

Phân tích và thiết kế Hệ thống

Giảng viên: Nguyễn Bá Ngọc

Chương 3

Hà Nội-2021


Chương 3

Mô hình hóa cấu trúc

Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập

Nội dung

- 
- Các khái niệm
 - Các kỹ thuật
 - Các sơ đồ
 - Mô tả ràng buộc với OCL
 - Đặc tả lớp với thẻ CRC
 - Mẫu phân tích
 - Bài tập

Lĩnh vực vấn đề

- Lĩnh vực vấn đề (*Problem domain*) - Vùng nhu cầu nghiệp vụ của người dùng trong phạm vi hệ thống
 - Bao gồm tất cả thông tin, các thứ như hóa đơn, sản phẩm v.v..
 - .. người dùng gặp khi tác nghiệp.

Lớp lĩnh vực

- Lớp lĩnh vực (*domain class*) mô tả các đối tượng của lĩnh vực vấn đề, các thứ mà người dùng gặp khi thực hiện các hoạt động nghiệp vụ
 - Chứa các thông tin cần được ghi nhớ và xử lý bởi hệ thống
 - Ví dụ, sản phẩm, hóa đơn, thông tin khách hàng, khoản thanh toán, v.v.
- (*Lớp lĩnh vực là 1 tầng kiến trúc hệ thống*)

Làm sao để xác định các lớp lĩnh vực?

Các bài toán

- Xác định các thành phần dữ liệu cần lưu trữ?
- Kết hợp các thành phần và xác định các đối tượng?
- Xác định các thuộc tính và các lớp lĩnh vực?
- Xác định các mối quan hệ giữa các đối tượng?
- Xây dựng mô hình cấu trúc?

Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập

Các kỹ thuật

- Kỹ thuật danh từ (Noun): Tìm, phân loại, và thiết lập danh sách danh từ có trong các nguồn như thảo luận hoặc tài liệu. Sau đó phân tích và xác định các lớp lĩnh vực và thuộc tính, loại bỏ những thứ không cần ghi nhớ/chỉ giữ lại những thứ cần ghi nhớ.
- Tổng hợp kiến thức (BrainStorming): Tất cả mọi người cùng tham gia đóng góp hiểu biết, biên soạn danh sách các mục dữ liệu quan trọng.
 - Thông tin có thể được tổng hợp và phân loại theo danh mục và được sử dụng làm cơ sở xác định các lớp lĩnh vực.

Kỹ thuật danh từ

- Thường dẫn đến 1 danh sách dài với nhiều danh từ, bao gồm cả những thứ không cần lưu trong hệ thống.
 - Danh từ có thể là tên của thuộc tính, đối tượng, lớp và cả những thứ khác
- Có thể đem lại kết quả ban đầu đáng kể, kết quả thu được có thể tiếp tục được mở rộng, ví dụ dựa trên kinh nghiệm, hiểu biết về lĩnh vực vấn đề.

Các bước áp dụng kỹ thuật danh từ

1. Sử dụng các mô hình chức năng (đặc tả ca sử dụng, sơ đồ ca sử dụng v.v..), và các tài liệu khác về hệ thống (bao gồm các thông tin nhập và xuất), xác định tất cả các danh từ và lập danh mục danh từ.
2. Sử dụng các nguồn thông tin khác hiện có: Các biểu mẫu, các báo cáo hiện có v.v., để mở rộng danh mục.
3. Kiểm tra các danh từ đã tìm được.
4. Kiểm tra danh mục và ghi chú phân loại từng danh từ cần được thêm vào, có thể giữ, loại bỏ hoặc cân nhắc thêm v.v..
5. Cùng kiểm tra danh sách với người dùng, các bên liên quan, và các thành viên thực hiện dự án và sau đó lập danh sách các thứ trong lĩnh vực nghiệp vụ.

Một số câu hỏi hỗ trợ phân loại danh từ

Các câu hỏi có thể được sử dụng để thực hiện bước 3

- Để quyết định giữ 1 danh từ
 - Nó có nằm trong phạm vi của hệ thống đang được phát triển?
 - Hệ thống có cần ghi nhớ nhiều hơn 1 thứ tương tự?
- Để quyết định loại 1 danh từ
 - Nó có phải 1 từ đồng nghĩa với 1 thứ khác đã được xác định?
 - Chỉ là đầu ra do hệ thống cung cấp từ những thông tin khác đã được xác định?
 - Chỉ là đầu vào dẫn tới việc lưu thông tin khác đã được xác định?
- Để xác định thuộc tính và đối tượng
 - Nó có phải thành phần của thứ khác đã được xác định?
 - Nó có phải thứ người dùng có thể cần nếu các giả thuyết thay đổi?

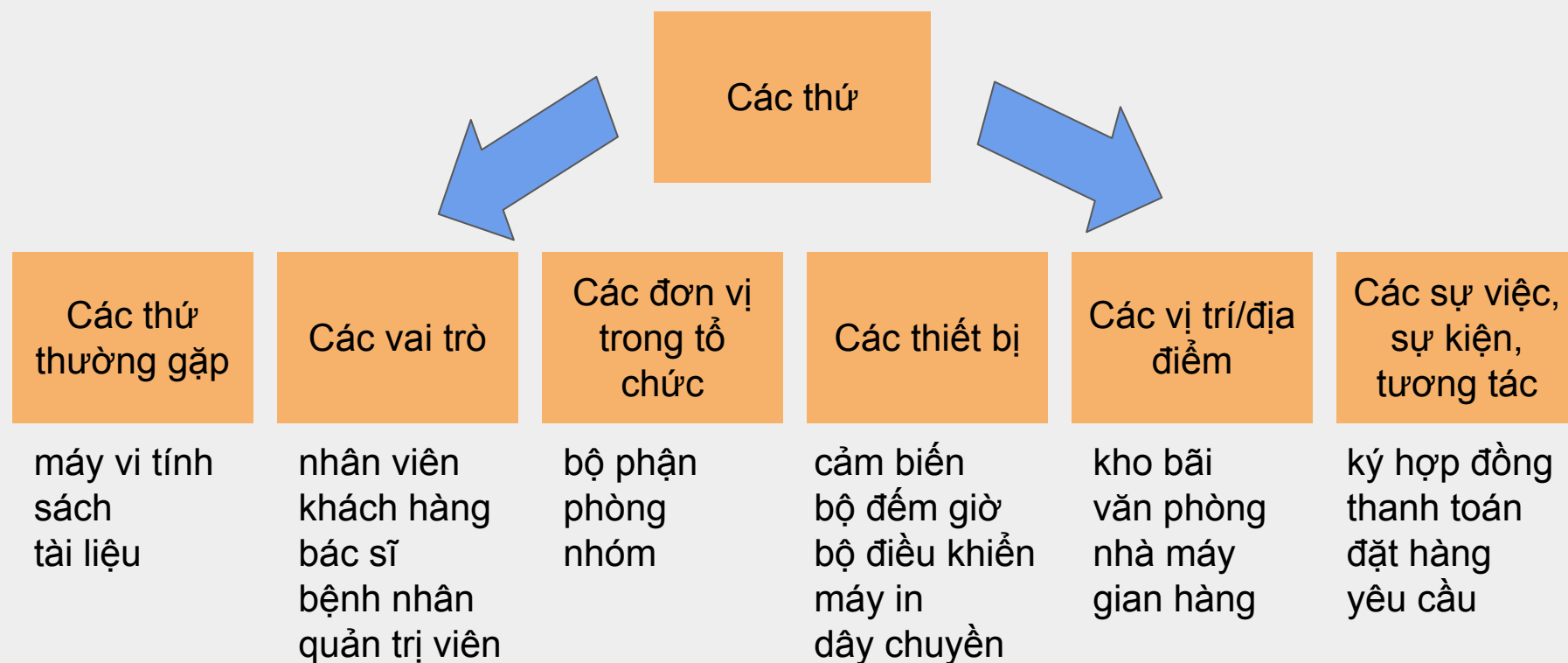
Ví dụ 3.1. Các danh từ

Các danh từ đã được phát hiện	Ghi chú
Khoản thanh toán	
Đơn hàng	
Ngân hàng	
Danh mục	
Sản phẩm	
Yêu cầu hoàn trả	
Giá bán	
Thông tin xác nhận	
Thông tin thẻ tín dụng	
Khách hàng	
Thông tin quản lý	
Thông tin quảng cáo	
Bản báo cáo doanh thu	

Các bước tổng hợp kiến thức

1. Lựa chọn người dùng và 1 tập ca sử dụng liên quan
2. Phỏng vấn người dùng và xác định các thứ liên quan đến ca sử dụng và thông tin về chúng cần được quản lý bởi hệ thống.
3. Sử dụng các danh mục để quản lý và đưa ra các câu hỏi: Bạn có lưu thông tin về những thứ quan trọng nào? Các hoạt động có được gắn với địa điểm nào không? Vai trò nào của người dùng cần được ghi nhớ?
4. Tiếp tục tổng hợp kiến thức từ tất cả các nhóm người dùng và những người liên quan để mở rộng kết quả.
5. Hợp nhất các kết quả, loại trùng lặp, và lập danh sách các thứ.

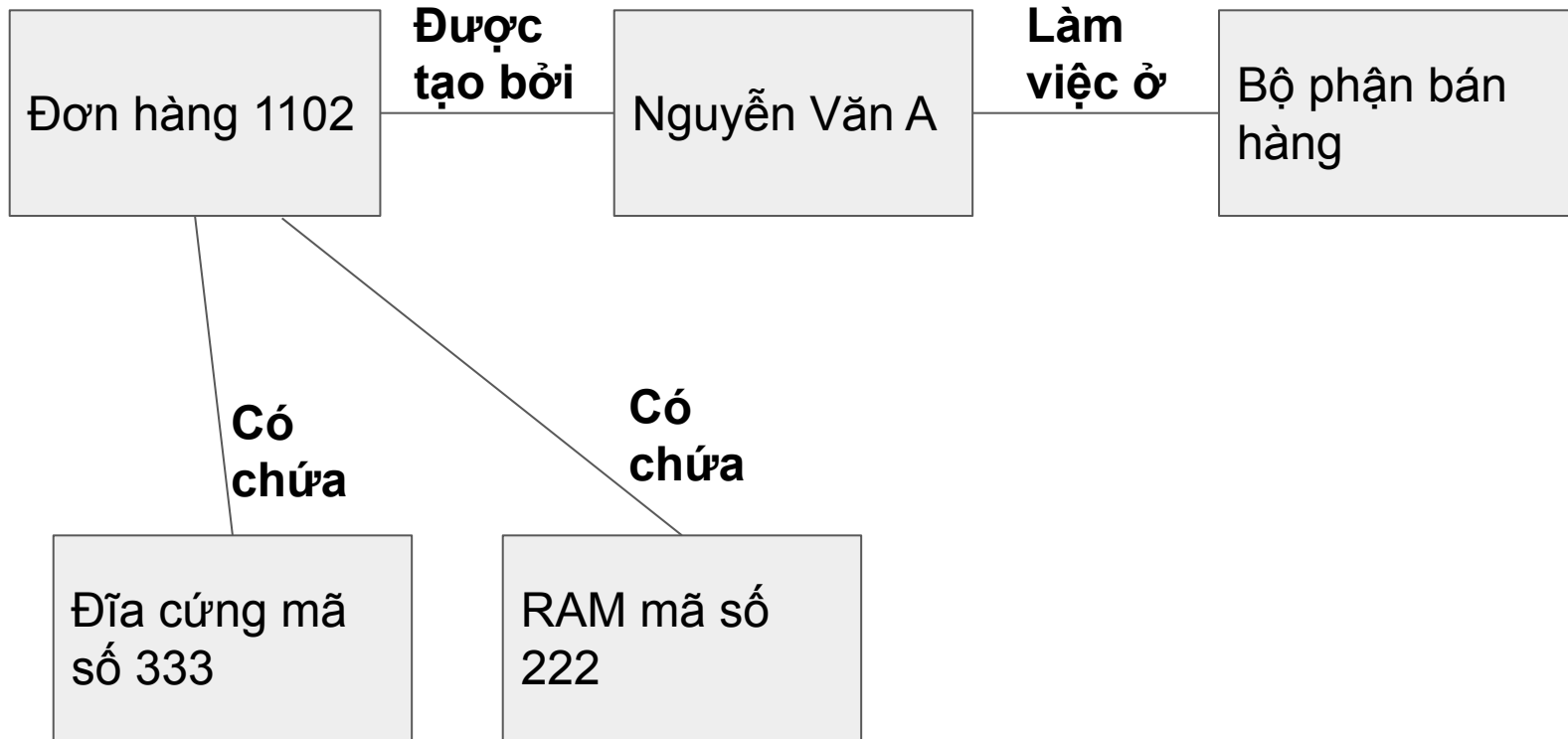
Ví dụ 3.2. Danh mục phân loại



Ví dụ 3.3. Các thuộc tính và giá trị

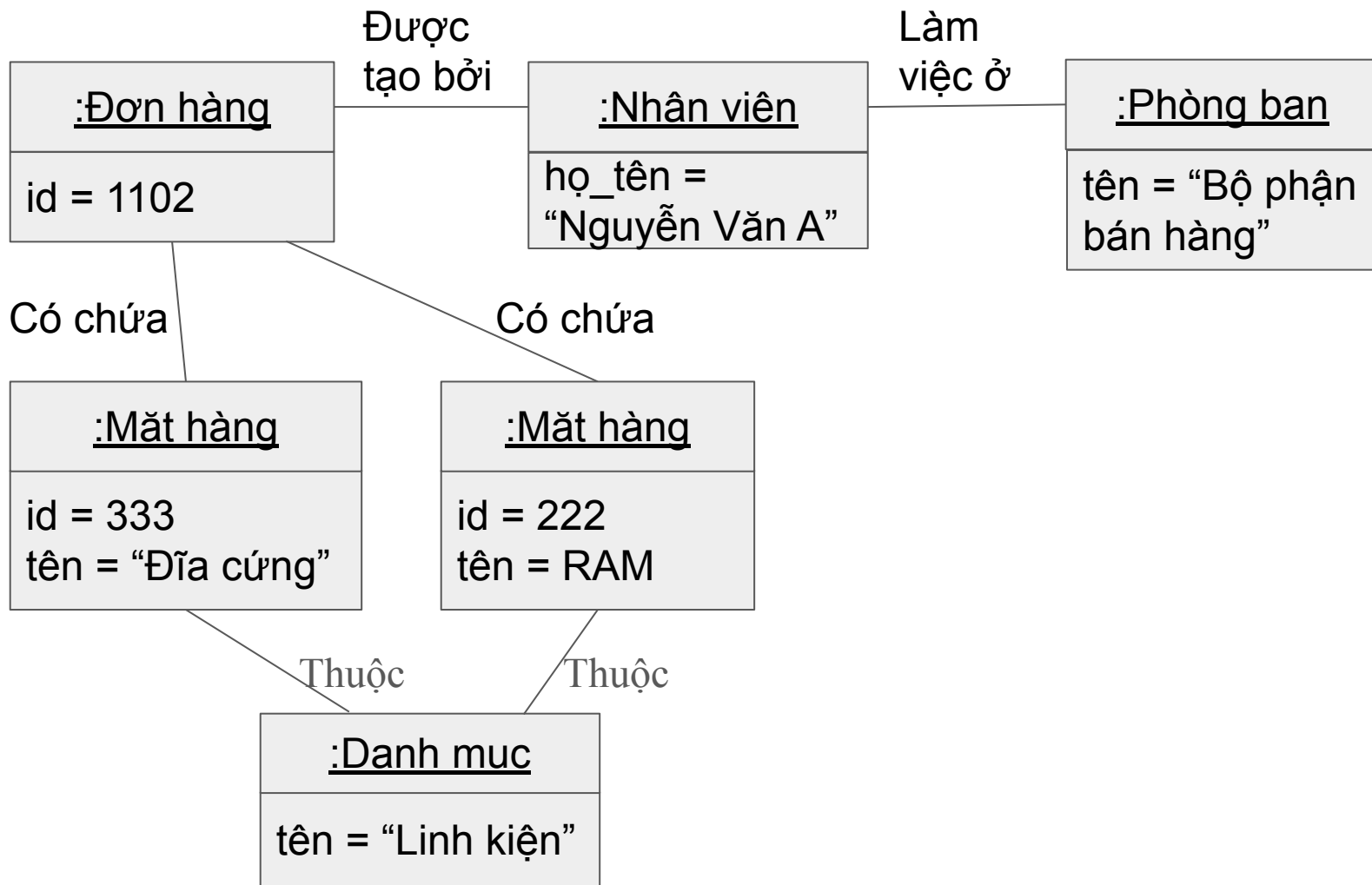
Tất cả khách hàng đều có các thuộc tính	Giá trị của các thuộc tính		
ID	101	102	103
Họ-đệm	Nguyễn Văn	Trần Thị	Vũ Văn
Tên	A	C	D
SĐT	0988888888	0967777777	097333333

