

BÀI GIẢNG ĐỒ ÁN TỐT NGHIỆP

PHÂN CÔNG NHÓM ĐỒ ÁN



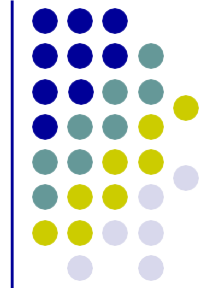
Phân chia sinh viên vào các nhóm
Mỗi nhóm 2 – 4 thành viên

LỰA CHỌN ĐỀ TÀI



Sinh viên lựa chọn đề tài theo các chủ đề:

- **WEB, THƯƠNG MẠI ĐIỆN TỬ**
- **ỨNG DỤNG ANDROID**
- **ỨNG DỤNG WINFORM**
- **KIỂM THỬ PHẦN MỀM**
- **TỰ CHỌN**

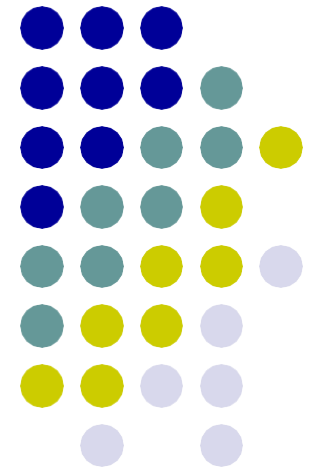


GIẢI ĐOẠN 2:

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Bước 1:

Khảo sát hệ thống





Mục tiêu

- + Khảo sát hiện trạng của hệ thống hiện tại
- + Tìm điểm yếu kém và đưa phương án giải quyết
- + Xác định yêu cầu của hệ thống trong tương lai.

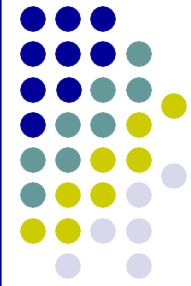
Kết quả:

- + Dự án khả thi, hồ sơ xác lập dự án
- + Yêu cầu chức năng, phi chức năng
- + Hồ sơ vào/ra, tài nguyên
- + Dự trù thiết bị phần cứng, phần mềm
- + Lịch trình làm việc sơ bộ

NỘI DUNG

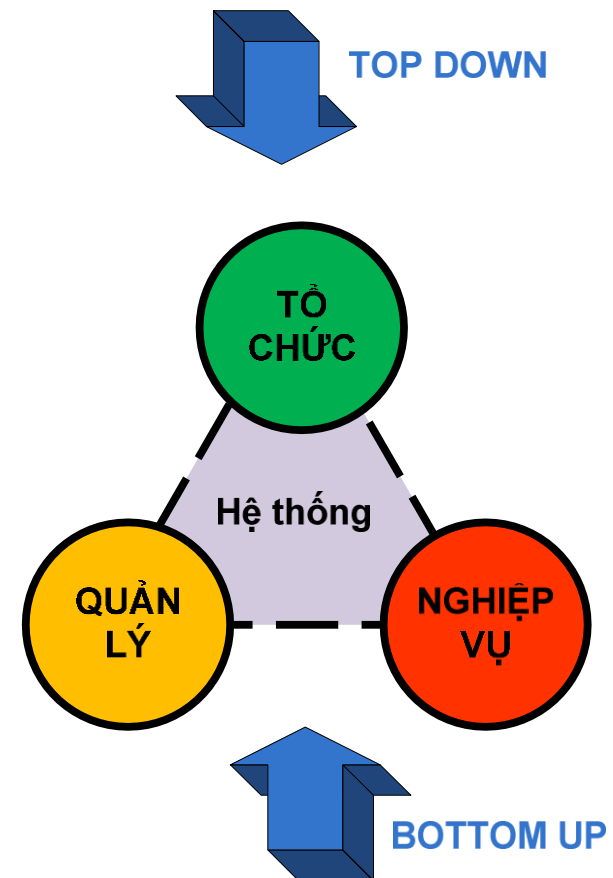


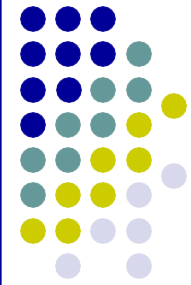
- **Cách tiếp cận, khảo sát hệ thống thực**
 - Cách tiếp cận một tổ chức
 - Các giai đoạn khảo sát
 - Quy trình khảo sát
 - Nội dung khảo sát
 - Yêu cầu đặt ra đối với người phân tích
- **Phương pháp thu thập xác định yêu cầu**
 - Phương pháp truyền thống
 - Phương pháp hiện đại
- **Xây dựng dự án**



TIẾP CẬN MỘT TỔ CHỨC

- Ba đặc trưng cơ bản
 - Mô hình tổ chức
 - Mô hình quản lý
 - Hoạt động nghiệp vụ
- Hai chiến lược tiếp cận
 - Từ trên xuống (top down)
 - Từ dưới lên (bottom up)

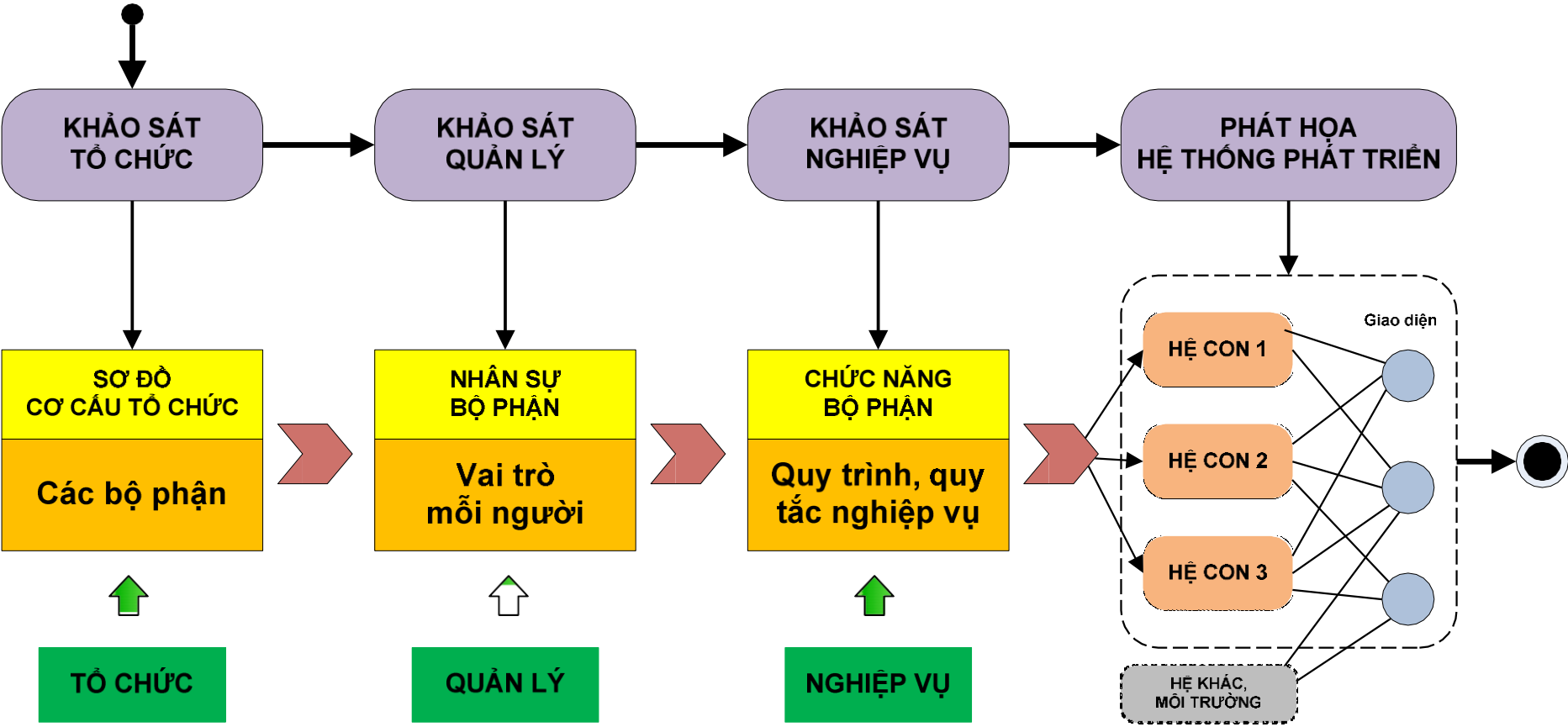




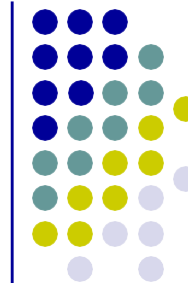
TIẾP CẬN MỘT TỔ CHỨC

- **Nguyên tắc từ trên xuống**
 - Tổ chức: **bộ phận** cao nhất → thấp nhất
 - Quản lý: **người quản lý** cao nhất → thấp nhất
 - Nghiệp vụ: **chức năng** chung nhất → cụ thể
- **Nguyên tắc từ dưới lên**
 - Từ chỗ làm việc
 - Từ công việc cụ thể
 - Từ một người cụ thể
 - Tích hợp dần lên

TIẾN TRÌNH TIẾP CẬN TỔ CHỨC



CÁC GIAI ĐOẠN KHẢO SÁT



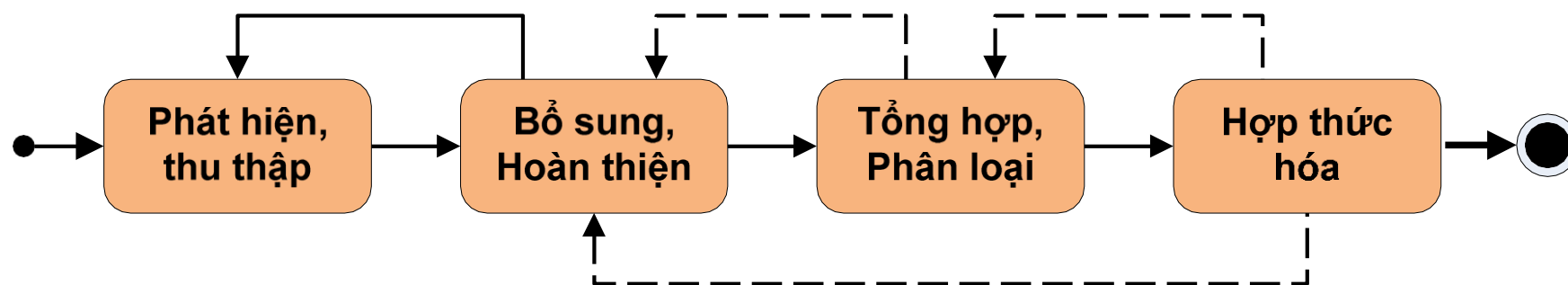
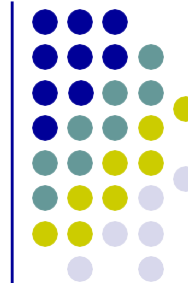
- **Khảo sát sơ bộ**

- Mục tiêu: hình thành dự án phát triển hệ thống
- Nội dung: xác định
 - Chức năng chính
 - Ràng buộc chính
 - Môi trường hệ thống

- **Khảo sát chi tiết**

- Mục tiêu: chi tiết hóa các nội dung đã khảo sát sơ bộ
- Nội dung: thu thập thông tin, dữ liệu chi tiết

QUY TRÌNH KHẢO SÁT



NỘI DUNG KHẢO SÁT



- Cơ cấu tổ chức
- Chức năng, nhiệm vụ, phân cấp quyền hạn
- Các loại tài liệu và đặc trưng sử dụng
- Các quy tắc nghiệp vụ, quy trình xử lý
- Các chính sách và hướng dẫn
- Các nguồn lực
- Điều kiện môi trường
- Sự mong đợi về hệ thống mới

YÊU CẦU VỚI PHÂN TÍCH VIÊN



- **Phẩm chất cần có**
 - Tính xông xáo
 - Tính chủ động
 - Sự nghi ngờ
 - Khả năng quan sát
 - Có cách nhìn đa chiều
- **Kết quả cần hình thành theo mẫu và các chuẩn mực**

PHƯƠNG PHÁP XÁC ĐỊNH YÊU CẦU



- Phương pháp truyền thống
 - Phỏng vấn (*interview*)
 - Quan sát tại chỗ (*observe*)
 - Điều tra bằng hỏi (*questionnaire*)
 - Nghiên cứu tài liệu viết (*studying documents*)



PHƯƠNG PHÁP XÁC ĐỊNH YÊU CẦU



- **Phương pháp hiện đại**
 - Thiết kế ứng dụng liên kết
 - Hệ thống trợ giúp nhóm
 - Công cụ CASE
 - Làm bản mẫu

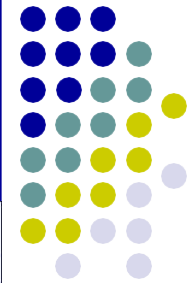
PHỎNG VẤN



- **Hỏi trực tiếp người liên quan để thu thông tin**
 - Là cách đơn giản và quan trọng nhất
 - Thu được nhiều thông tin, khó tổng hợp
- **Các yếu tố ảnh hưởng**
 - Sự chuẩn bị
 - Chất lượng câu hỏi
 - Phương pháp, phương tiện ghi
 - Kinh nghiệm và khả năng giao tiếp



PHỎNG VẤN – CHUẨN BỊ



- **Chuẩn bị**

- Lập danh sách và chọn người cần hỏi
- Hẹn gặp
- Tìm hiểu, xác định câu hỏi thích hợp
- Chuẩn bị phương tiện hỗ trợ
- Lập kế hoạch



PHỎNG VẤN – CÂU HỎI



- **Câu hỏi**

- **Câu hỏi đóng:** câu trả lời đã xác định
- **Câu hỏi mở:** tùy người trả lời
- **Câu hỏi chiến lược/chéo:** từ các nội dung khác nhau để biết điều nằm sau đó

- **Cách hỏi**

- **Bắt đầu:** tìm hiểu với câu hỏi mở
- **Kết thúc:** chốt lại với câu hỏi đóng

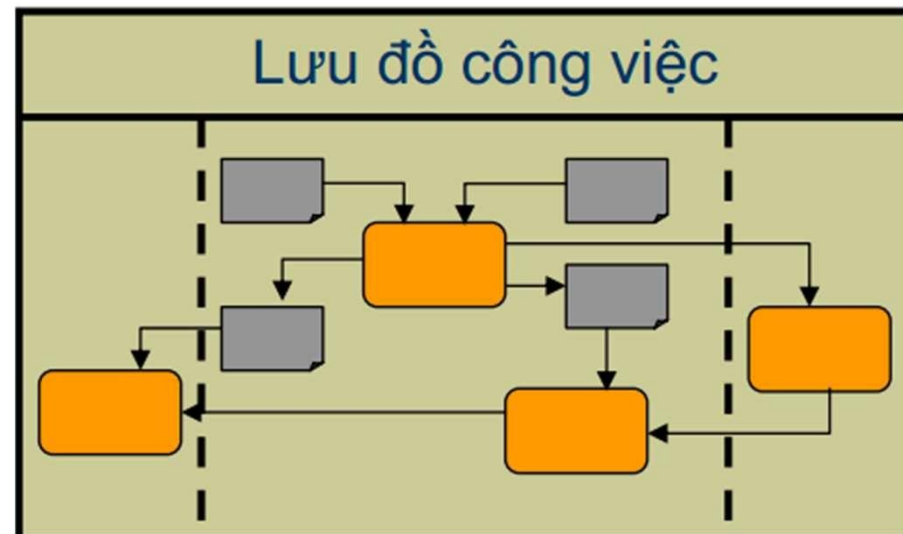


PHỎNG VẤN – MẪU BIỂU

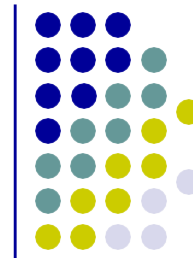


| Kế hoạch phỏng vấn | |
|------------------------------------|-----------|
| Địa điểm | Thời gian |
| Đối tượng | Yêu cầu |
| Chương trình 1.nn... 2.nn... | |
| câu 1 | 1 phút |
| câu 2 | 2 phút |
| ... | ... |
| câu n | 7 phút |
| tổng thời gian: n phút | |

| Phiếu phỏng vấn | |
|-----------------|----------------|
| Người hỏi | Người được hỏi |
| câu hỏi: | trả lời: |
| bình luận | |



PHỎNG VẤN: TIẾN HÀNH



- **Tiến hành**

- Nhóm 2 người: 1 hỏi, 1 ghi



- Phân công người ghi bằng biểu đồ, ký hiệu

- **Kinh nghiệm**

- Nắm, hiểu thuật ngữ nghiệp vụ, văn hóa
- Quan sát, lắng nghe, thay đổi thích hợp
- Tránh hỏi chuyện cá nhân, nội bộ
- Hỏi ngắn gọn, trực tiếp, không áp đặt
- Từng bước thiết lập sự thân thiện, tin cậy

QUAN SÁT TẠI CHỖ



- **Tiến hành**

- Nhìn vào đối tượng để thu thông tin
- Hai cách: **trực tiếp & qua phương tiện**



- **Đặc điểm**

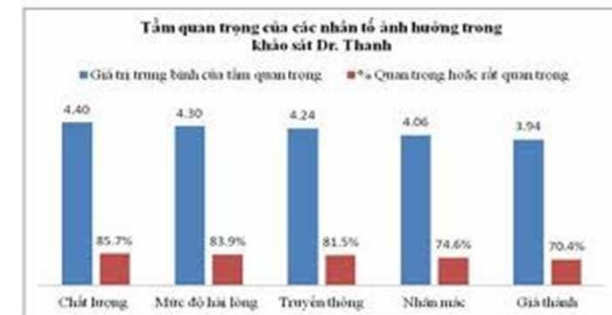
- Dùng bổ sung và chính xác hóa thông tin
- Có tính bộ phận, bề ngoài, bị động
- Hạn chế thời gian, phạm vi, đối tượng quan sát

ĐIỀU TRA BẢNG HỎI



- **Mục đích**

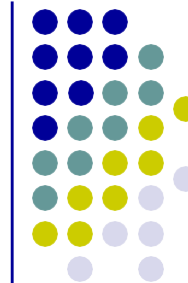
- Thăm dò dư luận
- Quan điểm, ý kiến chung
- Đặc trưng đại chúng rộng rãi



- **Đặc điểm**

- Nhanh, dễ tổng kết, có sẵn công cụ
- Đào tạo người điều tra ít tốn kém
- Độ chính xác thấp, mang tính trung bình

