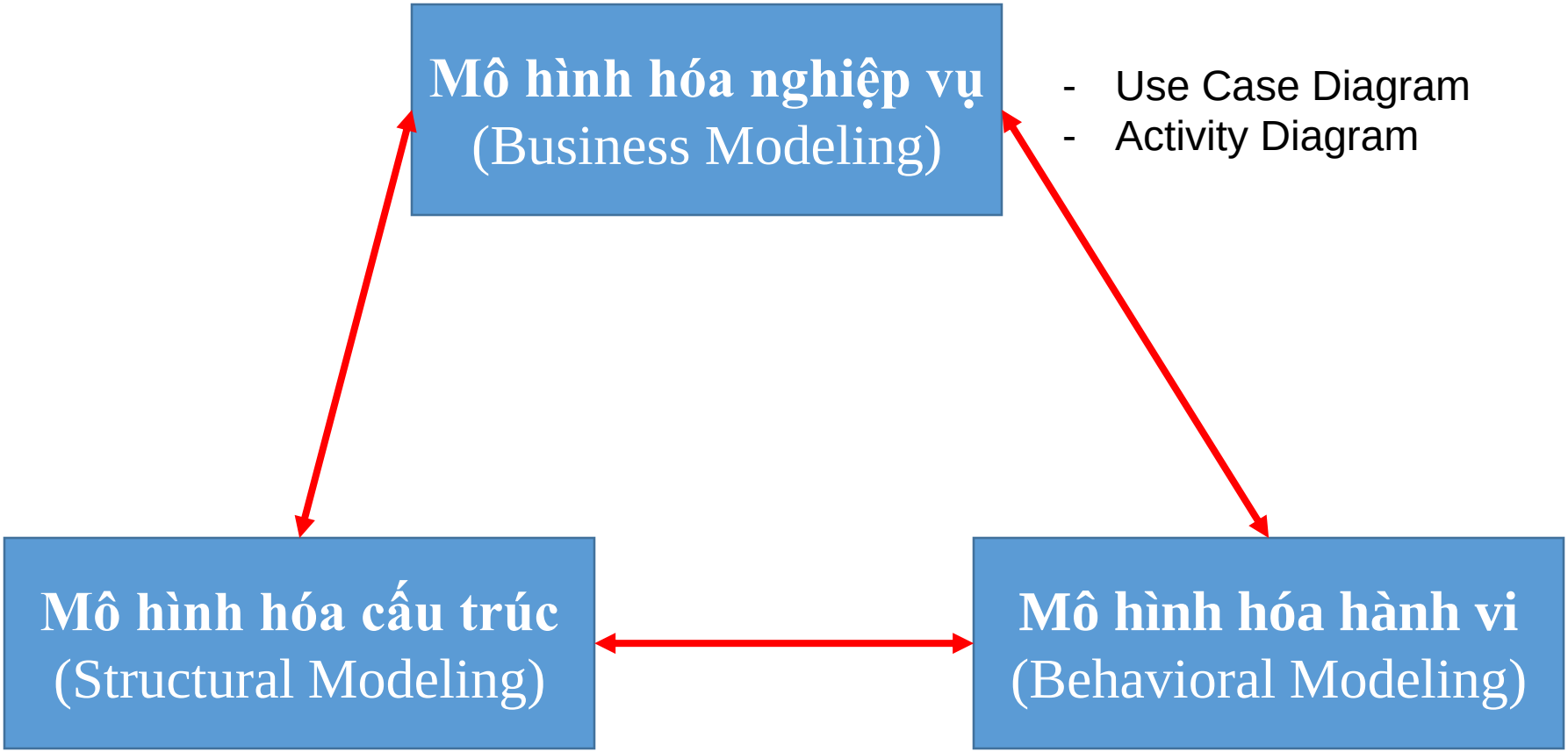
HIENLTH

Chủ đề 5: Mô hình hóa cấu trúc

Mô hình hóa nghiệp vụ



Mục đích của mô hình cấu trúc:

- Mô tả cấu trúc của dữ liệu được sử dụng trong hệ thống.
- 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

Các mô hình cấu trúc:

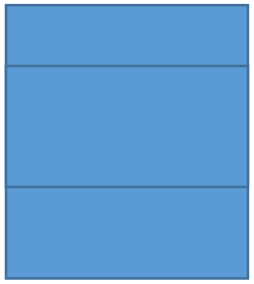
- *CRC cards*, *class diagrams*, *object diagrams*.

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 mướn, 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.
- Cú 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

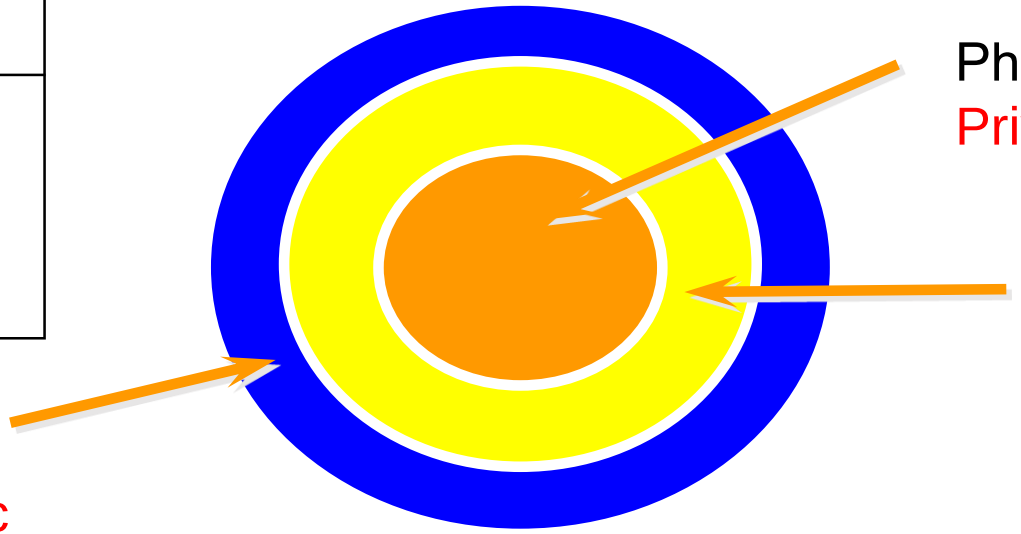
Tên class

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()



Tầm vực

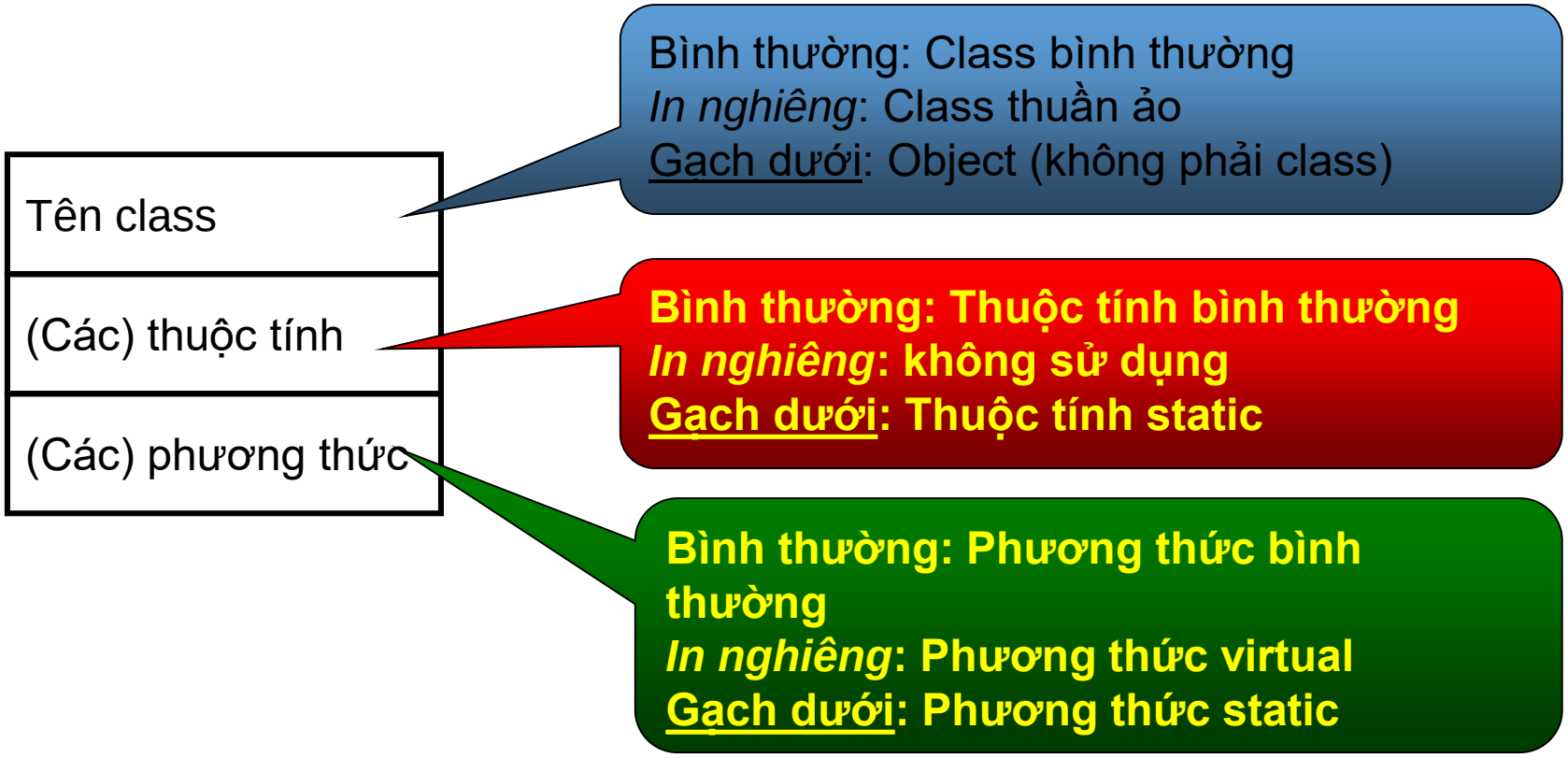
- Xác định số lượng thể hiện của thuộc tính / phương thức

Class
- classifierScopeAttribute - instanceScopeAttribute
classifierScopeOperation() 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

Các giai đoạn của mô hình hóa đối tượng bằng biểu đồ lớp

- Tìm kiếm các lớp
- Xác định liên kết giữa các lớp
- Xác định các thuộc tính
- Tổ chức và đơn giản hóa các lớp bằng cách sử dụng quan hệ thừa kế
- Xóa các liên kết thừa
- Kiểm tra xem biểu đồ đã bao gồm tất cả các yêu cầu của tài liệu hay chưa?
- Lặp lại và làm mịn mô hình
- Nhóm các lớp thành các modules (gói)

Tips

- 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

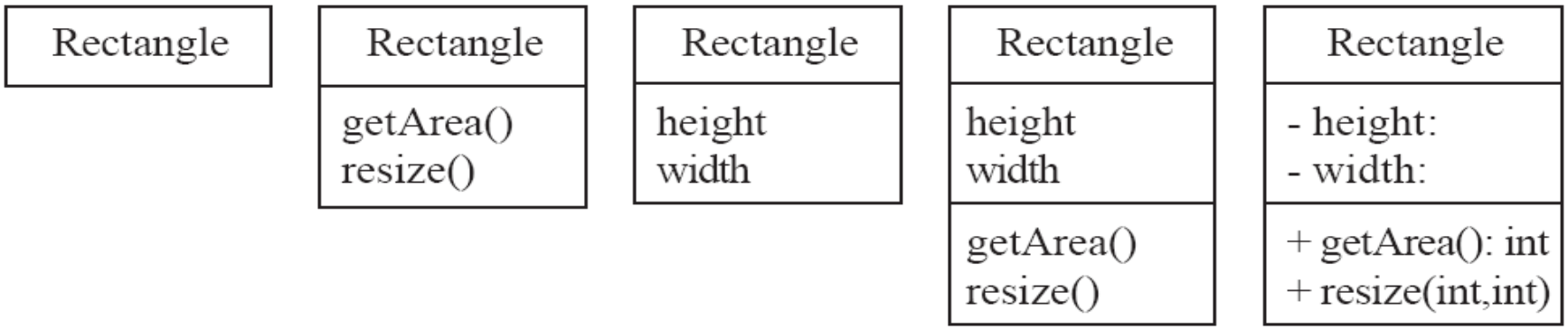
Tìm kiếm lớp như thế nào?

- 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...