Ví dụ 3.4. Các mối liên hệ



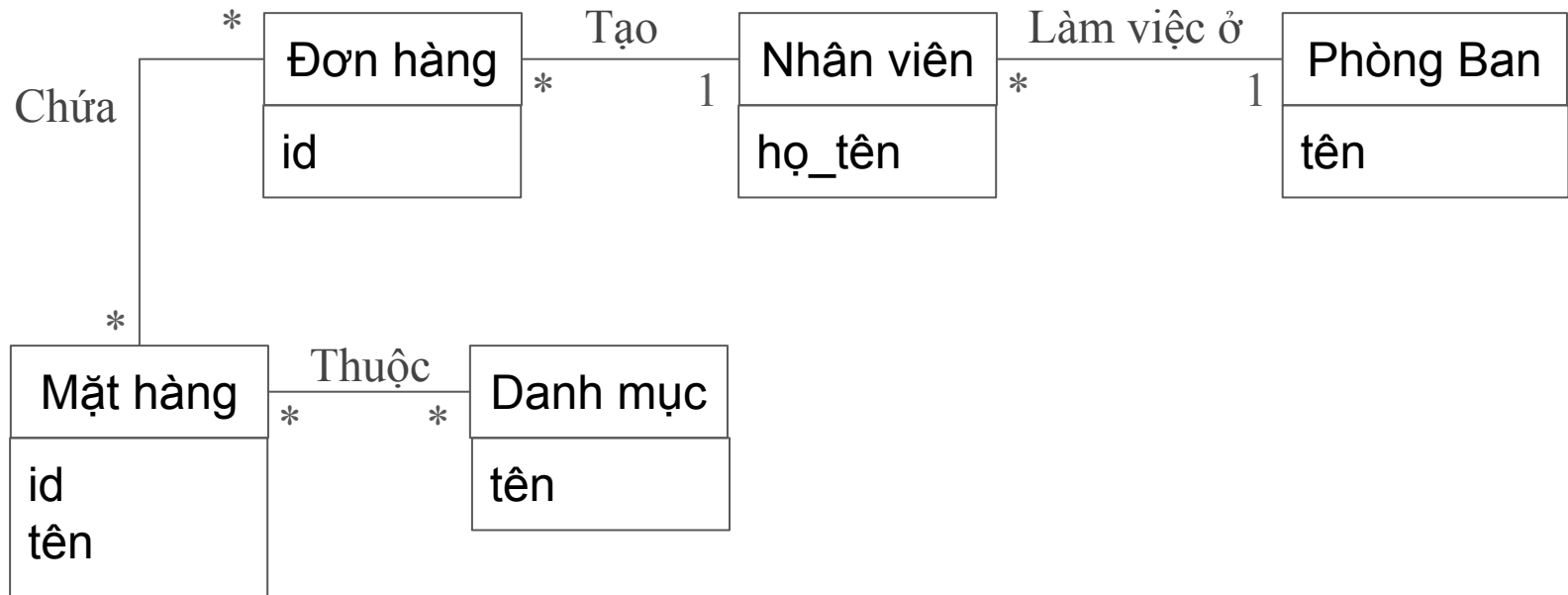
!Lưu ý: Ví dụ này không phải 1 sơ đồ UML

Ví dụ 3.4. Các mối liên hệ₍₂₎



Sơ đồ đối tượng

Ví dụ 3.4. Các mối liên hệ₍₃₎



Sơ đồ lớp

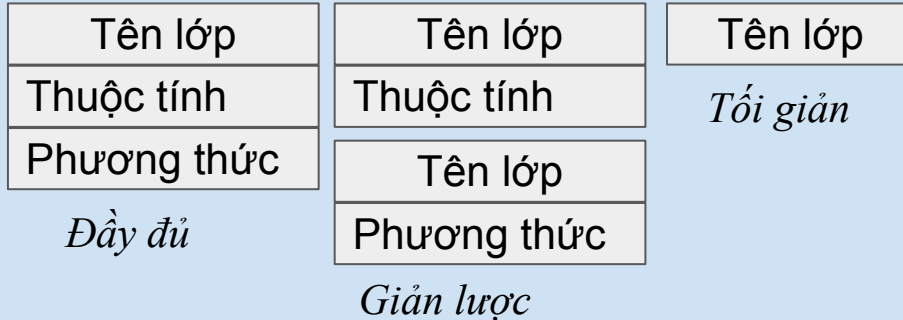
Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập

Sơ đồ lớp: Các thành phần thường gặp

Lớp

Các thành phần của lớp: Tên, thuộc tính, phương thức



(Có thể lược bỏ thuộc tính và/hoặc phương thức)

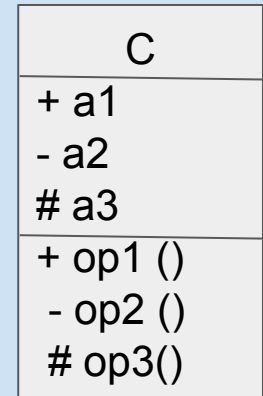
Giới hạn truy cập:

+: Công khai

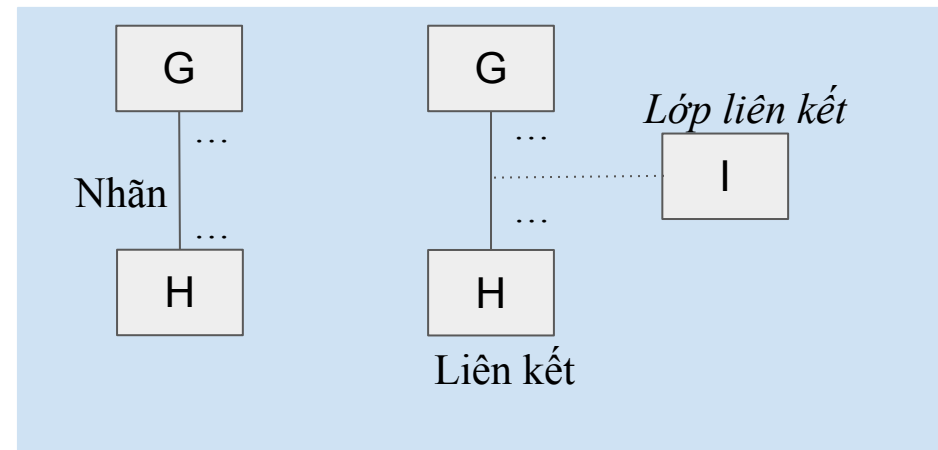
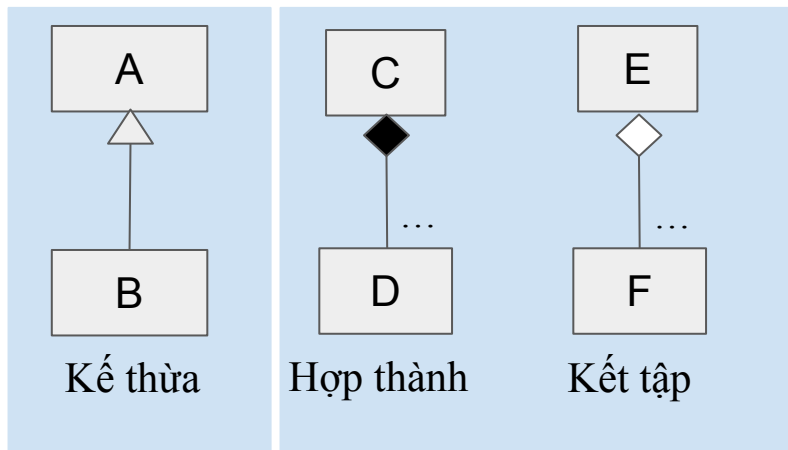
- : Riêng tư

: Bảo vệ

Ví dụ lớp



Quan hệ

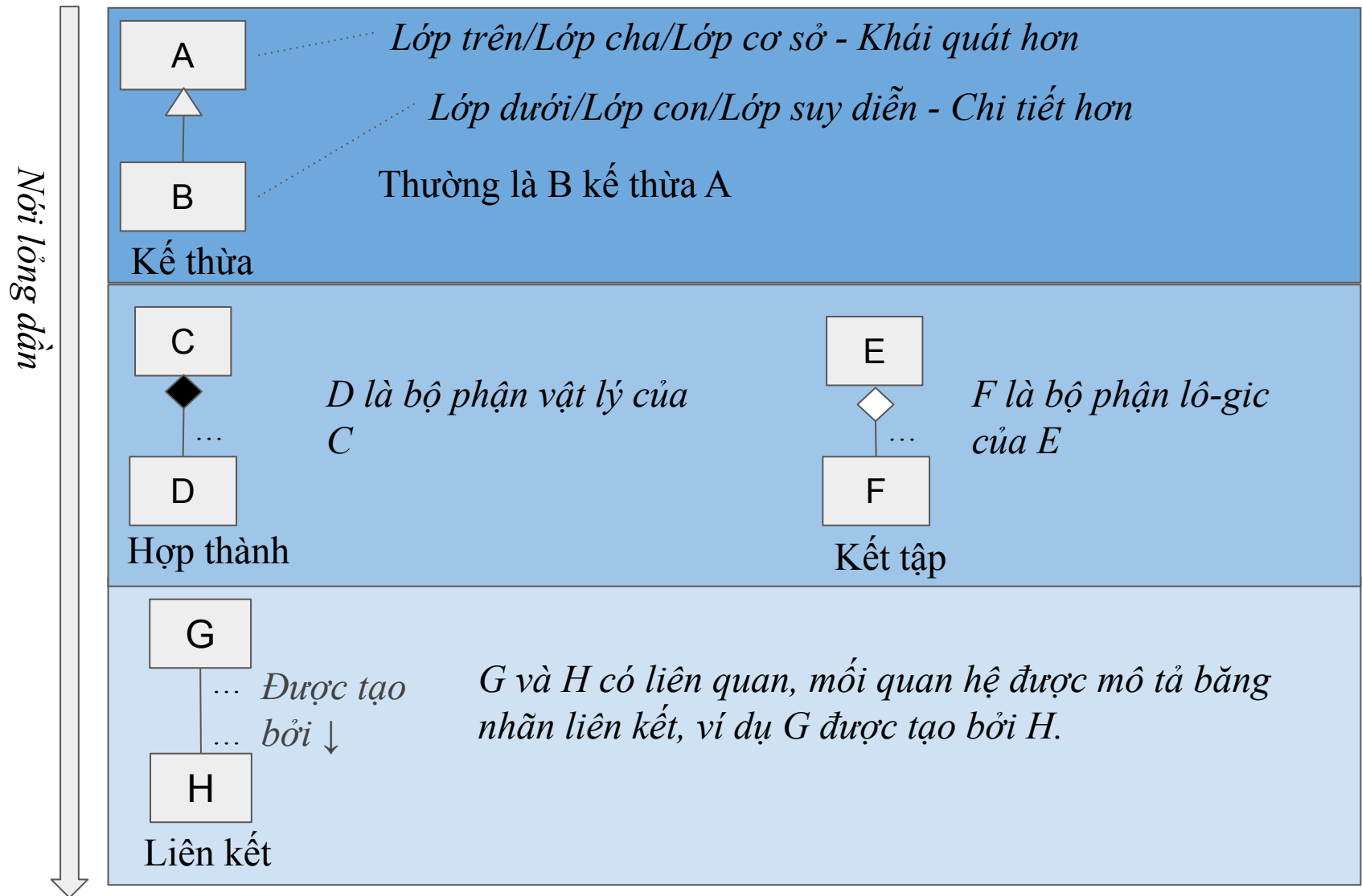


(ở vị trí ... là cơ số - ràng buộc số lượng đối tượng ở mỗi đầu của mỗi quan hệ)

Phân loại thuộc tính

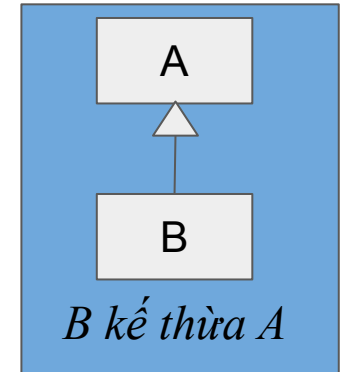
- Khóa
 - Một/bộ giá trị của 1/nhóm thuộc tính khóa xác định duy nhất 1 đối tượng.
 - Ví dụ mỗi mã khách hàng xác định đúng 1 khách hàng.
- Thuộc tính đơn trị
 - Giá trị của thuộc tính không được chia nhỏ thành các thành phần dữ liệu nhỏ hơn nữa.
 - Ví dụ, số lượng sản phẩm là 1 số nguyên.
- Thuộc tính đa trị
 - Bao gồm nhiều thành phần dữ liệu nhỏ hơn.
 - Ví dụ địa chỉ có thể bao gồm số nhà, đường phố, quận v.v..
- Thuộc tính suy diễn
 - Có thể tính được từ những giá trị của các thuộc tính khác.
 - Ví dụ tuổi của khách hàng có thể được tính từ ngày sinh.

Các mối quan hệ

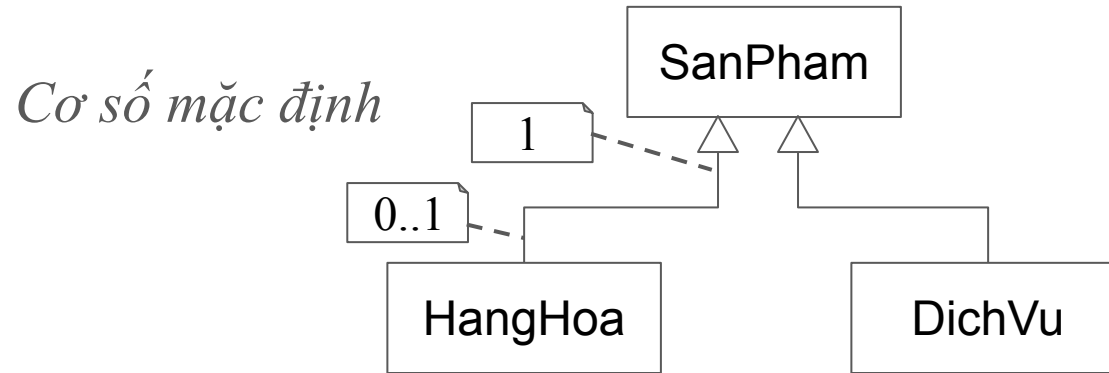


Quan hệ kế thừa

- Hướng mũi tên đi từ lớp dưới đến lớp trên.
- Cho phép tái sử dụng lớp trên, bổ xung thành phần mới và có thể định nghĩa lại phương thức.
 - Trong các triển khai, B thường sở hữu 1 đối tượng kiểu A.
 - B có các giao diện như A, nhưng hành vi có thể không tương thích (do có thể định nghĩa lại, dù không mong đợi).



Ví dụ 3.5. Quan hệ kế thừa



Hàng hóa và Dịch Vụ là các loại Sản Phẩm / HangHoa và DichVu kế thừa SanPham

Các quan hệ bộ phận-tổng thể

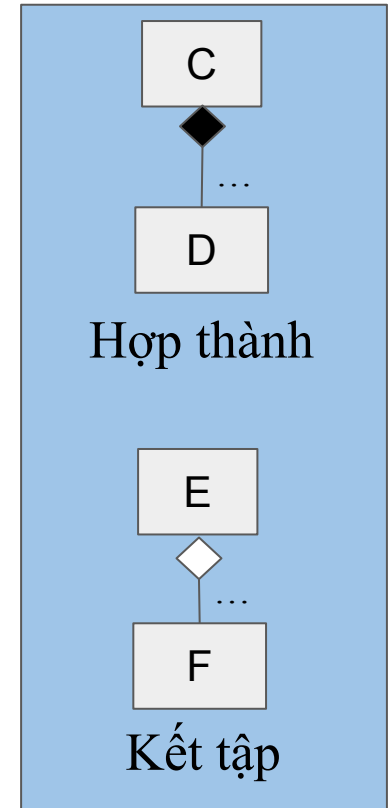
- Mũi tên đi từ bộ phận tới tổng thể.
- Gắn kết cái bộ phận với cái tổng thể.
- Được chia thành 2 mức

- Hợp thành - chặt hơn:

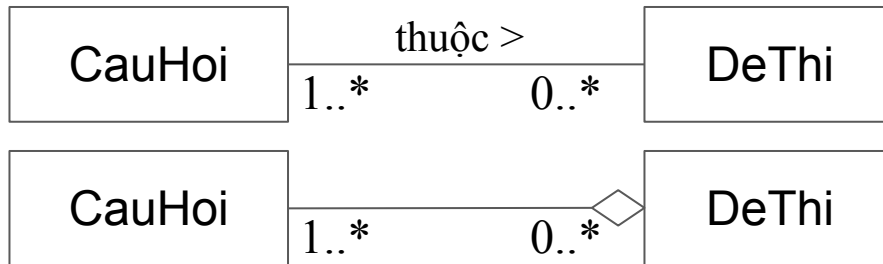
- Tổng thể sở hữu bộ phận
- Đối tượng bộ phận bị hủy khi hủy đối tượng tổng thể.
- Mỗi bộ phận chỉ có thể thuộc 1 tổng thể.
- Ý nghĩa: D là bộ phận vật lý của C.