NGHIÊN CỨU TÀI LIỆU VIẾT



- **Bao gồm**

- Xác định tài liệu, báo cáo cần thu thập
- Phân loại, sao chép, lên danh sách, bổ sung
- Ghi lại nội dung chính theo mẫu
- Phân tích làm rõ yêu cầu



NGHIÊN CỨU TÀI LIỆU VIẾT



- Nội dung cần chú ý
 - Chi tiết về tổ chức, chức năng, nhân sự, nguồn lực
 - Kế hoạch kinh doanh, sản phẩm, chính sách, môi trường
 - Công việc, quy trình, thời gian, chi phí, quy tắc hoạt động



XÂY DỰNG DỰ ÁN



- Phạm vi, khả năng, mục tiêu dự án
 - Xác định lĩnh vực, chức năng: 2 cách
 - Khoanh lĩnh vực hẹp giải quyết triệt để,
 - Giải quyết tổng thể.
 - Khả năng nguồn lực của đơn vị đầu tư:
 - Nhân lực
 - Thiết bị kỹ thuật
 - Tài chính
 - Giải quyết mong muốn của chủ đầu tư

XÂY DỰNG DỰ ÁN



- Phác họa giải pháp cân nhắc tính khả thi
 - Khả thi kỹ thuật
 - Yêu cầu kỹ thuật, công nghệ.
 - Khả thi kinh tế
 - Chi phí cho giải pháp
 - Lợi ích thu lại
 - Khả thi nghiệp vụ
 - Nhu cầu và nghiệp vụ của bên sử dụng
 - Cung cấp đúng thông tin, đúng lúc, đúng nơi

XÂY DỰNG DỰ ÁN



- Lập dự trù và kế hoạch triển khai
 - Hồ sơ điều tra, xác lập dự án
 - Tập hợp kết quả khảo sát
 - Ý kiến phê phán, đánh giá
 - Giải pháp đề xuất và quyết định lựa chọn
 - Dự trù thiết bị
 - Sơ bộ sự kiện
 - Thiết bị cần có
 - Điều kiện mua, lắp đặt

XÂY DỰNG DỰ ÁN



- Lập dự trù và kế hoạch triển khai
 - Kế hoạch triển khai dự án
 - Tiến độ triển khai
 - Người phụ trách: chuyên gia về tin học, về quản lý
 - Các nhân viên làm việc:
 - Phân tích viên
 - Lập trình viên
 - Những người khai thác

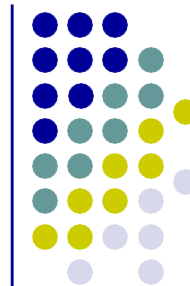
KẾT LUẬN



- **Kết quả thu được**
 - Dự án khả thi
 - Kế hoạch triển khai dự án.
- **Nhóm phân tích cần hoàn thành**
 - Hồ sơ khảo sát chi tiết
 - Mô hình tiến trình nghiệp vụ
 - Hồ sơ xác lập dự án
 - Từ điển dữ liệu (nếu cần)

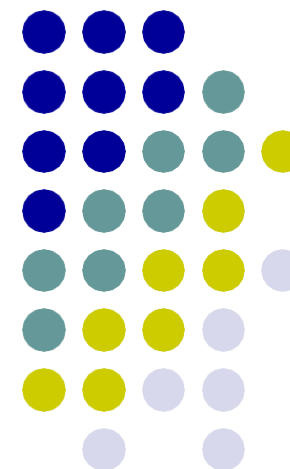
THẢO LUẬN

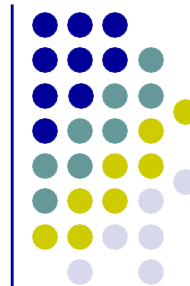




Bước 2:

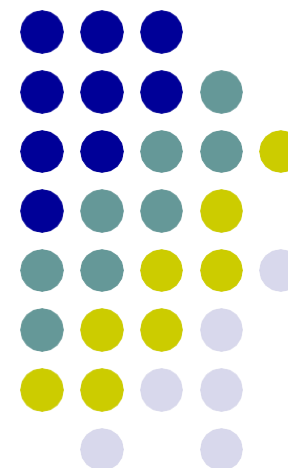
**ĐỀ XUẤT CHỨC NĂNG
CỦA HỆ THỐNG**





Bước 3:

PHÁT BIỂU BÀI TOÁN



Ví dụ Phát biểu bài toán

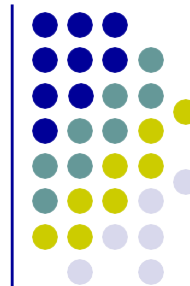


1. Các công ty hàng không (AirlineCompany) cung cấp nhiều chuyến bay (Flight) khác nhau.
2. Một chuyến bay mở và đóng đặt vé theo yêu cầu của công ty.
3. Một khách hàng có thể đặt vé một hoặc nhiều chuyến bay và cho nhiều hành khách khác nhau.
4. Một vé được đặt (Booking) liên quan đến một chuyến bay duy nhất và một hành khách duy nhất.
5. Vé được đặt (Booking) có thể được xác nhận hoặc bị hủy.
6. Một chuyến bay có một sân bay đi (departure airport) và sân bay đến (arrival airport).
7. Một chuyến bay có ngày và thời gian đi, và một ngày và thời gian đến.
8. Một chuyến bay có thể liên quan đến các chặng dừng(stopover) tại các sân bay.
9. Một chặng dừng có thời gian đến và thời gian đi.
10. Mỗi sân bay phục vụ một hoặc nhiều thành phố. (Each airport serves one or more cities).

VD 2: Phát biểu bài toán

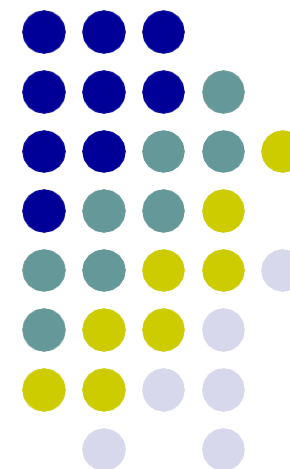


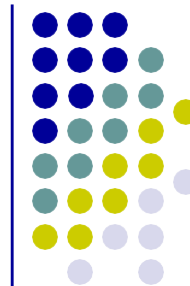
- 1) Quá trình đào tạo (training process) được bắt đầu khi người quản lý đào tạo (training manager) nhận được yêu cầu đào tạo từ một đại diện của học viên (behalf of an employee). Yêu cầu này được tiếp nhận bởi người phụ trách (person in charge), người phụ trách kiểm tra yêu cầu đào tạo xem có đáp ứng được không, và sau đó chuyển tiếp yêu cầu hoặc được chấp nhận hoặc từ chối đến những người liên quan.
- 2) Trong trường hợp yêu cầu được chấp nhận, người phụ trách tìm khóa học (training course) phù hợp với các yêu cầu trong danh mục của các khóa học (catalogue of registered courses). sau đó thông báo cho học viên về các nội dung khóa học và đưa ra danh sách các học kỳ tiếp theo (subsequent sessions). Khi các học viên đã quyết định, người quản lý đào tạo ghi danh những học viên và gửi đến bộ phận đào tạo liên quan
- 3) Nếu có yêu cầu hủy khóa học, các học viên phải thông báo cho người quản lý đào tạo trong thời gian sớm nhất có thể để hủy bỏ.
- 4) Khi kết thúc đào tạo, các học viên phải nộp một phiếu đánh giá cho người quản lý đào tạo về các khóa học mà họ đã hoàn thành, và tài liệu chứng minh các học viên đã tham dự khóa học
- 5) Người quản lý đào tạo kiểm tra các hóa đơn từ bộ phận đào tạo chuyển đến trước khi chuyển cho nhân viên kế toán



Bước 4:

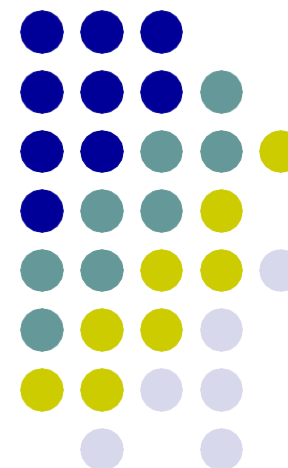
XÂY DỰNG CÁC BIỂU ĐỒ CỦA HỆ THỐNG





Bước 4.1:

BIỂU ĐỒ USE CASE



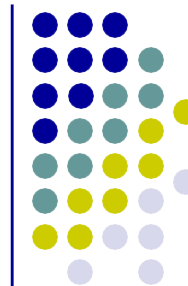
Mục đích của use case



Ca sử dụng biểu diễn những chức năng mà hệ thống cần làm

Các ca sử dụng cho phép:

- + Biết được hành vi của hệ thống mà không cần xác định làm thế nào hành vi này thực hiện
- + Định nghĩa những hạn chế chính xác của hệ thống
- + Cho người phát triển hiểu rõ hơn những gì mà khách hàng và người sử dụng chờ đợi



Mô hình ca sử dụng

Một biểu đồ ca sử dụng bao gồm:

Các tác nhân

Các ca sử dụng

Quan hệ giữa các tác nhân và các ca sử dụng

Một mô hình ca sử dụng được định nghĩa bởi:

Các biểu đồ ca sử dụng

Phần mô tả bằng lời các kịch bản sử dụng

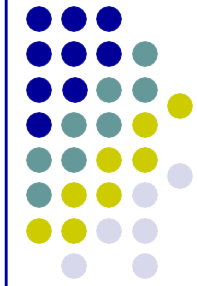
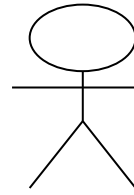
Phần mô tả các kịch bản dùng:

Biểu đồ tuần tự

Biểu đồ tương tác

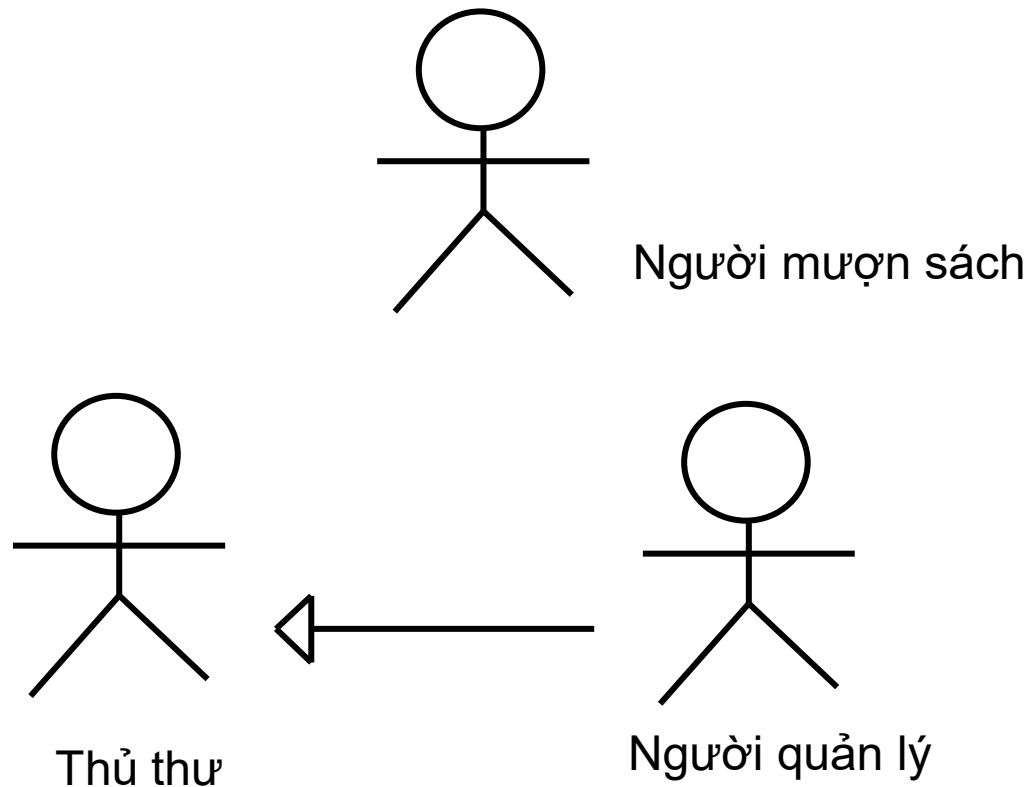
Biểu đồ hoạt động

Các tác nhân



Một tác nhân là một người hoặc một thiết bị, hệ thống bên ngoài có phản ứng với hệ thống

Quan hệ giữa các tác nhân: tổng quát hóa (thừa kế)



Tìm kiếm tác nhân như thế nào?



Trả lời các câu hỏi sau để tìm ra tác nhân hệ thống

Ai sẽ sử dụng chức năng chính của hệ thống?

Ai giúp hệ thống làm việc hàng ngày?

Ai quản trị, bảo dưỡng để hệ thống làm việc liên tục?

Hệ thống quản lý thiết bị phần cứng nào?

Hệ thống đang xây dựng tương tác với hệ thống khác nào?

Ai hay cái gì quan tâm đến kết quả hệ thống trả lại?

Ca sử dụng



Một ca sử dụng là một phương tiện để thể hiện các khả năng khác nhau sử dụng hệ thống

Nó biểu diễn một chuỗi tương tác giữa tác nhân và ứng dụng

Nó định nghĩa một chức năng có thể sử dụng được bởi tác nhân

Đặt trước

Thuê sách

Xem trạng thái

Tìm kiếm UC như thế nào?



□ Với mỗi tác nhân đã tìm ra, hãy trả lời các câu hỏi sau để tìm ra các Use case hệ thống

- + Tác nhân yêu cầu hệ thống thực hiện chức năng nào?
- + Tác nhân cần đọc, tạo lập, bãi bỏ, lưu trữ, sửa đổi các thông tin nào trong hệ thống?
- + Tác nhân cần thông báo cho hệ thống sự kiện xảy ra trong nó?
- + Hệ thống cần thông báo cái gì đó cho tác nhân?
- + Hệ thống cần vào/ra nào? Vào/ra đi đến đâu hay từ đâu?

□ Đặt tên UC hệ thống

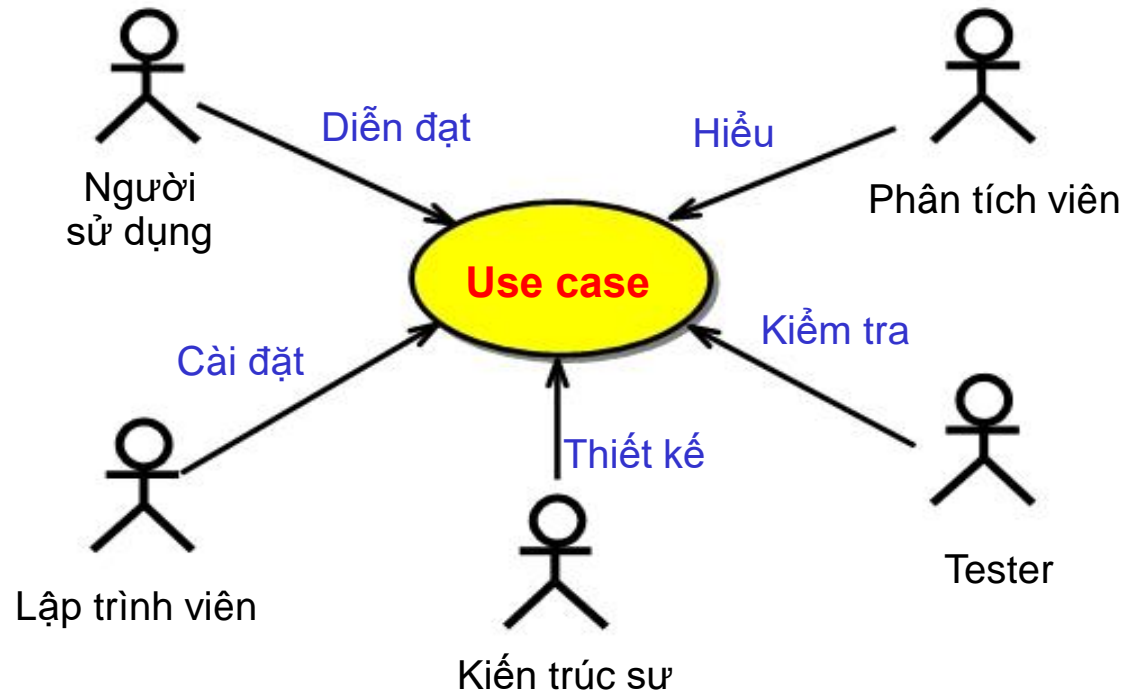
- + Theo khái niệm nghiệp vụ của tổ chức
- + Không sử dụng từ kỹ thuật, chuyên môn
- + Sử dụng các động từ, cụm từ ngắn gọn

□ Tùy theo tầm cỡ dự án mà mỗi hệ thống có từ 20-70 UC



Xây dựng UC để làm gì?

④ Ai quan tâm đến UC?