Biểu diễn lớp trong UML

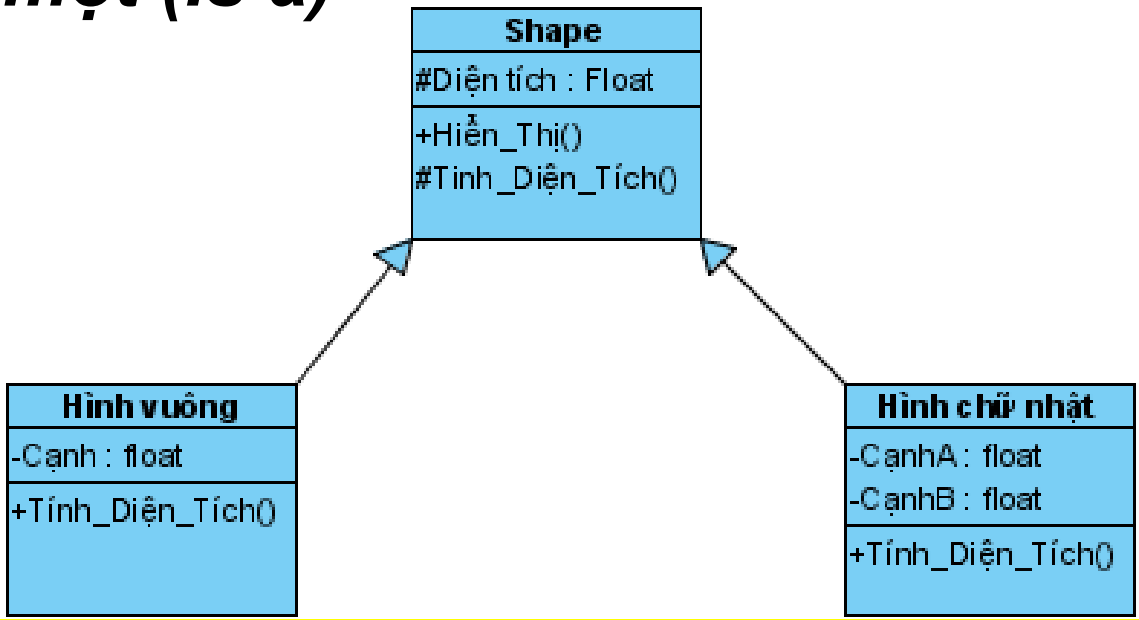


Mối quan hệ giữa các class

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

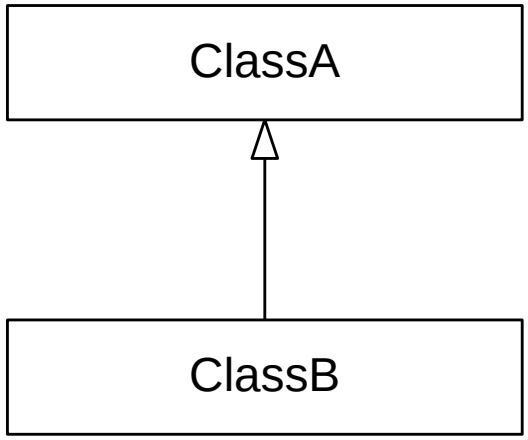
Các quan hệ trong biểu đồ lớp

- 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:



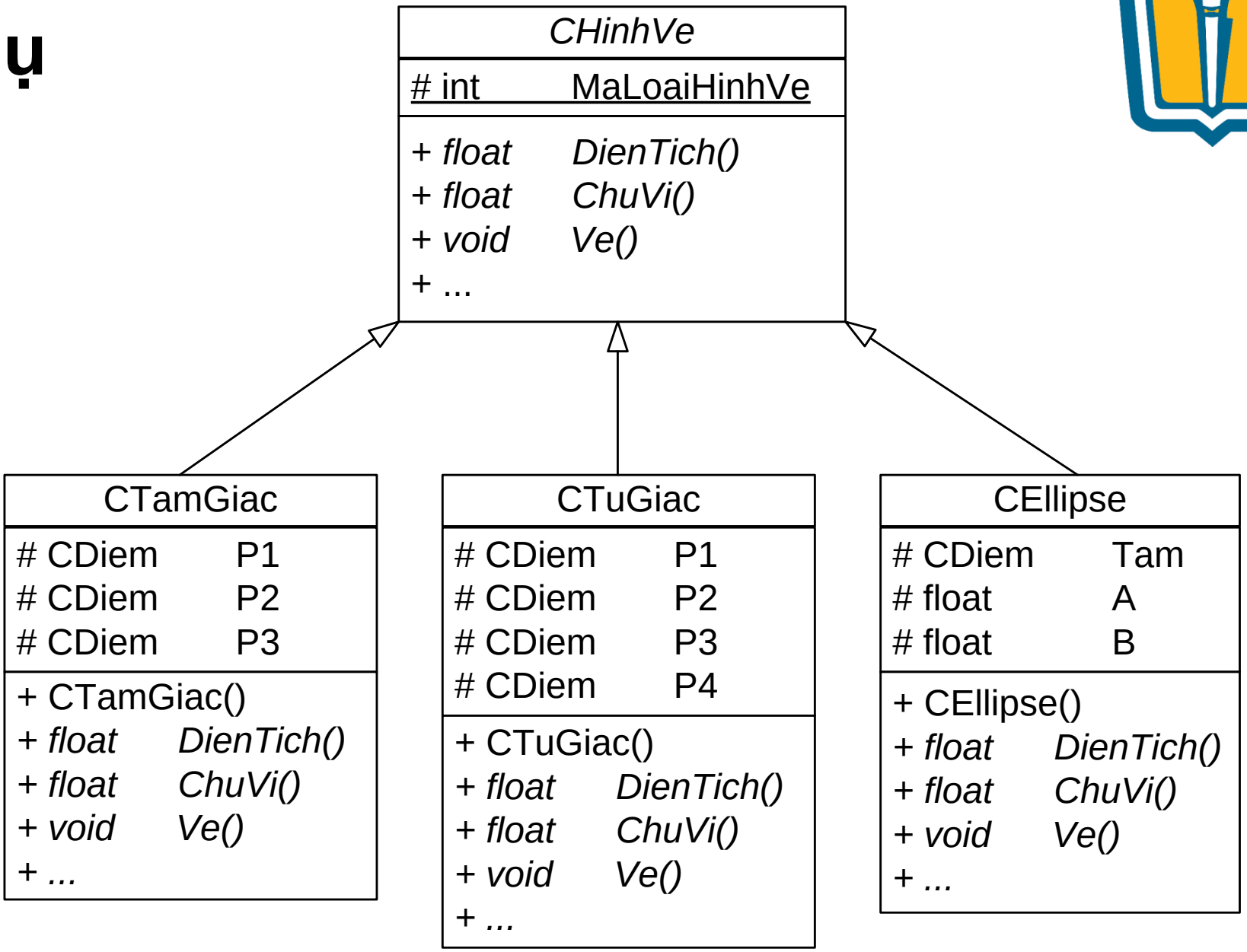
Quan hệ giữa các lớp đối tượng

- Quan hệ kế thừa



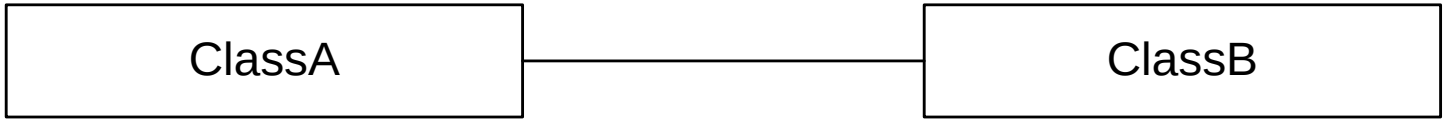
- ClassB kế thừa từ ClassA
- ClassB là một trường hợp đặc biệt của ClassA
- ClassA là trường hợp tổng quát của ClassB

Ví dụ


CHinhVe
int MaLoaiHinhVe
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CTamGiac
CDiem P1
CDiem P2
CDiem P3
+ CTamGiac()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CTuGiac
CDiem P1
CDiem P2
CDiem P3
CDiem P4
+ CTuGiac()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CEllipse
CDiem Tam
float A
float B
+ CEllipse()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...