- Kết tập - lỏng hơn:

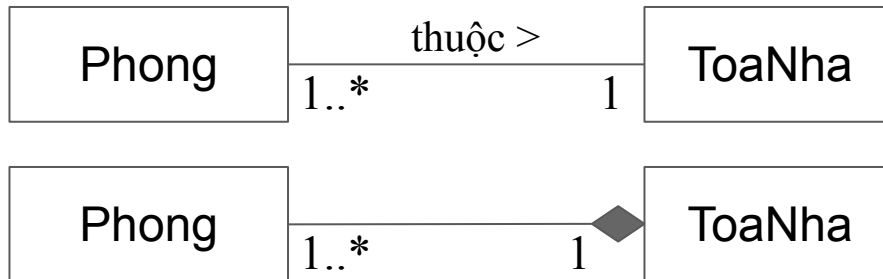
- Tổng thể không sở hữu bộ phận.
- Bộ phận vẫn có thể tồn tại sau khi hủy cái tổng thể.
- Mỗi bộ phận có thể thuộc nhiều tổng thể - chia sẻ
- Ý nghĩa: F là bộ phận lô-gic của E.



Ví dụ 3.6. Quan hệ bộ phận-tổng thể



Mỗi câu hỏi có thể thuộc 1 hoặc nhiều đề thi, và cũng có thể không thuộc đề thi nào.

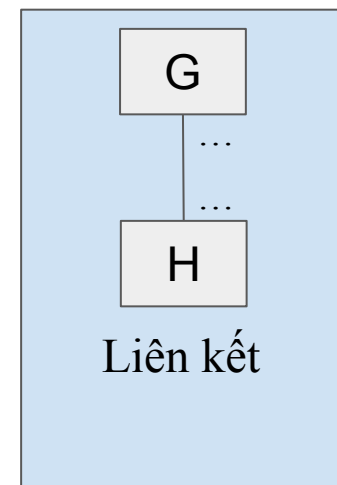


Mỗi phòng phải thuộc đúng 1 tòa nhà.

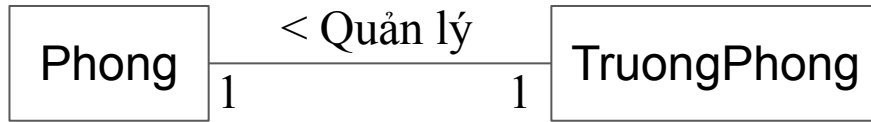
Các đầu mũi tên kim cương (cú pháp hoa mỹ) được ưa chuộng hơn các nhãn để biểu diễn quan hệ bộ phận-tổng thể.

Quan hệ liên kết

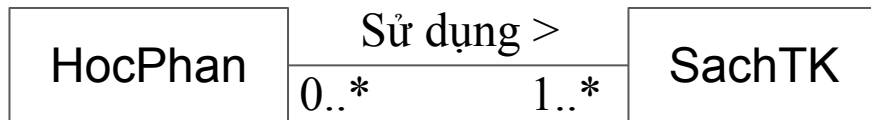
- Loại quan hệ lỏng nhất, ý nghĩa phụ thuộc vào trường hợp cụ thể.
- Được biểu diễn bằng đường nét liên kết nối 2 lớp cùng với tên quan hệ và các tô điểm đầu liên kết:
 - Tên quan hệ - Có thể khác nhau theo mỗi chiều - Có thể bổ xung ký tự ghi chú hướng đọc
 - Các tô điểm, mô tả đầu liên kết:
 - Vai trò của đối tượng ở mỗi đầu
 - Giới hạn hướng duyệt
 - Cơ số - số lượng đối tượng tham gia liên kết
 - Ràng buộc lưu trữ - các đối tượng được lưu như tập hợp, hay danh sách, v.v..



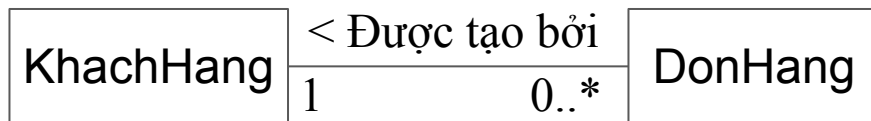
Ví dụ 3.7. Quan hệ liên kết



Mỗi trưởng phòng quản lý đúng 1 phòng.

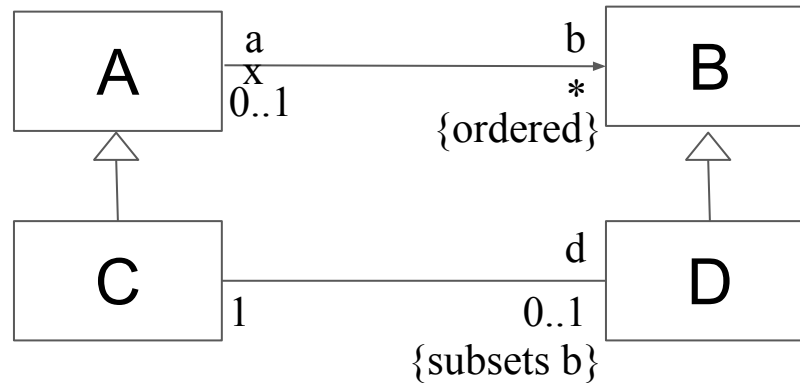


Mỗi học phần sử dụng ít nhất 1 sách tham khảo.



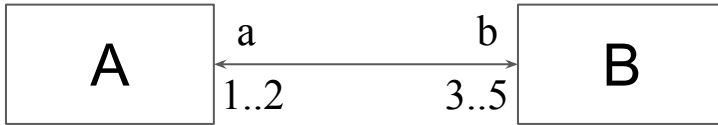
Mỗi đơn hàng được tạo bởi đúng 1 khách hàng.

Ví dụ 3.8. Các tô điểm cho đầu liên kết



- Ý nghĩa của các thành phần tô điểm liên kết:
 - a, b, d là các tên/vai trò của các đầu liên kết.
 - Các cơ số 0..1 đối với a, * đối với b, 1 ở đầu lớp C (không đặt tên), và 0..1 đối với d cho biết số lượng đối tượng có thể tham gia vào mỗi quan hệ với đối tượng ở đầu bên kia.
 - Đầu b là 1 tập có thứ tự.
 - Đầu d là tập con của đầu b, tương đương với:
context C inv: b->includesAll(d)
 - x - đối tượng lớp B không truy cập được đối tác của nó ở phía A; mũi tên - đối tượng lớp A có thể truy cập các đối tác của nó ở phía B; đường nét liên giữa C-D - hướng duyệt chưa xác định.

Ví dụ 3.9. Khả năng truy cập



Liên kết có thể truy cập cả 2 đầu



Liên kết không truy cập được đầu nào



Liên kết không mô tả truy cập



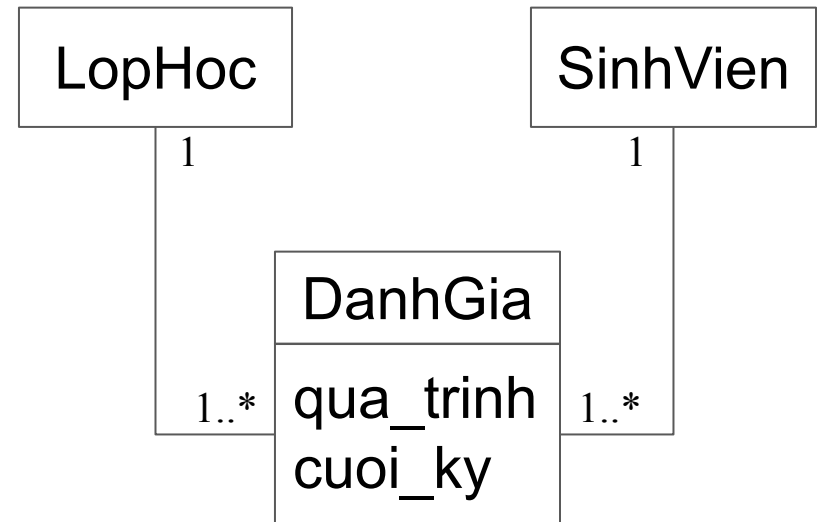
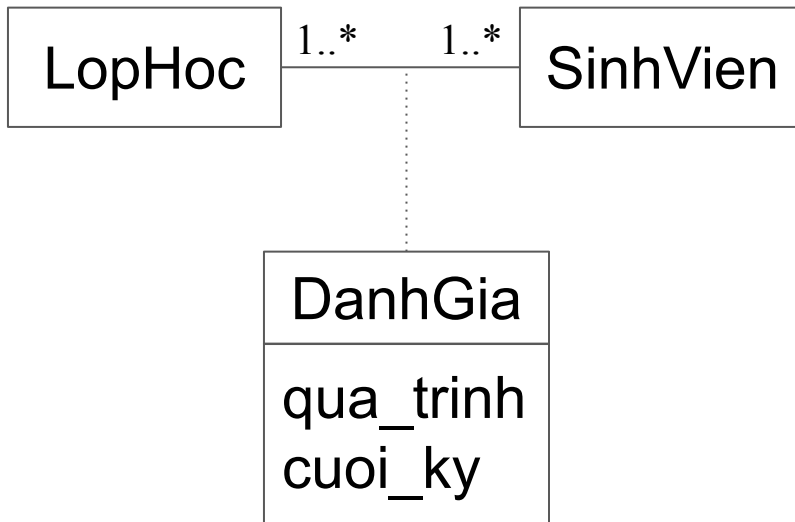
Liên kết có thể truy cập đầu h nhưng không truy cập được đầu g



Liên kết có thể truy cập đầu j, chưa xác định đầu i.

Ví dụ 3.10. Lớp liên kết

Biểu diễn các thuộc tính không phải thành phần của 2 đầu liên kết và đồng thời chỉ tồn tại trong liên kết.



Các phương thức

- Biểu diễn hành vi/trách nhiệm của đối tượng, những việc mà đối tượng có thể/cần thực hiện, những dịch vụ mà nó cung cấp.
- Để giản lược, giữ tính đơn giản của sơ đồ các phương thức có vai trò bổ trợ thường không được biểu diễn
 - Tạo hoặc hủy đối tượng
 - Hàm tạo
 - Hàm Hủy
 - Đọc hoặc thiết lập giá trị (get/set)
 - Lấy giá trị của thuộc tính
 - Thiết lập giá trị thuộc tính

Giới hạn truy cập

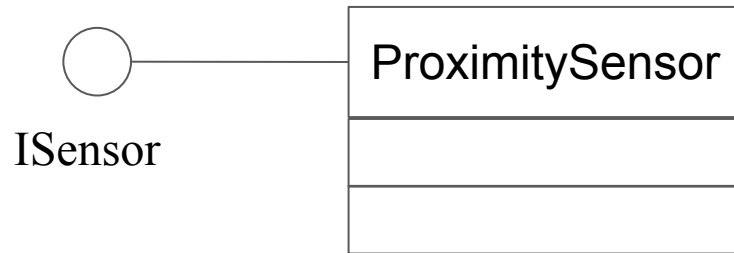
Thường được sử dụng để hỗ trợ đảm bảo tính nhất quán của các biểu diễn

- Công khai (+: public): Không giới hạn quyền truy cập
 - Thường được sử dụng cho phương thức
- Riêng tư (-: private): Giới hạn truy cập trong phạm vi lớp
 - Thường được sử dụng cho thuộc tính.
- Bảo vệ (#: protected): Giới hạn trong phạm vi cây kế thừa
 - *(Có thể có khác biệt giữa các môi trường, ví dụ Java vs. C++).*
 - Phạm vi nằm giữa riêng tư và công khai.

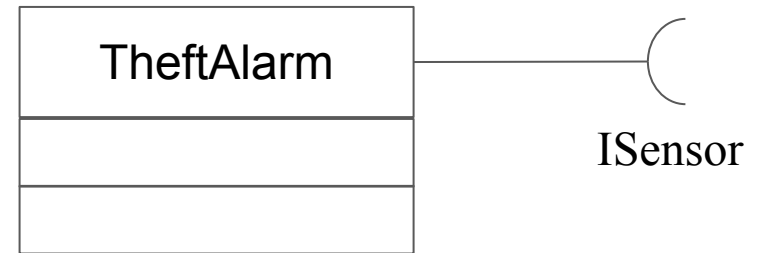
Lớp giao diện

- (Lớp) Giao diện là 1 kiểu lớp đặc biệt:
 - Các phương thức phải được thiết lập truy cập công khai.
 - Không có đối tượng cụ thể.
- Giao diện mô tả 1 tập tính năng nhưng không cung cấp triển khai của các tính năng đó.
 - Sau đó được triển khai trong các lớp cụ thể.
- Trong UML giao diện có thể được biểu diễn bằng hình chữ nhật với nhãn loài hoặc bằng biểu tượng riêng.

Ví dụ 3.11. Giao diện

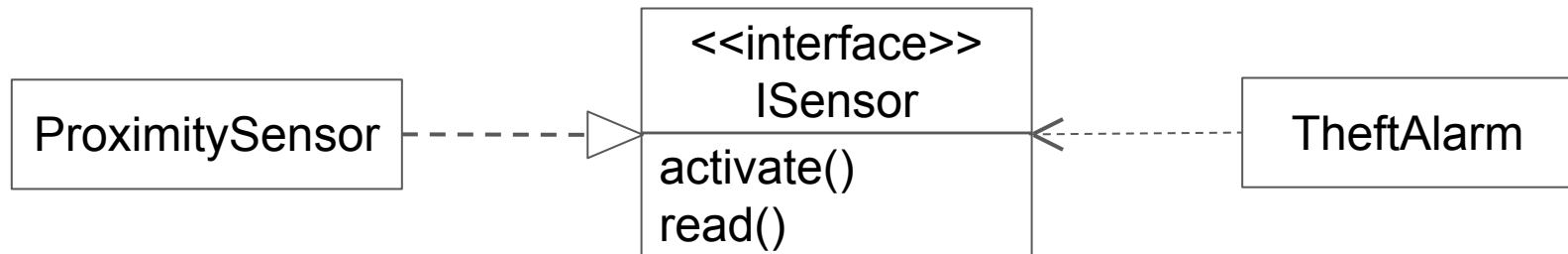


ProximitySensor triển khai/cung cấp giao diện ISensor

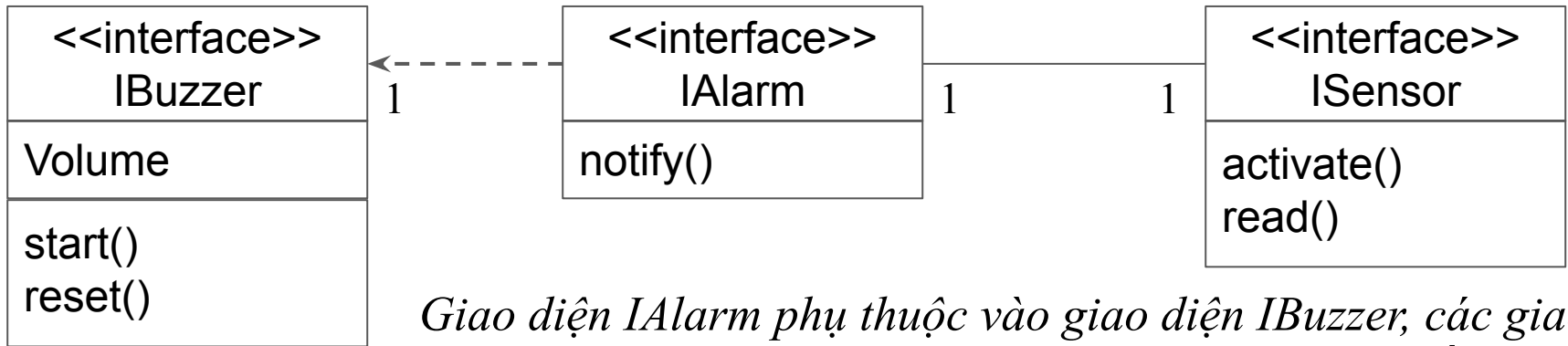


TheftAlarm phụ thuộc vào/sử dụng giao diện ISensor

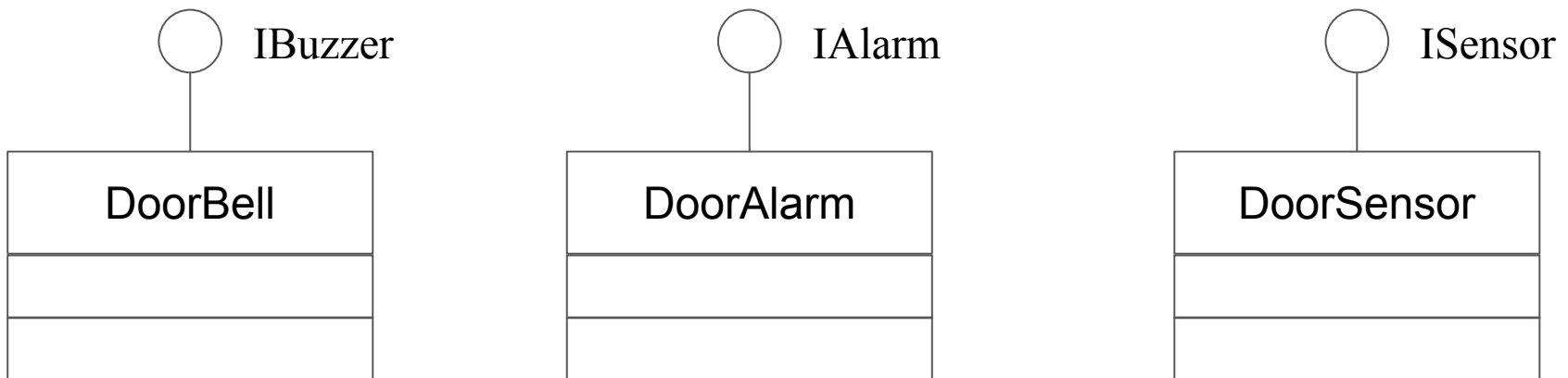
VS.



