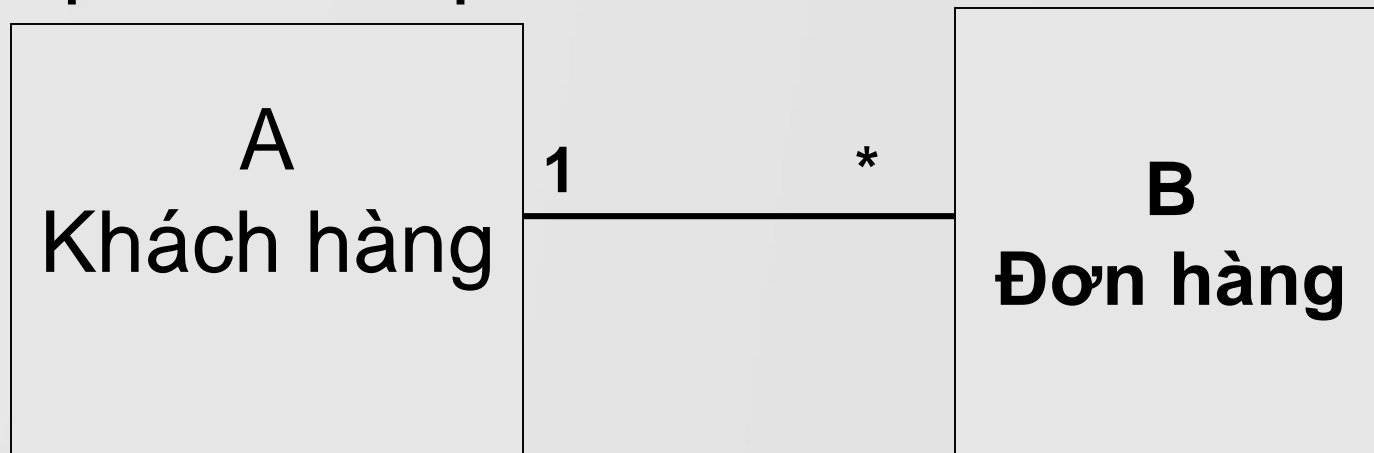
Một phần tử lớp A có tối đa 3 phần tử lớp B

Mỗi phần tử lớp B có đúng 1 phần tử lớp A

Bản số (Multiplicity)



- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



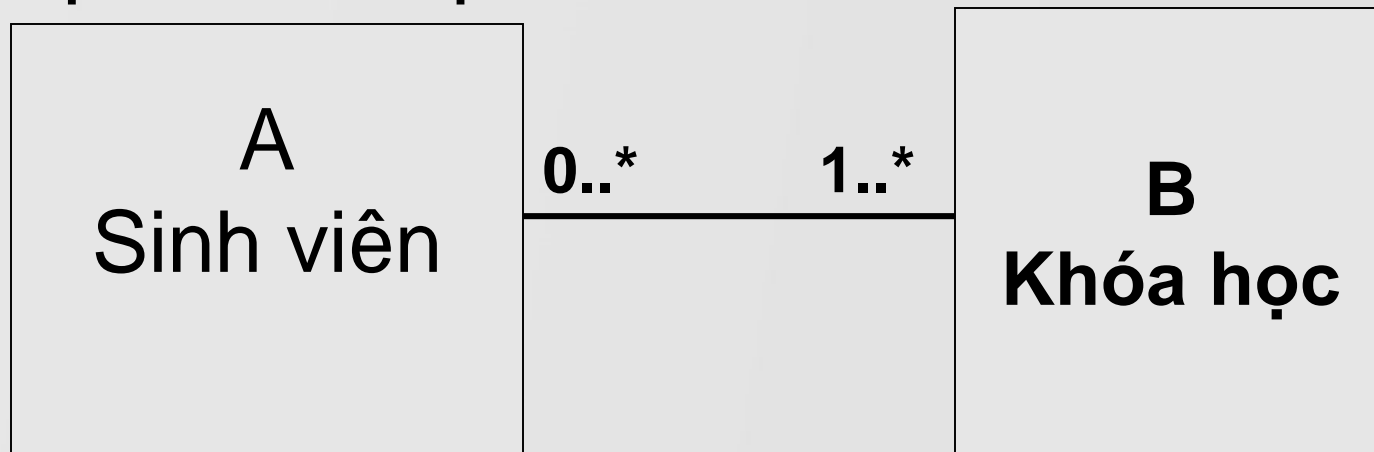
Một phần tử lớp A có nhiều phần tử lớp B

Mỗi phần tử lớp B có đúng 1 phần tử lớp A

Bản số (Multiplicity)



- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?