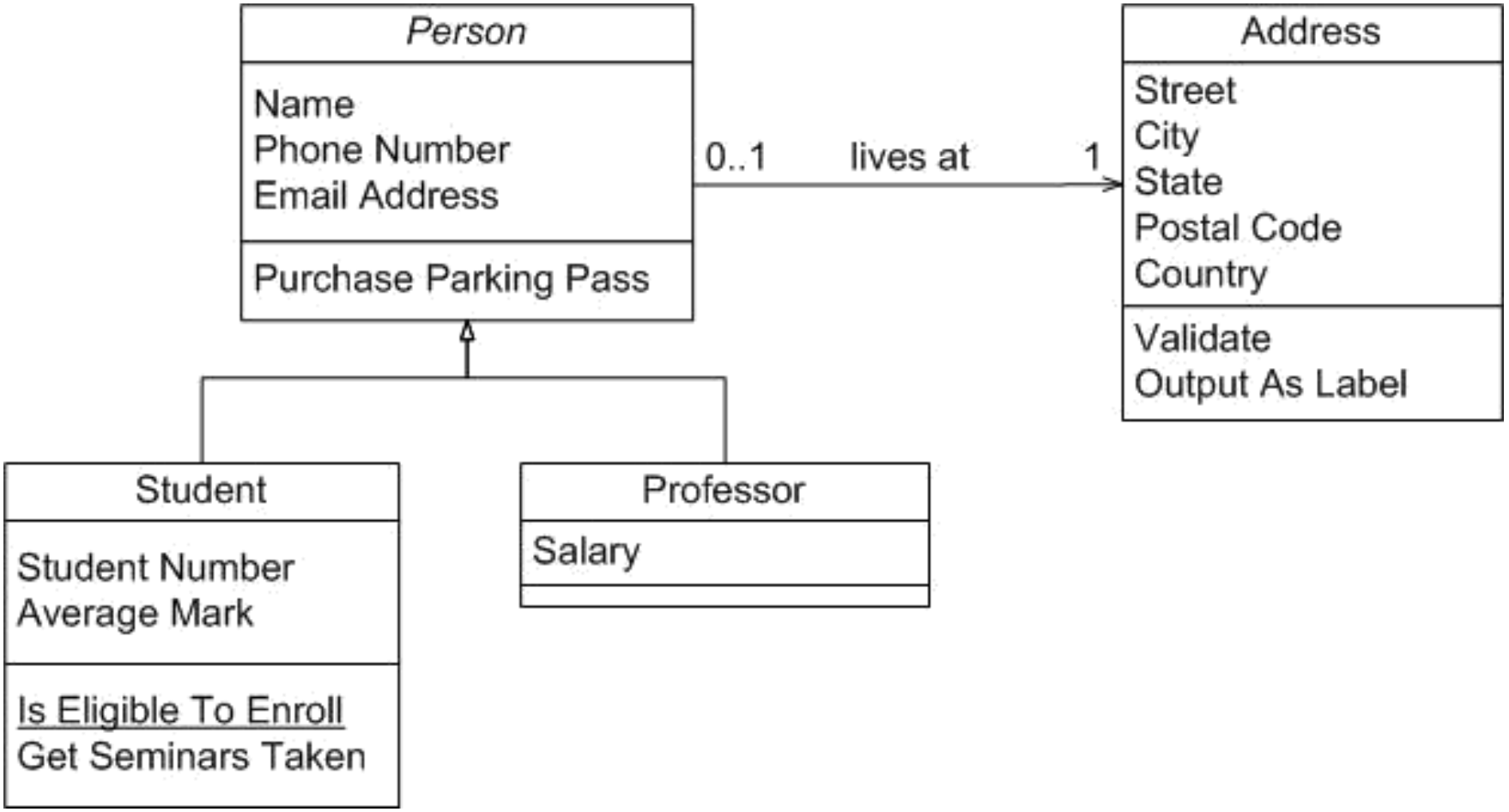
Quan hệ giữa các lớp đối tượng

- Quan hệ Association

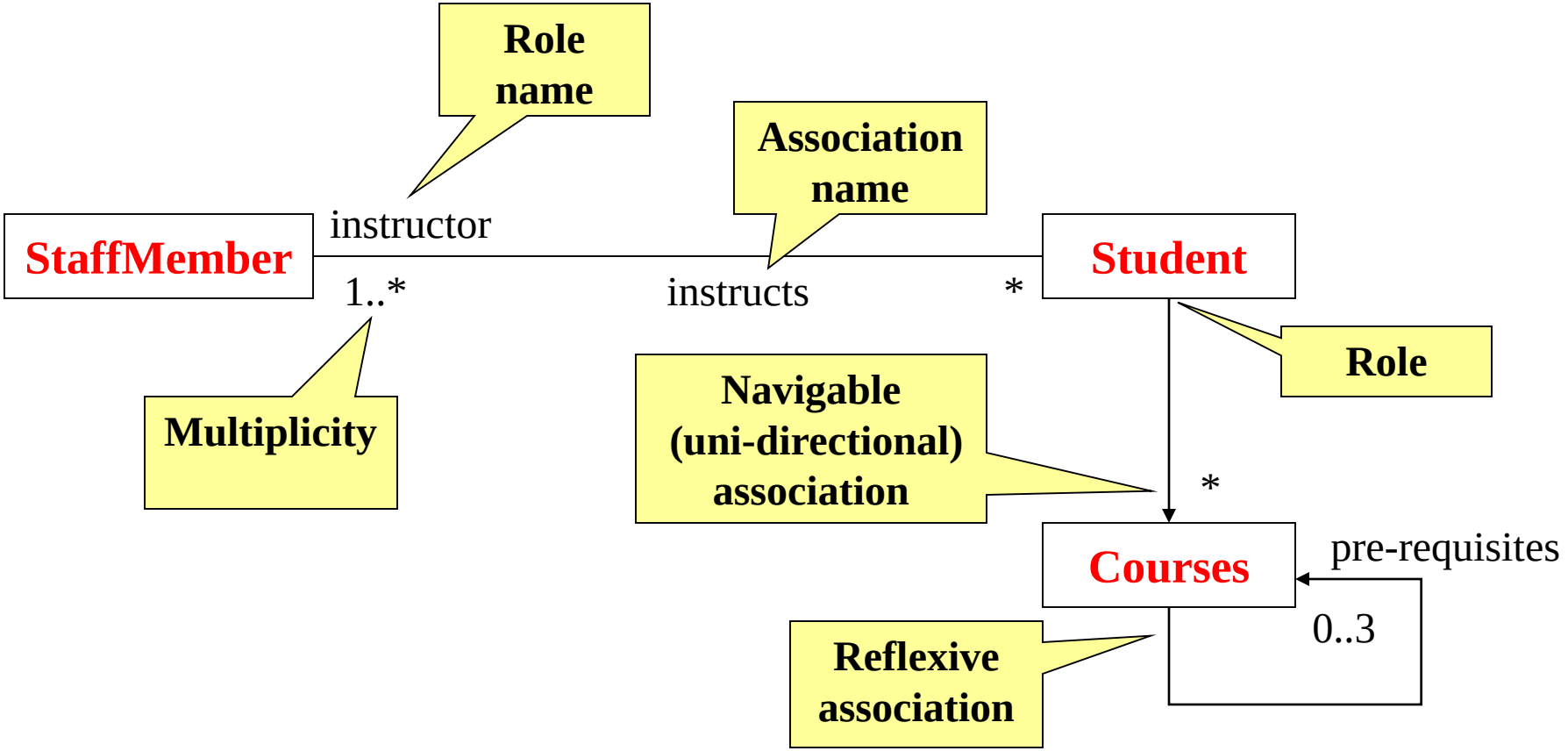


- Hoặc
  - Trong ClassA có thuộc tính có kiểu là ClassB
- Hoặc
  - Trong ClassB có thuộc tính có kiểu là ClassA
- Nhận xét: Về mặt lập trình, thuộc tính có thể được lưu trữ dạng biến đơn, biến mảng, hay biến con trỏ
- Ví dụ: ?

Ví dụ

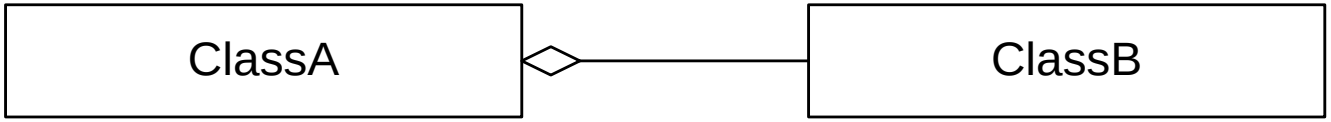

Person
Name
Phone Number
Email Address
Purchase Parking Pass
0..1
lives at
1
Address
Street
City
State
Postal Code
Country
Validate
Output As Label
Student
Student Number
Average Mark
Is Eligible To Enroll
Get Seminars Taken
Professor
Salary

Ví dụ


Role name
Association name
StaffMember
instructor
1..*
instructs
*
Student
Role
Multiplicity
Navigable (uni-directional) association
*
Courses
pre-requisites
0..3
Reflexive association

Quan hệ giữa các lớp đối tượng

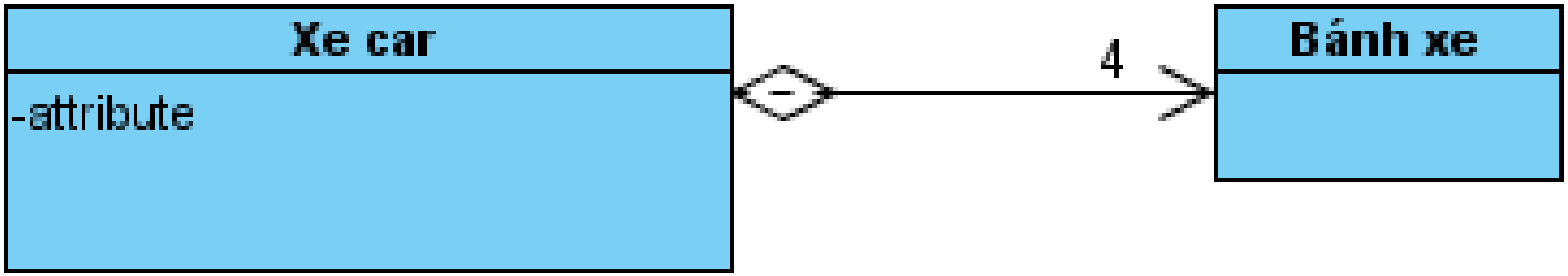
- Quan hệ Aggregation



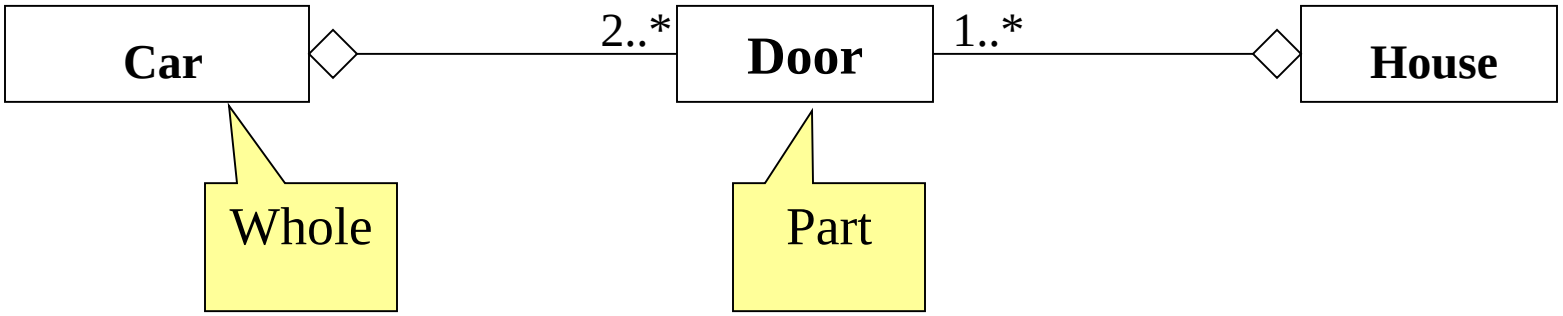
- Dùng để mô hình hóa quan hệ toàn thể - bộ phận giữa một kết tập và bộ phận của nó.
- Đã xác định được ClassA và ClassB có quan hệ Association với nhau
  - Xác định rõ hơn:
    - Trong object của ClassA có chứa (trong phần thuộc tính) object của ClassB
    - ObjectX của ClassA bị hủy thì ObjectY của 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:



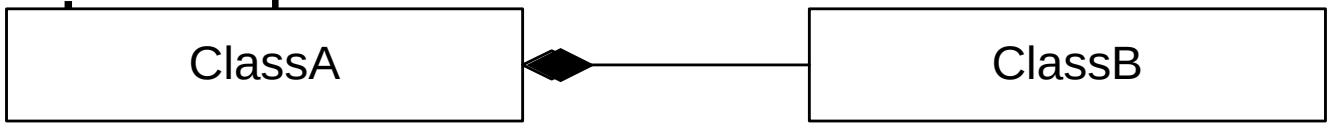
Quan hệ Aggregation



- Cụm từ "bộ phận của" (part of) được sử dụng để mô tả quan hệ?
  - Cánh cửa là một bộ phận của xe hơi
- Có phải một số hành vi của toàn thể được áp dụng tự động cho bộ phận của nó?
  - Xe hơi di chuyển, cửa di chuyển.
- Có phải một vài giá trị thuộc tính của toàn thể kéo theo một số thuộc tính của bộ phận?
  - Xe hơi màu xanh nên cửa màu xanh.
- Có tồn tại sự không đảo chiều giữa các lớp cho quan hệ kết tập?
  - Cửa là bộ phận của xe hơi. Xe hơi không là bộ phận của cửa.

Quan hệ giữa các lớp đối tượng

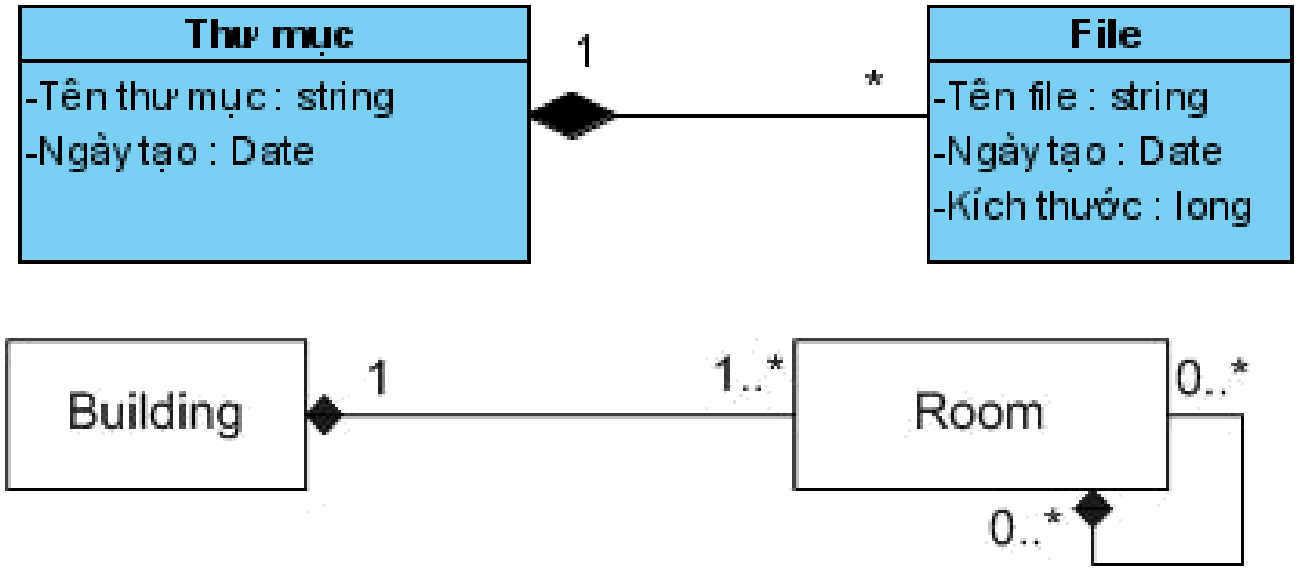
- Quan hệ Composition



- 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: ?

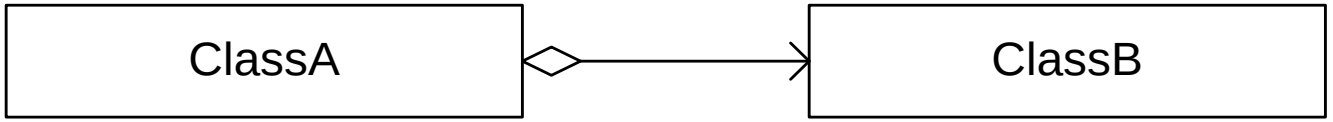
Quan hệ Composition

- 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

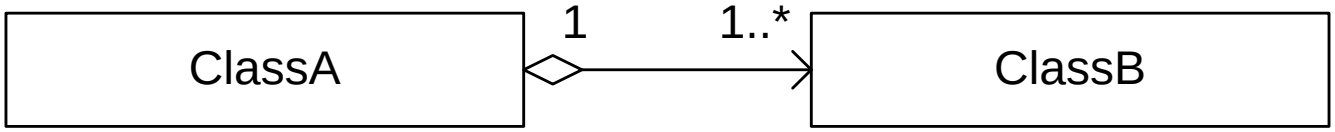
- Chiều của quan hệ (Association, Aggregation, Composition)



- 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)