Ví dụ 3.12. Kết hợp nhiều giao diện



Giao diện IAlarm phụ thuộc vào giao diện IBuzzer, các giao diện IAlarm và ISensor hình thành 1 giao thức 2 chiều.



Các lớp DoorBell, DoorAlarm và DoorSensor tuy được biểu diễn độc lập, nhưng các đối tượng của chúng vẫn có thể tương tác thông qua các giao diện.

Sơ đồ đối tượng

- Biểu diễn các đối tượng với các giá trị cụ thể của các thuộc tính và quan hệ cụ thể giữa các đối tượng
 - Đối tượng = Trường hợp cụ thể (phần tử) của lớp
- Có thể hỗ trợ tiến trình xây dựng sơ đồ lớp: Phân tích tương tác giữa các đối tượng thường dễ hơn phân tích mối quan hệ giữa các lớp.

Ví dụ 3.13. Biểu diễn đối tượng

Đối tượng w thuộc
lớp Window

<u>w:Window</u>
size = (100, 200) visible = true

Lớp Window

<u>obj:</u>
size visible
display() hide()

Tên đối tượng Tên lớp

<u>obj:Type</u>
id = 001 attrib = value

Đối tượng không tên
(thuộc lớp Customer)

<u>:Customer</u>
id = 11-22-333 active = true

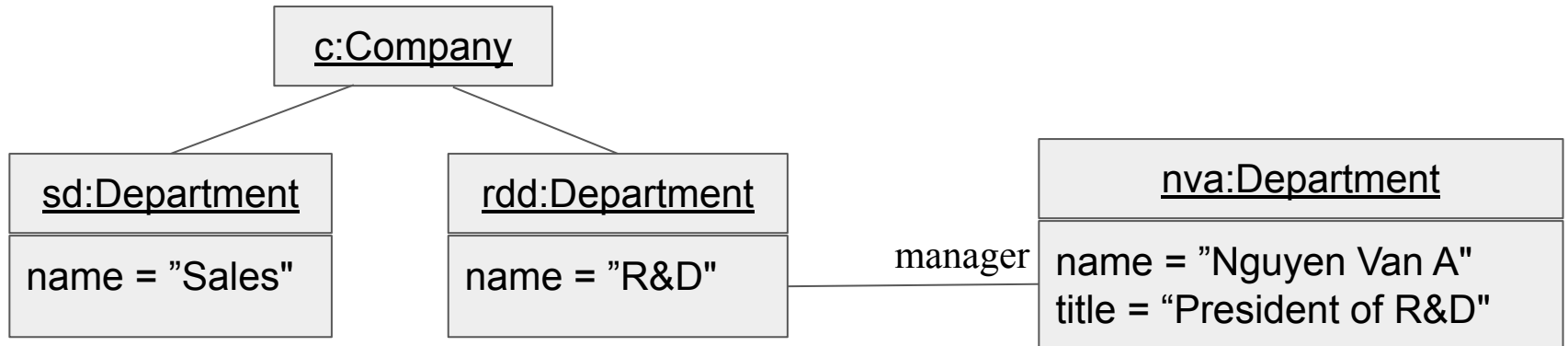
Đối tượng không lớp
(chưa xác định lớp)

<u>obj:</u>
id = 21-23-111 active = true

Đối tượng không tên
không lớp

<u>:</u>
id = 01-12-131 active = true

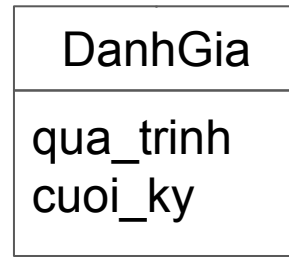
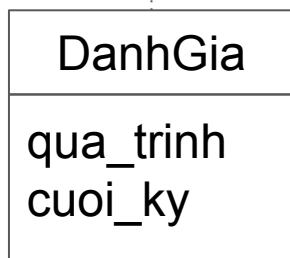
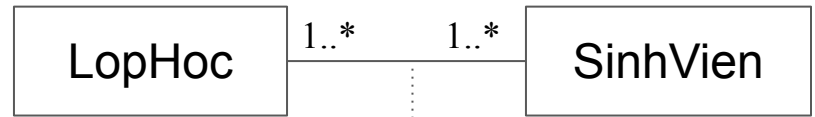
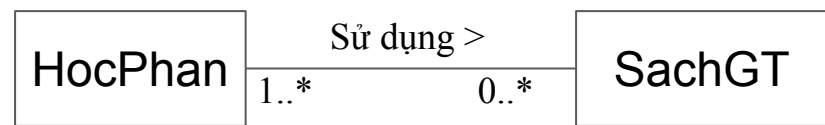
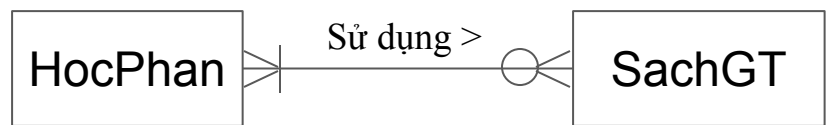
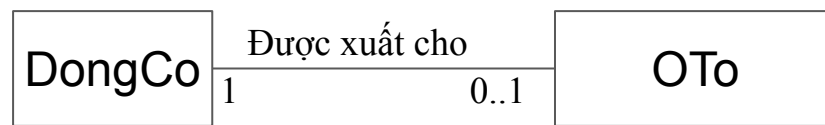
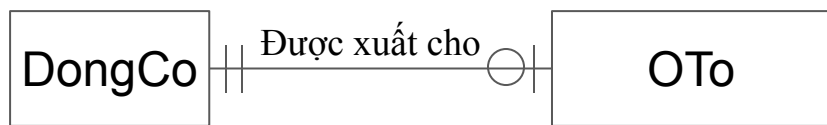
Ví dụ 3.14. Sơ đồ đối tượng



Sơ đồ thực thể liên kết vs. Sơ đồ lớp

- Tổng quan:
 - Sơ đồ thực thể-liên kết (ERD) mô tả các kiểu thực thể cùng với quan hệ giữa các thực thể.
 - Có nhiều hệ ký hiệu khác nhau;
 - Thường được sử dụng trong thiết kế CSDL quan hệ.
 - Sơ đồ lớp mô tả các lớp cùng với quan hệ giữa các đối tượng.
 - Là 1 sơ đồ UML, hệ ký hiệu trong quy chuẩn quốc tế.
 - Được sử dụng cho nhiều mục đích như mô hình hóa lĩnh vực, biểu diễn mô hình dữ liệu hướng đối tượng.
- Phạm vi diễn đạt:
 - Sơ đồ lớp có khả năng diễn đạt mạnh hơn. Có thể bảo toàn ý nghĩa khi chuyển đổi sơ đồ thực thể-liên kết sang sơ đồ lớp.
 - Kiểu thực thể => Lớp lĩnh vực chỉ chứa thuộc tính, không có phương thức.
 - Liên kết => Liên kết
 - Thực thể liên kết => Lớp liên kết.
 - V.V..

Ví dụ 3.15. ERD vs. sơ đồ lớp



(Cơ số trong ERD chỉ có thể là 1 trong 4 trường hợp {0..1, 1, 0..*, 1..*})

Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập

OCL là gì?

- Ngôn ngữ ràng buộc đối tượng - Object Constraint Language
 - Được phát triển từ năm 1995 ở IBM.
 - Sau đó IBM đề xuất và OMG thông qua quy chuẩn.
 - Được sử dụng để đặc tả UML
 - Phiên bản mới nhất song hành với UML:
 - OCL 2.4: <https://www.omg.org/spec/OCL/2.4/PDF>
 - UML 2.4.1: <https://www.omg.org/spec/UML/2.4.1>
- Được sử dụng để bổ xung các chi tiết mô tả cho sơ đồ lớp
 - Các bất biến đối với các thuộc tính, tiền điều kiện và hậu điều kiện đối với các phương thức, v.v..

Các ràng buộc và ngữ cảnh

- Ràng buộc là 1 giới hạn đối với 1 hoặc nhiều thành phần của 1 mô hình hướng đối tượng hoặc hệ thống
 - Tiêu biểu như: Bất biến, tiền điều kiện, hậu điều kiện.
- Ngữ cảnh là cái (lớp, phương thức, v.v..) được thiết lập ràng buộc
 - Ràng buộc được thiết lập ở mức lớp sẽ có hiệu lực cho các đối tượng thuộc lớp đó.
- Ràng buộc có thể được thêm vào như ghi chú trên sơ đồ UML hoặc tách biệt trong biểu mẫu đặc tả riêng.

Bất biến

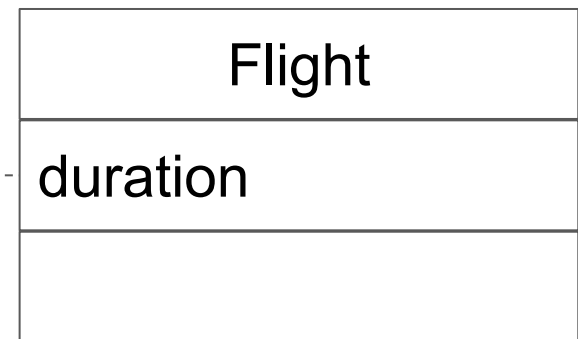
- Bất biến đối với đối tượng là ràng buộc phải đúng trong suốt thời gian tồn tại của đối tượng đó.
- Cú pháp:
context <lớp>
inv [<tên ràng buộc>]:
<Biểu thức ràng buộc OCL>

Các mô tả bất biến tương đương

- **context** Flight **inv**: self.duration < 4
 - self - Đối tượng đang được kiểm tra ràng buộc.
- **context** Flight **inv**: duration < 4
- **context** Flight **inv** flightDuration: duration < 4
 - flightDuration - Tên được đặt cho ràng buộc

*Ràng buộc OCL như
ghi chú trong sơ đồ
UML*

{duration < 4}



Tiền điều kiện và hậu điều kiện

- Được áp dụng cho các phương thức
- Tiền điều kiện phải đúng trước khi thực hiện phương thức.
- Hậu điều kiện phải đúng sau khi thực hiện phương thức.
- Cú pháp:

context <lớp>::<phương thức>(<các tham số>)

pre|post [<tên ràng buộc>]:

<Biểu thức ràng buộc OCL>

Ví dụ 3.16. Mô tả điều kiện

context Flight::shiftDeparture(t: Integer)

pre: $t > 0$

post: $\text{result} = \text{departureTime}@pre + t + \text{duration}$

-- Chúng ta muốn trả về thời gian đến đã cập nhật

-- @pre - chỉ được sử dụng trong hậu điều kiện, để chỉ định trạng thái ban đầu.

Flight
departTime /arrivalTime duration
shiftDeparture(t:Integer)

Kế thừa ràng buộc

Các hệ quả của nguyên lý khả thay Liskov

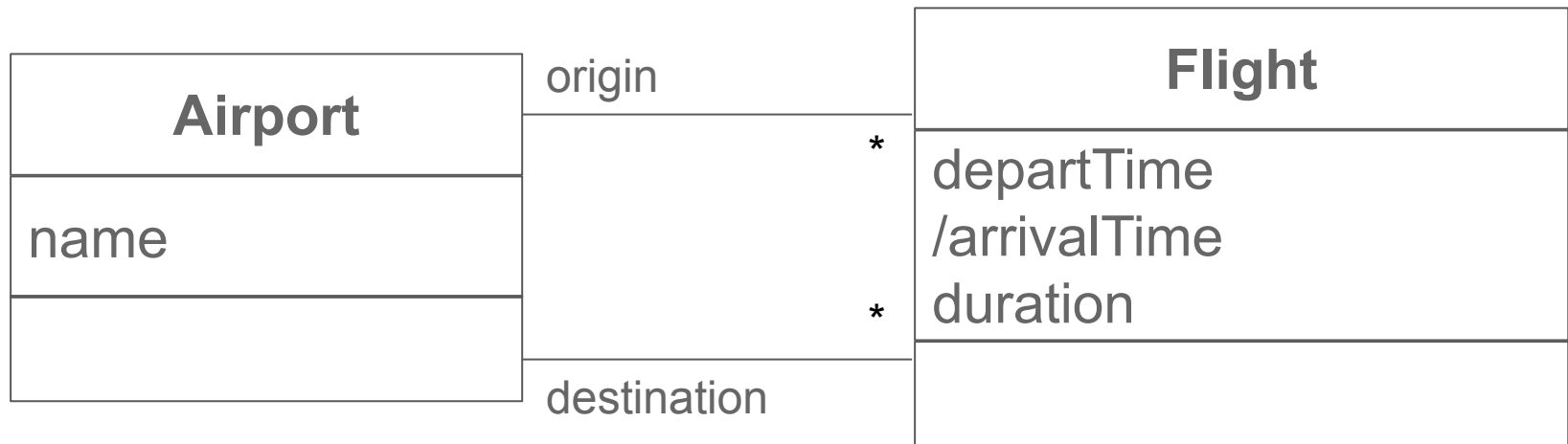
- Các lớp dưới kế thừa các bất biến của lớp trên
 - Lớp dưới có thể tăng cường các bất biến nhưng không được nói lỏng chúng.
- Lớp dưới có thể nói lỏng tiền điều kiện trong lớp trên
 - **context** SuperClass::foo(i:Integer) **pre:** $i > 100$
 - **context** SubClass::foo(i:Integer) **pre:** $i > 0$
 - \Rightarrow Lớp dưới vẫn có thể xử lý dữ liệu đầu vào như lớp trên.
- Lớp dưới có thể tăng cường hậu điều kiện trong lớp trên
 - **context** SuperClass::foo(): Integer **post:** $result > 0$
 - **context** SuperClass::foo(): Integer **post:** $result > 10$
 - \Rightarrow Biểu thức gọi phương thức bằng giao của diện lớp trên vẫn thu được kết quả > 0 dù đối tượng thuộc lớp dưới.