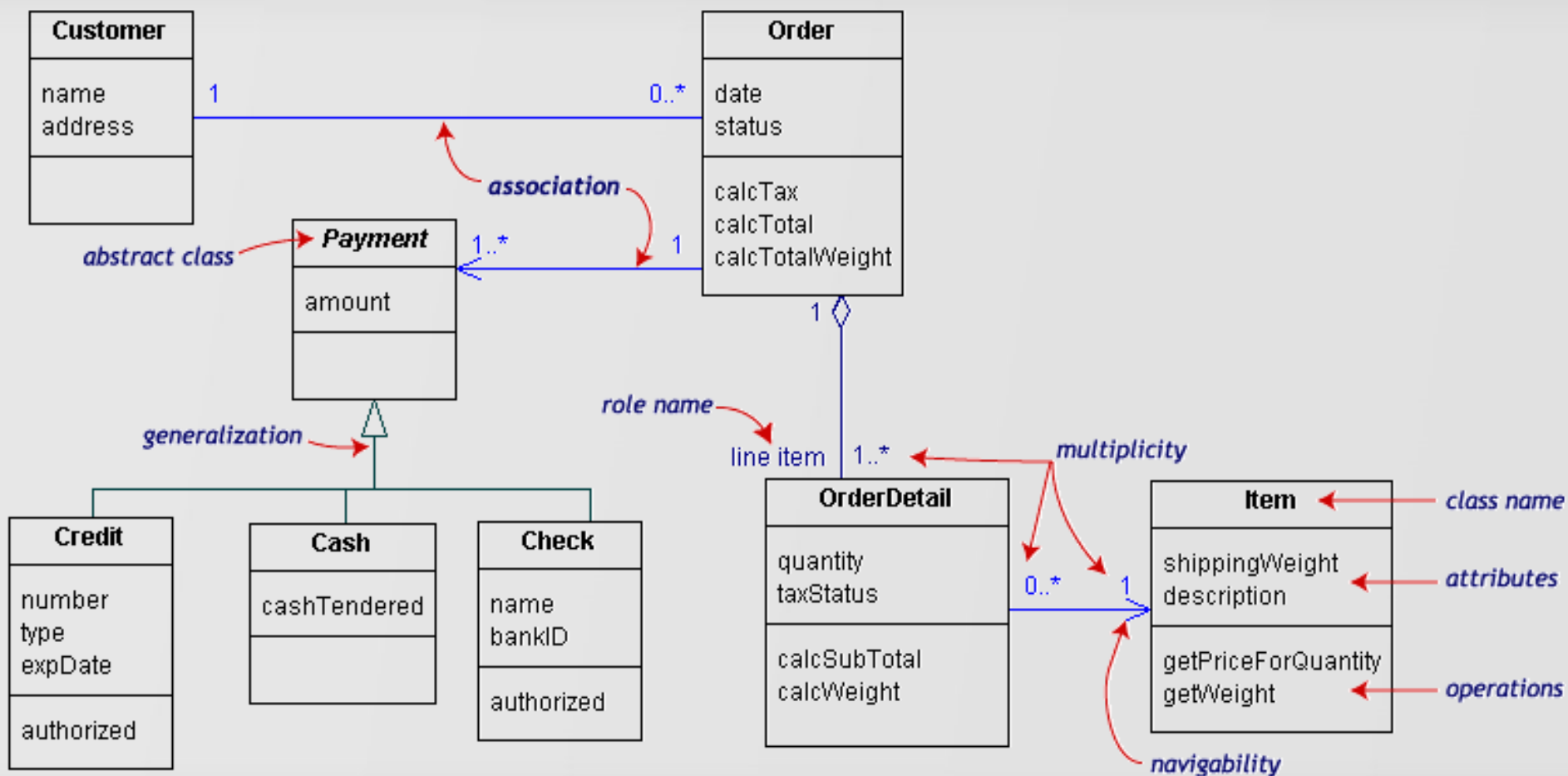


Mỗi sinh viên tham gia ít nhất 1 khóa học

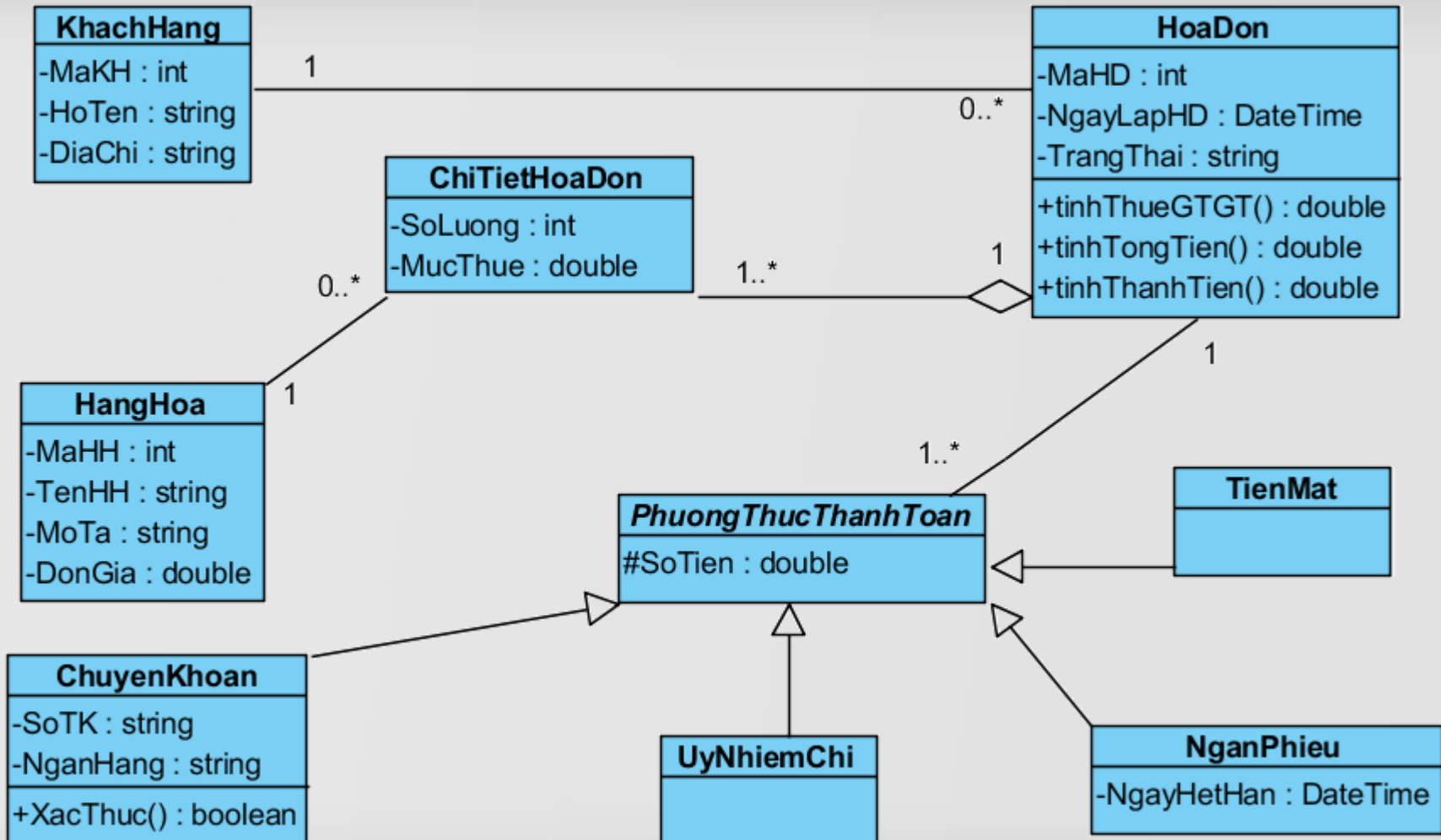
Mỗi khóa học có thể có 0 hoặc nhiều sv tham gia



Ví dụ



Ví dụ





Xây dựng sơ đồ lớp ở mức phân tích

CLASS DIAGRAM

Class Diagram



- Được xây dựng và hiệu chỉnh trong suốt quá trình phát triển
- Mục tiêu:
 - Đặt tên và lập mô hình các khái niệm trong hệ thống
 - Đặc tả sự cộng tác
 - Đặc tả sơ đồ CSDL
- Được phát triển bởi **phân tích viên, thiết kế viên, lập trình viên**

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
- ...

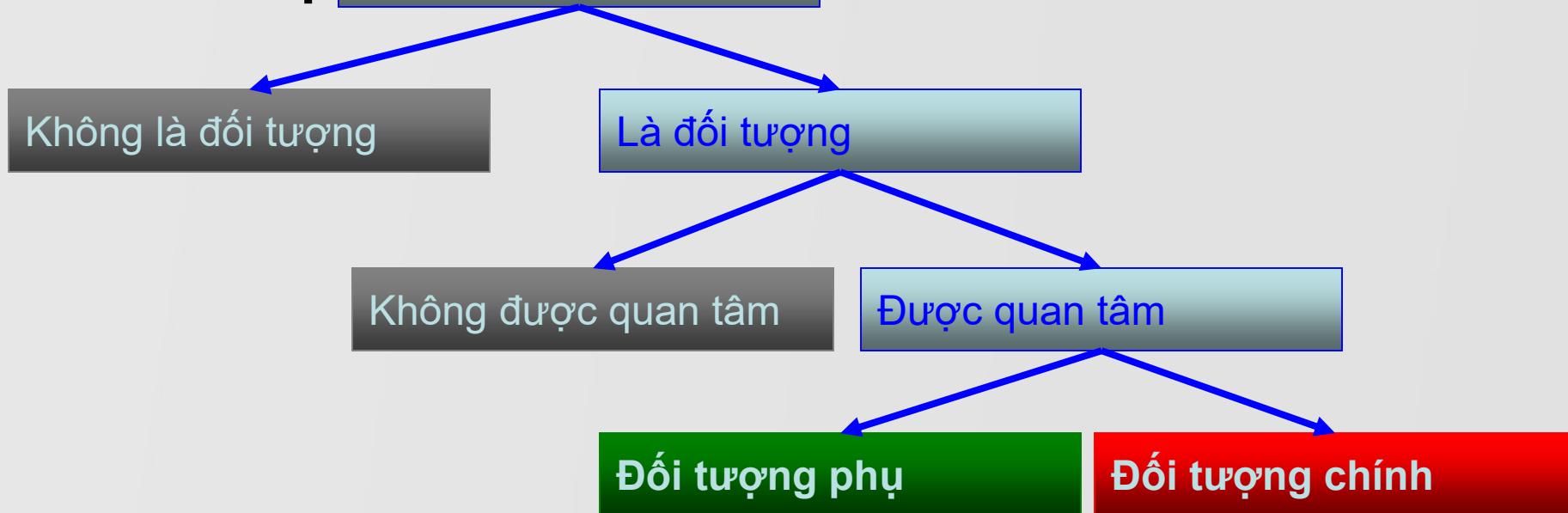
- **Đề nghị:**

- 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 **Đối tượng đề nghị**



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



Ví dụ

- 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 viên	BGH	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?**

Ví dụ



- **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...



Ví dụ

- **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 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)
- Bước 6:
 - Bổ sung các trách nhiệm (phương thức) vào các lớp đối tượng ở mức phân tích



Kết quả

- 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
...