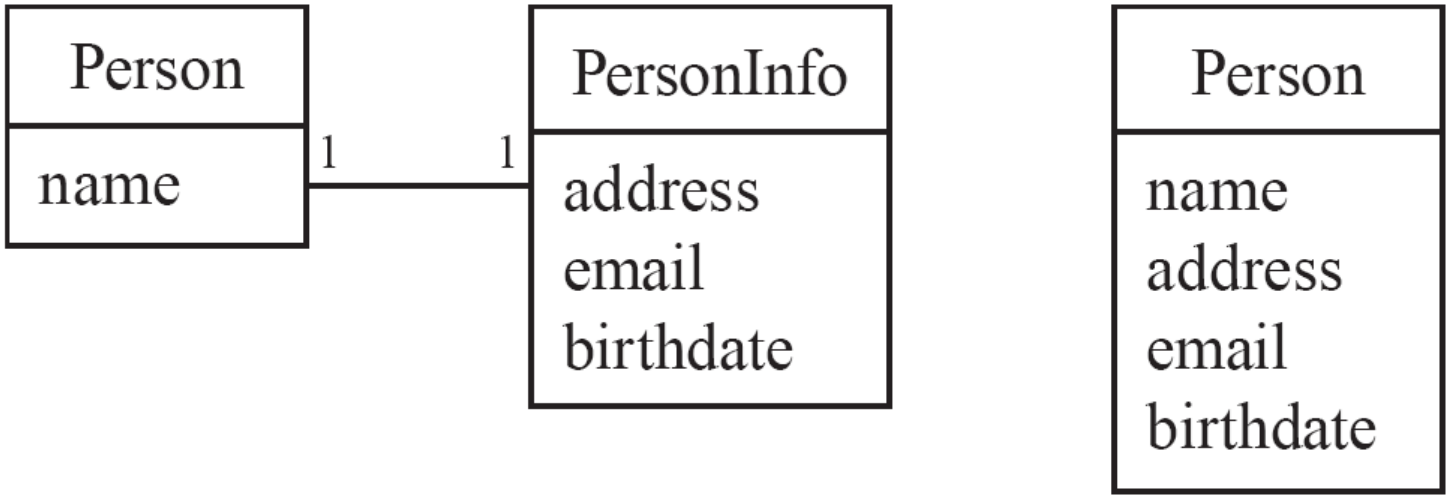


  - Ý nghĩa
  - Ví dụ:
    - 1
    - 2
    - 1..*
    - 0..*
    - *
    - 1, 3, 5..9

Chỉ có 1 đối tượng	1
0 hoặc nhiều (unlimited)	* (0..*)
1 hoặc nhiều	1..*
0 hoặc 1 (optional association)	0..1
Khoảng xác định	2..4
Nhiều khoảng	2, 4..6, 8

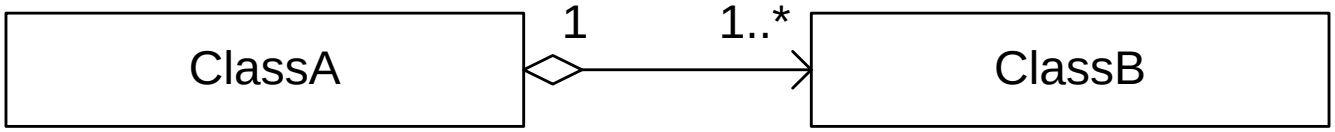
Quan hệ giữa các lớp đối tượng

- Tránh sử dụng quan hệ 1-1 không cần thiết trong biểu đồ lớp



Quan hệ giữa các lớp đối tượng

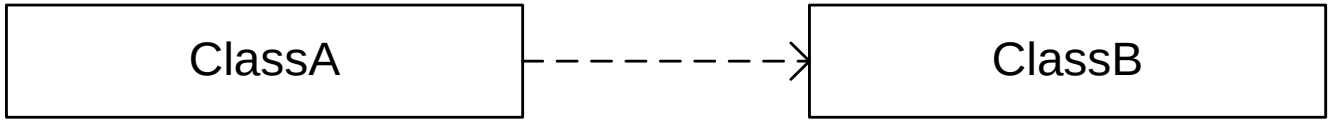
- Bản số - Multiplicity (Association, Aggregation, Composition)