Các thành phần của biểu thức ràng buộc

- Các kiểu cơ sở:
 - Boolean
 - Integer
 - Real
 - String
- Các lớp từ mô hình UML và các thành phần của chúng
 - Thuộc tính
 - Thao tác truy xuất
- Liên kết từ mô hình UML
 - Bao gồm tên vai trò ở mỗi đầu liên kết

Các biểu thức duyệt

- Duyệt theo liên kết - Được sử dụng để truy cập các đối tượng liên quan, bắt đầu từ đối tượng ngữ cảnh



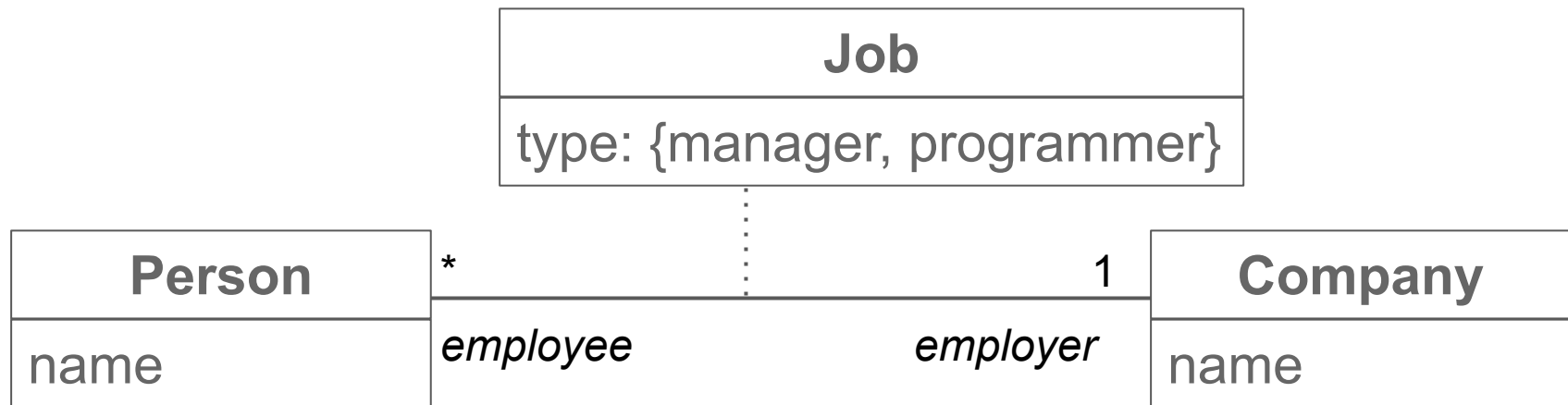
context Flight

inv: origin != destination

inv: origin.name == "Hanoi"

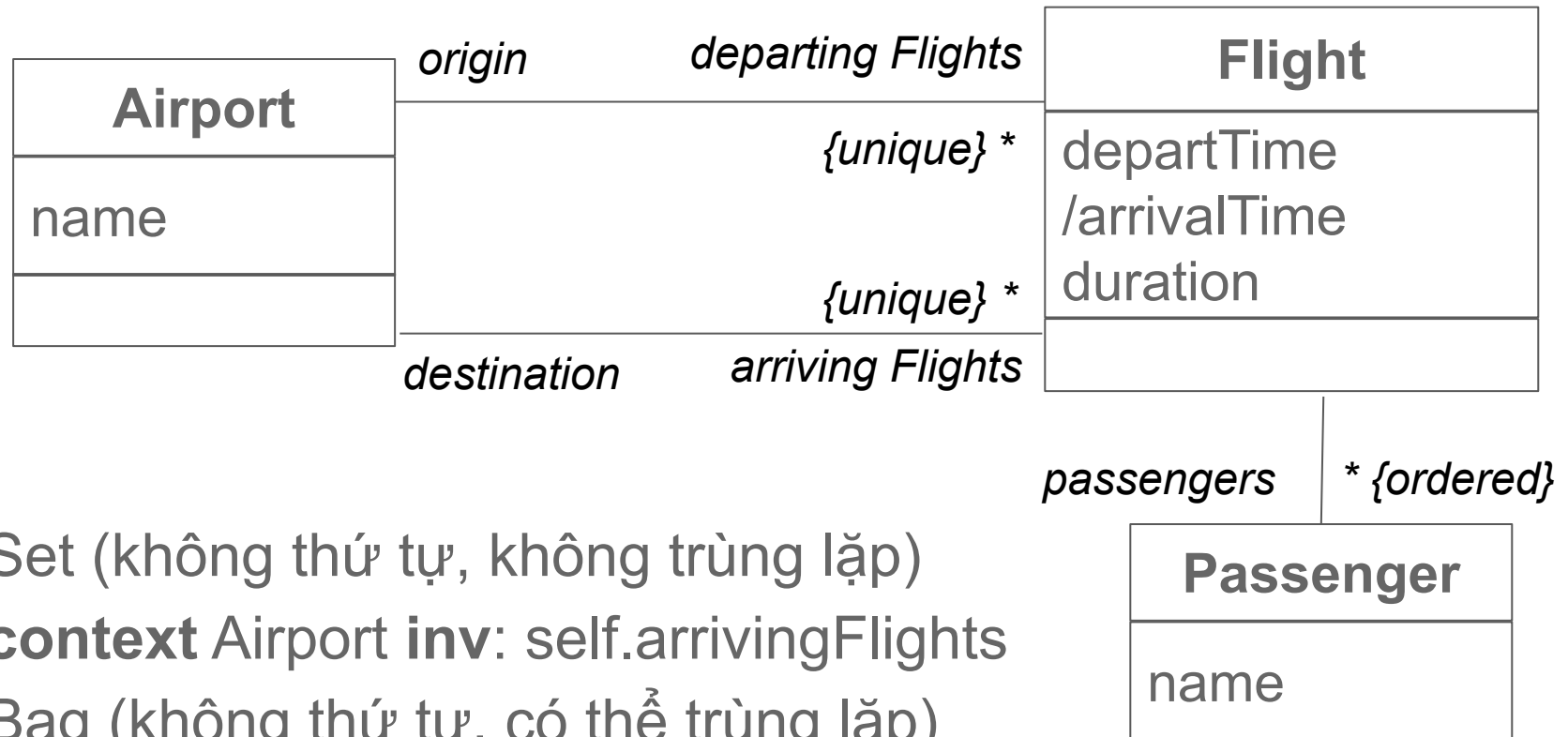
Duyệt lớp liên kết

- Lớp liên kết không có tên vai trò, vì vậy biểu thức OCL phải sử dụng tên lớp, bắt đầu với chữ thường.



```
context Person inv:
if self.name = "Ivan Ivanov" then
    job.type = #manager
else
    job.type = #programmer
endif
```

Các cấu trúc lưu trữ trong OCL



- Set (không thứ tự, không trùng lặp)
context Airport **inv**: self.arrivingFlights
- Bag (không thứ tự, có thể trùng lặp)
context Airport **inv**: self.arrivingFlights.passengers.name
- Sequence (có thứ tự, có thể trùng lặp)
context Flight **inv**: self.passengers

Các thao tác với cấu trúc lưu trữ

- **collect:** `collection->collect(expr)` hoặc `collection.expr`
 - Trả về 1 túi (bag) giá trị của biểu thức `expr` trên các phần tử của `collection`.
- **select:** `collection->select(expr)`
 - Trả về tập con của `collection` bao gồm các phần tử có `expr` đúng.
- **forAll:** `collection->forAll(expr)`
 - Thao tác `forAll` đúng nếu `expr` đúng với tất cả các phần tử của `collection`
- **exists:** `collection->exists(expr)`
 - Đúng nếu `expr` đúng với ít nhất 1 phần tử trong `collection`

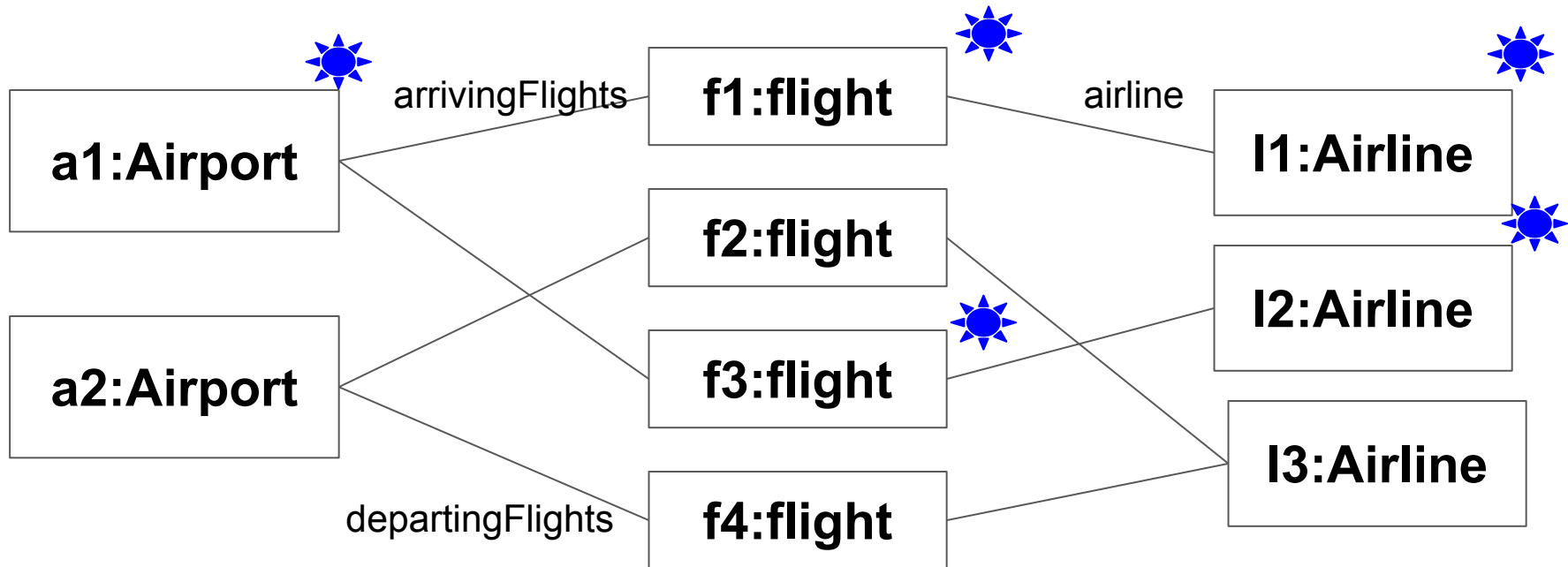
...

Ví dụ 3.17. Thao tác *collect*



context Airport **inv**:

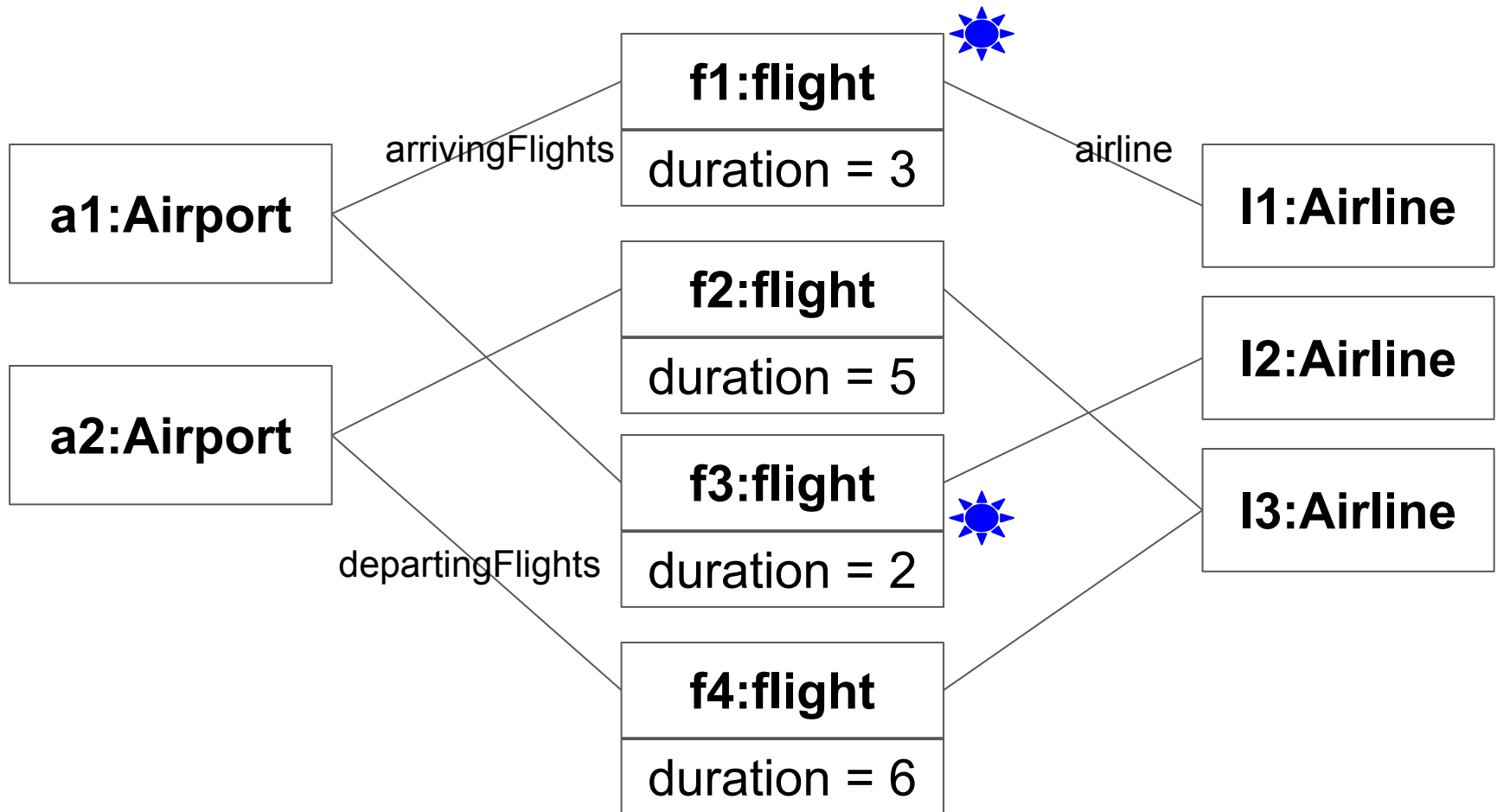
`self.arrivingFlights->collect(airline)->notEmpty()`



Ví dụ 3.18. Thao tác *select*

context Airport **inv**:

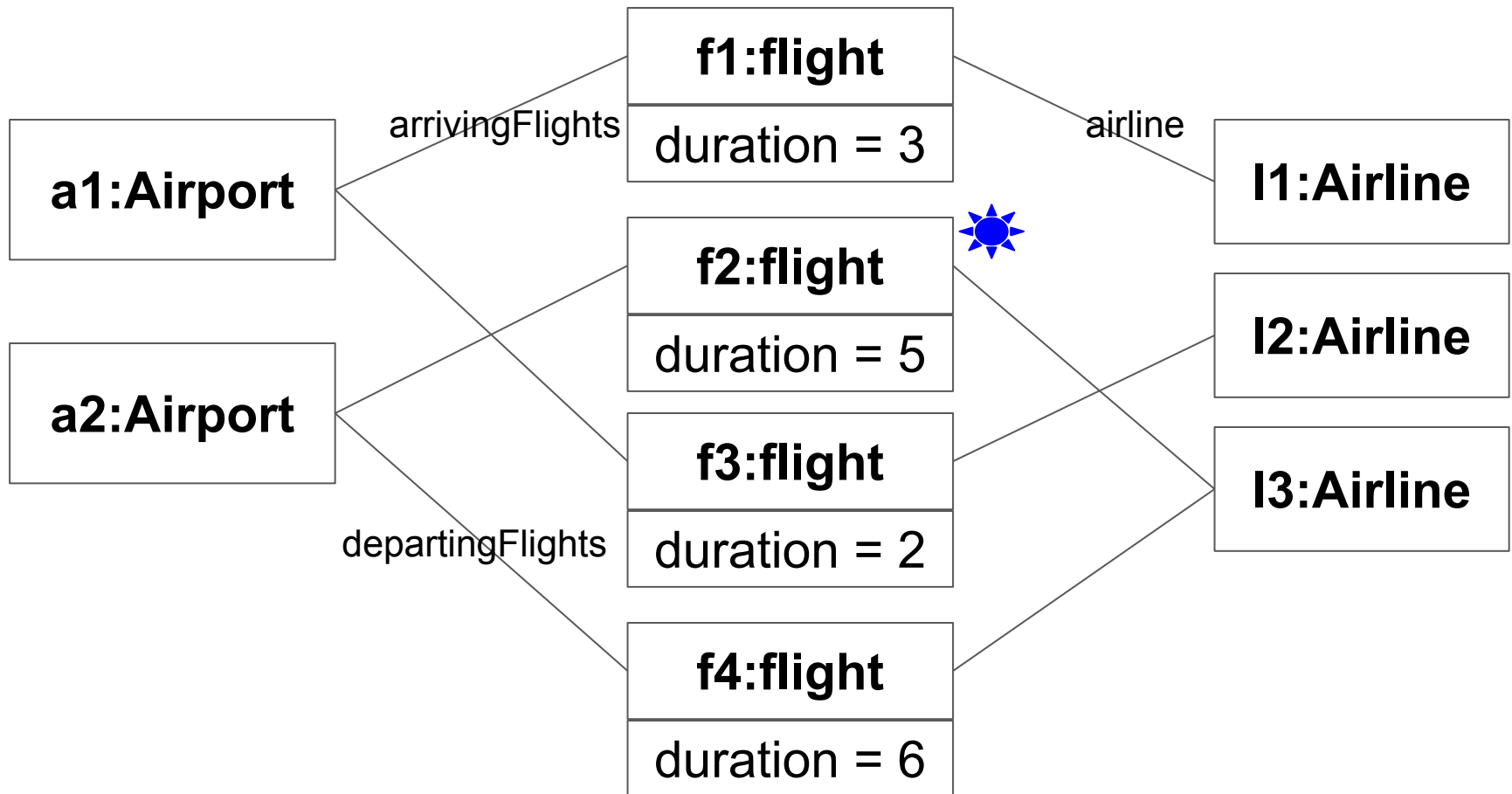
`self.departingFlights->select(duration < 4)->notEmpty()`