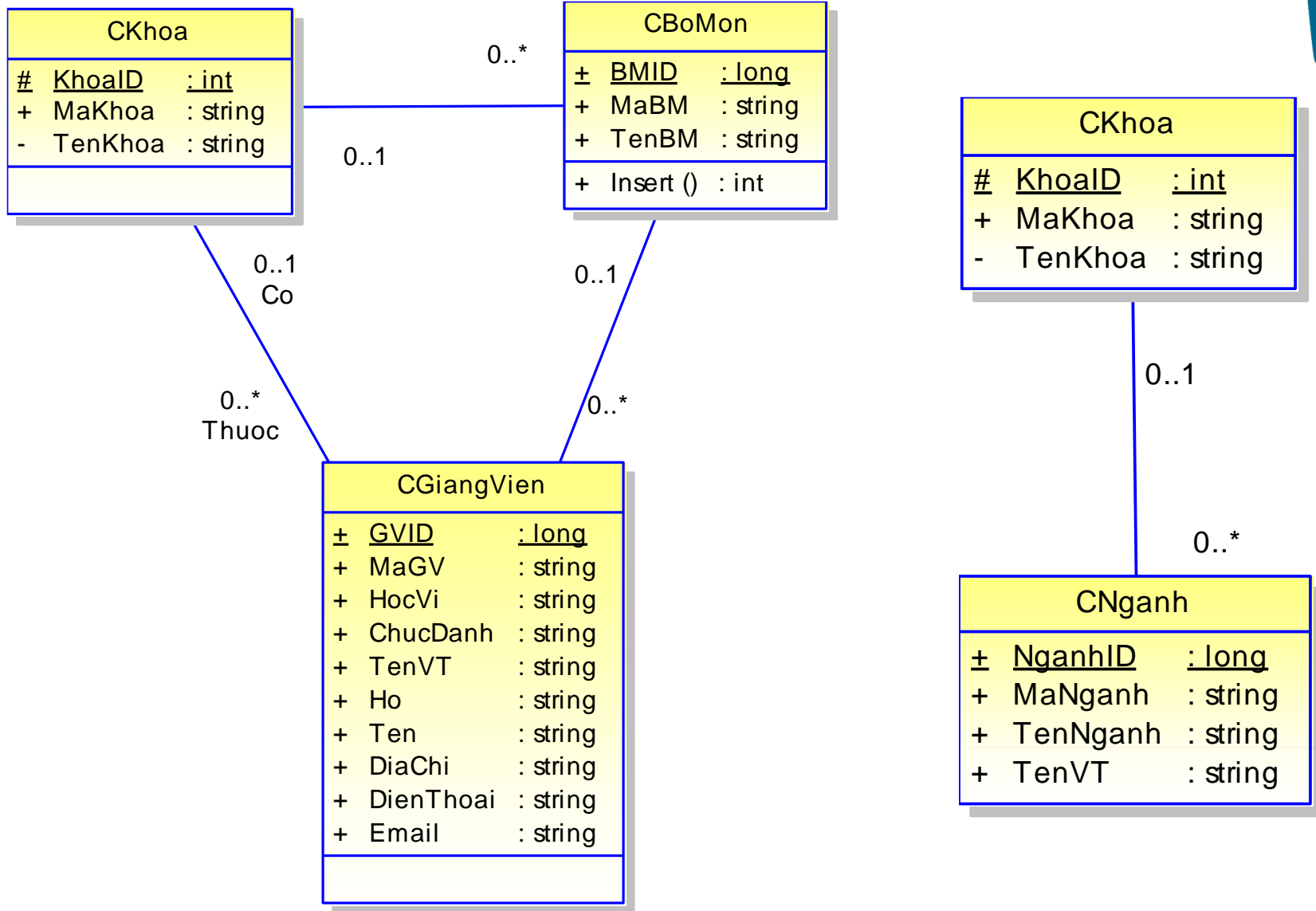
Áp dụng



- 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

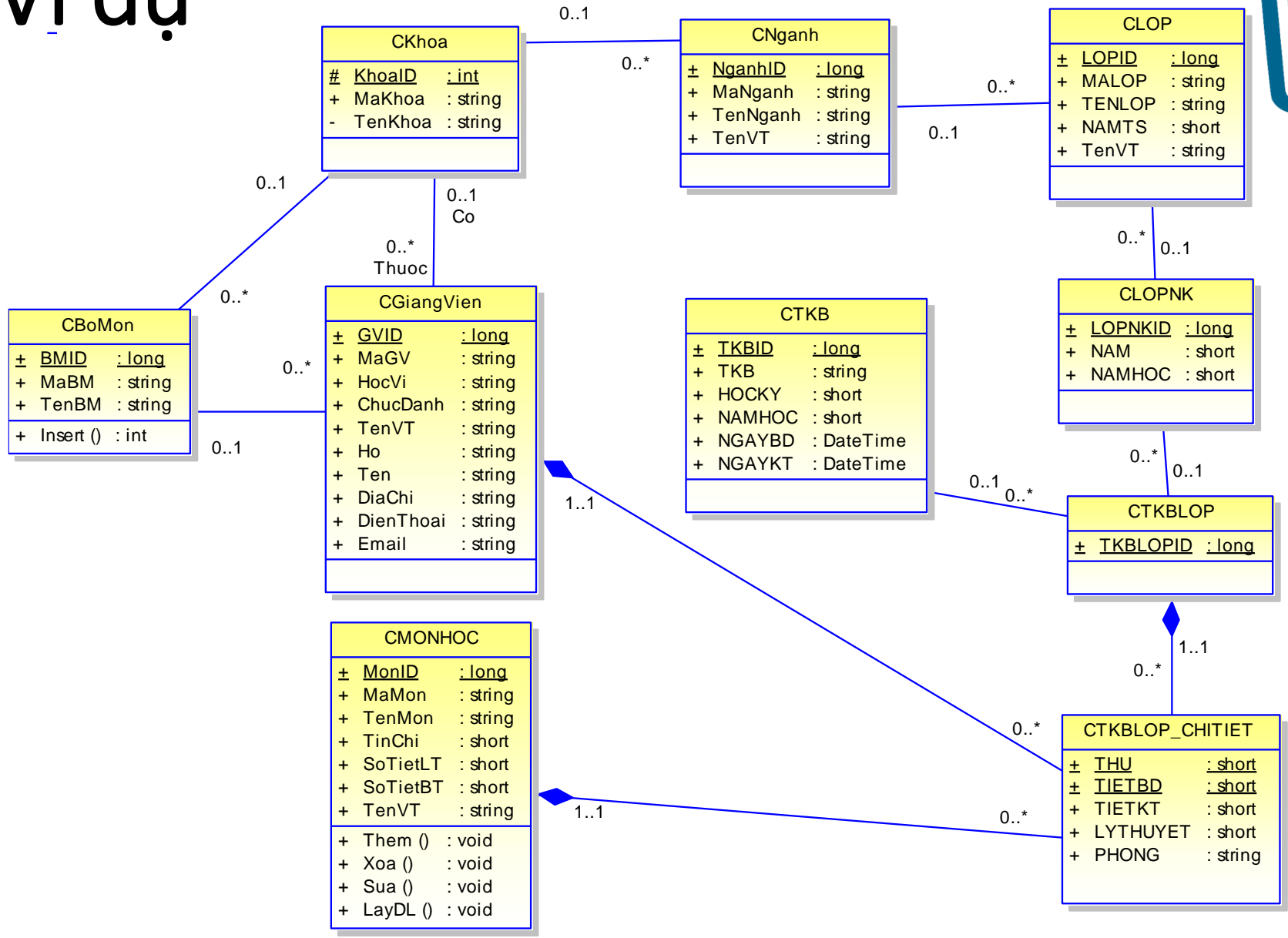


Ví dụ

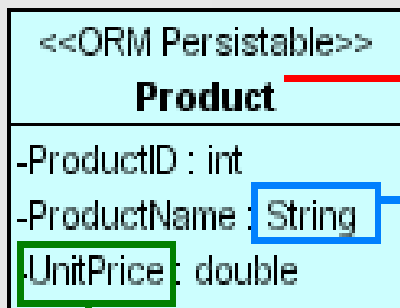




Ví dụ

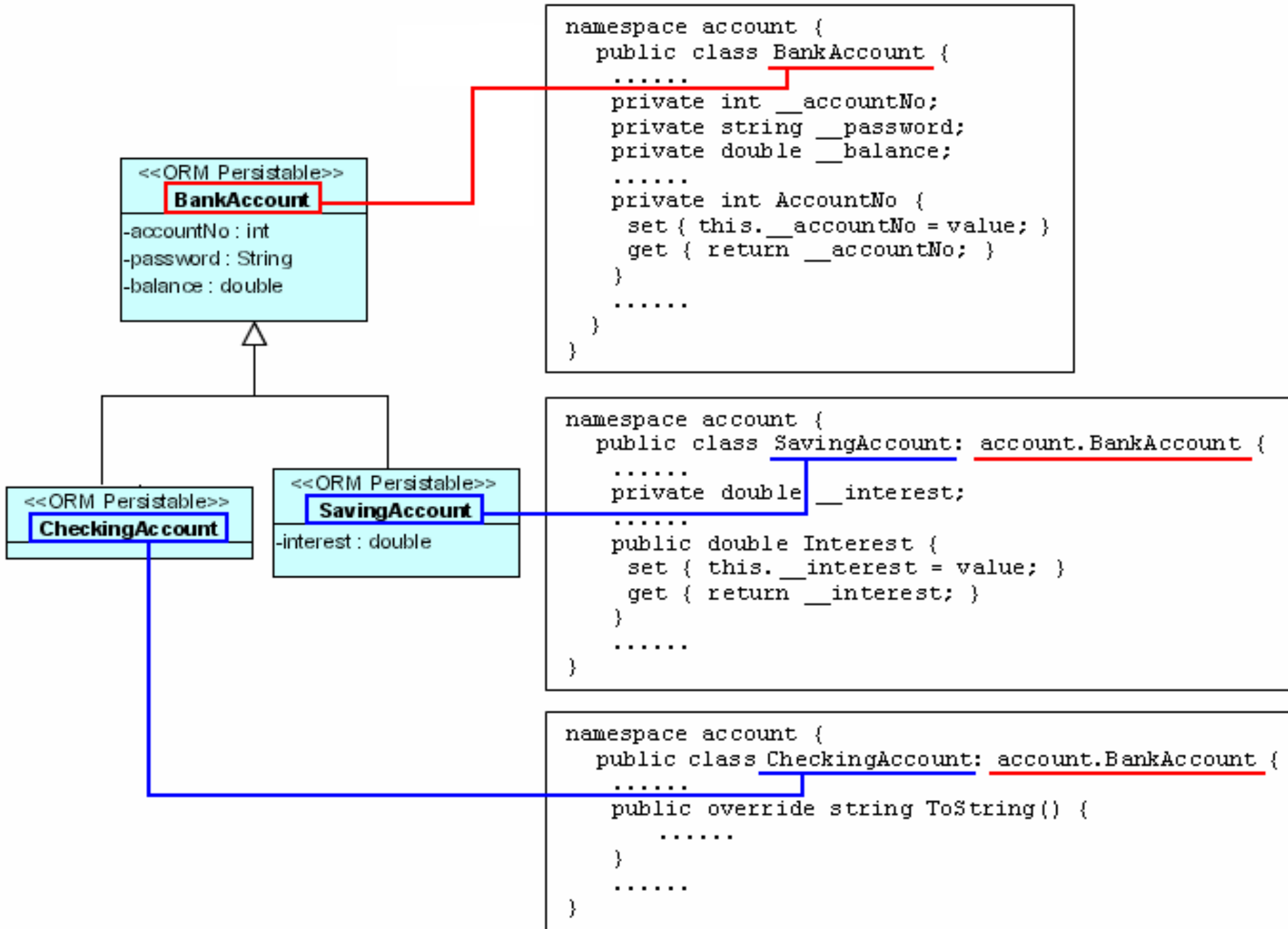


Ánh xạ biểu đồ sang Code



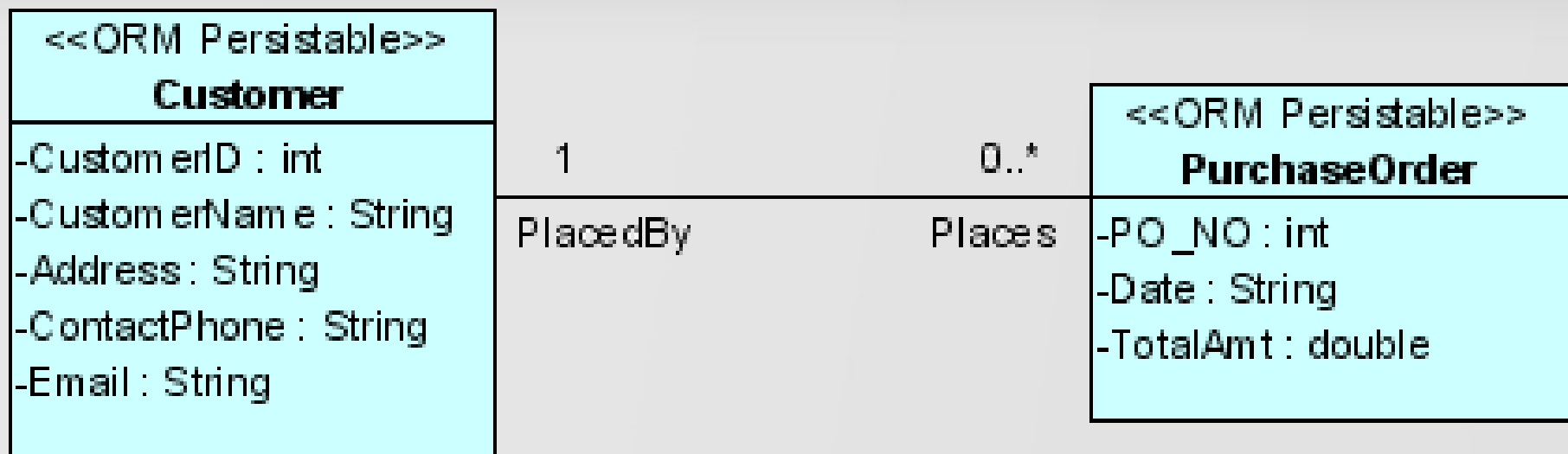
```
/// <summary>  
/// ORM-Persistable Class  
/// <summary>  
public class Product {  
    .....  
    private int __ProductID;  
    private string __ProductName;  
    private double __UnitPrice;  
    .....  
}
```

- Mapping Classes, Attributes and Data Type

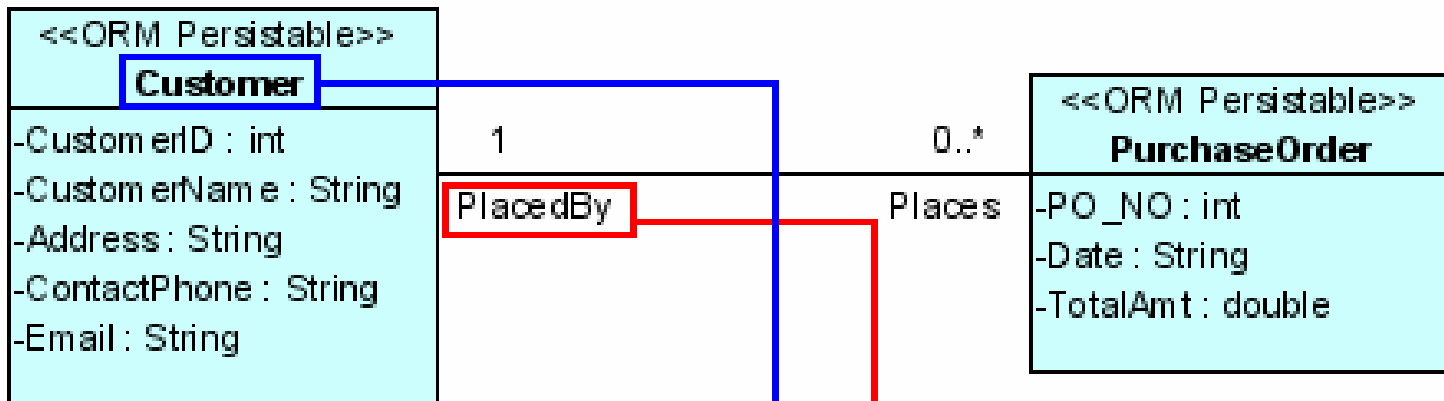


Mapping generalization

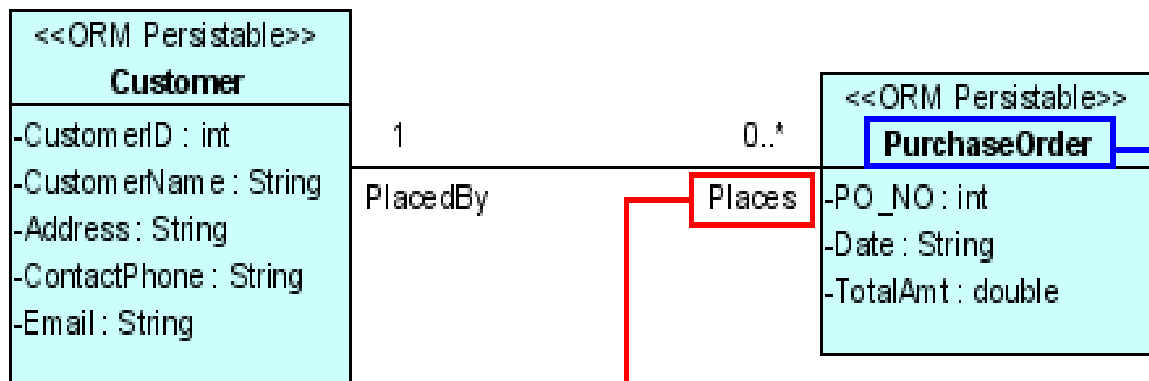
Ảnh xạ khách hàng- đơn hàng



The Classes with association and multiplicity



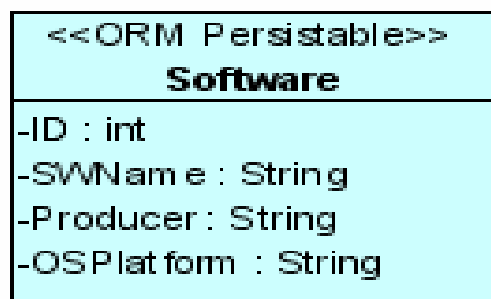
```
public class PurchaseOrder {  
    .....  
    private int __PO_NO;  
    private string __Date;  
    private double __TotalAmt;  
    private shoppingcart.Customer __PlacedBy;  
    .....  
    public shoppingcart.Customer PlacedBy {  
        set { value.Places.Add(this); }  
        get { return __PlacedBy; }  
    }  
    .....  
}
```



```
public class Customer {
    .....
    private int __CustomerID;
    private string __CustomerName;
    private string __Address;
    private string __ContactPhone;
    private string __Email;
    .....
    public readonly