  - 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()
    {
     …
    }
};
```

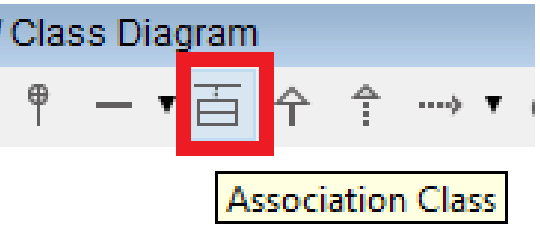
Biến cục bộ

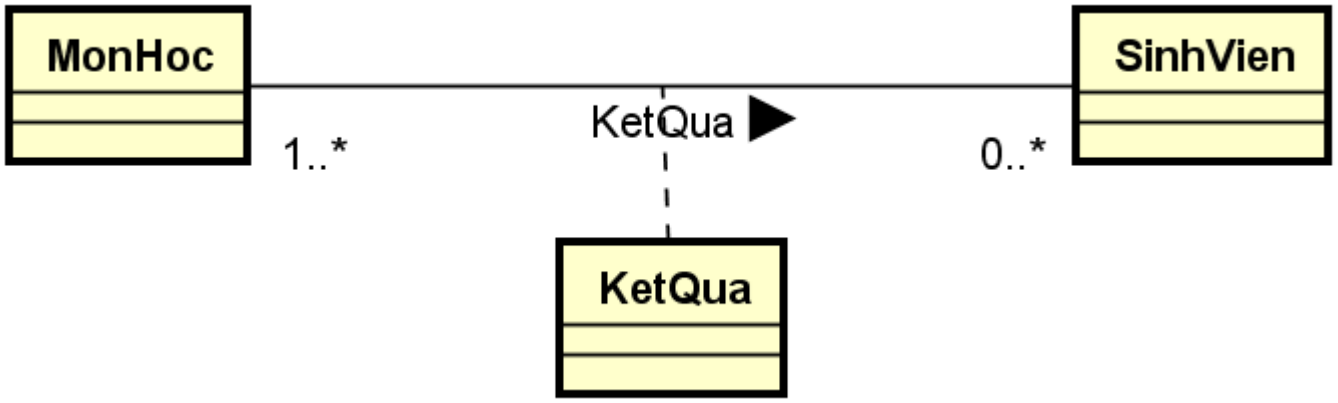
```
class A
{
    void    F()
    {
        B    x;
    }
};
```

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

Lớp kết hợp (Association Classes)

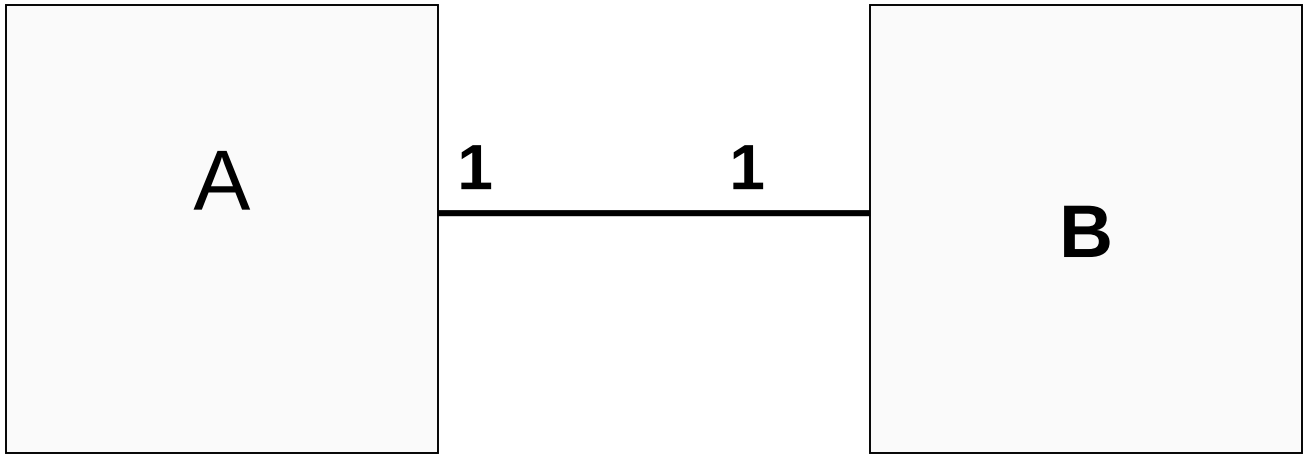
- Khi một mối kết hợp có các đặc trưng (thuộc tính, hoạt động và các mối kết hợp), chúng ta tạo ra một lớp để chứa các thuộc tính đó và kết nối với mối quan hệ, lớp này được gọi là lớp kết hợp.





Bản số (Multiplicity)

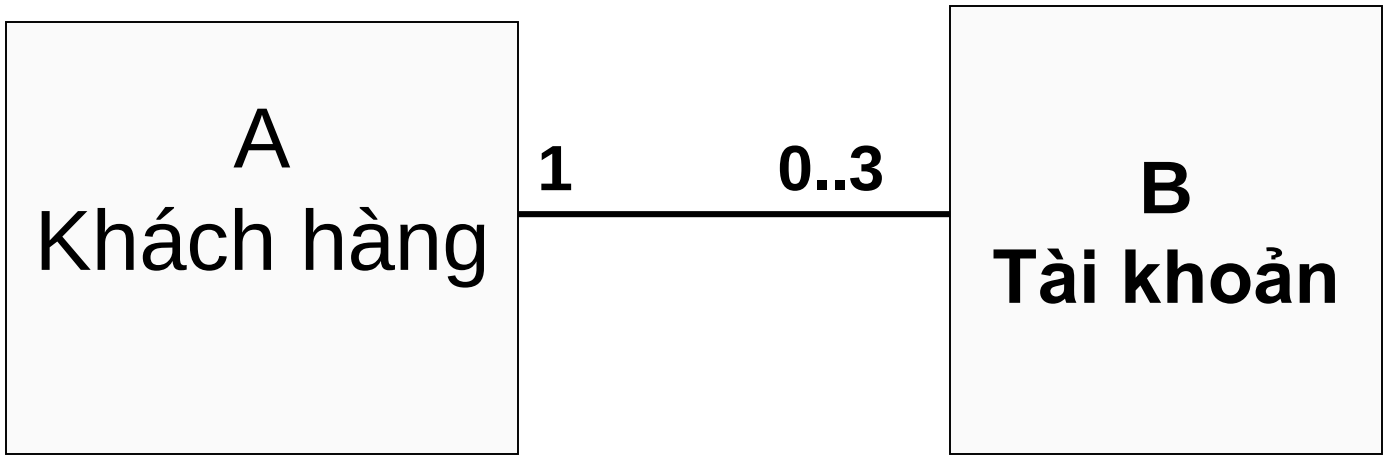
- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



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?

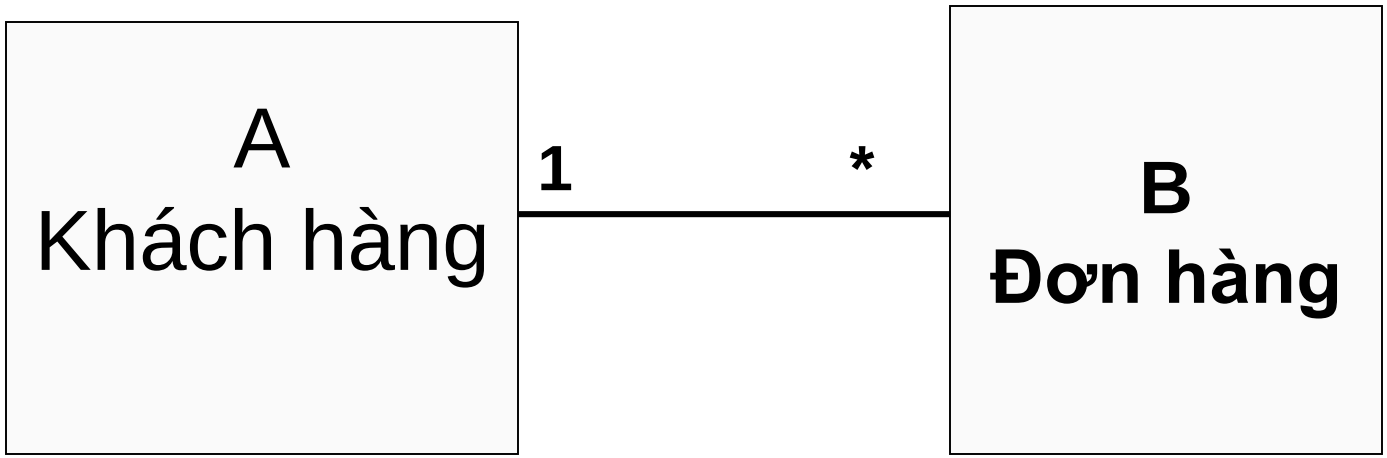


Một phần tử lớp A có tối đa 3 phần tử lớp B

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?

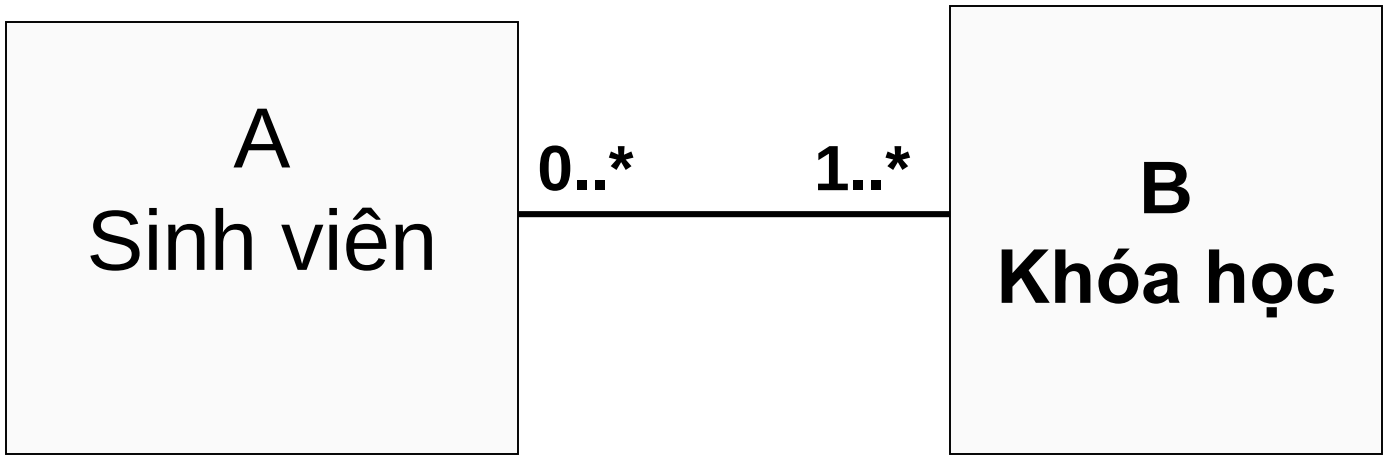


Một phần tử lớp A có nhiều phần tử lớp B

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

Bản số (Multiplicity)

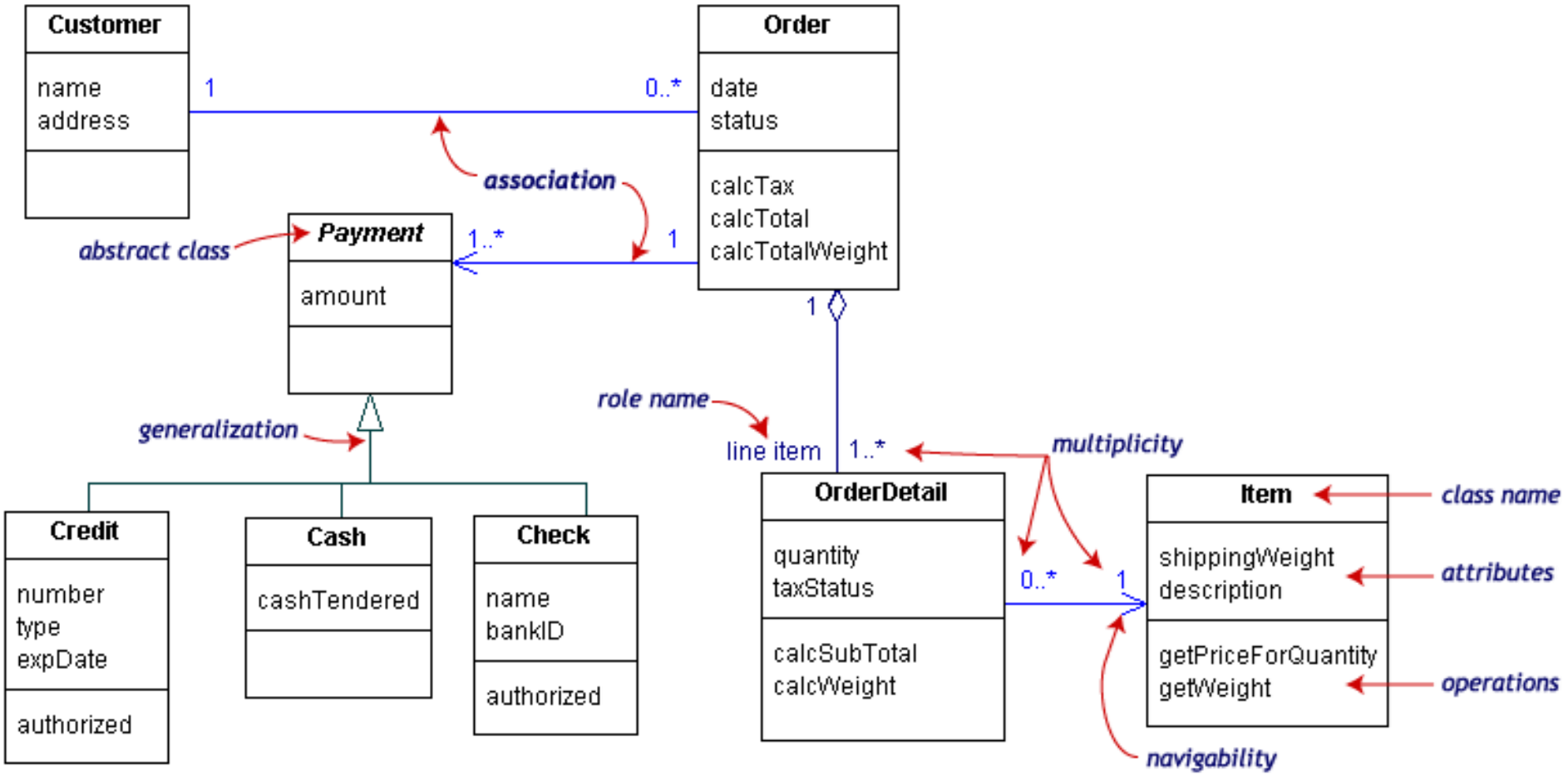
- Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?



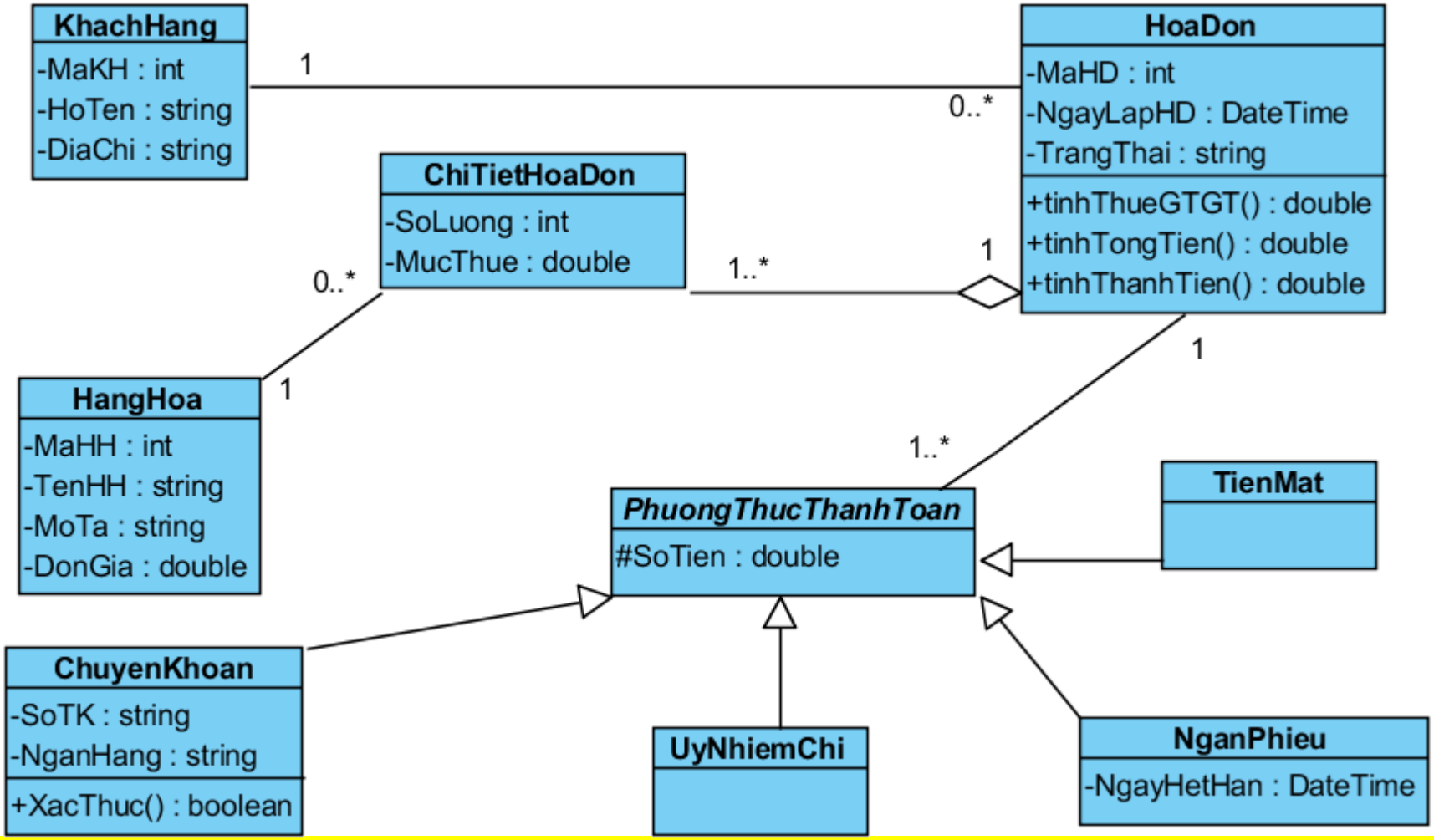
Mỗi sinh viên tham gia ít nhất 1 khóa học

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

Ví dụ


Customer
name
address
1
0..*
Order
date
status
calcTax
calcTotal
calcTotalWeight
association
abstract class
Payment
amount
1..*
1
1
generalization
role name
line item
1..*
multiplicity
Credit
number
type
expDate
authorized
Cash
cashTendered
Check
name
bankID
authorized
OrderDetail
quantity
taxStatus
calcSubTotal
calcWeight
0..*
1
Item
shippingWeight
description
getPriceForQuantity
getWeight
class name
attributes
operations
navigability