Ví dụ 3.19. Thao tác *exists*

context Airport **inv**:

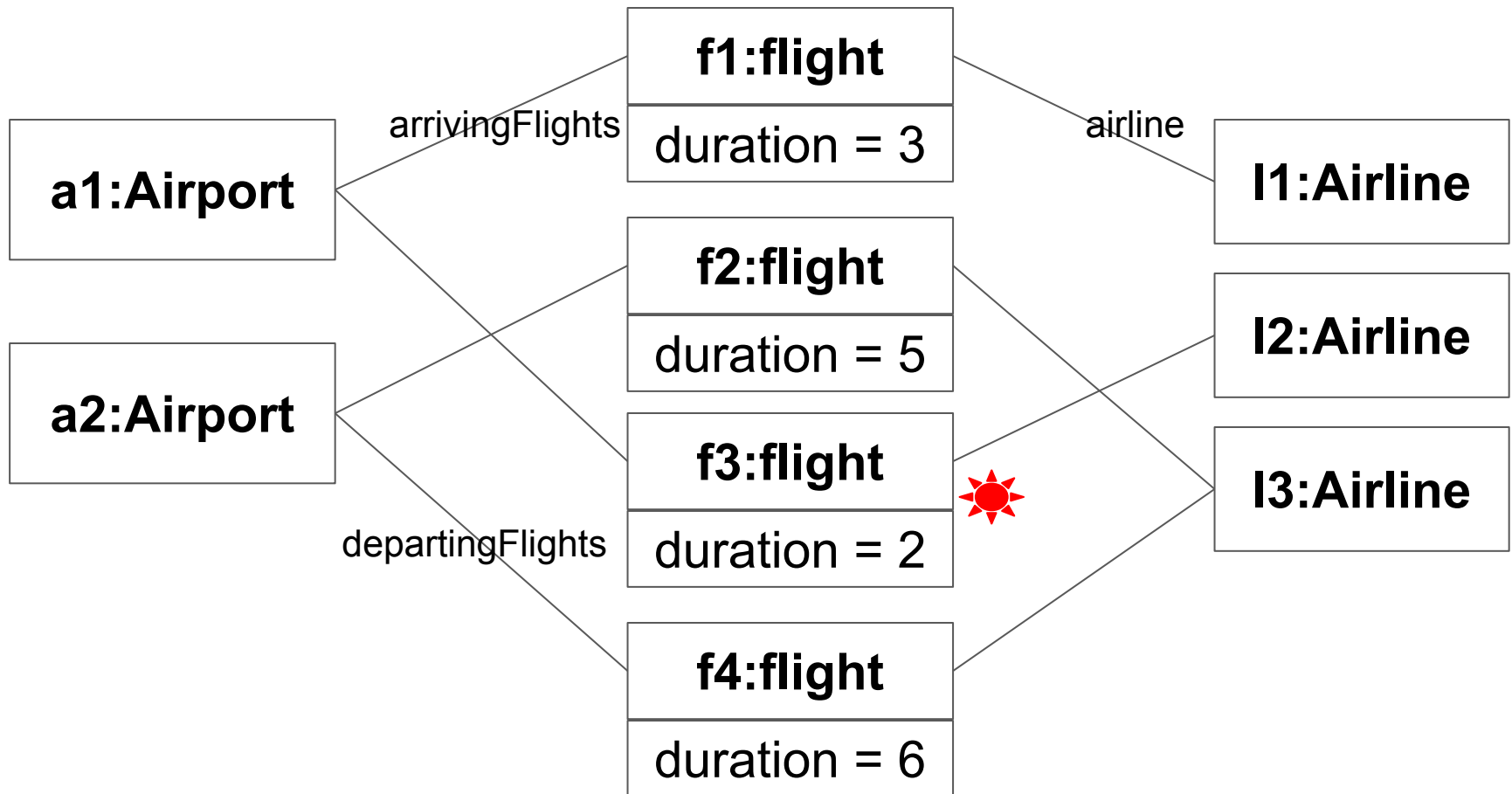
`self.departingFlights->exists(duration = 5)`



Ví dụ 3.20. Thao tác *forall*

context Airport **inv**:

self.departingFlights->forall(duration > 2)



Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập

Thẻ CRC

Class Responsibilities and Collaborations

- Được sử dụng để đặc tả lớp
 - Còn có thể được sử dụng để đóng vai đối tượng trong kiểm tra các kịch bản ca sử dụng
- Các trách nhiệm:
 - Được biểu diễn như các phương thức
 - Biết gì? - các thao tác tra cứu thông tin
 - Làm gì? - những thao tác xử lý
- Lưu dữ liệu gì:
 - Danh sách thuộc tính

Thẻ CRC₍₂₎

- Đối tác:
 - Các đối tượng làm việc cùng nhau để đáp ứng 1 yêu cầu
 - Đối tượng yêu cầu (khách)
 - Đối tượng phản hồi (chủ)
 - Các tương tác được quy định theo hình thức tương tự hợp đồng
 - Nội dung chi tiết được trình bày trong phần thiết kế


Mẫu thẻ CRC

Tên lớp:	ID:	Kiểu:
Mô tả:		CSD:
Các trách nhiệm		Các đối tác
Các thuộc tính		
Các mối quan hệ		
Kế thừa: Bộ phận-tổng thể: Liên kết:		

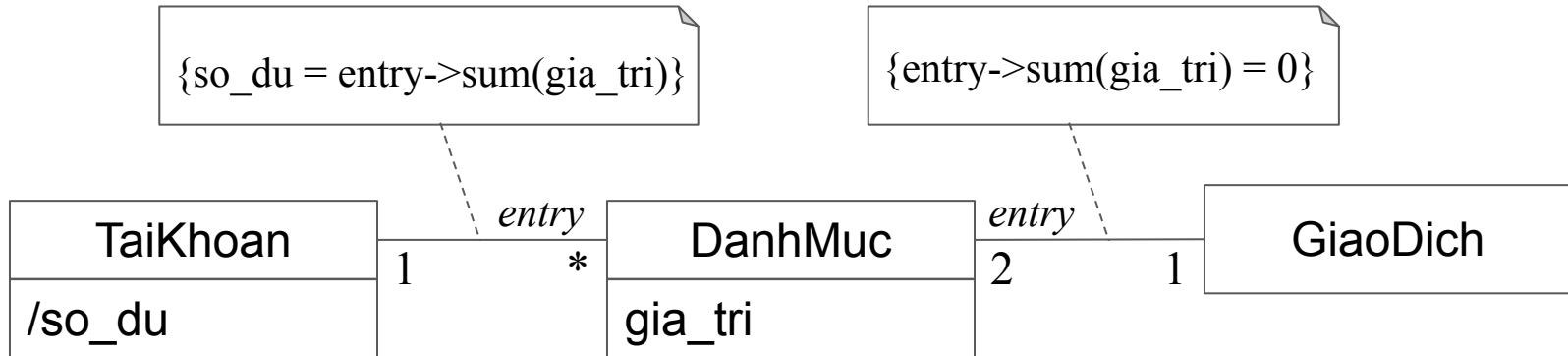
Ví dụ 3.21. Đặc tả lớp với thẻ CRC

Tên lớp: Bản khảo sát	ID: 1	Kiểu: Lĩnh vực
Mô tả: Danh sách câu hỏi và câu trả lời được sử dụng để xác định dịch vụ.		CSD: UC01
<p style="text-align: center;">Các trách nhiệm</p> <p>Hiển thị câu hỏi Lưu câu trả lời</p>		<p style="text-align: center;">Các đối tác</p> <p>Câu hỏi khảo sát --</p>
<p style="text-align: center;">Các thuộc tính</p> <p>Mã số Danh sách câu hỏi Danh sách câu trả lời</p>		
<p style="text-align: center;">Các mối quan hệ</p> <p>Kế thừa: -- Bộ phận-tổng thể: Bao gồm các câu hỏi khảo sát Liên kết: Khách hàng, nhu cầu dịch vụ</p>		

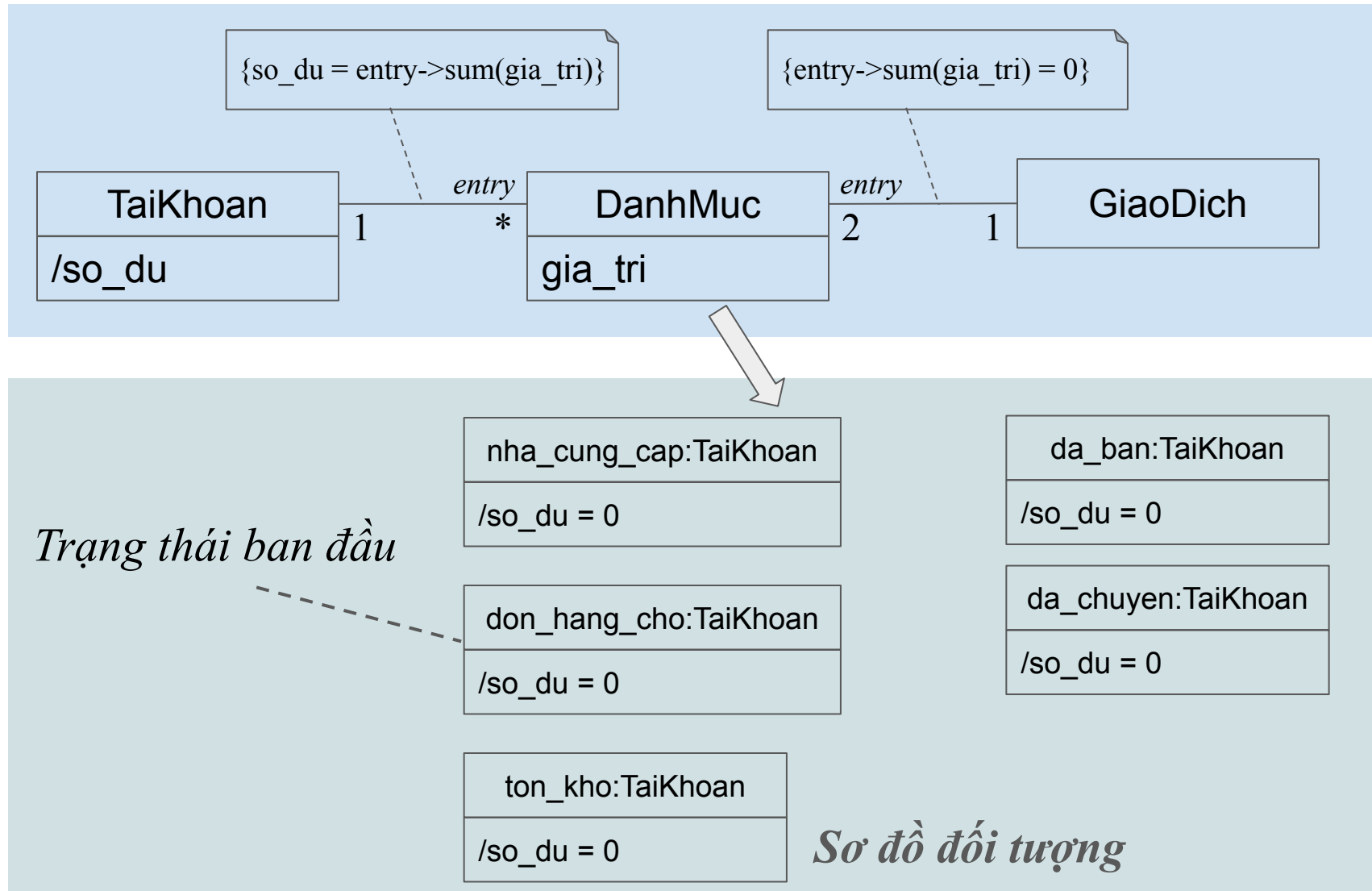
Nội dung

- Các khái niệm
 - Các kỹ thuật
 - Các sơ đồ
 - Mô tả ràng buộc với OCL
 - Đặc tả lớp với thẻ CRC
 - Mẫu phân tích
 - Bài tập
- 