        shoppingcart.PurchaseOrderSetCollection Places;
    .....
}
```

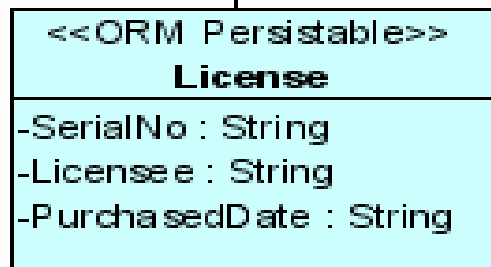
```
public class PurchaseOrderSetCollection
: Orm.Util.ORMSet {
    .....
    public void Add(PurchaseOrder value) {
        ..... }
    public void Remove(PurchaseOrder value) {
        ..... }
    public bool Contains(PurchaseOrder value) {
        ..... }
    .....
}
```

Mapping the One-to-many Associations

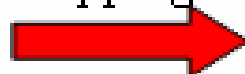


1 belongsTo

1 contains



Mapping



```
public class Software{
    private int __ID;
    private string __SWName;
    private string __Producer;
    private string __OSPlatform;
    private License __contains;
    .....
    public License Contains{
        set {
            ..... }
        get {
            ..... }
        .....
    }
}
```

```
public class License{
    private string __SerialNo;
    private string __Licensee;
    private string __PurchasedDate;
    private Software __belongsTo;
    .....
    public Software BelongsTo{
        set {
            ..... }
        get {
            ..... }
        .....
    }
}
```

Mapping the One-to-one Relationship



Sơ đồ trạng thái

STATECHART DIAGRAM

Statechart Diagram



- Ghi nhận các hành vi động (even – oriented)
- Mục tiêu:
 - Mô hình hóa chu kỳ sống của đối tượng
 - Mô hình hóa các đối tượng phản hồi (user interfaces, devices, ...)