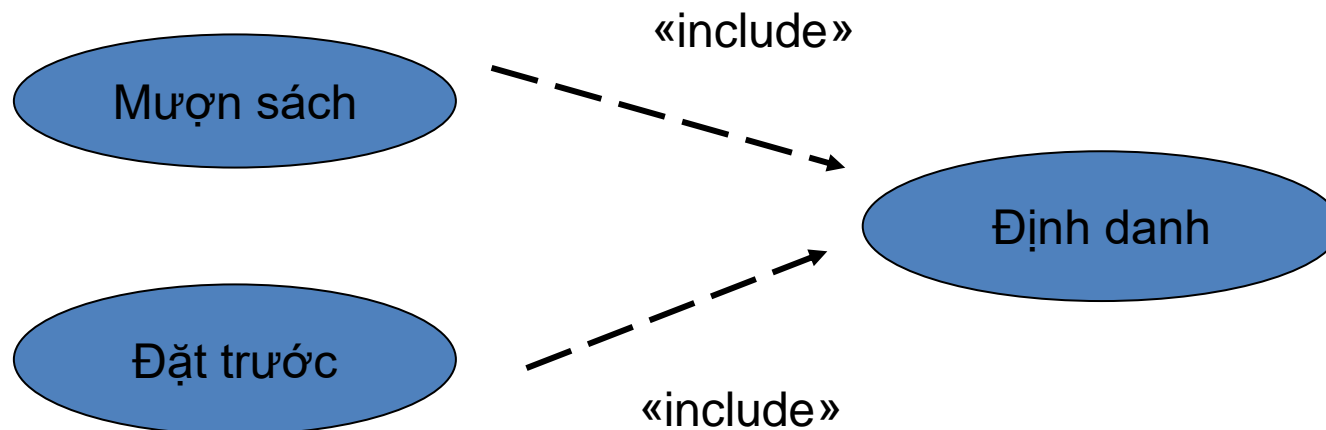


TỔ chức các ca sử dụng: include

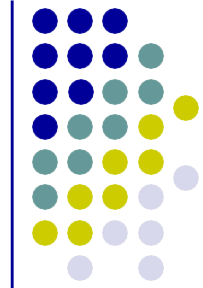


Quan hệ *«include»* biểu diễn một ca sử dụng chứa hành vi định nghĩa trong một ca sử dụng khác

Quan hệ này cho phép biểu diễn phần chung các hành vi của nhiều ca sử dụng

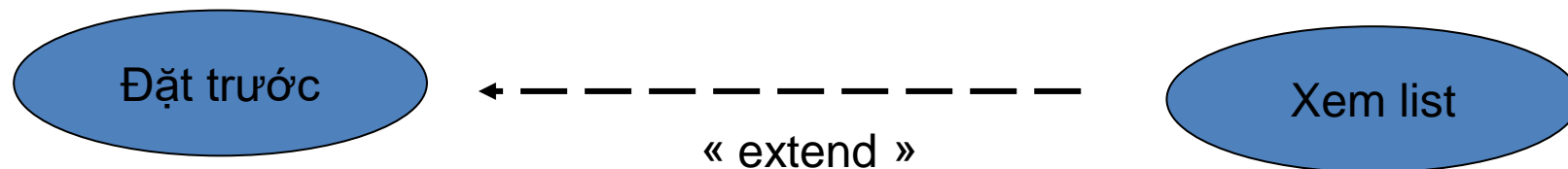


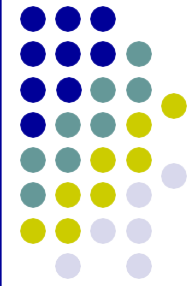
Tổ chức các ca sử dụng: extend



Một UC tùy ý mở rộng chức năng do UC khác cung cấp

Sử dụng để mô hình hóa một vài chức năng dùng chung, sử dụng lại giữa hai hay nhiều UC

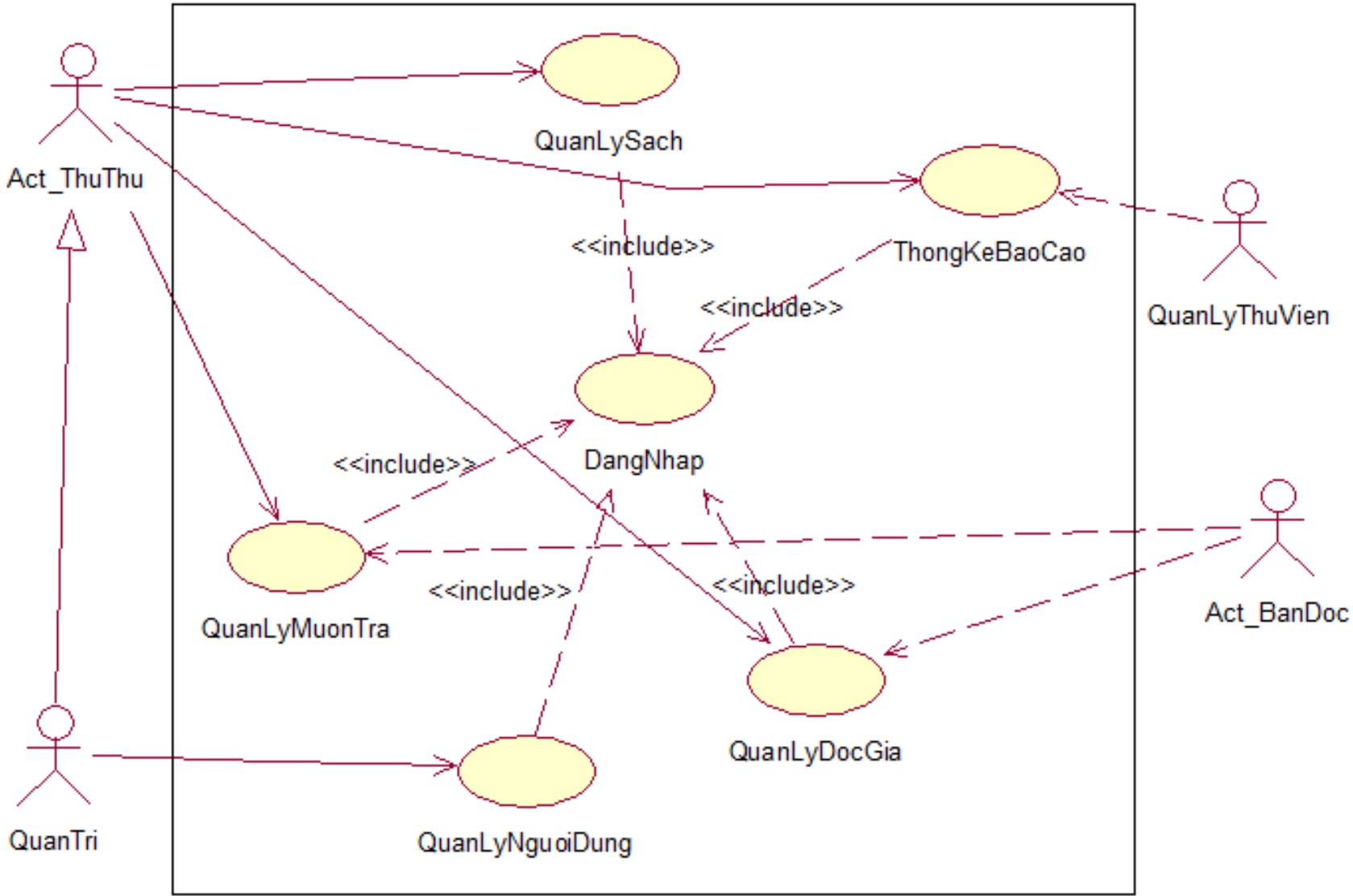


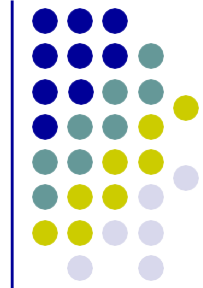


Mô hình hóa một hệ thống: xác định ca sử dụng

- Xác định các tác nhân sử dụng
- **Với mỗi tác nhân**, tìm kiếm các **ca sử dụng** với hệ thống. Đặc biệt những gì thay đổi trạng thái của hệ thống hoặc chờ đợi phản ứng từ hệ thống
- Tổ chức lại các ca sử dụng với các **quan hệ sử dụng**, **mở rộng...**

Biểu đồ ca sử dụng





Viết kịch bản cho use case

Luồng sự kiện trong UC



Tài liệu luồng sự kiện (flow of events) mô tả hành vi của UC

Mô tả luồng logic đi qua UC

Mô tả người sử dụng làm gì, hệ thống làm gì

Trong một UC có nhiều luồng sự kiện: luồng chính, luồng phụ

Kịch bản (Scenario)

Một luồng sự kiện trong một hiện thực của UC

Là trình tự hành động cụ thể để mô tả hành vi

Kịch bản đi xuyên suốt UC theo nhánh chính, nhánh phụ, nhánh đặc biệt

Tài liệu luồng sự kiện

Tài liệu luồng sự kiện bao gồm

Mô tả vắn tắt UC

Mô tả ngắn gọn UC làm gì?

Những ai sử dụng UC?

Nó trả lại kết quả gì?

Tiền điều kiện (pre-condition)

Điều kiện cần thực hiện trước khi UC khởi động

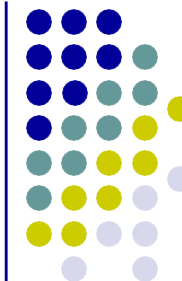
Không phải UC nào cũng có tiền điều kiện

Luồng sự kiện chính và luồng sự kiện rẽ nhánh

Hậu điều kiện (post-condition)

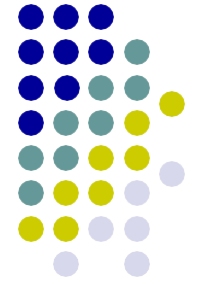


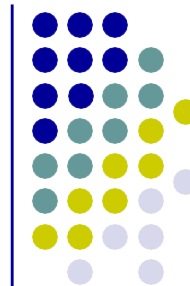
VD: Kịch bản của một ca sử dụng Quản lý sách



| | |
|---|---|
| Tên use case | QuanLySach |
| Tên Actor | Thủ thư |
| Mức | 1 |
| Tiền điều kiện | Thủ thư phải đăng nhập hệ thống |
| Đảm bảo tối thiểu | |
| Đảm bảo thành công | |
| Kích hoạt | Thủ thư yêu cầu chức năng QL sách |
| Hành động tác nhân | Phản ứng hệ thống |
| 1. Thủ thư yêu cầu chức năng | <ol style="list-style-type: none">1. Hệ thống lấy về thông tin các chuyên ngành2. Lấy về thông tin ngày giờ hệ thống.3. Hệ thống chọn chuyên ngành mặc định.4. Hệ thống lấy về các đầu sách thuộc chuyên ngành, hiển thị lên giao diện |
| | |
| 1. Thực hiện các chức năng theo yêu cầu | |

THẢO LUẬN





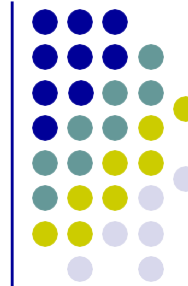
Bước 4.2: BIỂU ĐỒ LỚP

Biểu đồ lớp



- + Biểu diễn một cách trực quan các lớp (mức quan niệm, chỉ có thuộc tính, ko có phương thức) hoặc các đối tượng (thế giới thực, hay còn gọi là các thực thể) trong phạm vi bài toán (problem domain).
- + Mô tả hướng nhìn tĩnh của một hệ thống bằng các khái niệm lớp và mối liên hệ giữa chúng với nhau.
- + Mục đích là tạo nền tảng cho các biểu đồ khác, thể hiện các khía cạnh khác của hệ thống (ví dụ như trạng thái của đối tượng hay cộng tác động giữa các đối tượng, được chỉ ra trong các biểu đồ động).
- Cung cấp 3 thông tin:
 - + Các lớp (Conceptual classes hoặc Domain objects).
 - + Các mối liên hệ giữa các lớp.
 - + Các thuộc tính của các lớp.

Cách tìm lớp



Nghiên cứu kỹ tất cả các use case và scenario để tìm ra các danh từ có vai trò nào đó trong các scenario (kịch bản).

Các danh từ này sẽ trở thành các lớp ứng cử viên.

- Loại bỏ các lớp ứng cử viên không thích hợp:

+ Lớp dư thừa: do có hai hay nhiều danh từ cùng chỉ một thực thể nên ta chỉ cần giữ lại một từ duy nhất và loại bỏ các từ khác.

+ Danh từ không thích hợp: đó là các danh từ không liên quan đến phạm vi của bài toán.

+ Danh từ mô tả những lớp không rõ ràng: đó là các danh từ hoặc không biểu diễn một thực thể cụ thể hoặc các khái niệm không rõ nghĩa.

+ Các danh từ chỉ là một vai trò (role) trong mối liên hệ với một lớp khác.

Mối quan hệ giữa các lớp



Liên hệ là một sự nối kết giữa các lớp, một liên quan về ngữ nghĩa giữa các đối tượng của các lớp tham gia. Liên hệ thường thường mang tính hai chiều, có nghĩa khi một đối tượng này có liên hệ với một đối tượng khác thì cả hai đối tượng này nhận thấy nhau.

Các mối quan hệ giữa các lớp

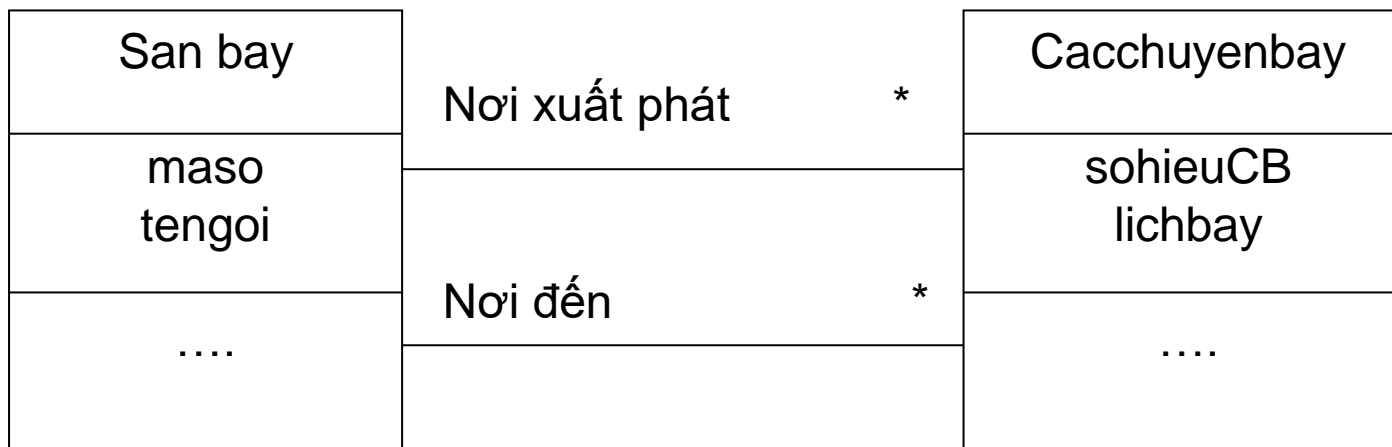
Giữa các lớp có bốn quan hệ cơ bản:

- ❑ *Quan hệ kết hợp (Association),*
- ❑ *Quan hệ kết tập (Aggregation),*
- ❑ *Quan hệ tổng quát hóa, kế thừa (Generalization, Inheritance)*
- ❑ *Quan hệ phụ thuộc (Dependency).*

3.1. Quan hệ kết hợp (Association)

Booch đã mô tả vai trò của mỗi liên kết giữa các đối tượng như sau:

“Một liên kết chỉ rõ sự kết hợp mà qua đó, một đối tượng được một đối tượng khác phục vụ hoặc một đối tượng có thể điều khiển đối tượng kia”.



Xác định mối quan hệ kết hợp association:

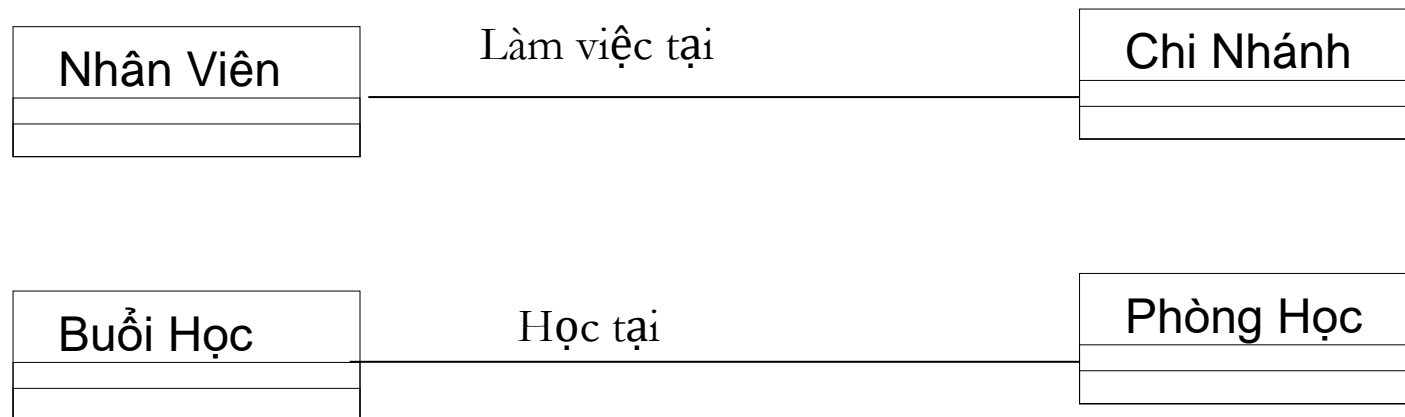
- Hướng dẫn xác định mối kết hợp:
- Một sự phụ thuộc giữa hai hay nhiều lớp có thể thiết lập thành mối kết hợp. Mối kết hợp thường tương ứng với một động từ hoặc cụm giới từ như là *thành phần của, làm việc cho, chứa trong, ...*
- Một tham chiếu từ một lớp đến một lớp khác là một mối kết hợp.