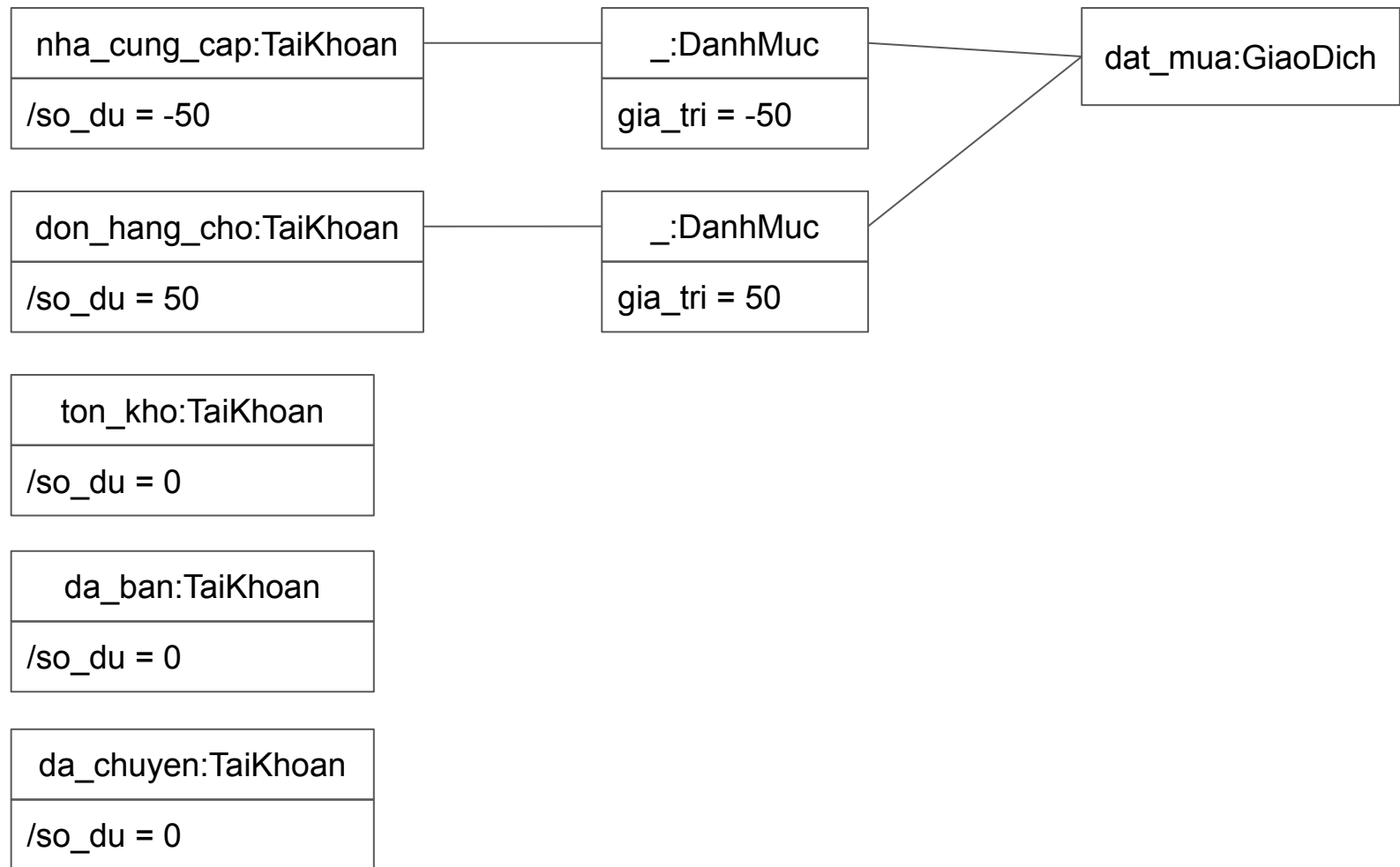
Mẫu tài khoản/giao dịch



Ví dụ 3.22. Các đối tượng tài khoản/giao dịch

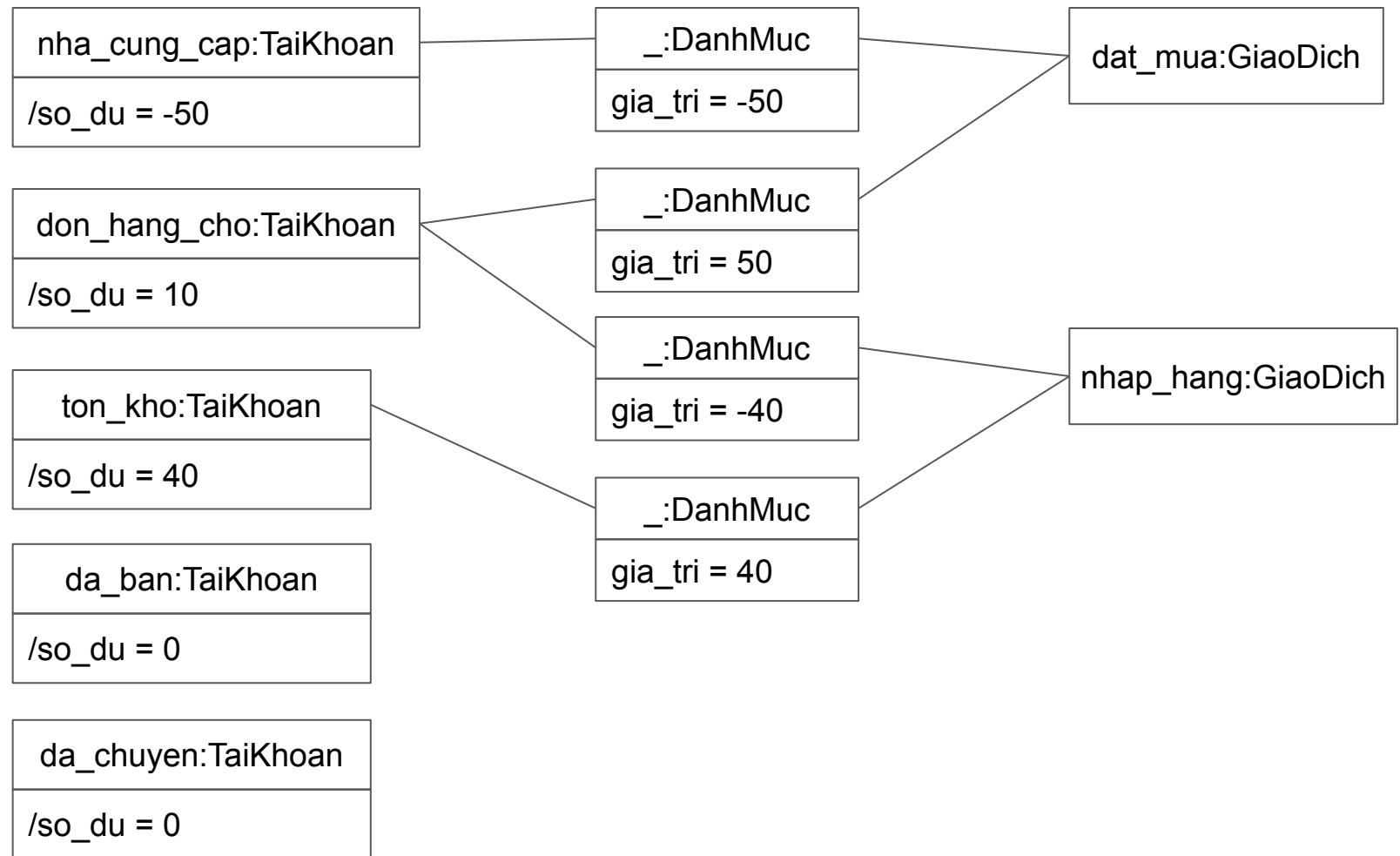


Ví dụ 3.22. Các đối tượng tài khoản/giao dịch₍₂₎



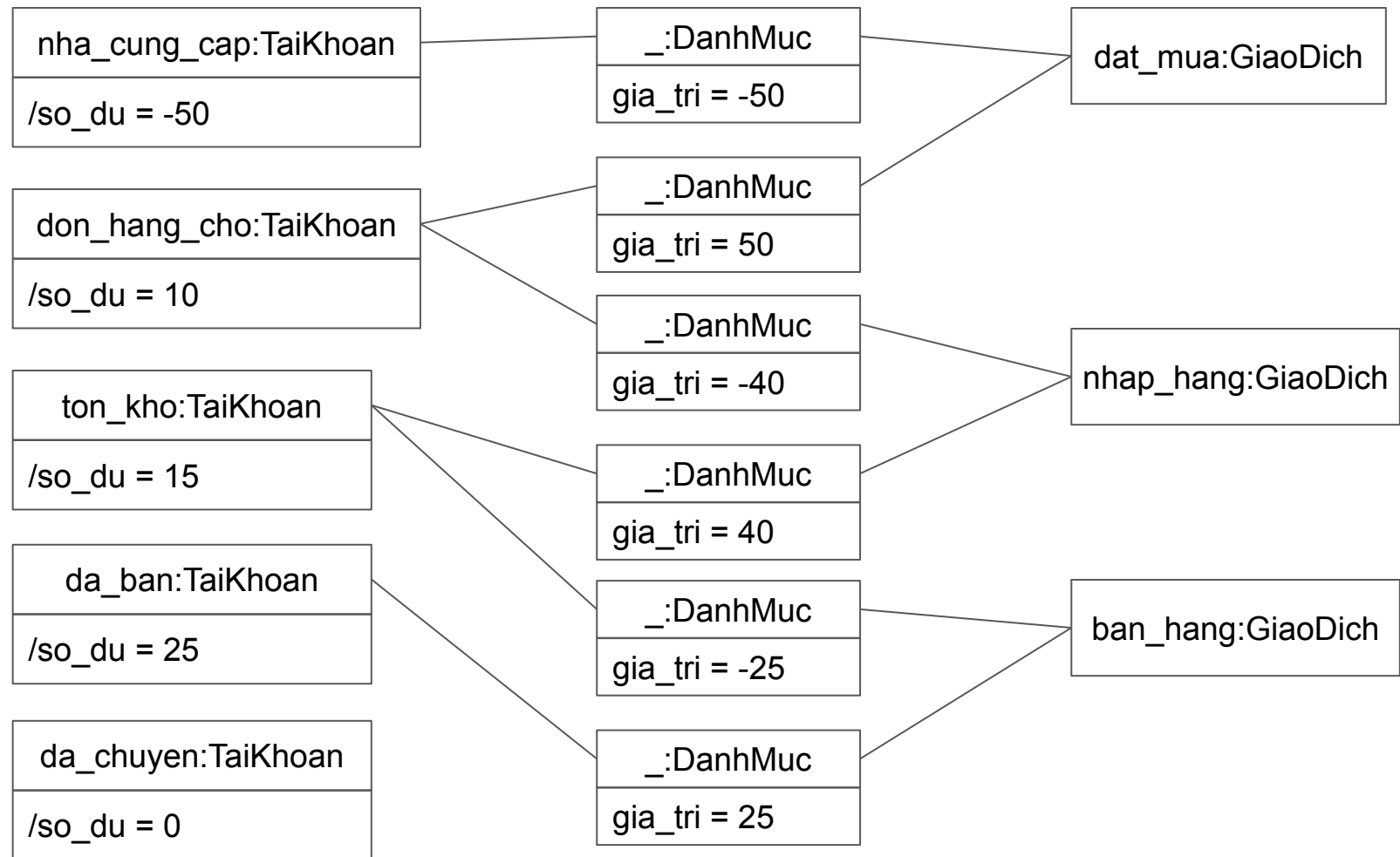
Trạng thái sau khi đặt mua từ nhà cung cấp

Ví dụ 3.22. Các đối tượng tài khoản/giao dịch⁽³⁾



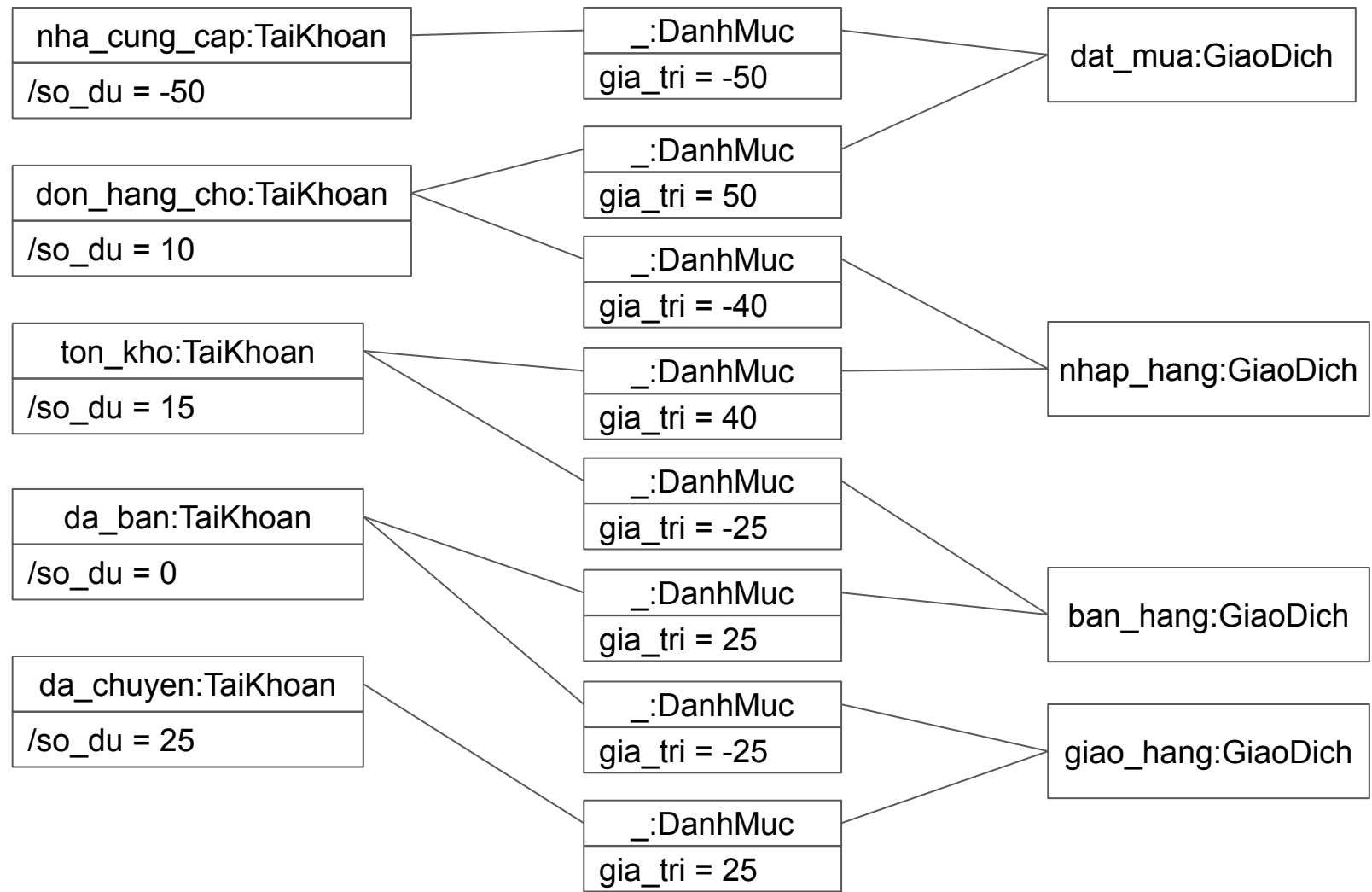
Trạng thái sau khi nhập 1 phần hàng từ nhà cung cấp