Các khái niệm



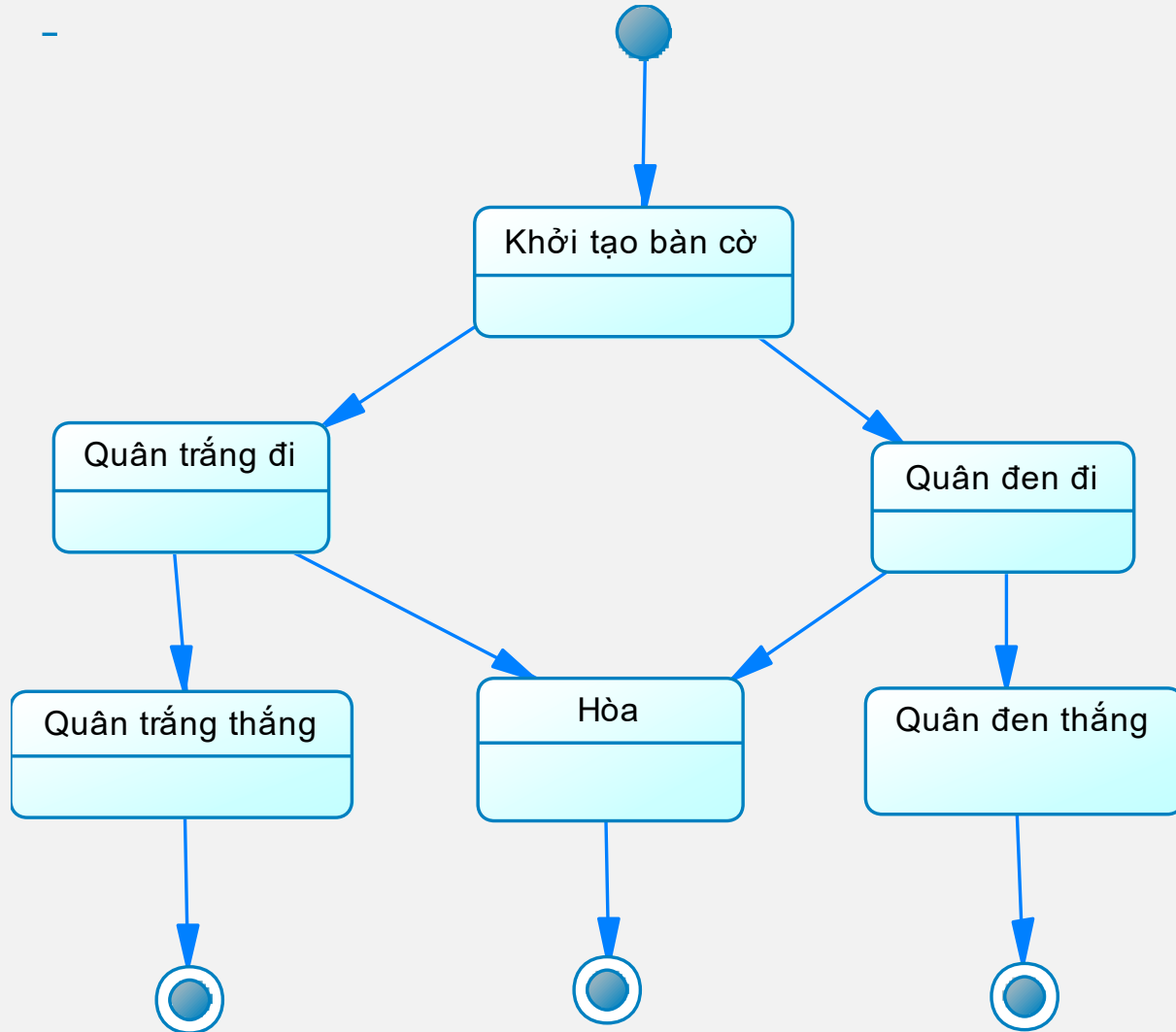
- Sơ đồ trạng thái

- 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.

State Diagram Bàn cờ



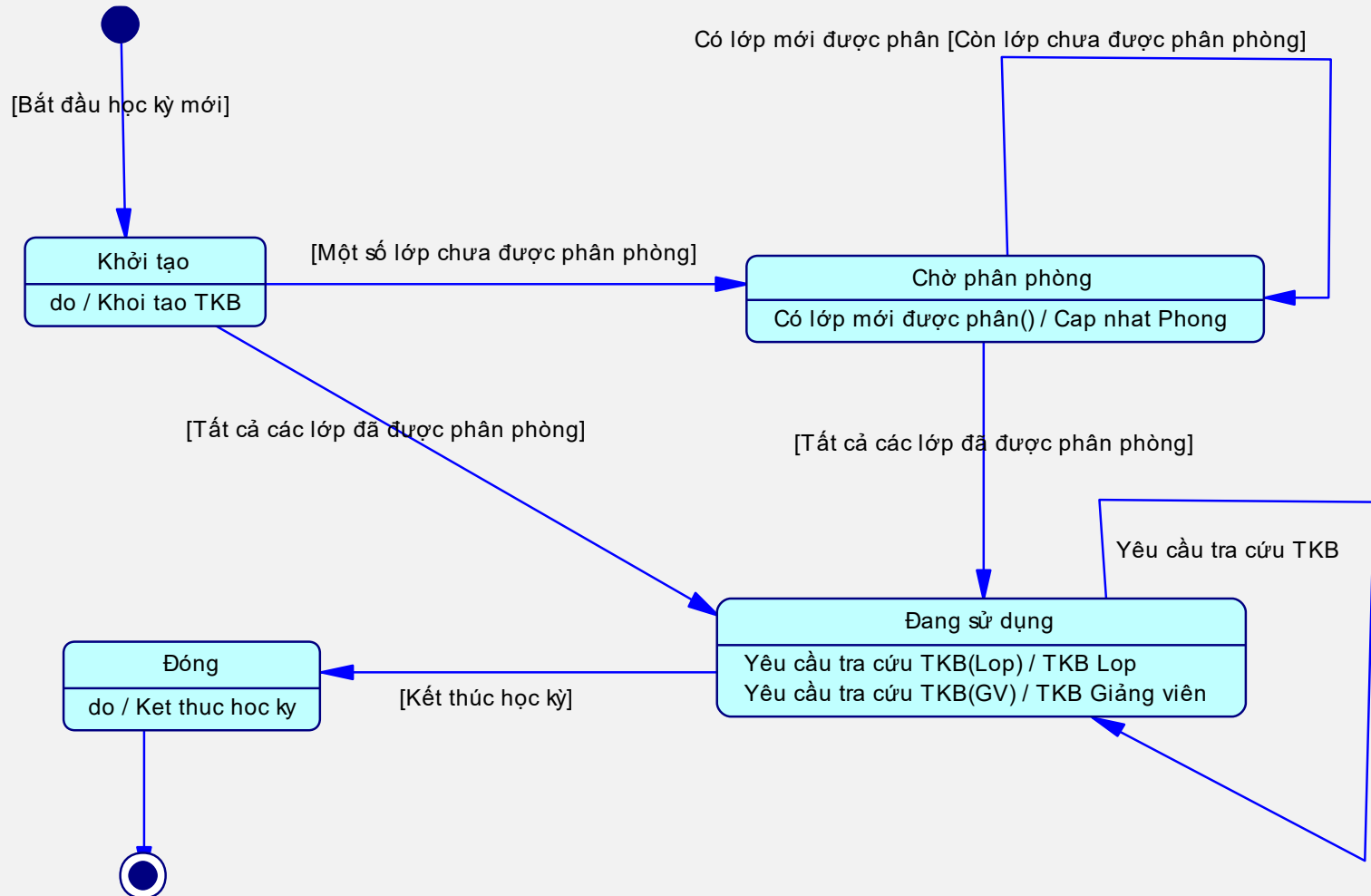
Ví dụ: Xét ứng dụng quản lý TKB



- Hệ thống cho phép bộ phận giáo vụ Khoa nhập thời khóa biểu, cập nhật thời khóa biểu của tất cả các lớp
- Hệ thống cho phép sinh viên tra cứu thời khóa biểu theo từng học kỳ của lớp
- Hệ thống cho phép giảng viên tra cứu thời khóa biểu giảng dạy trong học kỳ