- Xác định mối kết hợp association:

- Các mẫu xác định mối kết hợp:

- Mối kết hợp vị trí (location): *liên kết tới, thành phần của, làm việc tại,*

- Ví dụ:

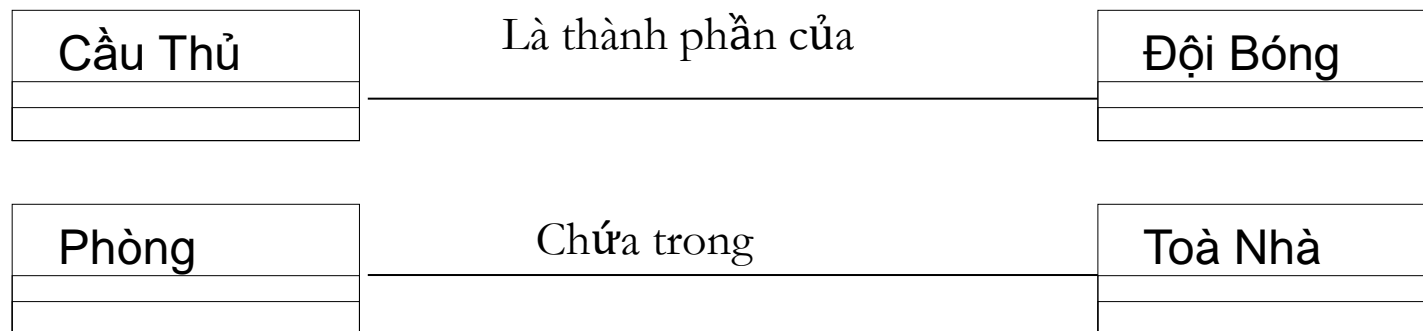


- Xác định mối kết hợp association:

- Các mẫu xác định mối kết hợp:

- Mối kết hợp sở hữu: *của, có, thuộc, ...*

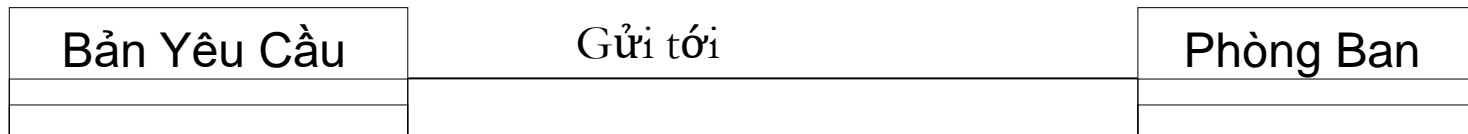
- Ví dụ:



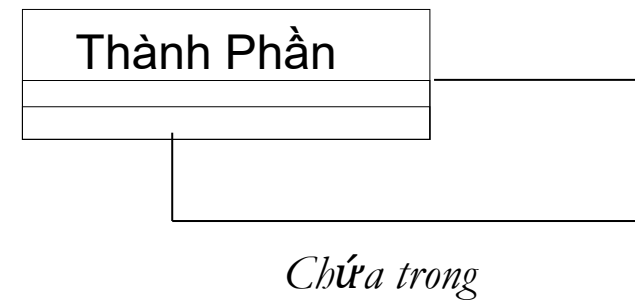
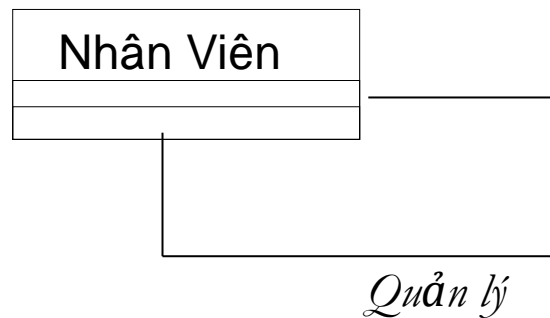
- Xác định mối kết hợp association:

- Các mẫu xác định mối kết hợp:

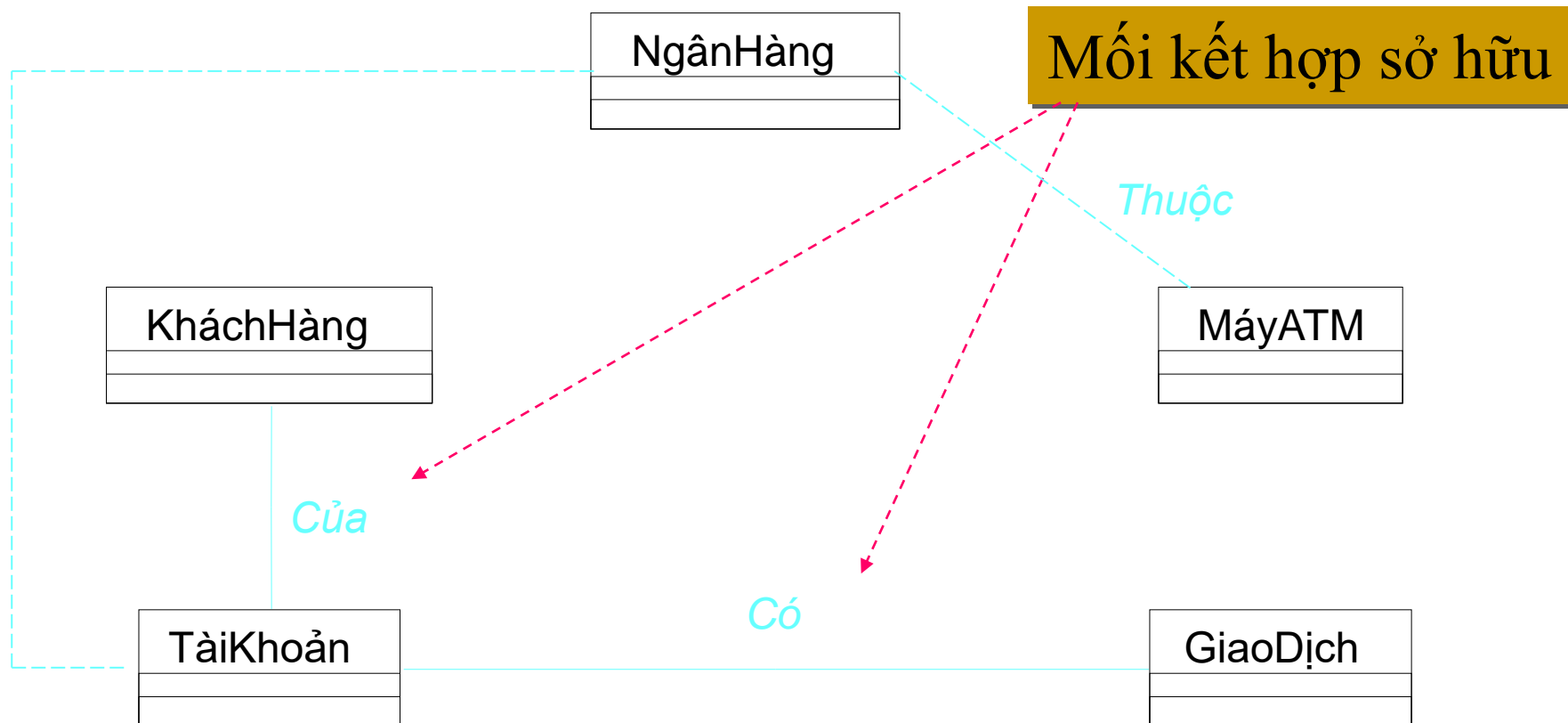
- Mối kết hợp truyền thông, liên lạc (communication): *đặt tới, trao đổi với, gửi cho, tiếp nhận từ, ...*



- Xác định mối kết hợp association:
 - Mối kết hợp phản thân: là mối quan hệ được thiết lập giữa một đối tượng của một lớp với một đối tượng khác cũng thuộc lớp đó.

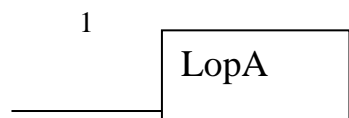


- Xác định mối kết hợp association:
 - Ví dụ: hệ thống ATM

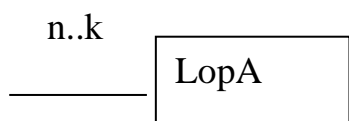


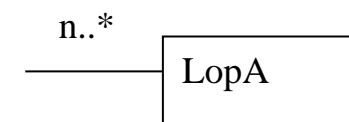
Xác định mối kết hợp association:

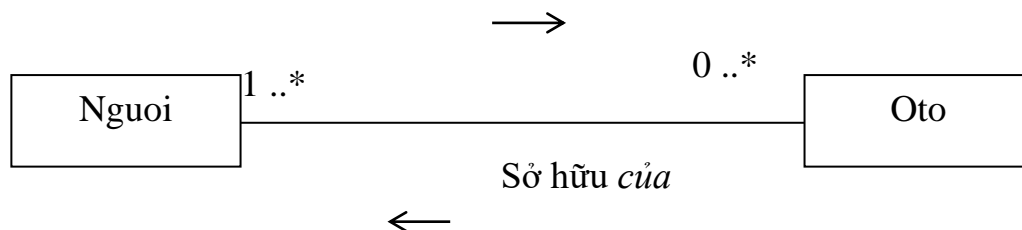
Bội số (Multiplicity): xác định số lượng các thể hiện (đối tượng) của một lớp trong quan hệ kết hợp với lớp khác.

1  Chính xác 1 (nếu không nhập nhằng có thể không điền số 1)

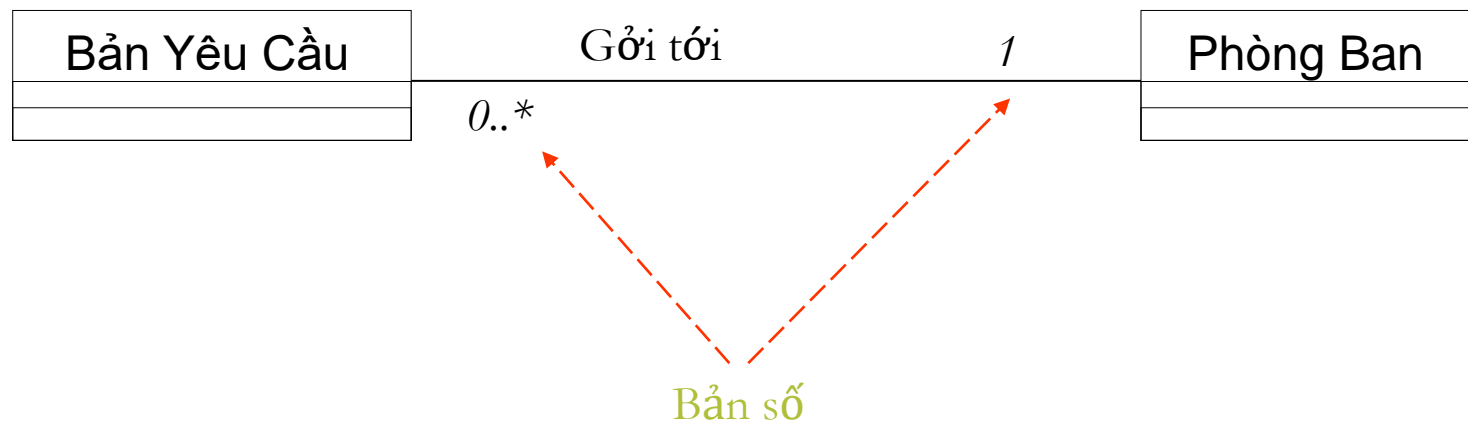
*  Nhiều (không hoặc nhiều)

n..k  Số lượng được xác định giữa số n và k (≥ 0).

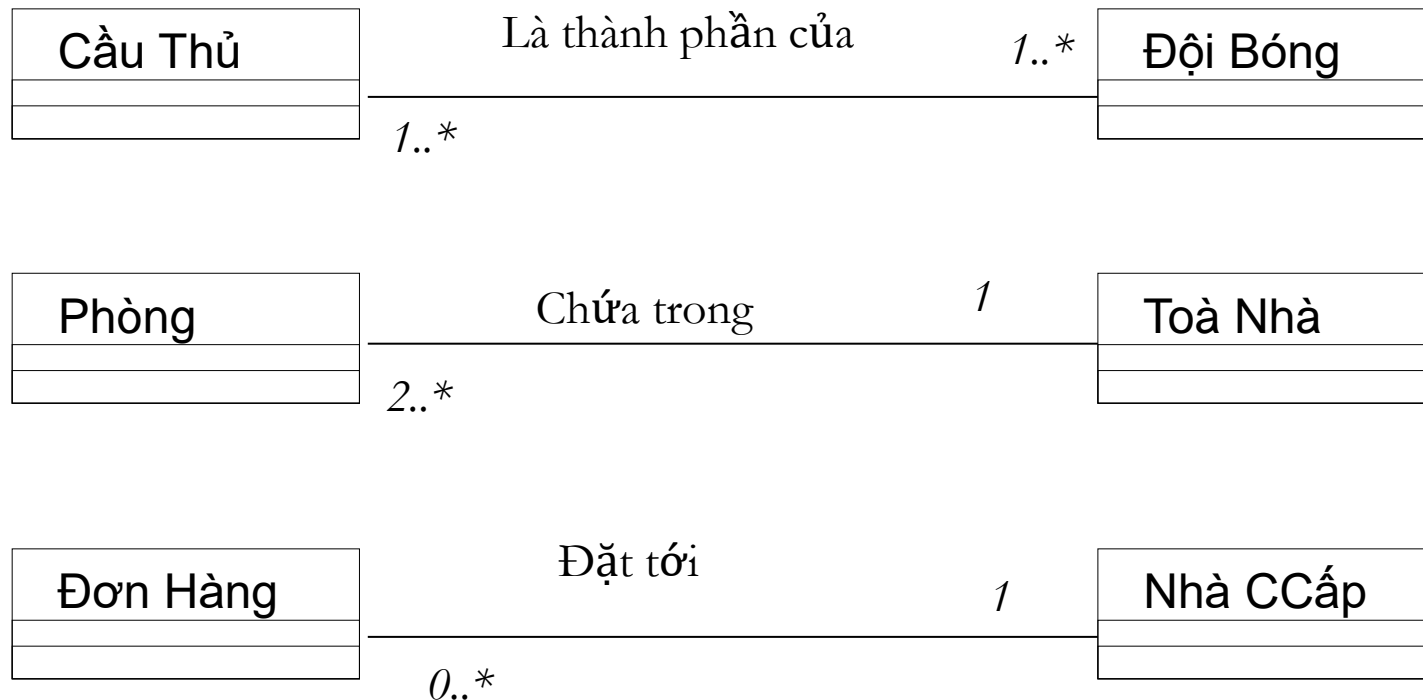
n..*  Số lượng được xác định bởi số n cho đến nhiều ($n \geq 0$).



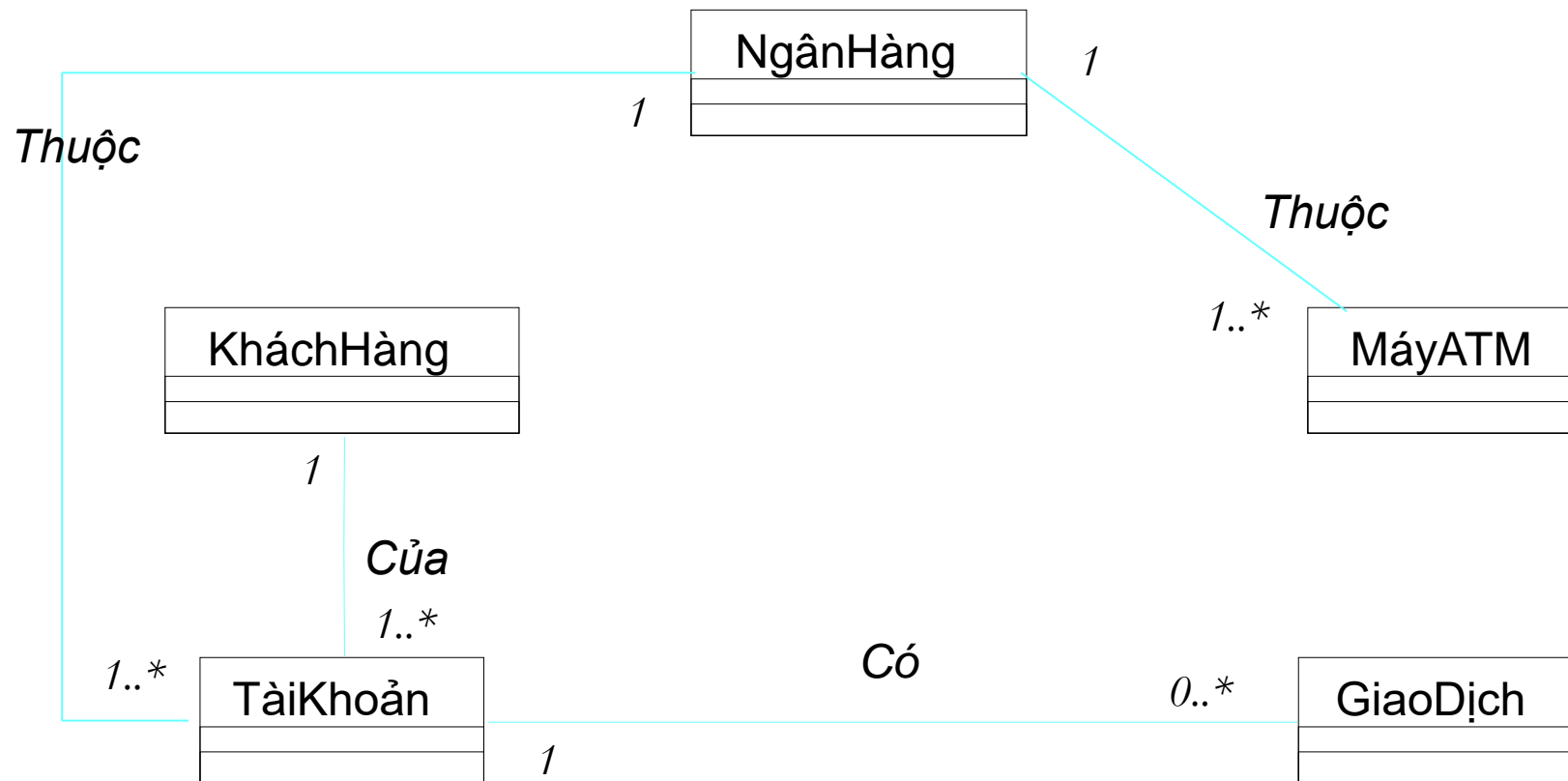
- Xác định mối kết hợp association:
 - Xác định bản số cho mỗi kết hợp: (min, max)
 - 1; 0..1;
 - 1..*;
 - 0..*;
 - a..* : a là hằng



- Xác định mối kết hợp association:
 - Xác định bản số cho mỗi kết hợp: (min, max)
 - Ví dụ:



- Xác định mối kết hợp association:
 - Xác định bản số cho mỗi kết hợp: (min, max)
 - Ví dụ: hệ thống ATM