Ví dụ 3.22. Các đối tượng tài khoản/giao dịch⁽⁴⁾



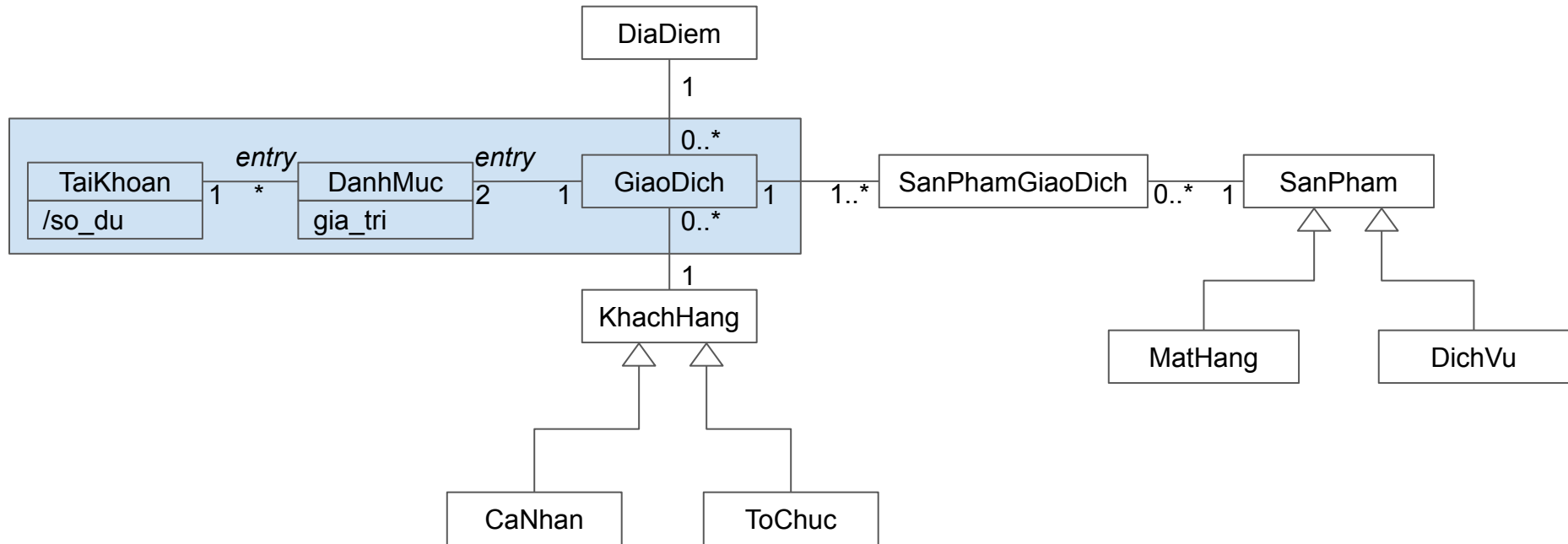
Trạng thái sau khi bán cho khách hàng

Ví dụ 3.22. Các đối tượng tài khoản/giao dịch⁽⁵⁾

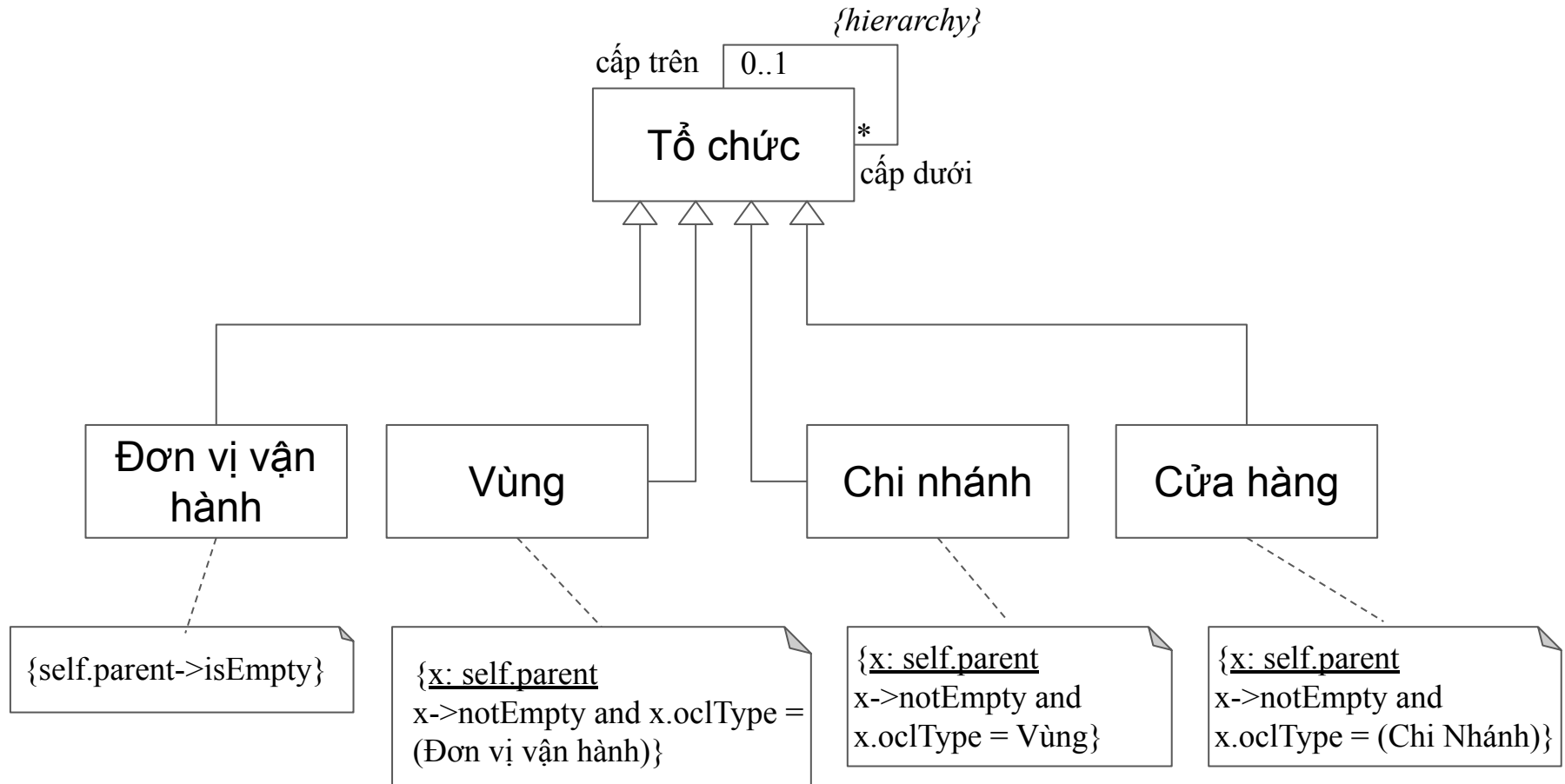


Trạng thái sau khi giao hàng cho khách

Ví dụ 3.23. Sử dụng mẫu tài khoản/giao dịch

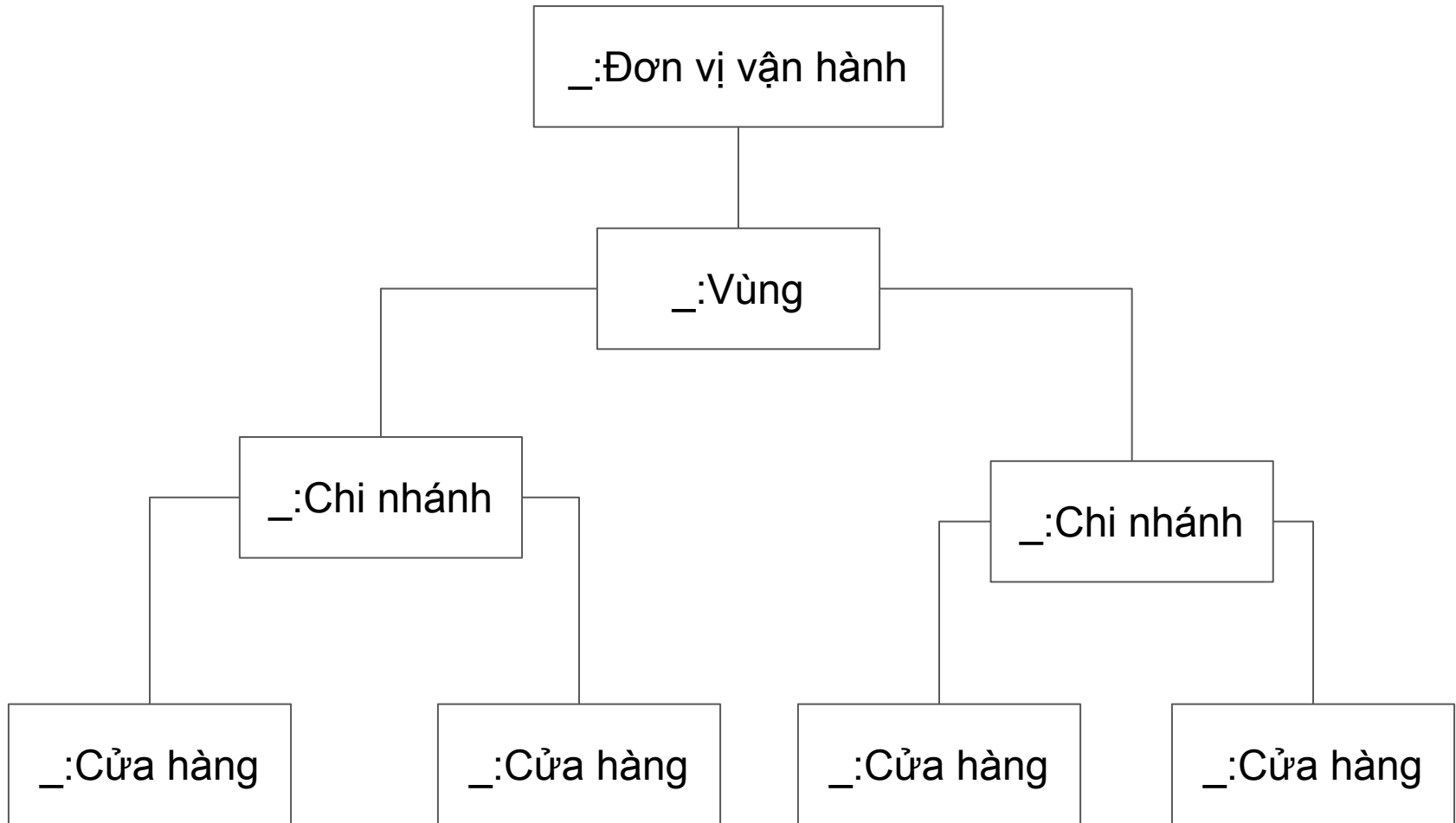


Mẫu phân cấp tổ chức



*Có ích trong trường hợp cần thay đổi cấu trúc phân cấp, ví dụ
xóa mức vùng*

Ví dụ 3.24. Các đối tượng phân cấp tổ chức



Ví dụ 3.25. Mẫu đơn hàng



Nội dung

- Các khái niệm
- Các kỹ thuật
- Các sơ đồ
- Mô tả ràng buộc với OCL
- Đặc tả lớp với thẻ CRC
- Mẫu phân tích
- Bài tập



Bài tập

Xác định các thành phần thông tin, biểu diễn dữ liệu bằng đối tượng, vẽ sơ đồ lớp.

Công ty cổ phần Thanh Hà
Số 123, Phường Yên Hòa,


Mẫu số 02-VT
(Ban hành kèm thông tư số 133/2016/TT-BTC
Ngày 26/08/2016 của Bộ Tài Chính)

PHIẾU XUẤT KHO

Ngày 19 tháng 3 năm 2001
Số: XK0026

Nợ 156
Có 135

- Họ tên người nhận: Nguyễn Văn A
- Địa chỉ:
- Lý do xuất kho: Bán hàng theo hóa đơn 0000023
- Xuất tại kho: K1 Địa điểm:

STT	Tên sản phẩm	Mã số	Đơn vị	Số lượng 		Đơn giá	Thành tiền
				Yêu cầu	Thực xuất		
1	Xi măng HP	HP-30	Bao	6		50 000	300 000
2	Thép cây	P12	Cây	10		200 000	2000 000
	Tổng						2 300 000

- Tổng số tiền (viết bằng chữ): Hai triệu ba trăm nghìn đồng chẵn
- Số chứng từ gốc kèm theo:

Người lập phiếu
(Ký, họ tên)

Người nhận hàng
(Ký, họ tên)

Thủ kho
(Ký, họ tên)

Kế toán trưởng
(Ký, họ tên)

Giám đốc
(Ký, họ tên)

Xác định đối tượng và vẽ sơ đồ lớp

Xác định các thành phần thông tin, biểu diễn dữ liệu bằng đối tượng, vẽ sơ đồ lớp.

Công ty cổ phần Thanh Hà
Số 123, Phường Yên Hòa,

Mẫu số 02-VT
(Ban hành kèm thông tư số 133/2016/TT-BTC
Ngày 26/08/2016 của Bộ Tài Chính)

PHIẾU XUẤT KHO

Ngày 19 tháng 3 năm 2001

Số: XK0026

Nợ 156

Có 135

- Họ tên người nhận: Nguyễn Văn A
- Địa chỉ:
- Lý do xuất kho: Bán hàng theo hóa đơn 0000023
- Xuất tại kho: K1 Địa điểm:

STT	Tên sản phẩm	Mã số	Đơn vị	Số lượng		Đơn giá	Thành tiền
				Yêu cầu	Thực xuất		
1	Xi măng HP	HP-30	Bao	6		50 000	300 000
2	Thép cây	P12	Cây	10		200 000	2000 000
	Tổng						2 300 000

- Tổng số tiền (viết bằng chữ): Hai triệu ba trăm nghìn đồng chẵn
- Số chứng từ gốc kèm theo:

Người lập phiếu
(Ký, họ tên)

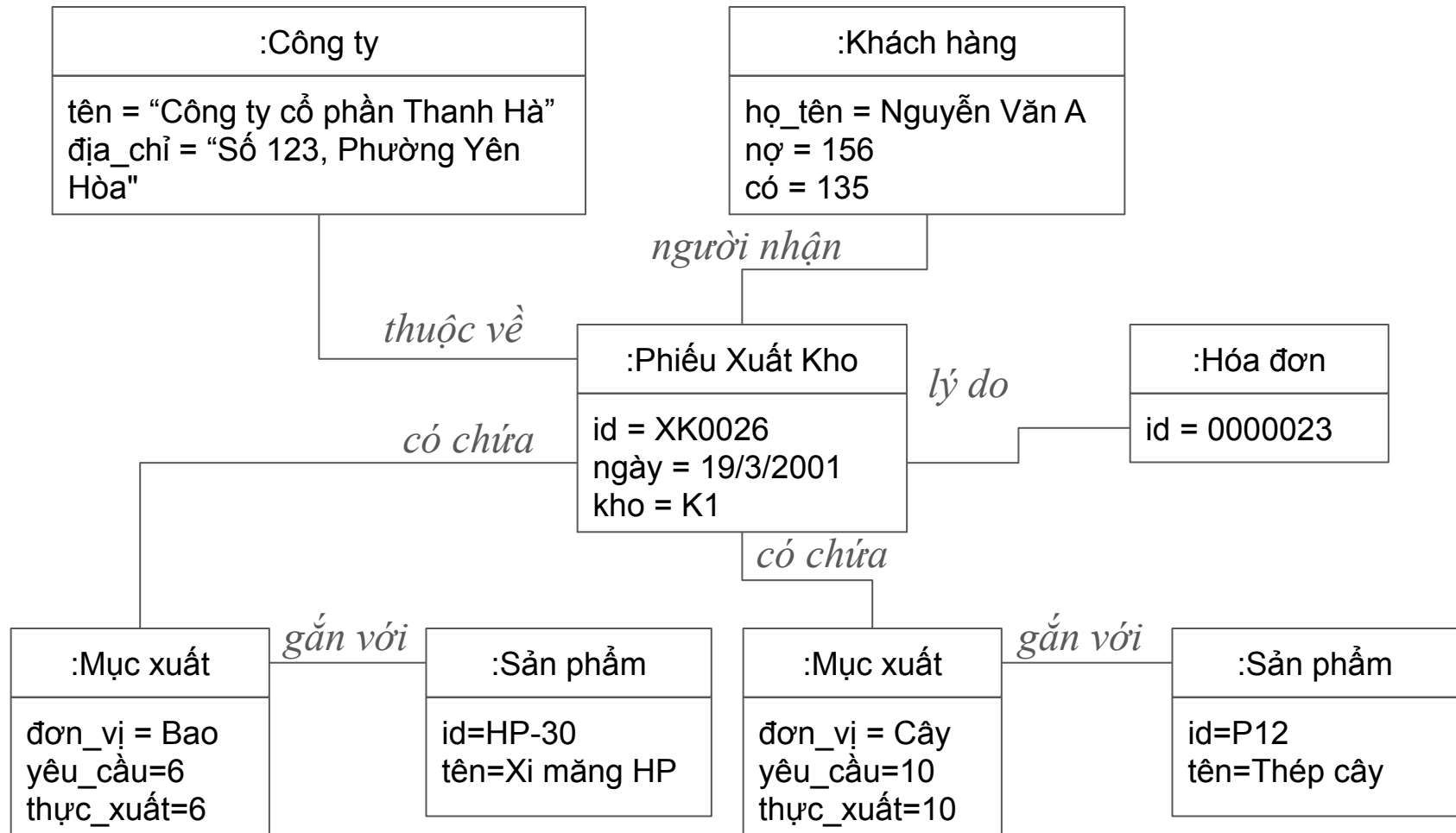
Người nhận hàng
(Ký, họ tên)

Thủ kho
(Ký, họ tên)

Kế toán trưởng
(Ký, họ tên)

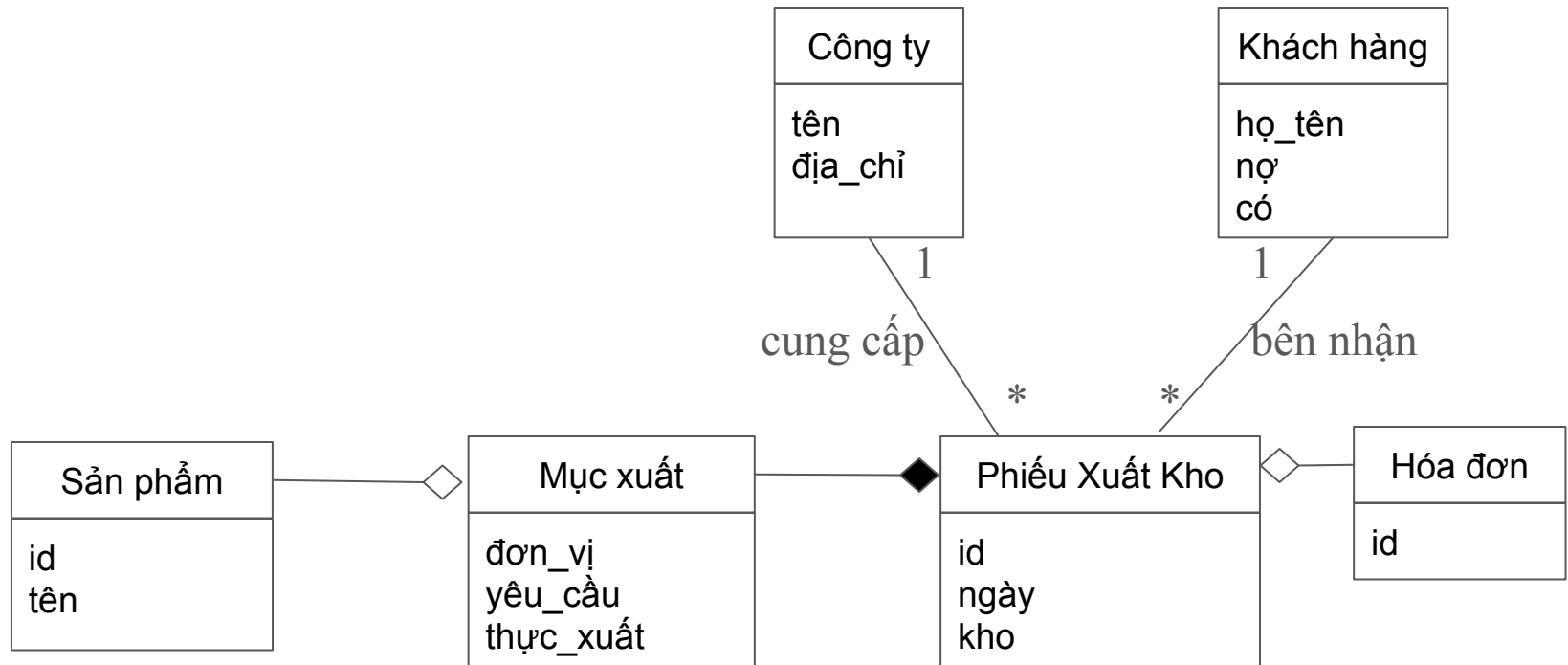
Giám đốc
(Ký, họ tên)

Sơ đồ đối tượng



!Lưu ý: Lời giải để tham khảo, không phải mẫu.

Sơ đồ lớp



!Lưu ý: Lời giải đề tham khảo, không phải mẫu.