Ví dụ minh họa

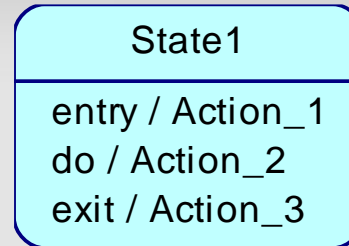
- State Diagram mô tả trạng thái TKB





Các thành phần trong State diagram

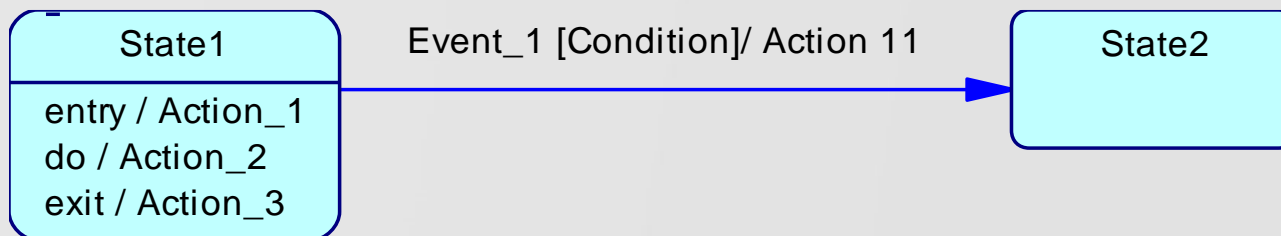
- Trạng thái – State



- Sự kiện – Event

- Hành động – Action

- Mối liên hệ giữa các 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

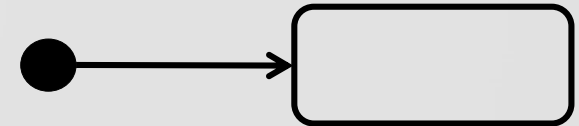
Các trạng thái đặc biệt



• 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

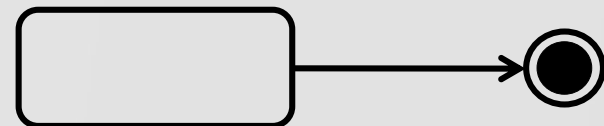


• Trạng thái kết thúc

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



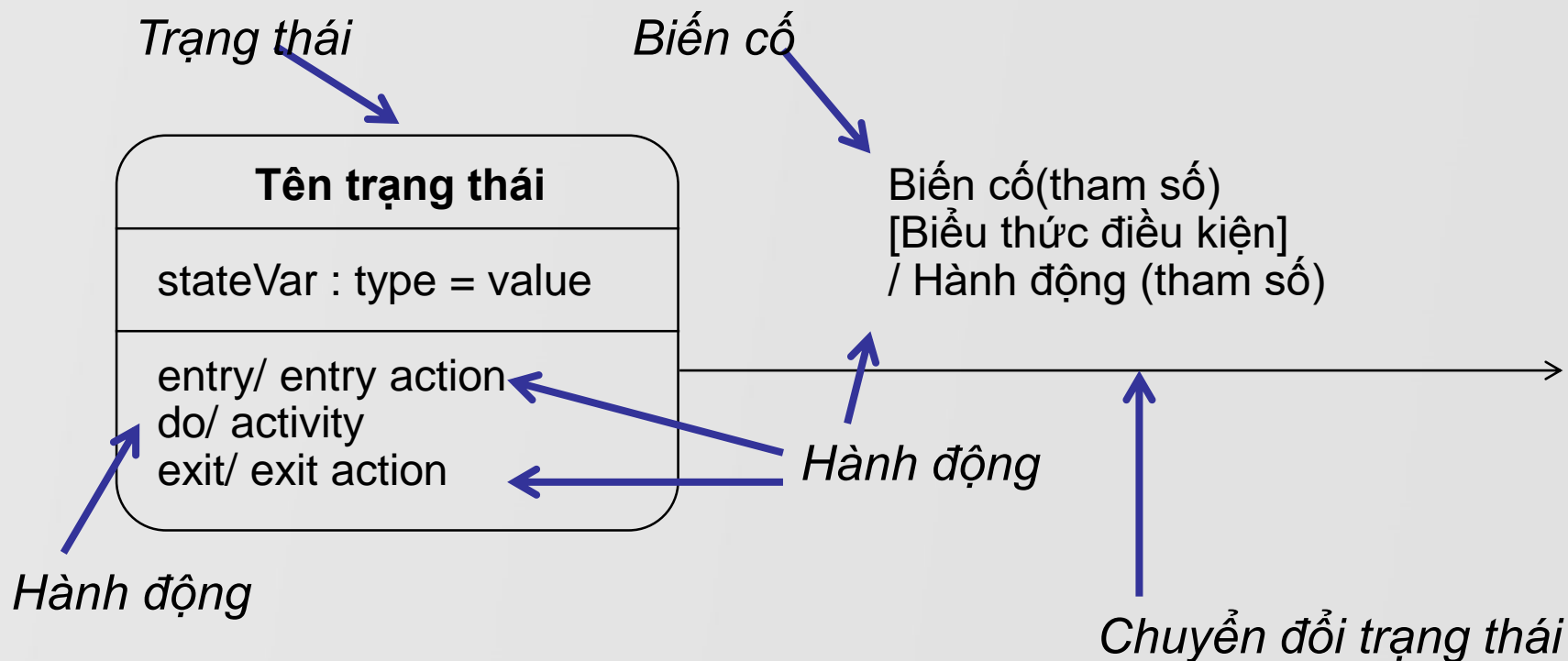
Tr/thái kết thúc





Sơ đồ trạng thái

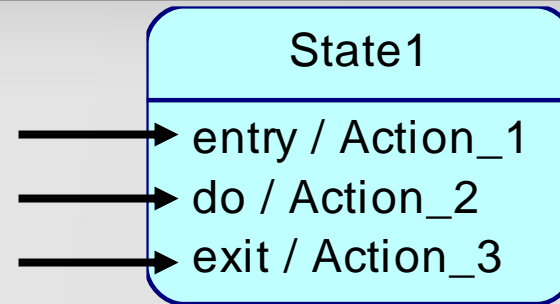
- Là đồ thị có hướng với các node là các trạng thái nối với nhau bởi các cung mô tả việc chuyển đổi trạng thái



Các thành phần trong State diagram



- Sự kiện – **Event**



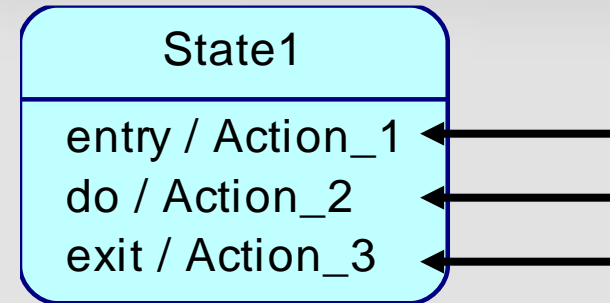
- Event

- **Entry**: sự kiện phát sinh khi đối tượng bắt đầu nhận trạng thái
- **Exit**: sự kiện phát sinh khi đối tượng kết thúc trạng thái thúc trạng thái
- **Do**: sự kiện phát sinh khi user thực hiện một hành động thông qua bàn phím/chuột.

Các thành phần trong State diagram



- Trạng thái – **State**



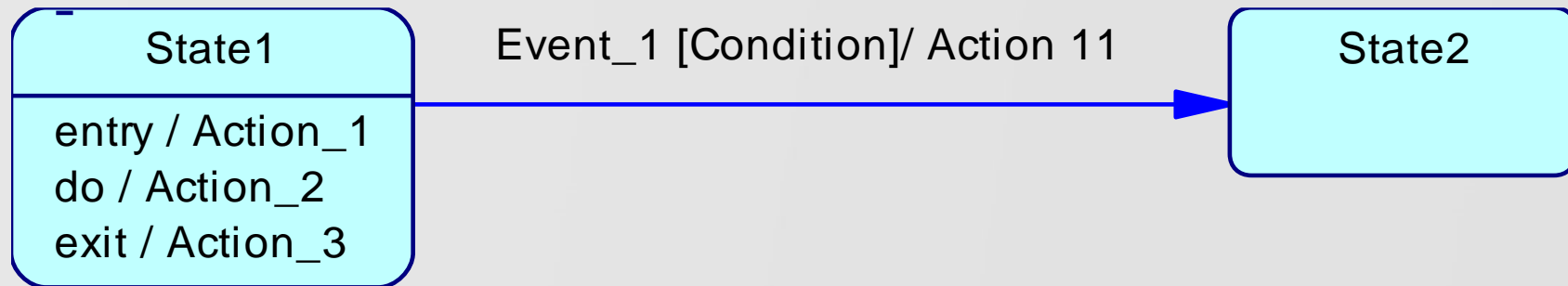
- Action

- **Entry**: hành động được thực hiện khi đối tượng bắt đầu trạng thái
- **Do**: tập các hành động có thể thực hiện với trạng thái
- **Exit**: hành động được thực hiện khi đối tượng kết thúc trạng thái



Các thành phần trong State diagram

- Mỗi liên hệ giữa các trạng thái –Transition



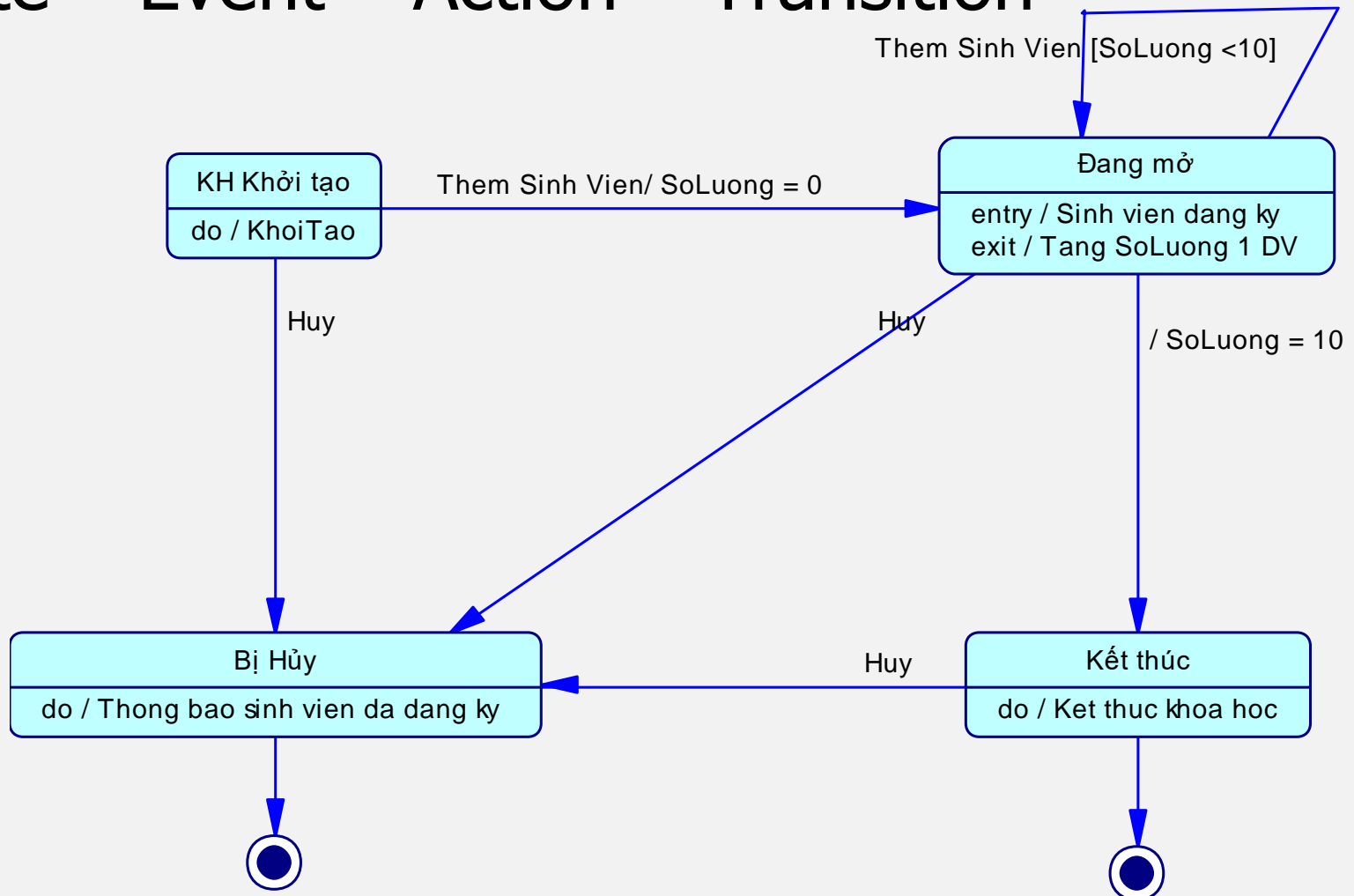
– Event

– Action

– **Condition**: điều kiện cho phép chuyển từ trạng thái này sang trạng thái khác