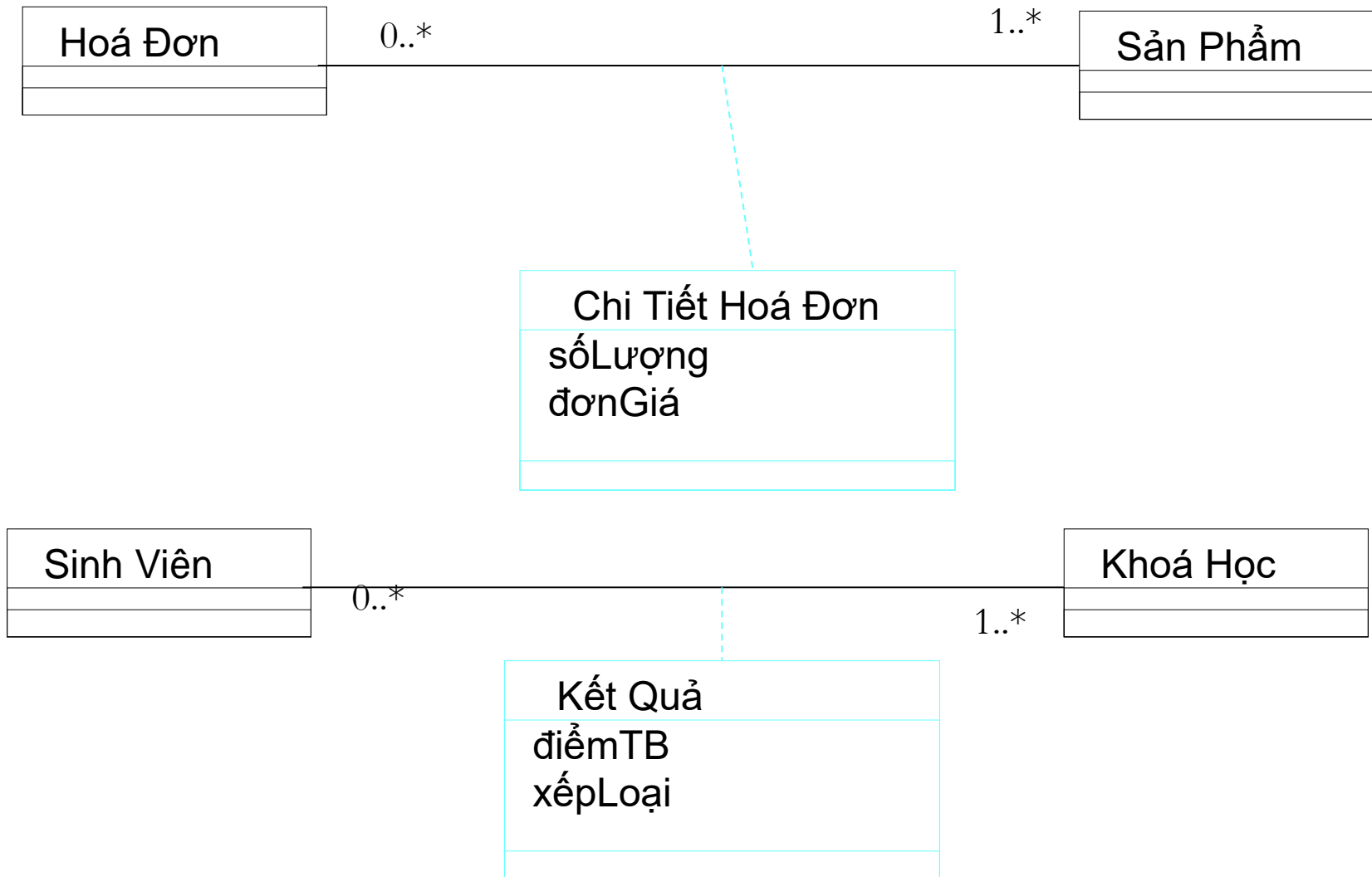


■ Xác định lớp kết hợp (Association class)

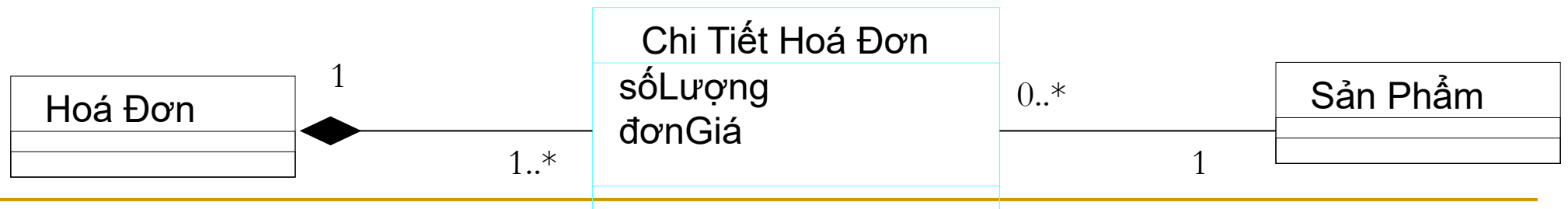
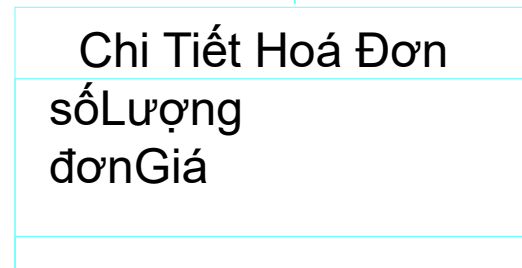
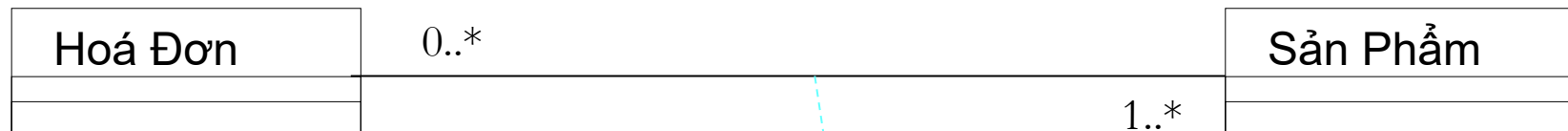
Lớp liên hệ (Association Class)

Một lớp có thể được đính kèm theo một liên hệ, trong trường hợp này nó sẽ được gọi là một lớp liên hệ. Một lớp liên hệ không được nối tới bất kỳ một lớp nào của mỗi liên hệ, mà tới chính bản thân mỗi liên hệ. Cũng giống như một lớp bình thường, lớp liên hệ có thể có thuộc tính, Phương thức và các quan hệ khác. Lớp liên hệ được sử dụng để bổ sung thêm thông tin cho nối kết (link), ví dụ như thời điểm nối kết được thiết lập. Mỗi nối kết của liên hệ gắn liền với một đối tượng của lớp liên hệ.

■ Xác định lớp kết hợp (Association class)



■ Xác định lớp kết hợp (Association class)



2. Quan hệ kết tập (Aggregation)

- **Kết tập** là một loại của quan hệ kết hợp, tập trung thể hiện quan hệ giữa tổng thể và bộ phận (*Whole / part*).
- **Kết tập** thường biểu diễn cho quan hệ “có” (*has-a*), “là bộ phận của” (*is-a-part-of*), hoặc “bao gồm” (*contains*)
- Người ta chia quan hệ kết tập thành 2 loại:
 - + **Kết tập chia sẻ** (*Shared Aggregation*)
 - + **Kết tập hợp thành hay quan hệ hợp thành** (*Composition*).

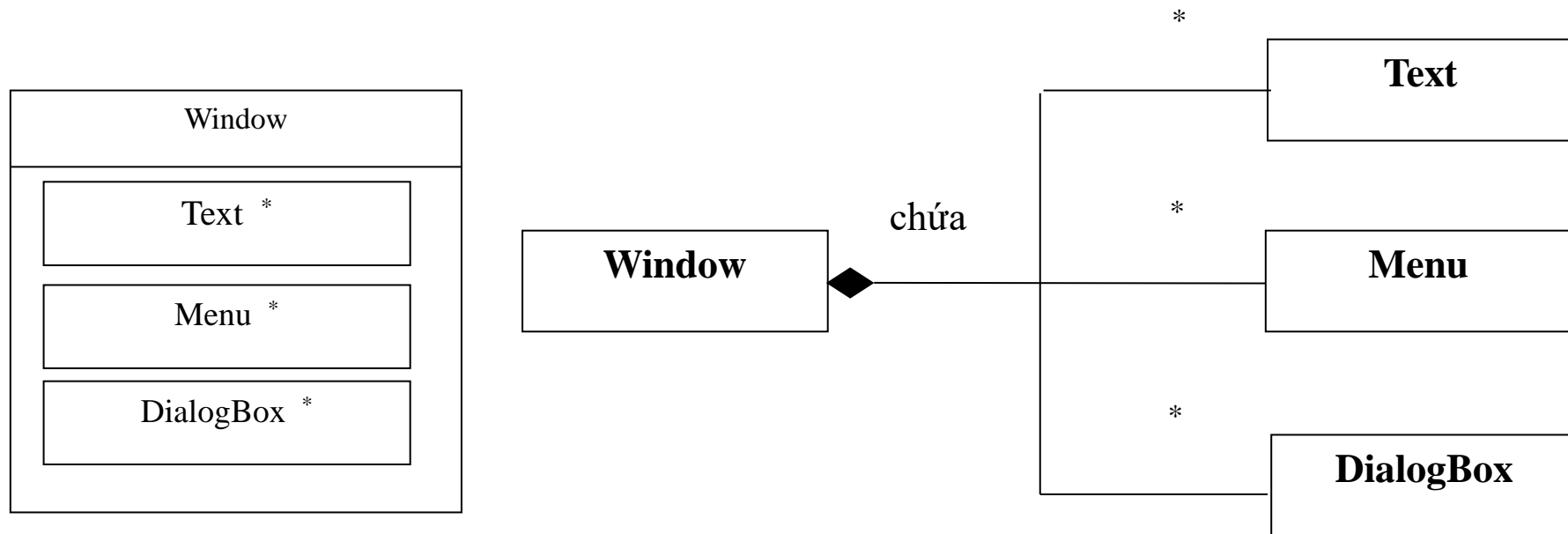
2.1. Kết tập chia sẻ:

- Quan hệ *kết tập chia sẻ* là loại kết tập, trong đó phía bộ phận có thể tham gia vào nhiều phía tổng thể.



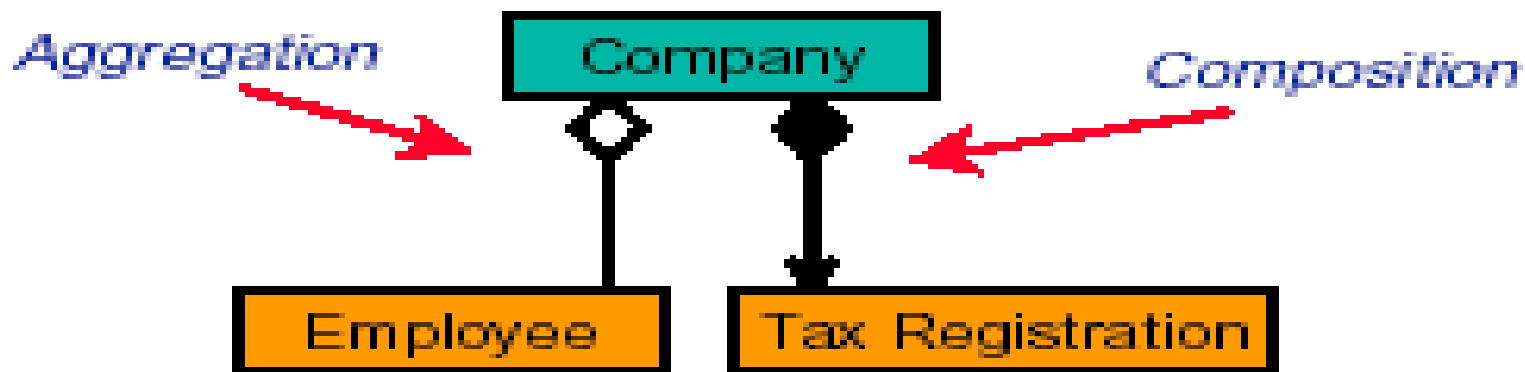
2.2. Hợp thành:

□ **Quan hệ hợp thành** chỉ ra một vật có chứa một số bộ phận và các bộ phận đó tồn tại vật lý bên trong vật tổng thể. Do vậy khi thực hiện huỷ bỏ, thiết lập mới bên tổng thể thì các bộ phận bên thành phần cũng sẽ bị huỷ bỏ hoặc phải được bổ sung.

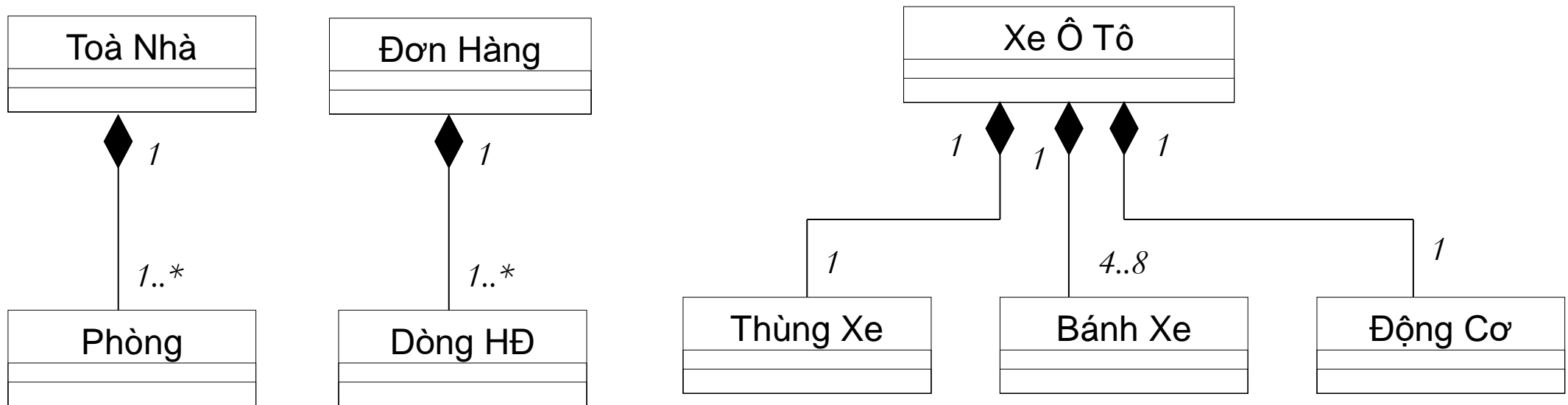


Quan hệ hợp thành (Compositions)

- Hợp thành là trường hợp đặc biệt của kết tập :
 - Đối tượng bộ phận chỉ thuộc 1 đối tượng tổng thể.
 - Các bộ phận cùng sống, chết cùng thời gian với đối tượng tổng thể.
- Ví dụ:
 - **Aggregation**: Một Công ty có nhiều nhân viên. Các nhân viên có thể thay đổi công ty làm việc.
 - **Composition**: Công ty có một số thuế. Mã số thuế luôn gắn chặt với Công ty, khi Công ty bị giải thể thì mã số thuế cũng bị hủy theo.

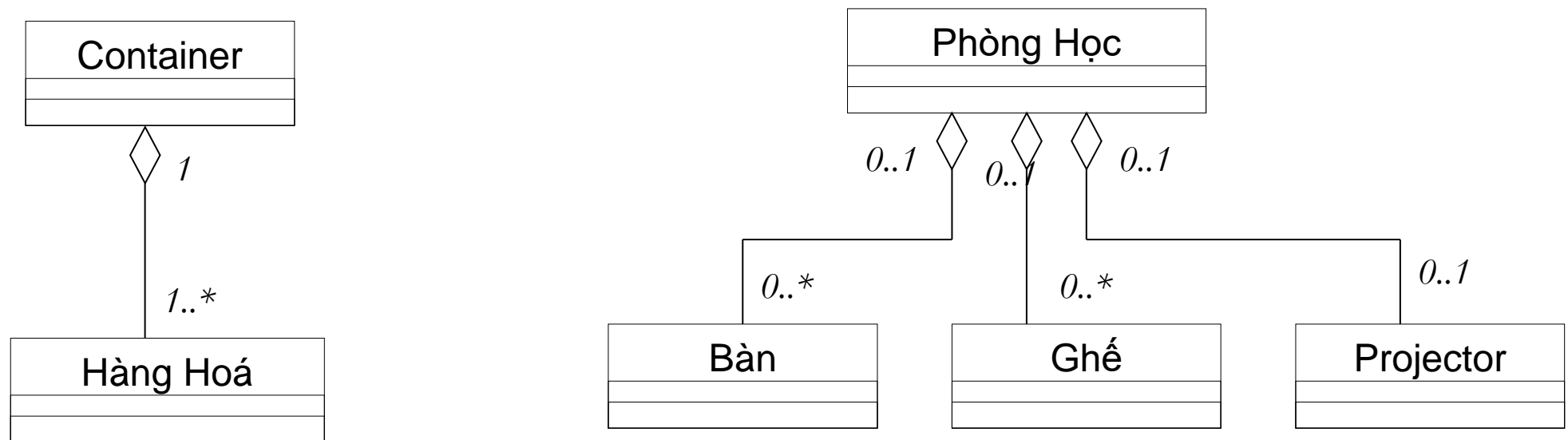


- ❑ Xác định mỗi kết hợp thành phần (a-part-of, aggregation)
- Tập hợp: một đối tượng vật lý được hình thành từ các đối tượng vật lý thành phần khác



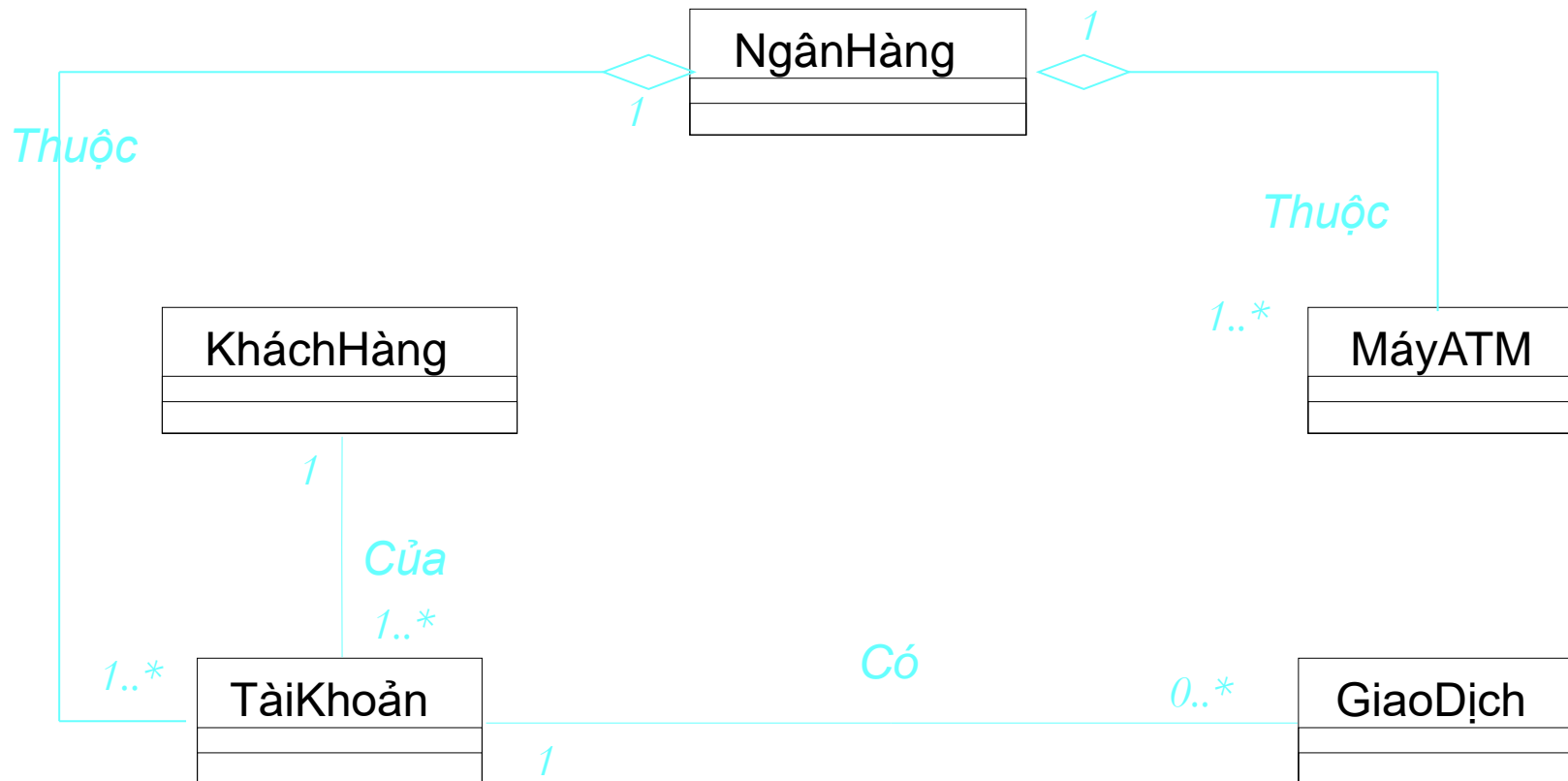
- Xác định mối kết hợp thành phần (a-part-of, aggregation)

- **Vật chứa**: một đối tượng vật lý chứa đựng các thành phần nhưng không được cấu tạo bởi các thành phần



Ví dụ:

- ❑ Xác định mối kết hợp thành phần
- ❑ hệ thống ATM



3. Quan hệ tổng quát hoá:

- *Tổng quát hoá* và *chuyên biệt hoá* là hai cách nhìn dưới / lên (*bottom – up*) và trên / xuống (*top-down*) để phân cấp các lớp, nó có khả năng quản lý cấp độ phức tạp của hệ thống bằng cách trừu tượng hoá các lớp.
- *Tổng quát hoá* là đi từ các lớp dưới lên sau đó hình thành lớp tổng quát (lớp trên, lớp cha), tức là cây cấu trúc các lớp từ lá đến gốc.
- *Chuyên biệt hoá* là quá trình ngược lại của *tổng quát hoá*, nó cho các lớp dưới (lớp con) khác nhau của lớp cha.