Ví dụ


KhachHang
-MaKH : int
-HoTen : string
-DiaChi : string
1
0..*
HoaDon
-MaHD : int
-NgayLapHD : DateTime
-TrangThai : string
+tinhThueGTGT() : double
+tinhTongTien() : double
+tinhThanhTien() : double
ChiTietHoaDon
-SoLuong : int
-MucThue : double
0..*
1..*
1
HangHoa
-MaHH : int
-TenHH : string
-MoTa : string
-DonGia : double
1
1
1..*
PhuongThucThanhToan
#SoTien : double
TienMat
ChuyenKhoan
-SoTK : string
-NganHang : string
+XacThuc() : boolean
UyNhiemChi
NganPhieu
-NgayHetHan : DateTime

Xây dựng sơ đồ lớp ở mức phân tích

CLASS DIAGRAM

Class Diagram

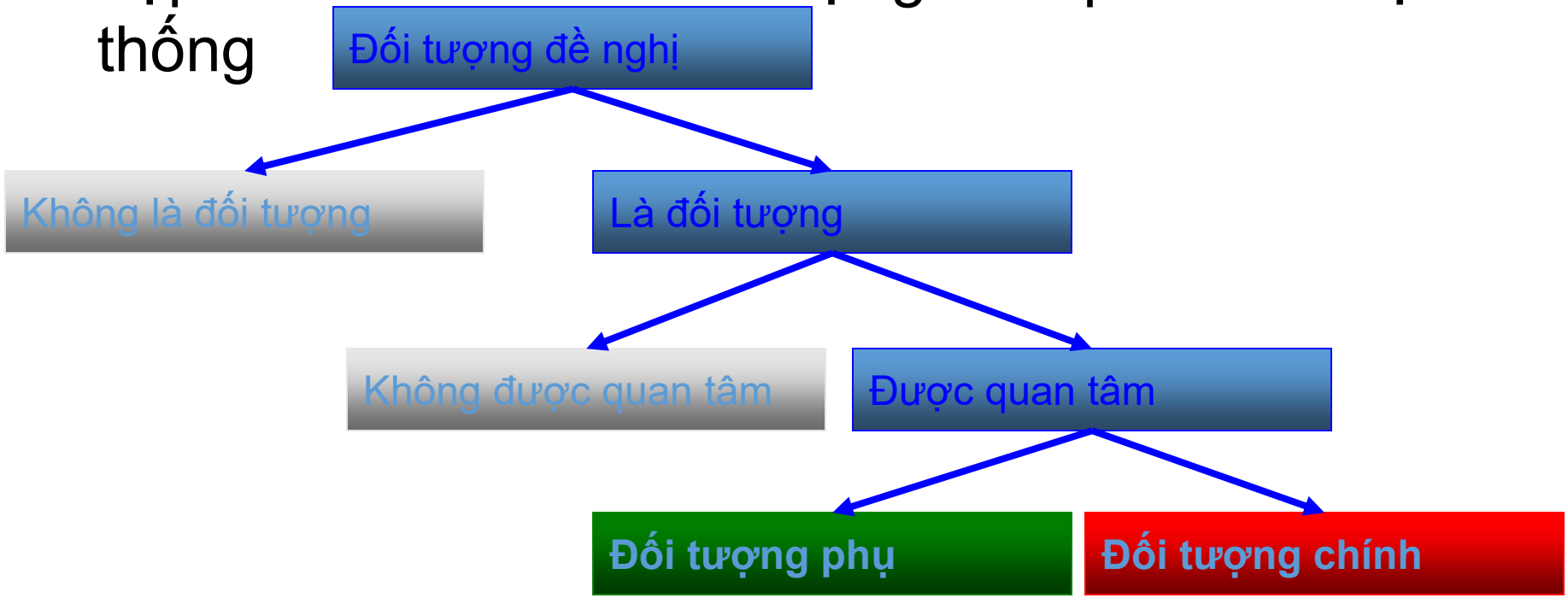
- Được xây dựng và hiệu chỉnh trong suốt quá trình phát triển
- Mục tiêu:
  - Đặt tên và lập mô hình các khái niệm trong hệ thống
  - Đặc tả sự cộng tác
  - Đặc tả sơ đồ CSDL
- Được phát triển bởi phân tích viên, thiết kế viên, lập trình viên

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 thống



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
...	...	...	...	...

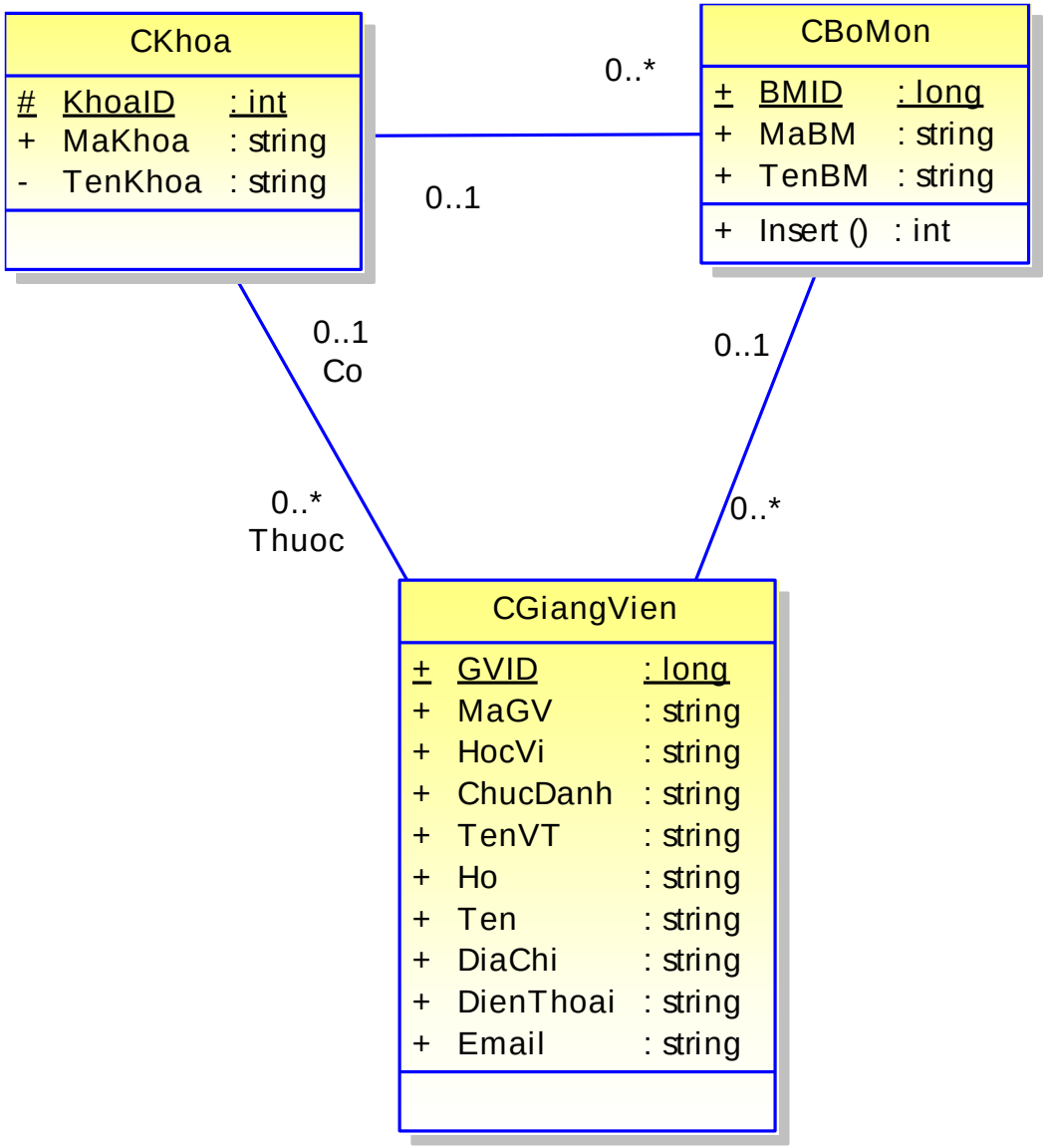
    - Danh sách các trách nhiệm chính
  - Với mỗi quan hệ:

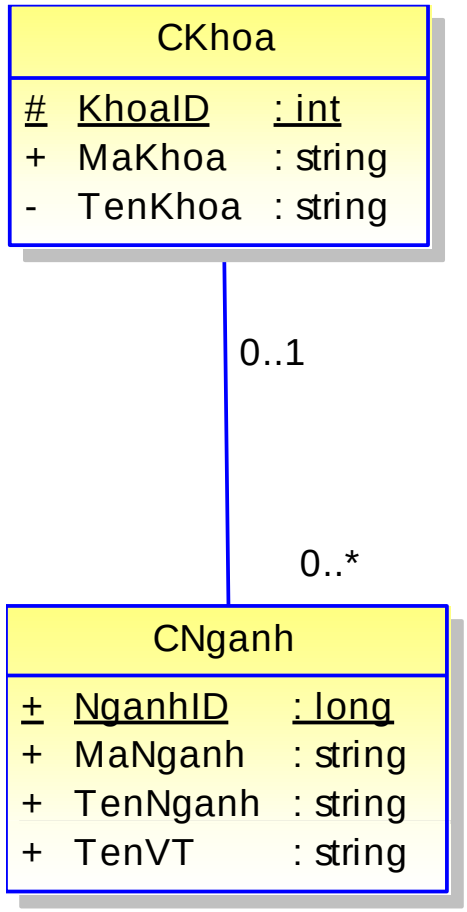
STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
...	...	...	...	...

Áp dụng

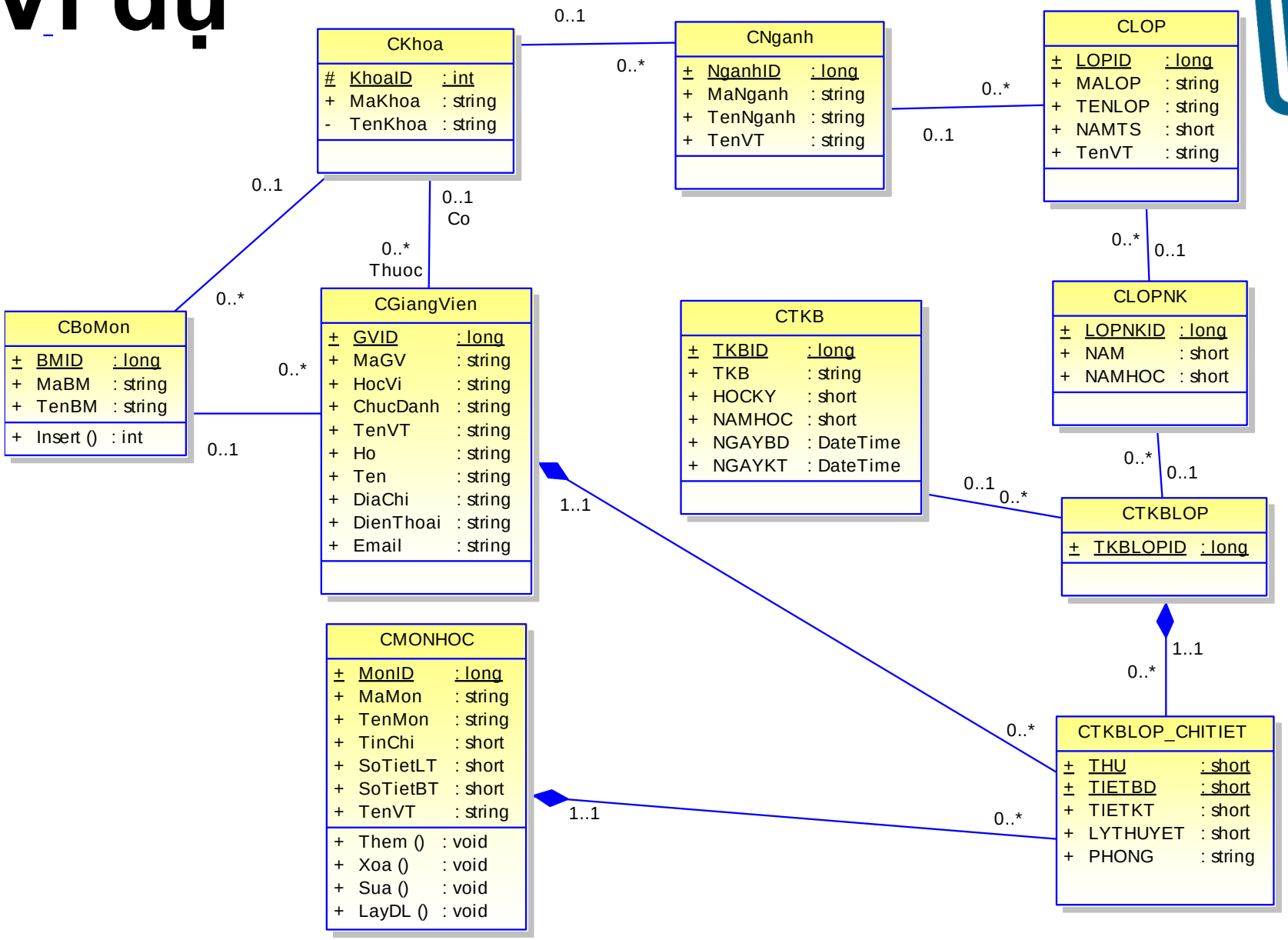
- Áp dụng thực tế vào các bài tập
  - Xác định các lớp đối tượng chính
  - Xác định các thông tin và hành động/trách nhiệm của mỗi lớp đối tượng chính
  - Xác định các quan hệ chính
  - Xác định các lớp đối tượng phụ, các danh mục

Ví dụ

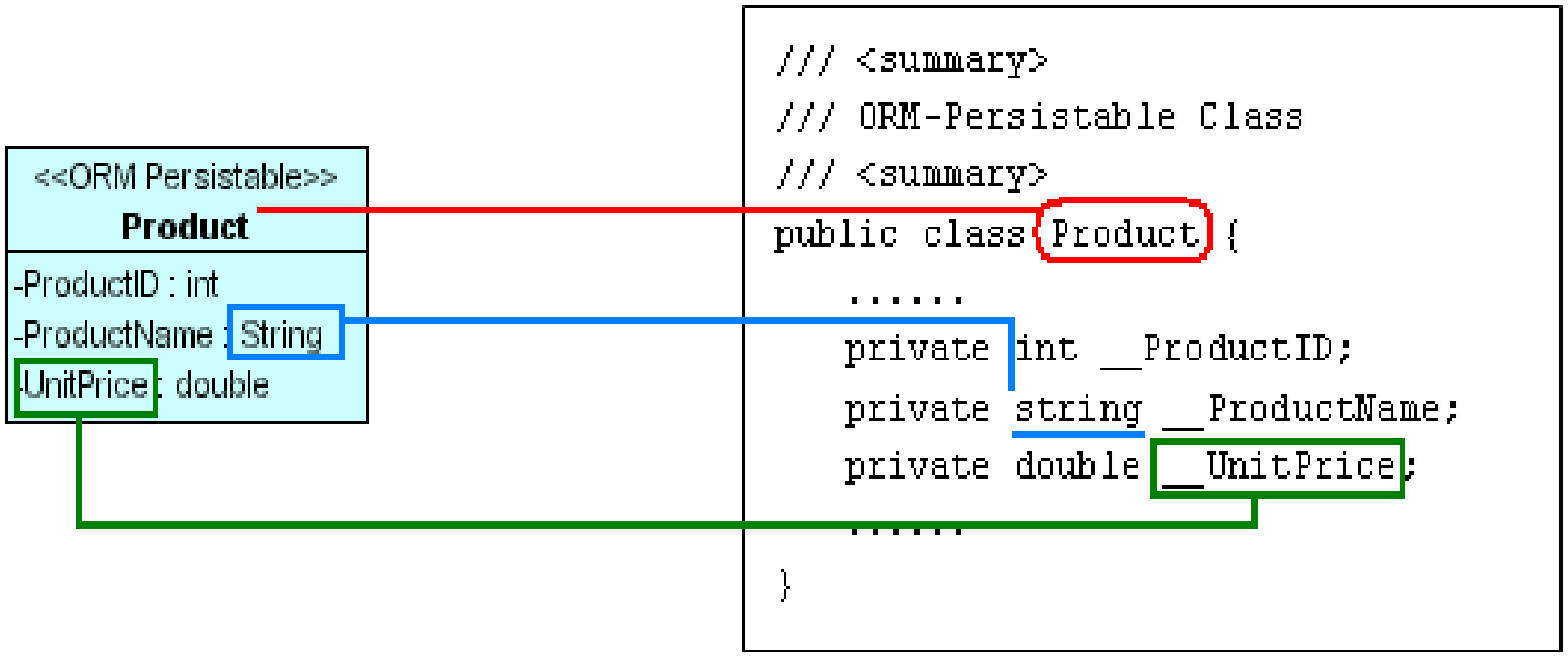

CKhoa
KhoaID : int
+ MaKhoa : string
- TenKhoa : string
CBoMon
+ BMID : long
+ MaBM : string
+ TenBM : string
+ Insert () : int
0..*
0..1
0..1
Co
0..1
0..*
Thuoc
0..*
CGiangVien
+ GVID : long
+ MaGV : string
+ HocVi : string