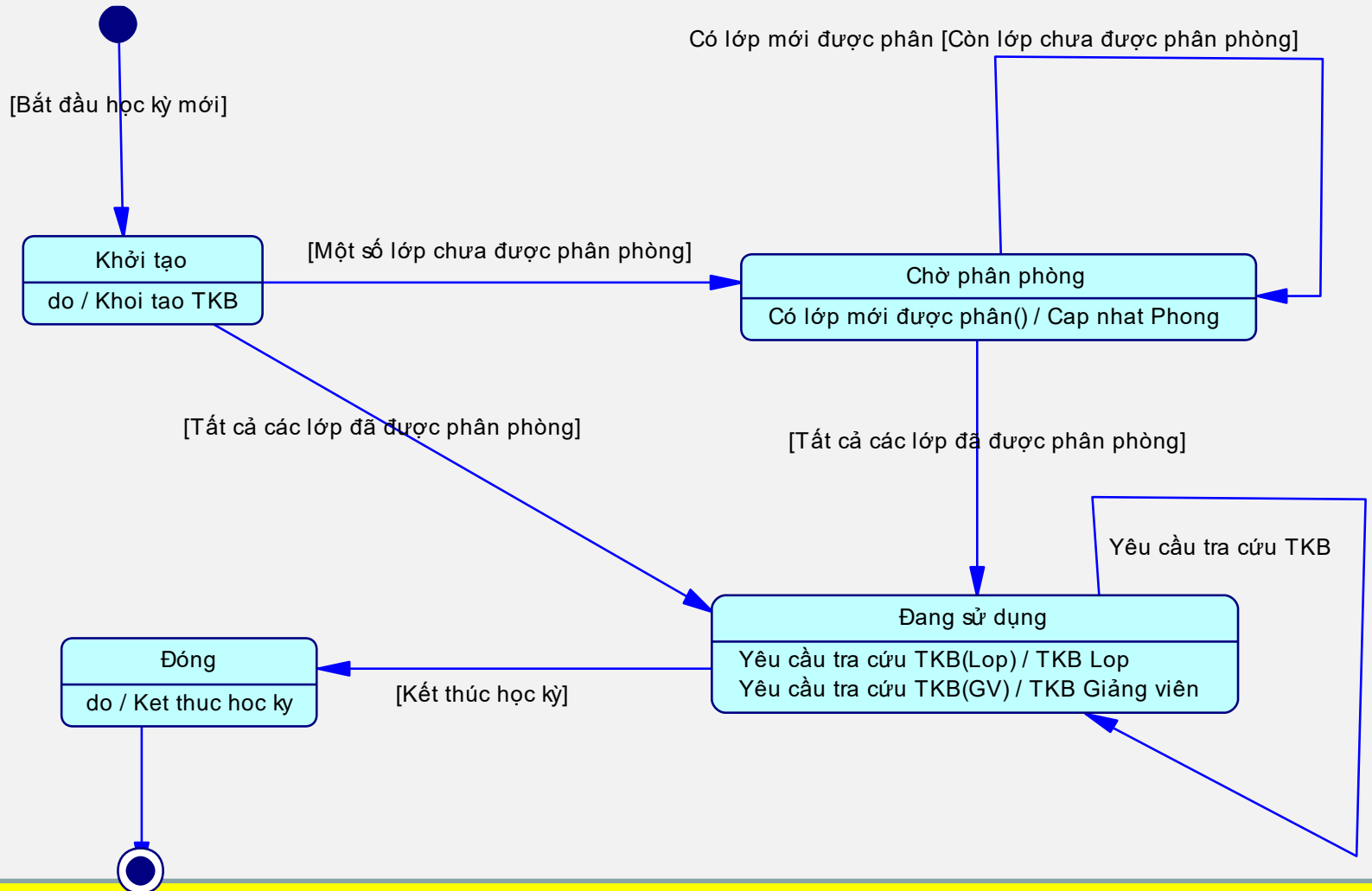
Các thành phần trong State diagram

- State – Event – Action – Transition



Ví dụ minh họa

- State Diagram mô tả trạng thái TKB



Ví dụ minh họa

- **State Diagram** mô tả trạng thái màn hình quản lý danh mục người dùng

The screenshot shows a Windows-style application window titled "Quan ly Nguoi dung". It contains a form for adding or editing user information and a table listing existing users.

Thông tin người dùng

Ma	<input type="text" value="2"/>	Ho ten	<input type="text" value="Quan tri he thong"/>
Ten nguoi dung	<input type="text" value="admin"/>	Dia chi	<input type="text" value="123 Truong"/>
Mat khau	<input type="text" value="admin"/>	Email	<input type="text" value="admin@yahoo.com"/>

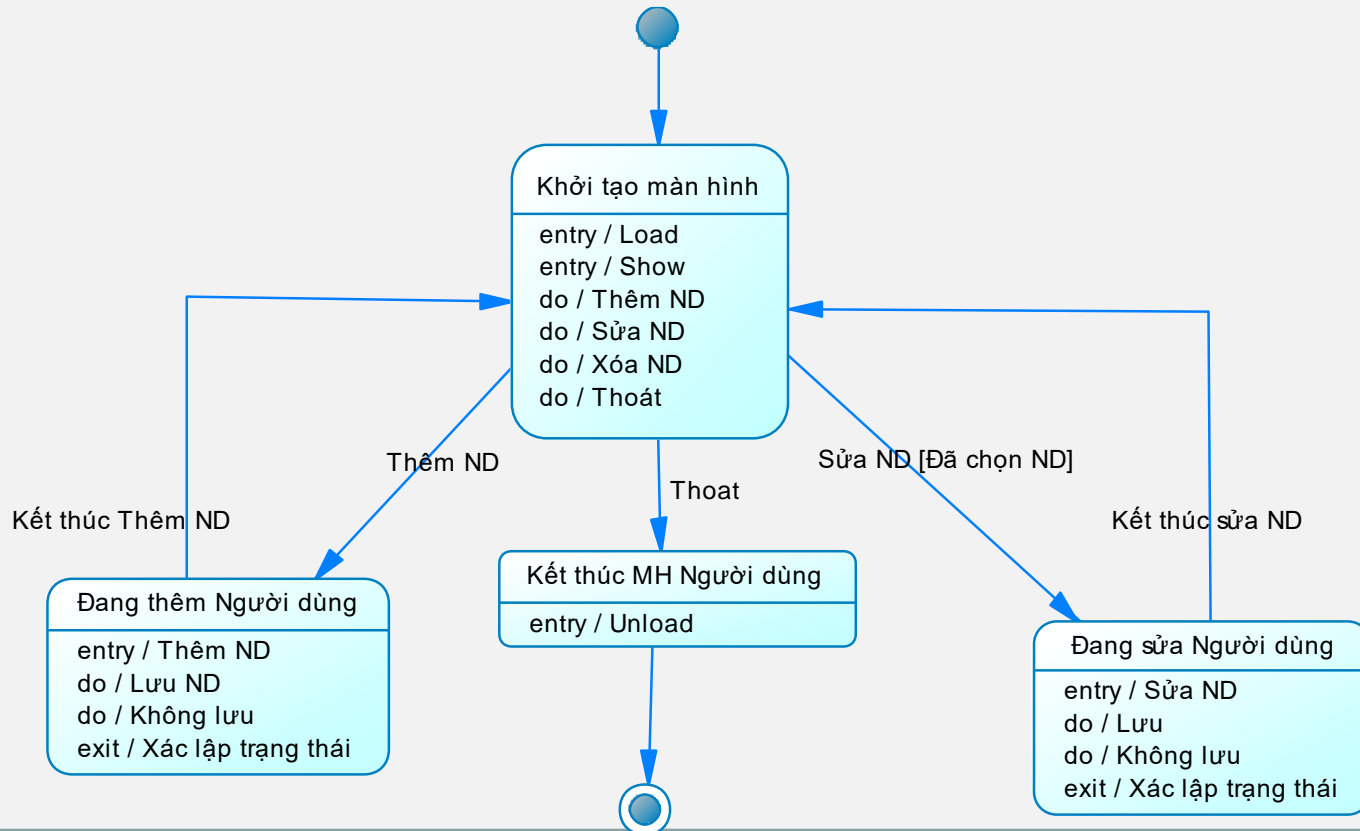
Danh sach nguoi dung

Ma ND	Ten ND	Mat Khau	Ho Ten	Dia Chi	Email
4	SV1	555555	Doan Van	i Hong Bang	@gmail.com
2	admin	admin	Quan tri he t	123 Truong	admin@yahi
3	Hienlth	hienlth	DHSP	uong Vuong	dhsp@yaho