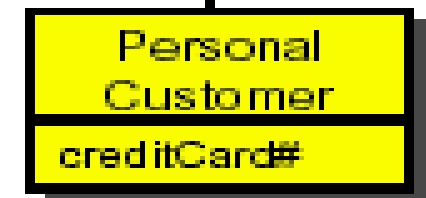
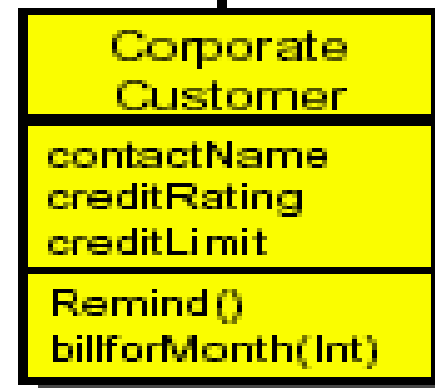
Quan hệ tổng quát hoá/ chuyên biệt hóa

- **Generalization:** Nghĩa là tập hợp những đặc tính giống nhau của các đối tượng con để tạo ra lớp cha.
- **Specialization:** Nghĩa là quá trình phân nhóm và làm mịn dần

Similarities are placed in a general superclass.



The differences are separated in specialized subclasses.

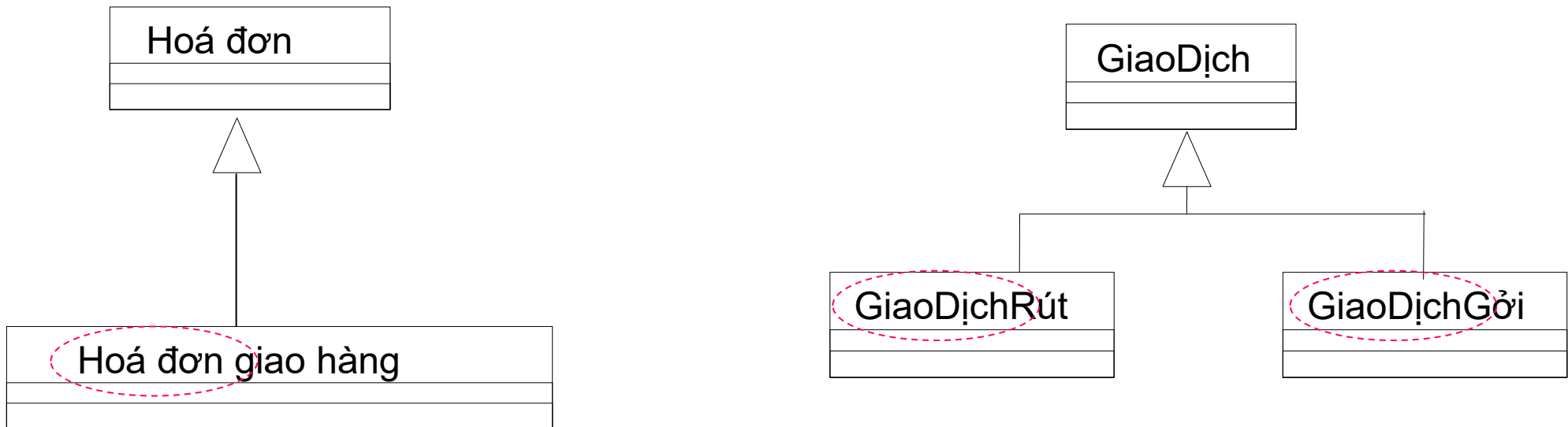


- ❑ Xác định mối kết hợp tổng quát – chuyên biệt (generalization): Thể hiện quan hệ kế thừa giữa các lớp và một cấu trúc phân cấp xác định những dòng kế thừa này

- **Tiếp cận top-down:**

- ❑ Từ một lớp chúng ta tìm kiếm cụm danh từ chứa tên lớp và tính từ (hoặc danh từ). Đánh giá xem cụm danh từ này có thể là một trường hợp đặc biệt cần được quản lý trong hệ thống không
- ❑ Tìm kiếm xem có những đặc trưng riêng của lớp
- ❑ Xây dựng mối kết hợp chuyên biệt từ lớp này đến lớp ban đầu

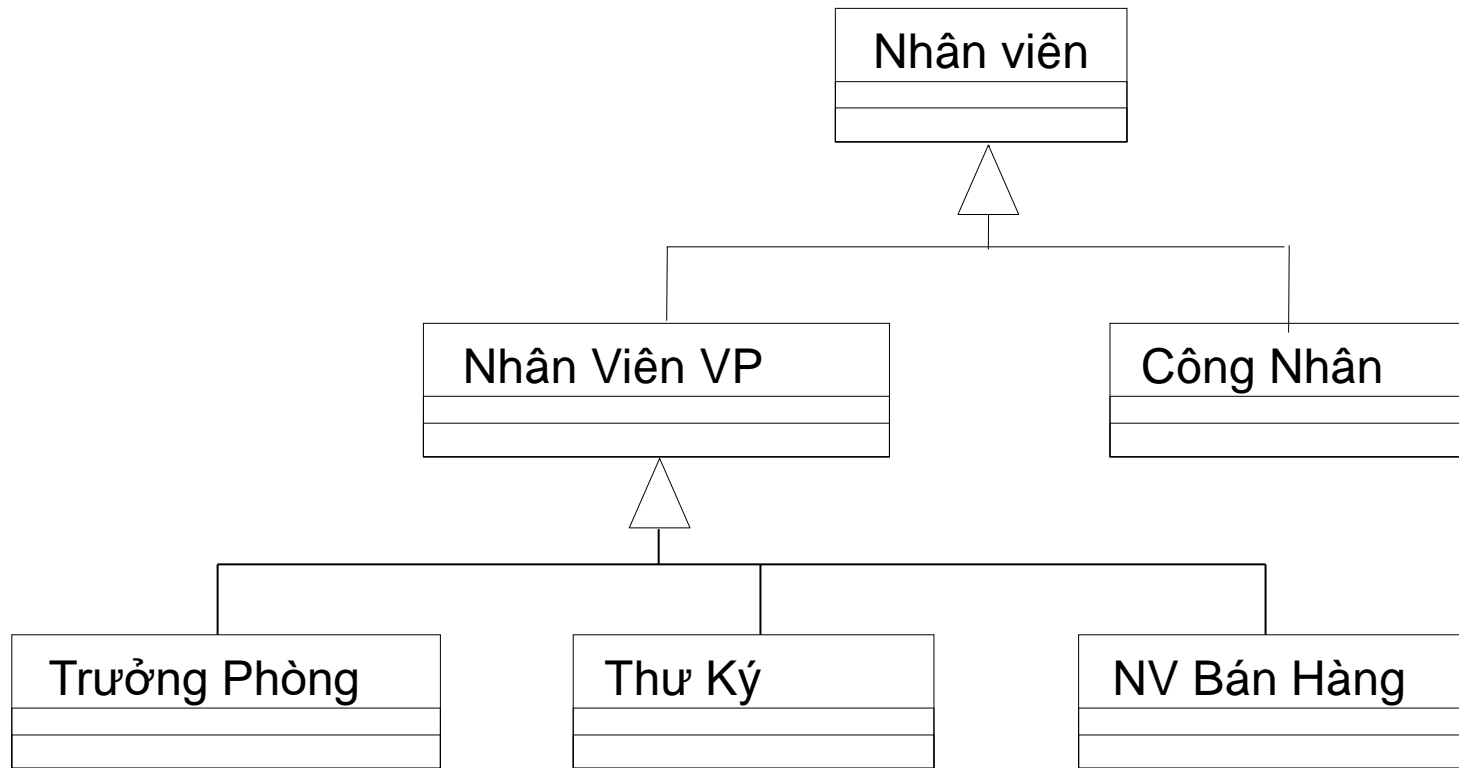
- ❑ Xác định mối kết hợp tổng quát – chuyên biệt (generalization):
 - Tiếp cận top-down – ví dụ:



Ghi chú: chỉ cần đưa vào các lớp chuyên biệt mà chúng ta xác định được các đặc trưng riêng (thuộc tính, method, liên kết) của nó trong hệ thống.

❑ Xác định mối kết hợp tổng quát – chuyên biệt (generalization):

- Tiếp cận top-down – ví dụ: phức tạp

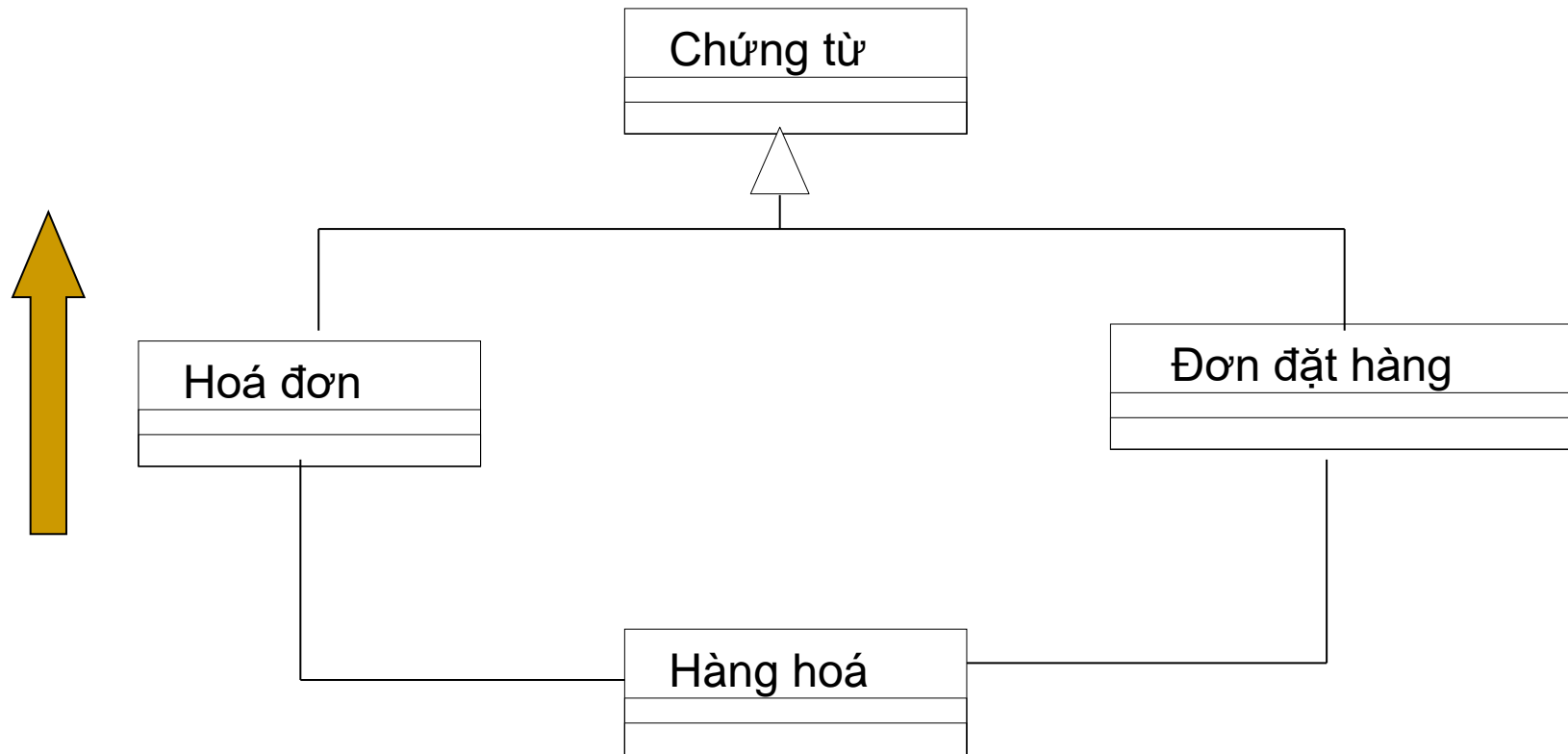


- ❑ Xác định mỗi kết hợp tổng quát – chuyên biệt (generalization):

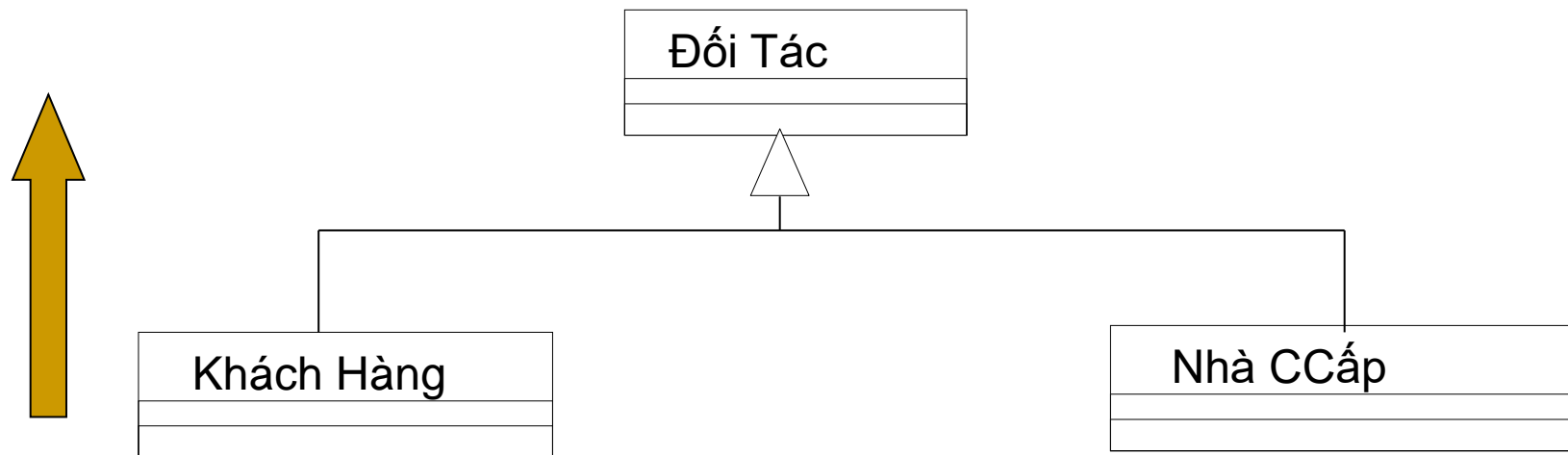
- Tiếp cận bottom-up:

- ❑ Tìm kiếm trong các lớp để xác định xem có các thuộc tính và phương thức giống nhau. Sau đó chúng ta có thể gom nhóm và đưa các thuộc tính và phương thức chung này lên một lớp tổng quát (trừu tượng)
- ❑ Tạo mỗi kết hợp tổng quát hoá từ các lớp này đến lớp tổng quát mới xác định

- ❑ Xác định mối kết hợp tổng quát – chuyên biệt (generalization):
 - Tiếp cận bottom-up – ví dụ:



- Xác định mỗi kết hợp tổng quát – chuyên biệt
 - Tiếp cận bottom-up – ví dụ:



Tìm thuộc tính cho lớp



Dựa trên tập các lớp đã được xác định, tiếp tục nghiên cứu kỹ các use case và scenario và trả lời các câu hỏi sau:

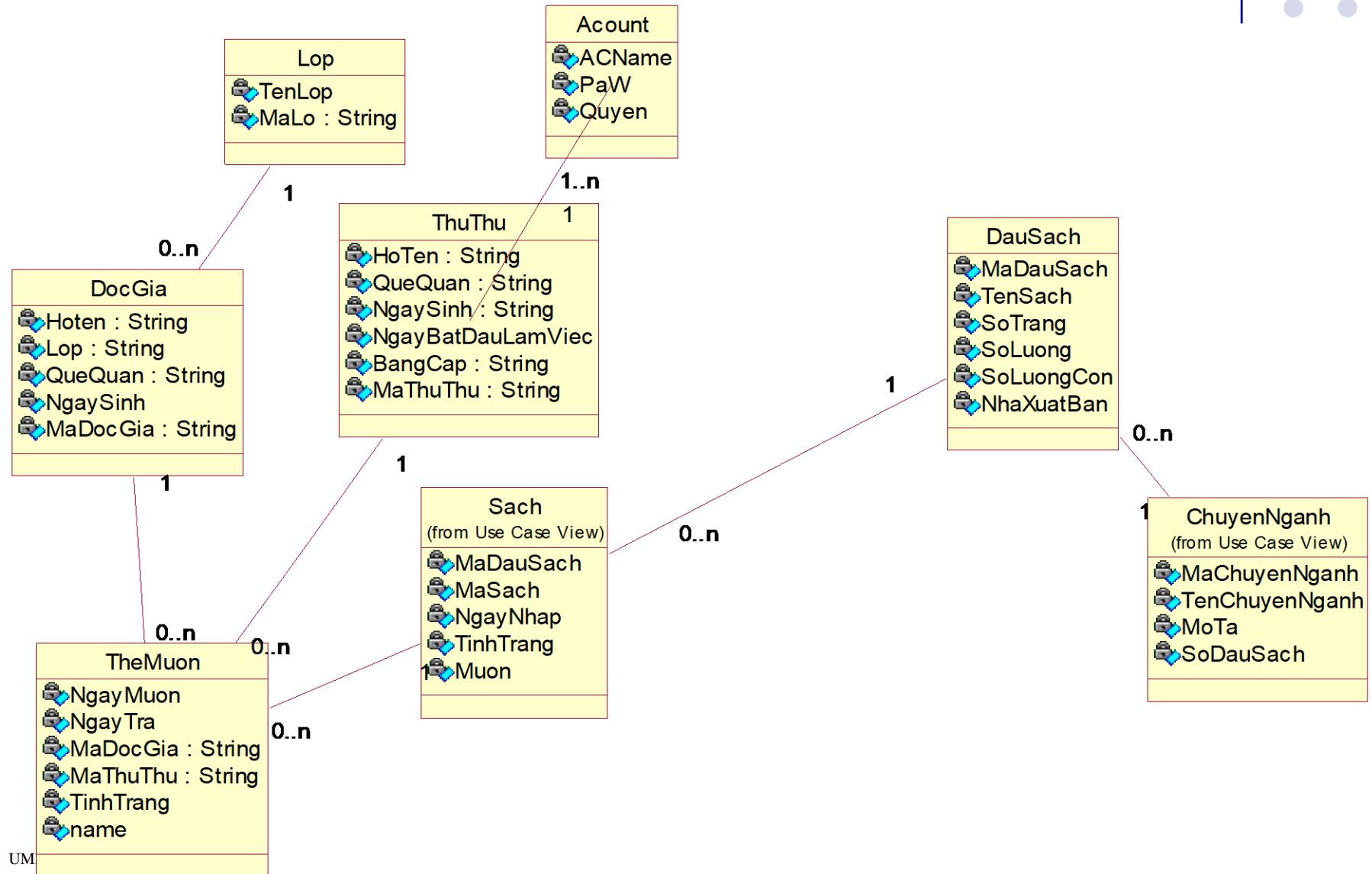
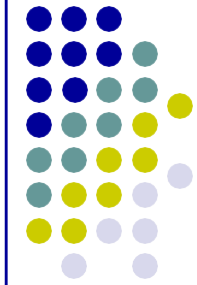
- + Với mỗi lớp, những danh từ nào mô tả thông tin của lớp đó tìm ra các thuộc tính.
- + Những thông tin nào của lớp thực sự liên quan đến lĩnh vực quan tâm của hệ thống loại bỏ các thuộc tính không cần thiết.
- + Những thông tin nào là thông tin riêng của lớp (các thuộc tính private), những thông tin nào có thể chia sẻ trong mối liên hệ với lớp khác (các thuộc tính protected hoặc public).

Các giai đoạn của mô hình hóa đối tượng



- 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)

Ví dụ biểu đồ lớp thực thể



UM

Phân tích UC – Xây dựng biểu đồ lớp

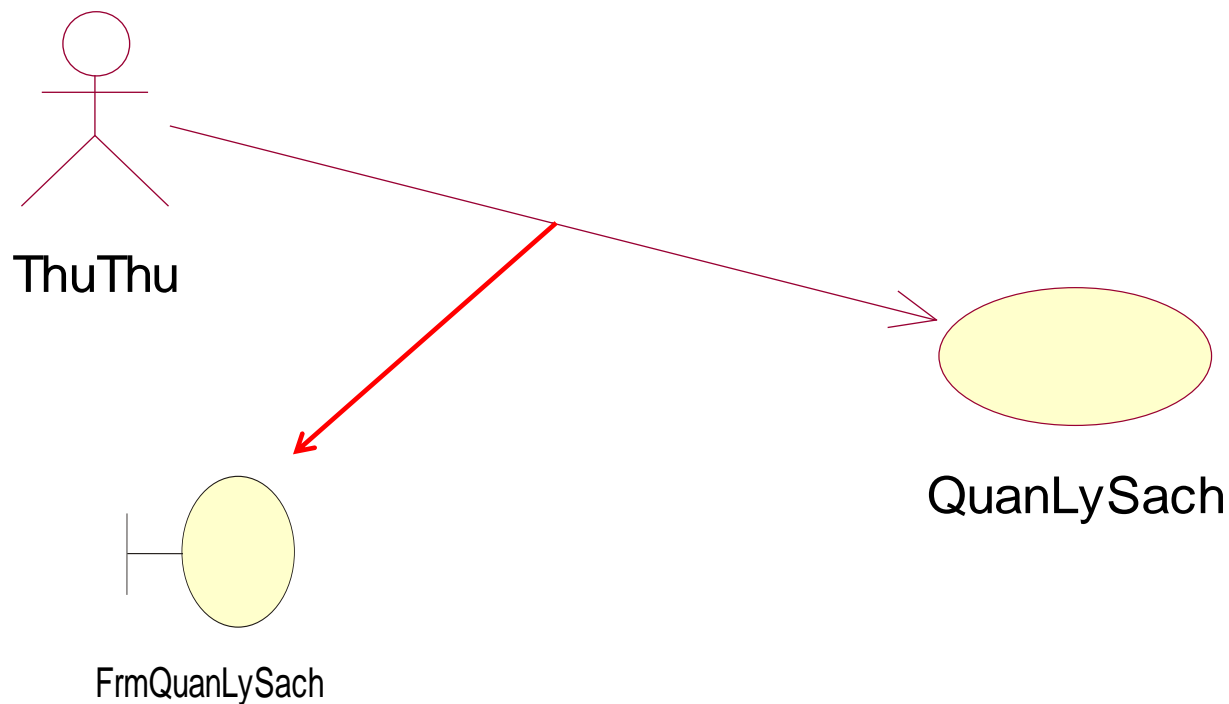


- Phân tích lặp lại cho tất cả các UC
- Đọc kịch bản UC
 - + Xác định lớp Bound
 - + Xác định lớp Control
 - + Xác định các lớp thực thể

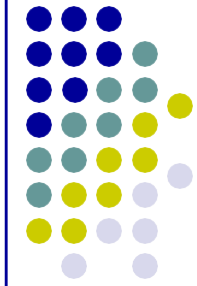
Phân tích UC – Xây dựng biểu đồ lớp



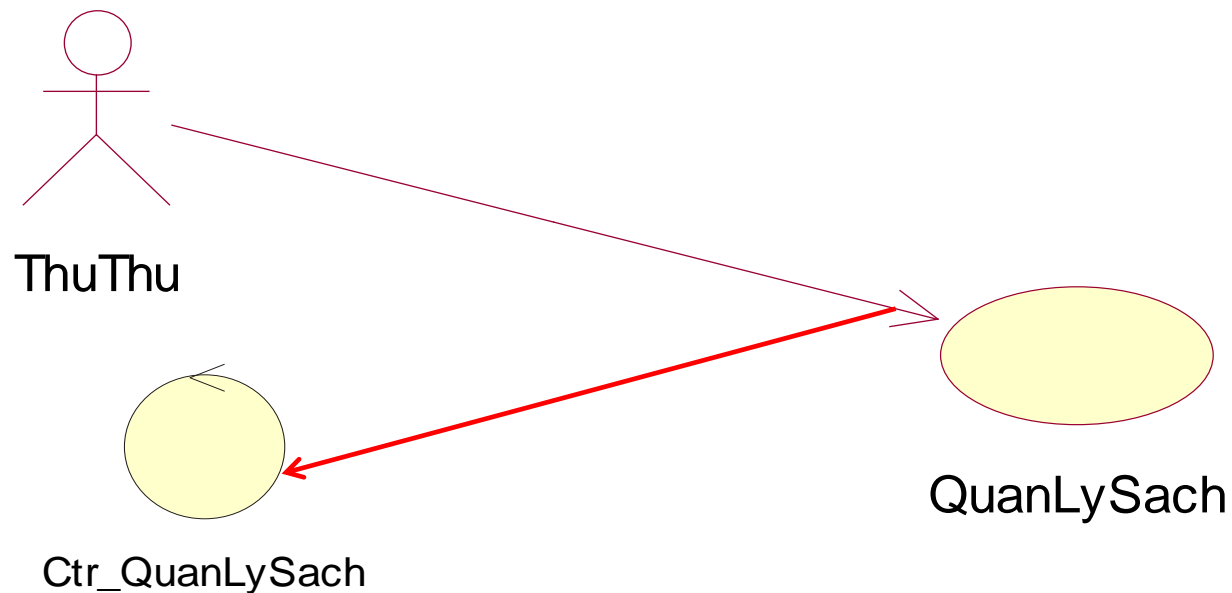
Xác định lớp boundary theo nguyên tắc: Mỗi một cặp Actor và UC có một lớp Bound.



Phân tích UC – Xây dựng biểu đồ lớp



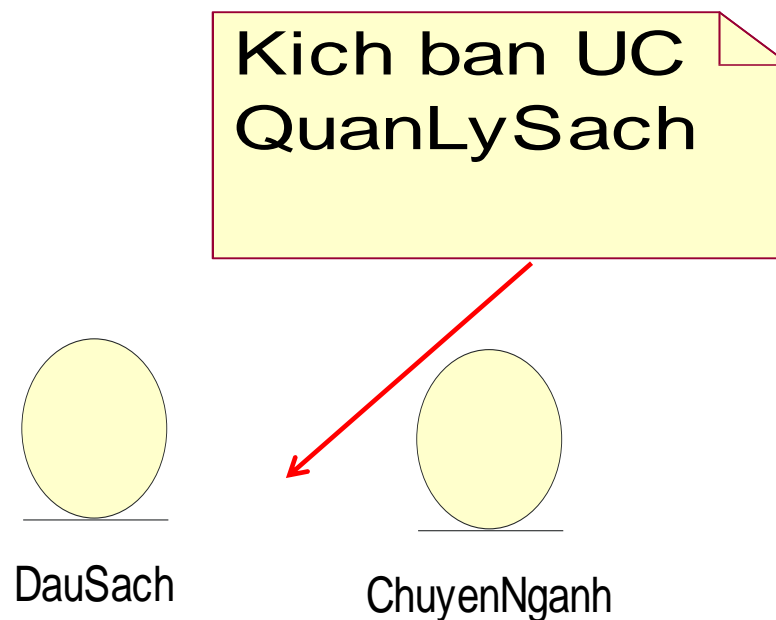
Xác định lớp control theo nguyên tắc: Mỗi một UC có ít nhất một lớp Control đảm nhận nghiệp vụ của UC



Phân tích UC – Xây dựng biểu đồ lớp



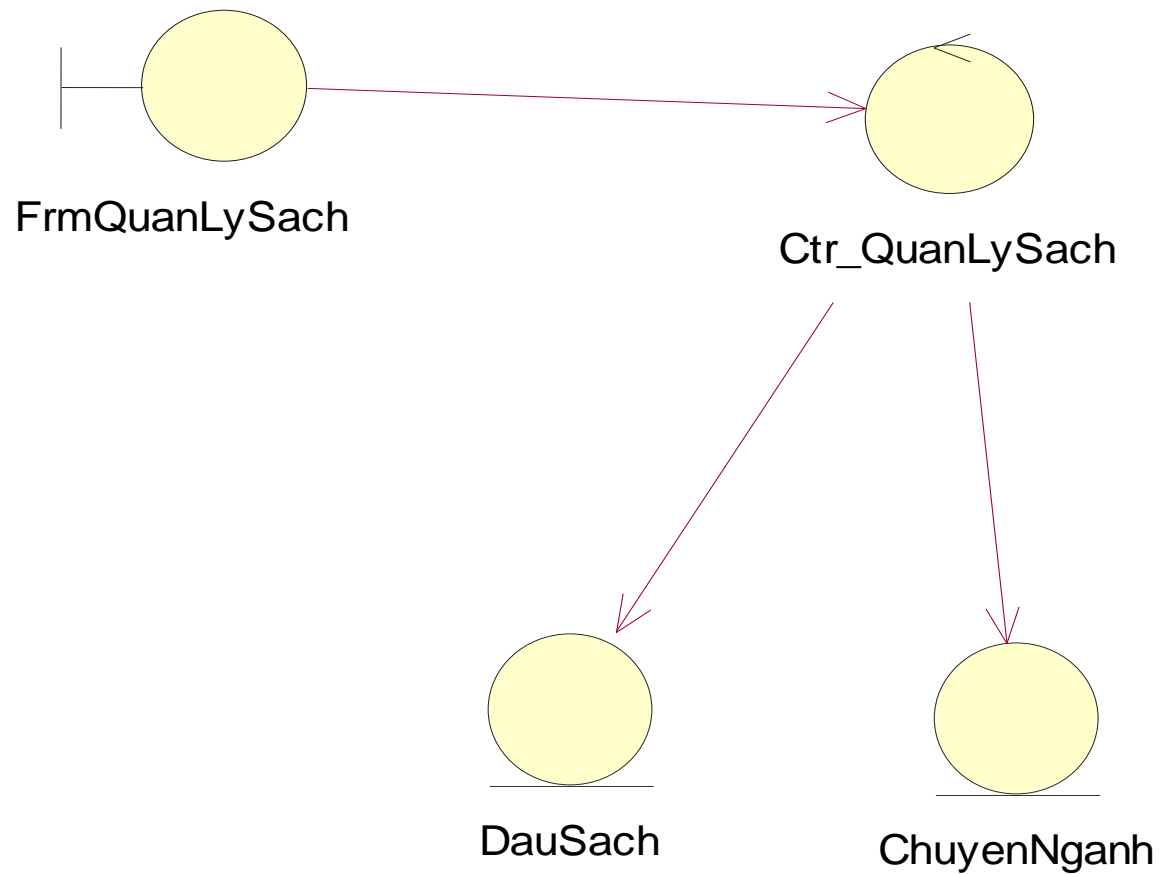
Xác định lớp Entity theo nguyên tắc: Đọc kịch bản UC xác định các danh từ, loại bỏ các danh từ lặp, loại bỏ danh từ là thuộc tính, loại bỏ danh từ là Actor

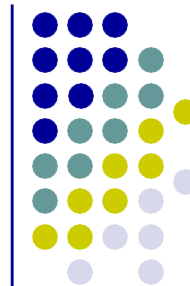


Phân tích UC – Xây dựng biểu đồ lớp



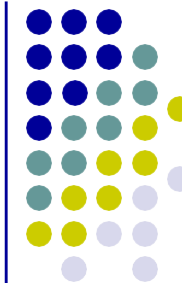
Biểu đồ lớp phân tích UC QuanLySach





THIẾT KẾ LỚP CHI TIẾT

XÁC ĐỊNH PHƯƠNG THỨC CHO LỚP

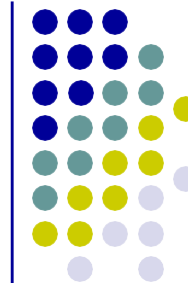


Xác định phương thức dựa vào thông điệp đến trong biểu đồ tương tác

Phương thức cần thiết chuyển đổi trạng thái

Hàm tạo, hàm hủy nếu cần thiết

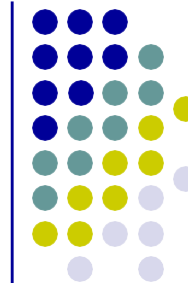
XÁC ĐỊNH MỐI QUAN HỆ GIỮA LỚP



Xác định dạng cụ thể quan hệ

Xác định lực lượng quan hệ

HOÀN CHỈNH BIỂU ĐỒ LỚP



Bổ sung lớp còn thiếu

Lớp Biên

Lớp trung gian: Tách từ mỗi quan hệ $m..n$

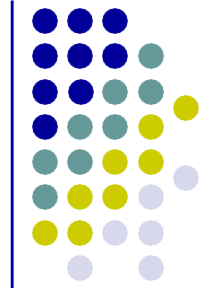
Lớp trừu tượng

Lớp điều khiển

Hiệu chỉnh mô tả thuộc tính và phương thức

Kiểm thử tính đúng đắn biểu đồ lớp

Thiết kế CSDL

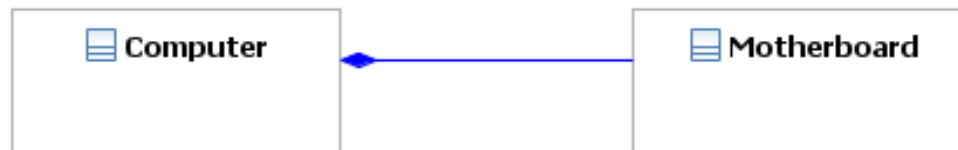


CSDL được ánh xạ từ biểu đồ lớp thực thể sang.

Quy tắc chuyển sang bảng CSDL



- Liên kết Association: 1-1 có mối quan hệ cộng gộp ví dụ như hình dưới chuyển khóa chính Motherboard thành khóa ngoại của Computer.



Quy tắc chuyển sang bảng CSDL

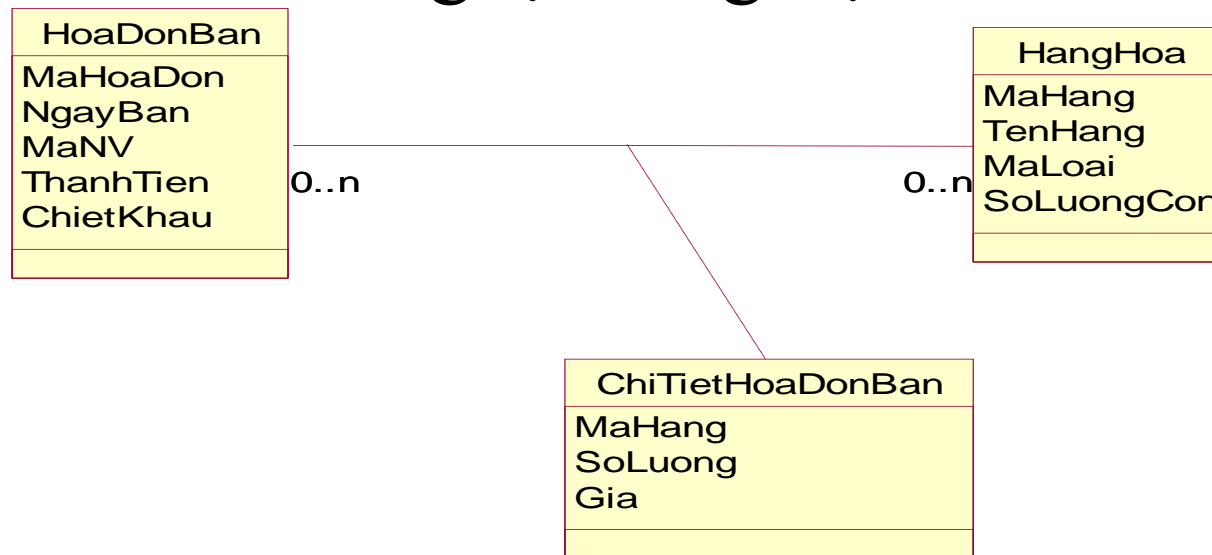


- Liên kết Association: 1-N chuyển hai lớp đầu liên kết thành hai bảng và chuyển khóa chính bên 1 sang bên nhiều

Quy tắc chuyển sang bảng CSDL



- Liên kết Association: M-N chuyển hai lớp đầu liên kết thành hai bảng, thêm bảng liên kết gồm các thuộc tính liên kết và chuyển khóa chính hai bảng thành khóa ngoại bảng liên kết.