+ ChucDanh : string
+ TenVT : string
+ Ho : string
+ Ten : string
+ DiaChi : string
+ DienThoai : string
+ Email : string


CKhoa
KhoaID : int
+ MaKhoa : string
- TenKhoa : string
0..1
0..*
CNganh
+ NganhID : long
+ MaNganh : string
+ TenNganh : string
+ TenVT : string

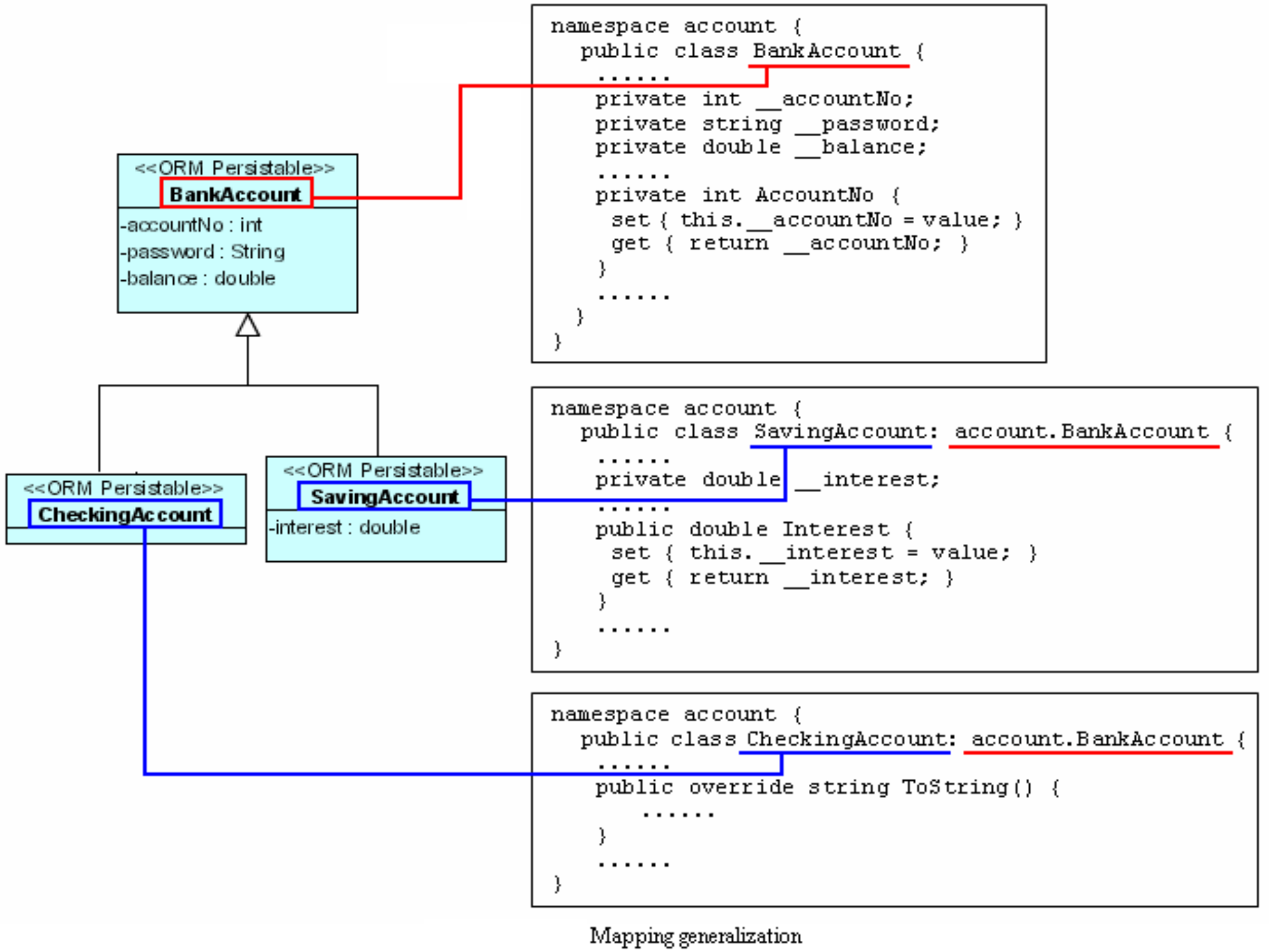
Ví dụ


CKhoa
KhoaID : int
+ MaKhoa : string
- TenKhoa : string
CNganh
+ NganhID : long
+ MaNganh : string
+ TenNganh : string
+ TenVT : string
CLOP
+ LOPID : long
+ MALOP : string
+ TENLOP : string
+ NAMTS : short
+ TenVT : string
CBoMon
+ BMID : long
+ MaBM : string
+ TenBM : string
+ Insert () : int
CGiangVien
+ GVID : long
+ MaGV : string
+ HocVi : string
+ ChucDanh : string
+ TenVT : string
+ Ho : string
+ Ten : string
+ DiaChi : string
+ DienThoai : string
+ Email : string
CTKB
+ TKBID : long
+ TKB : string
+ HOCKY : short
+ NAMHOC : short
+ NGAYBD : DateTime
+ NGAYKT : DateTime
CLOPNK
+ LOPNKID : long
+ NAM : short
+ NAMHOC : short
CTKBLOP
+ TKBLOPID : long
CMONHOC
+ MonID : long
+ MaMon : string
+ TenMon : string
+ TinChi : short
+ SoTietLT : short
+ SoTietBT : short
+ TenVT : string
+ Them () : void
+ Xoa () : void
+ Sua () : void
+ LayDL () : void
CTKBLOP_CHITIET
+ THU : short
+ TIETBD : short
+ TIETKT : short
+ LYTHUYET : short
+ PHONG : string
0..1
0..*
0..1
0..*
0..1
0..*
0..1
Co
0..*
Thuoc
0..*
0..1
0..*
0..1
1..1
0..1
0..*
0..*
0..1
1..1
0..*
1..1
0..*
0..*

Ánh xạ biểu đồ sang Code

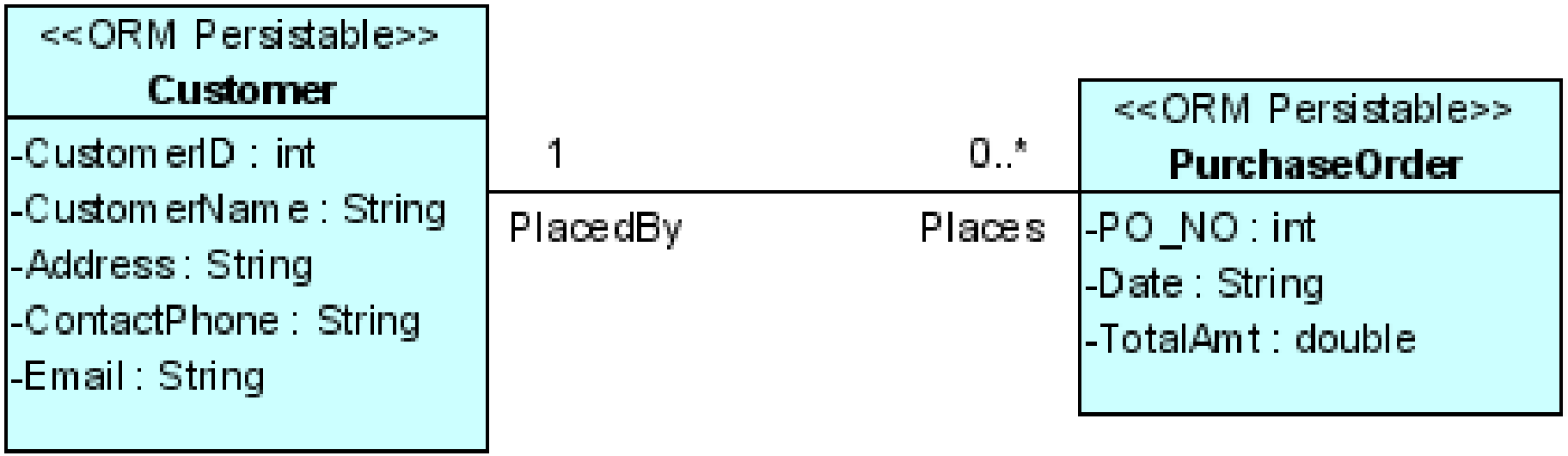


- Mapping Classes, Attributes and Data Type

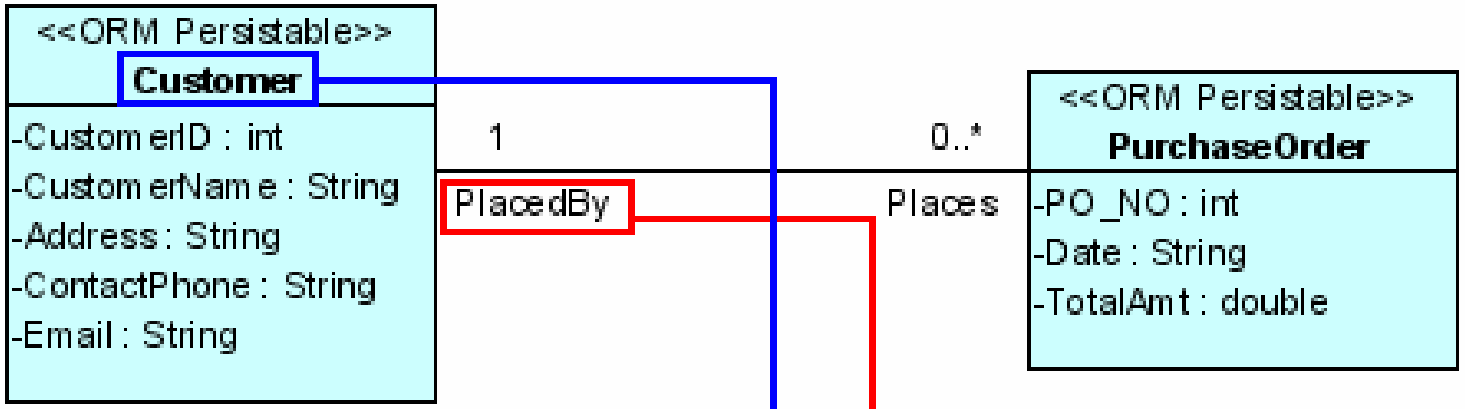


Mapping generalization

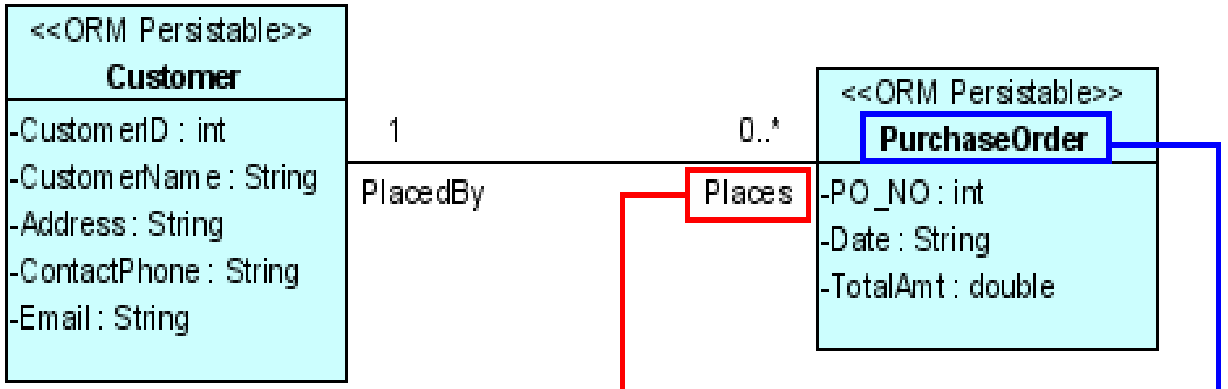
Ánh xạ khách hàng- đơn hàng



The Classes with association and multiplicity



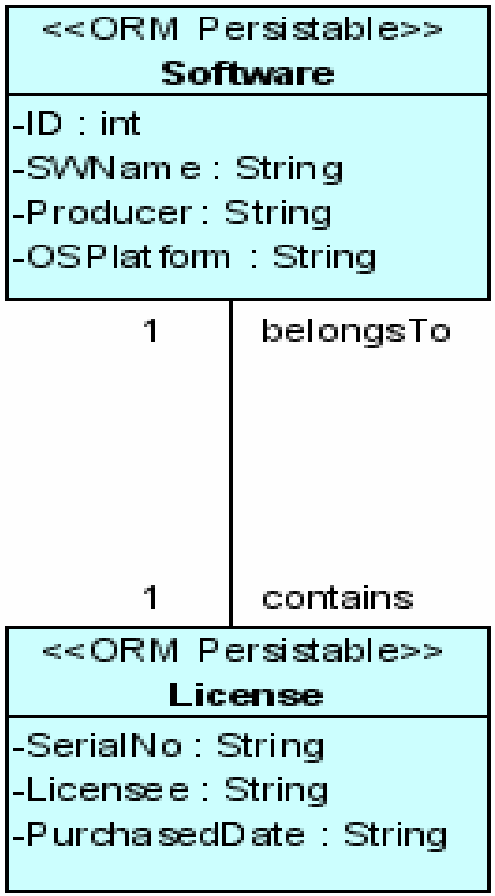