Buttons: Them, Xoa, Sua, Ghi, Khong, Thoat

Ví dụ minh họa

- **State Diagram** mô tả trạng thái màn hình quản lý danh mục người dùng



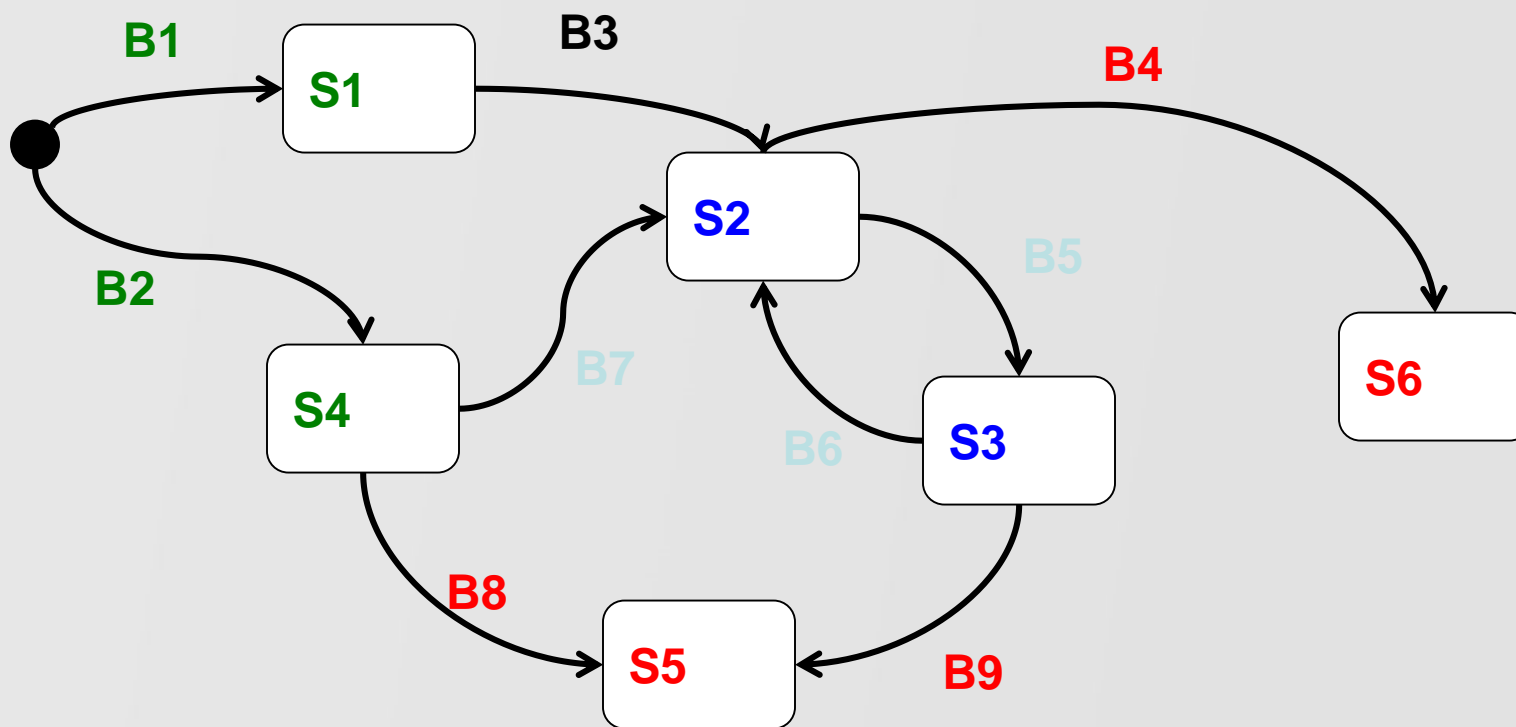


Sơ đồ trạng thái

B1, B2: biến cố sinh

B4, B8, B9: biến cố mất

B3, B5, B6, B7: biến cố hoạt động



Sơ đồ trạng thái

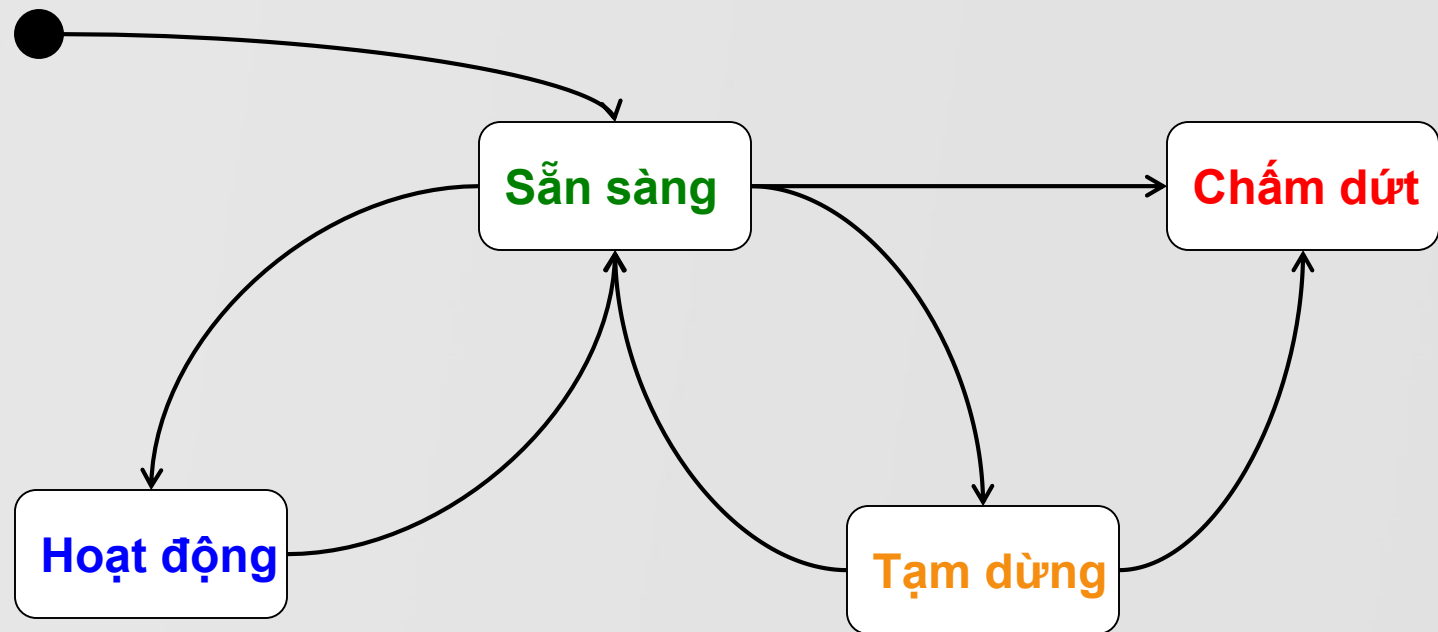


- Mỗi sơ đồ trạng thái sẽ ánh xạ thành **một thuộc tính** của đối tượng
- Các trạng thái của đối tượng trong 1 sơ đồ trạng thái **không giao nhau** và giá trị của thuộc tính tương ứng chỉ mang 1 trong các giá trị tương ứng
- Sơ đồ trạng thái càng chi tiết sẽ phục vụ:
 - Có những xử lý thích hợp trong 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

Sơ đồ trạng thái



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



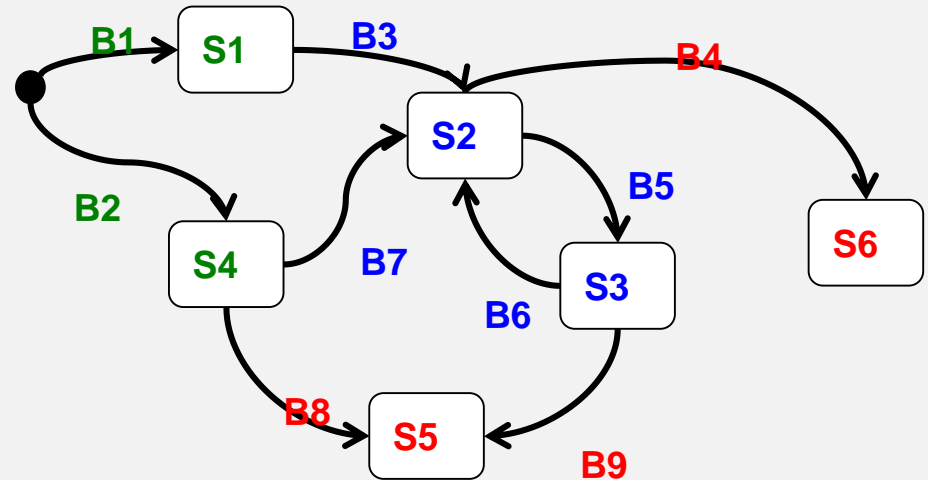
Cách xây dựng sơ đồ trạng thái



- 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
...

Tạo State Diagram trong PD



- Tạo mới State Diagram
- State
 - Start
 - End
 - State
- Transition
- Action
- Event

Bài tập



- 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

Bài tập



Vẽ sơ đồ trạng thái:

- Quản lý giải bóng đá vô địch quốc gia
- Quản lý bình chọn bài hát hay Làn sóng xanh
- Quản lý giáo vụ trường đại học
- Quản lý giáo vụ trường phổ thông
- Quản lý bán hàng
- Quản lý nhân sự - Tiền lương

Ôn tập



- State diagram là gì ?
- Event là gì ?
- Transition ?
- Action là gì ? Các loại Action ?
- Xây dựng State diagram cho một số ứng dụng trong phần bài tập

Câu hỏi và thảo luận