```
public class PurchaseOrder {
  ......
  private int __PO_NO;
  private string __Date;
  private double __TotalAmt;
  private shoppingcart.Customer __PlacedBy;
  ......
  public shoppingcart.Customer PlacedBy {
    set { value.Places.Add(this); }
    get { return __PlacedBy; }
  }
  ......
}
```




```
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



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 {
           ...... }
  ......
}
```

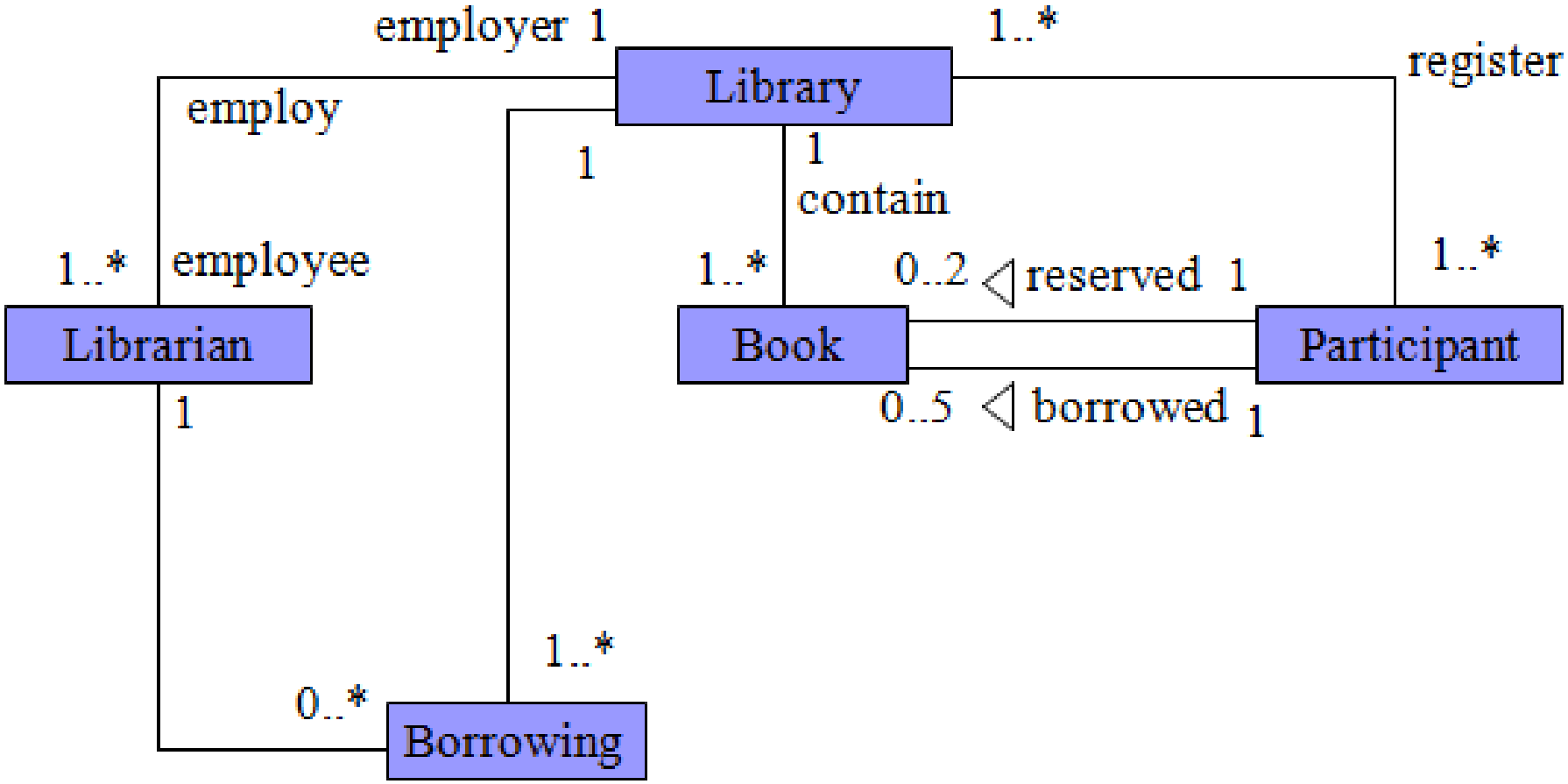
Mapping the One-to-one Relationship

Bài tập

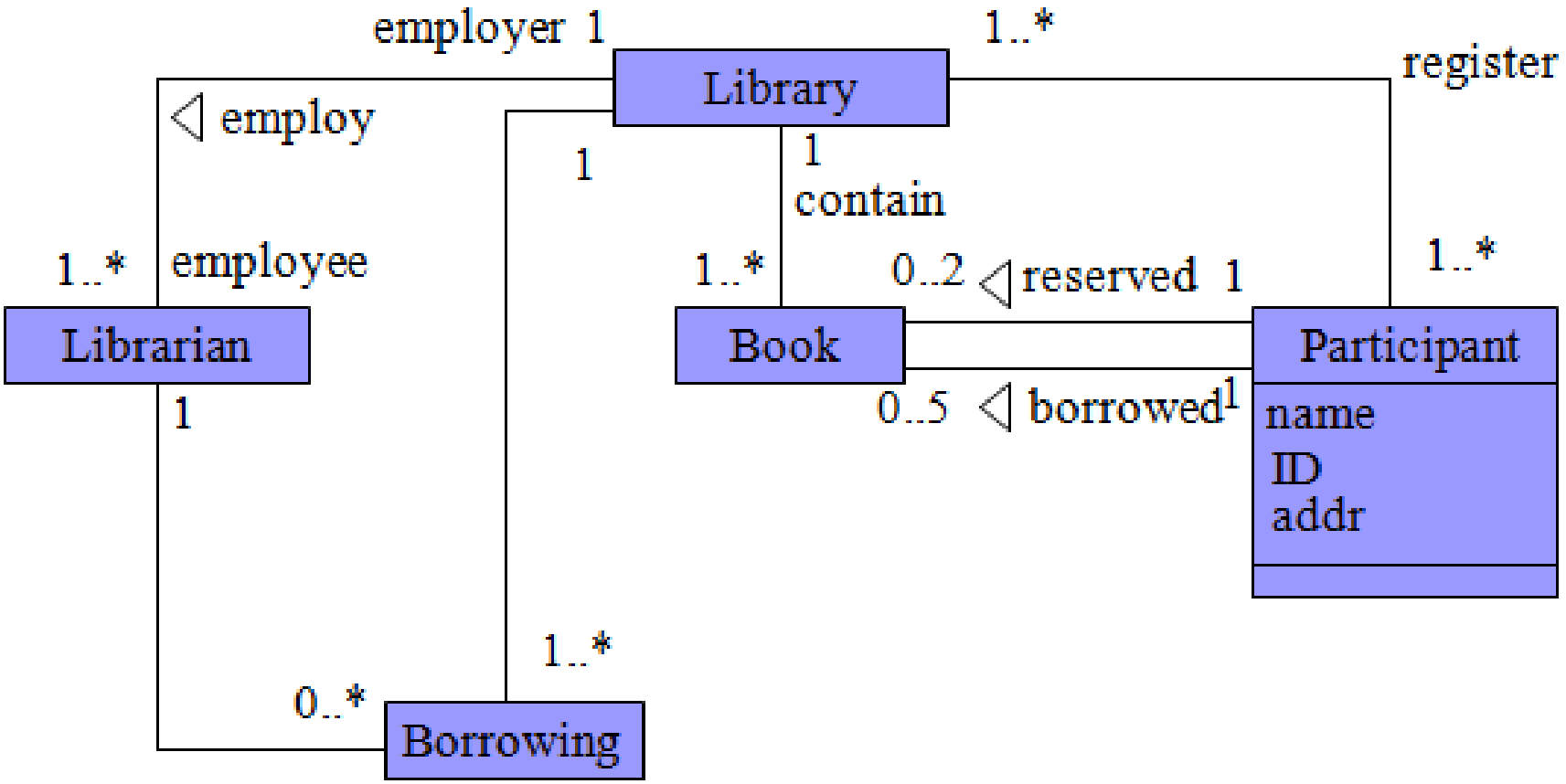
Mô hình hóa biểu đồ lớp cho hệ thống quản lý thư viện

- Người quản lý thư viện mong muốn tự động hóa việc mượn sách
- Họ yêu cầu một phần mềm cho phép người sử dụng biết sách hiện có, có thể đặt mượn 2 quyển sách, những người tham gia mượn sách có thể biết sách nào đã mượn hoặc đã đặt
- Những người tham gia mượn sách sở hữu một password để truy nhập
- Việc mượn sách được thực hiện bởi các thủ thư, sau khi xác định người mượn sách, họ biết được người này có được phép mượn hay không? (tối đa 5 quyển), người này được ưu tiên? (đã đặt trước)

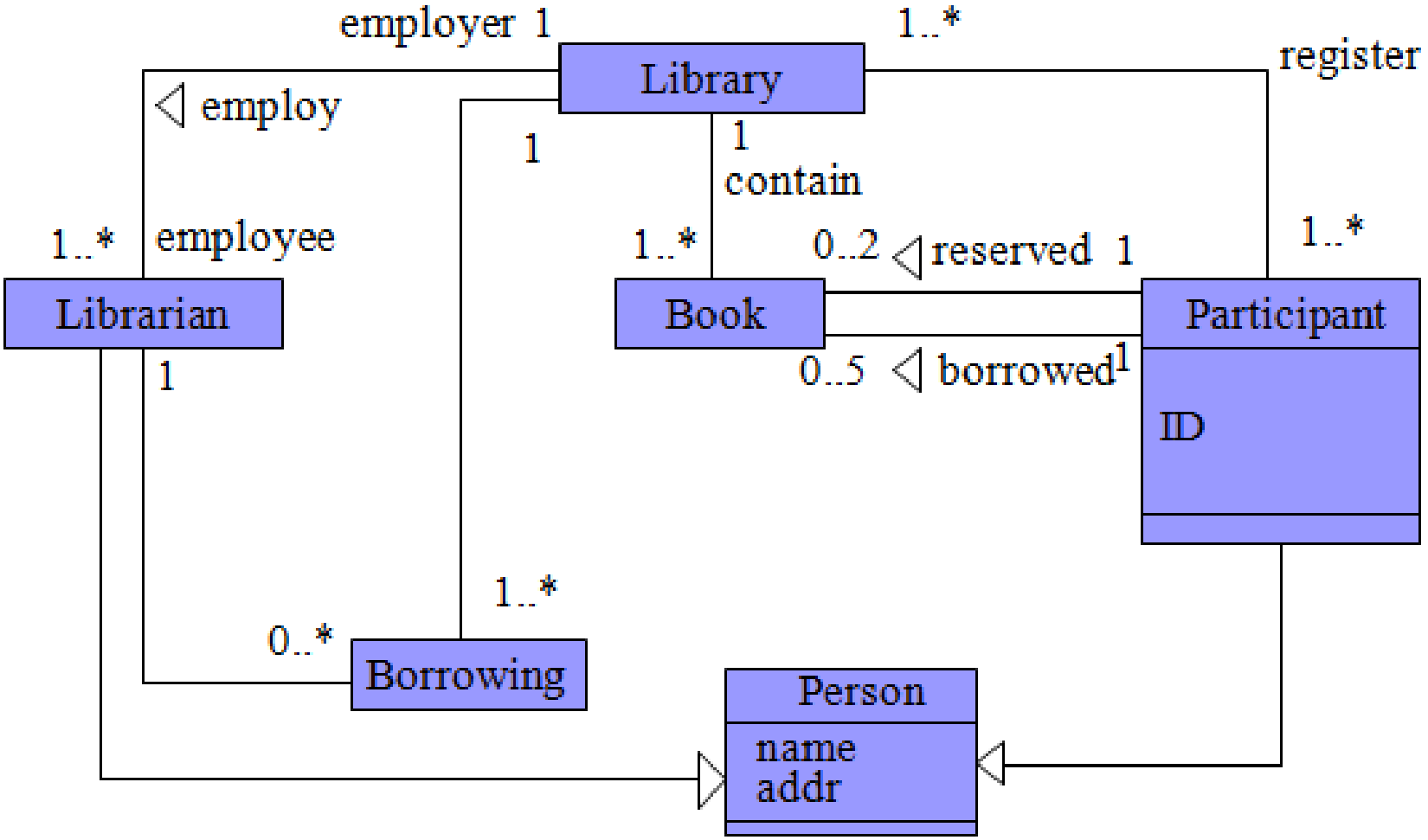
Xác định các liên kết



Xác định các thuộc tính



Tổng quát hóa bằng thừa kế



Câu hỏi và thảo luận

Thank you!!!