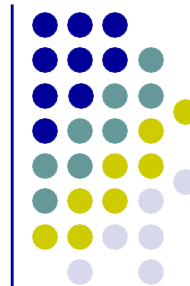
Quy tắc chuyển sang bảng CSDL



-Kế thừa: 2 cách chuyển

-Cách 1: Chuyển lớp cha thành bảng, bỏ các lớp con, thuộc tính của cha hợp các thuộc tính các lớp con. Thêm một thuộc tính loại (để xác định kiểu lớp con nào)

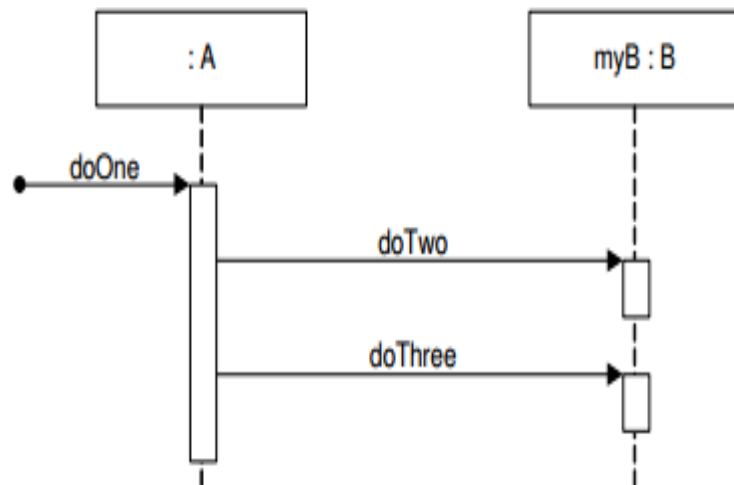
-Cách 2: Chuyển các lớp con thành các bảng, bỏ lớp cha.



Bước 4.3: BIỂU ĐỒ TUẦN TỰ

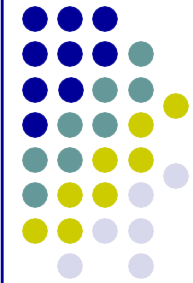


Sequence Diagram: là sơ đồ mô tả sự tương tác giữa các đối tượng theo hướng thời gian, nhấn mạnh thứ tự thực hiện các tương tác.

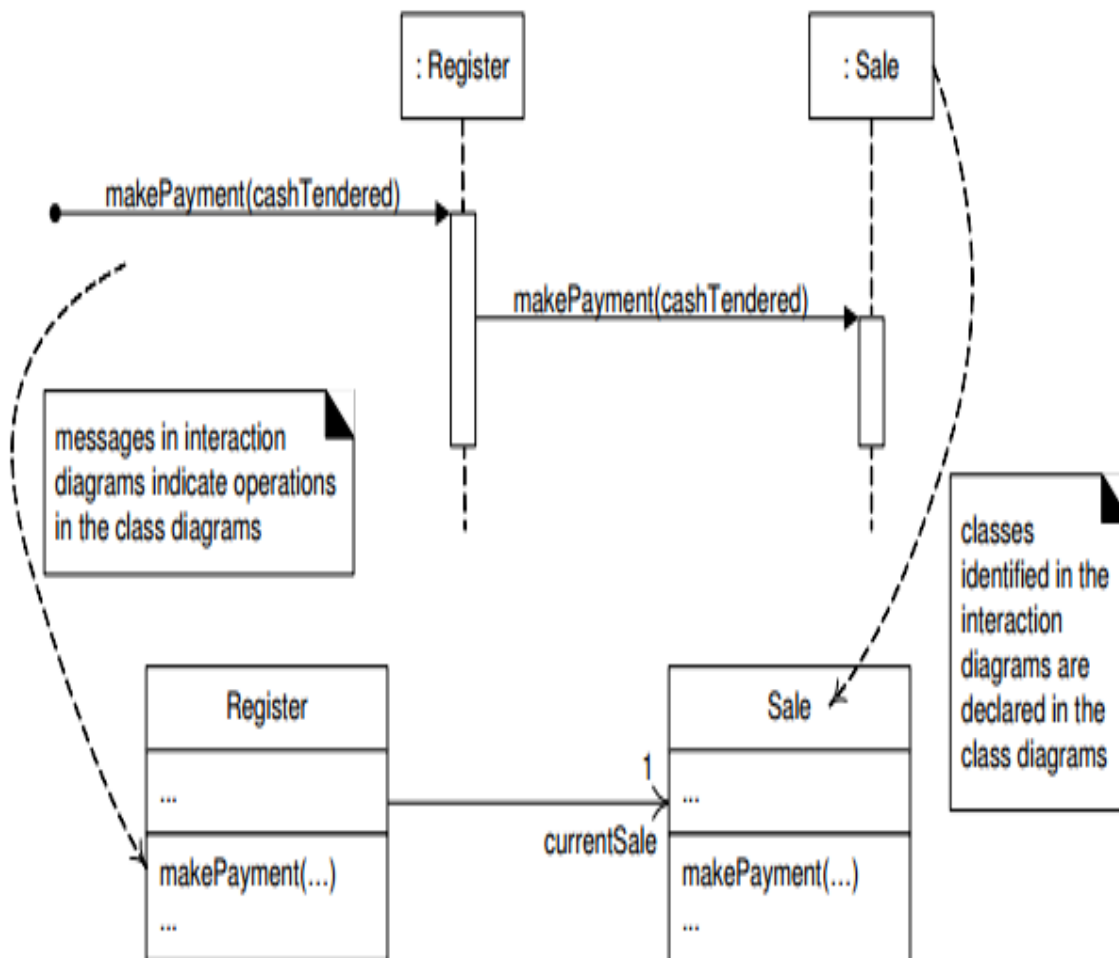


```
public class A {
    private B myB = new B();

    public void doOne() {
        myB.doTwo();
        myB.doThree();
    }
}
```



Mối quan hệ giữa biểu đồ tuần tự và biểu đồ lớp



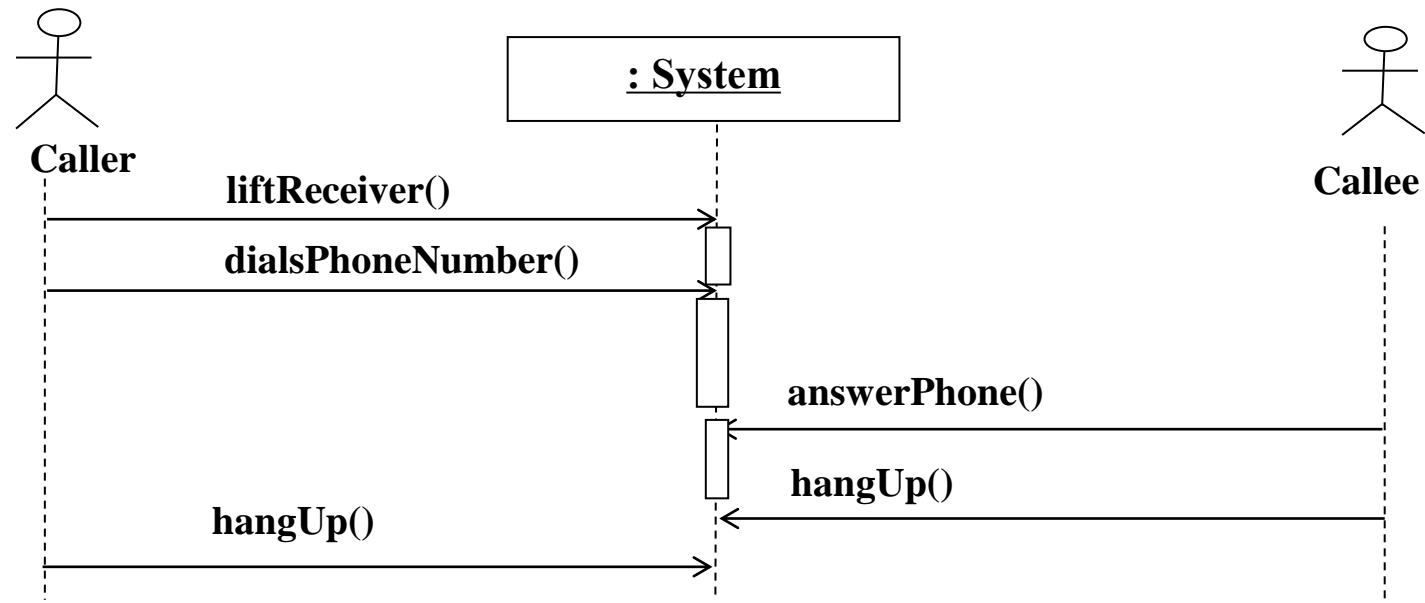
```
public class Register {  
    private Sale sale = new Sale();  
  
    public void makePayment(Money cashTendered) {  
        sale.makePayment(cashTendered);  
    }  
}
```

Biểu đồ trình tự

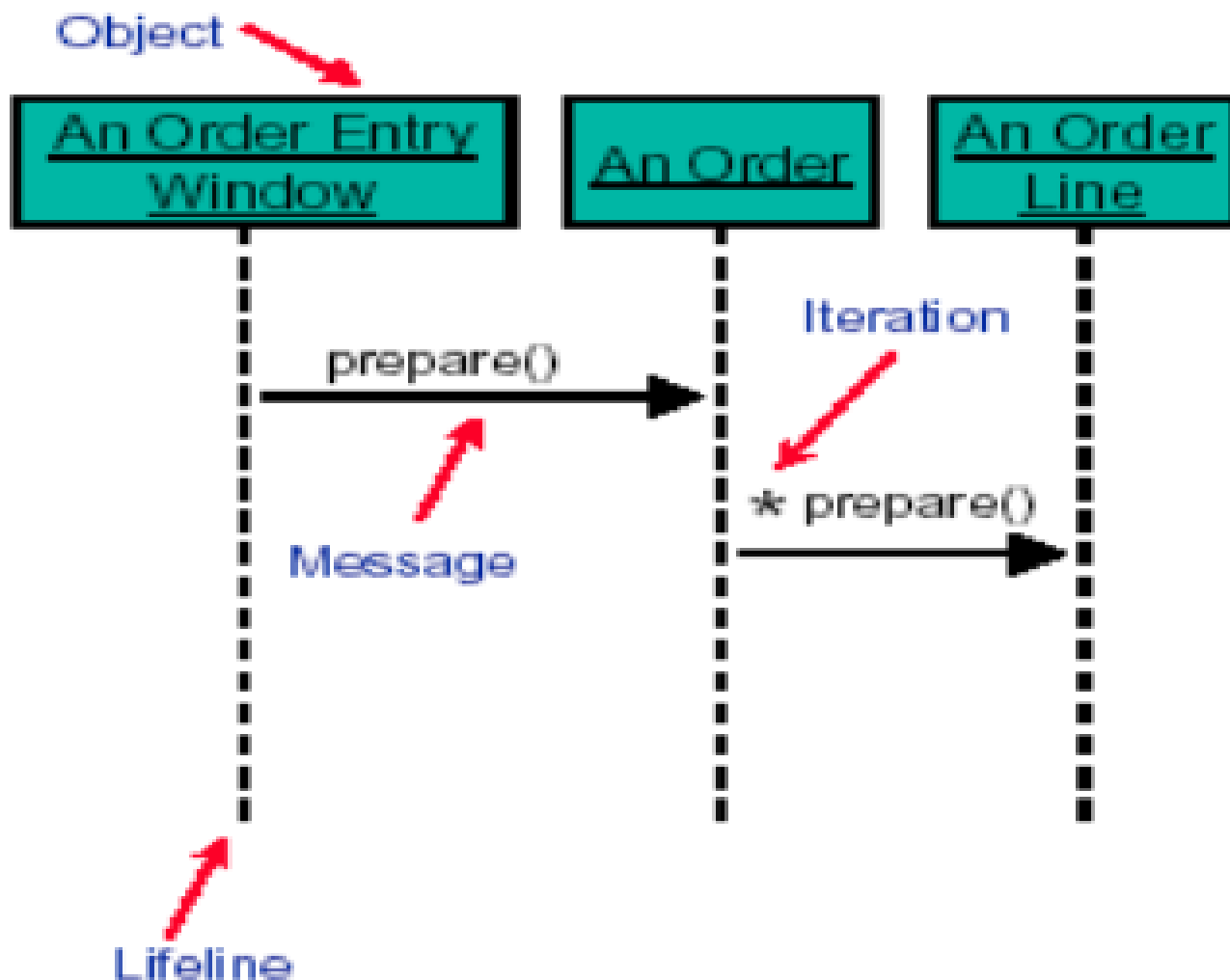
Các thành phần của biểu đồ trình tự

Biểu đồ trình tự được thể hiện theo hai trục:

- (i) **Trục dọc trên xuống chỉ thời gian** xảy ra các sự kiện, hay sự truyền thông điệp, được biểu diễn bằng các đường gạch - gạch thẳng đứng bắt đầu từ đỉnh đến đáy của biểu đồ.
- (ii) **Trục ngang từ trái qua phải là dãy các đối tượng tham gia** vào tham gia vào việc trao đổi các thông điệp với nhau theo chiều ngang, có thể có cả các tác nhân.

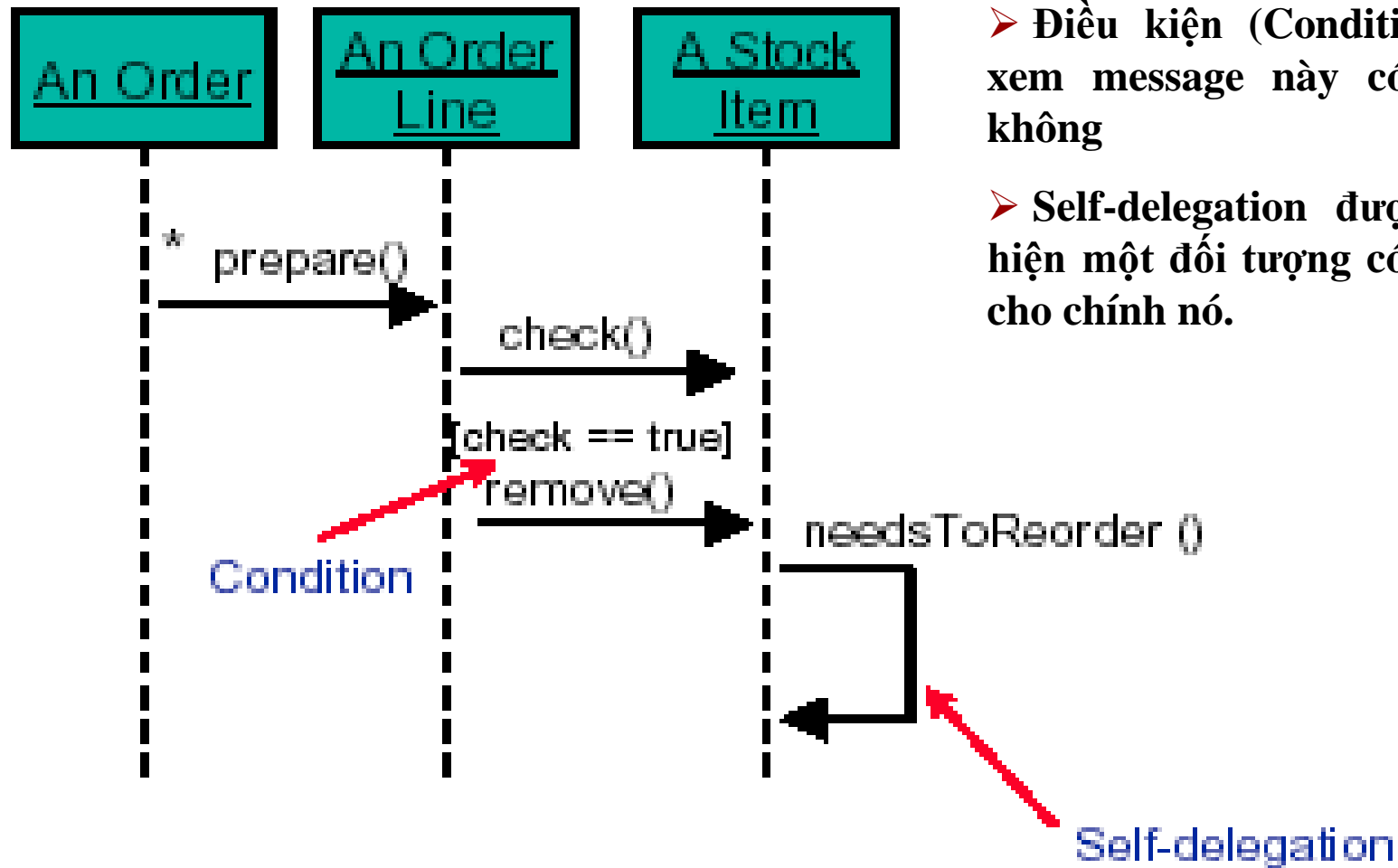


Biểu đồ trình tự (TT)



- Đối tượng và message thường được xác định được từ biểu đồ lớp
- Lifeline xác định thời gian sống của đối tượng khi tham gia vào UC
- Hai đối tượng trao đổi message với nhau chỉ khi chúng là 2 đối tượng của 2 lớp có quan hệ với nhau trong biểu đồ lớp
- Ký hiệu lặp (*) chỉ ra khả năng lặp lại việc trao đổi message đó một số lần.

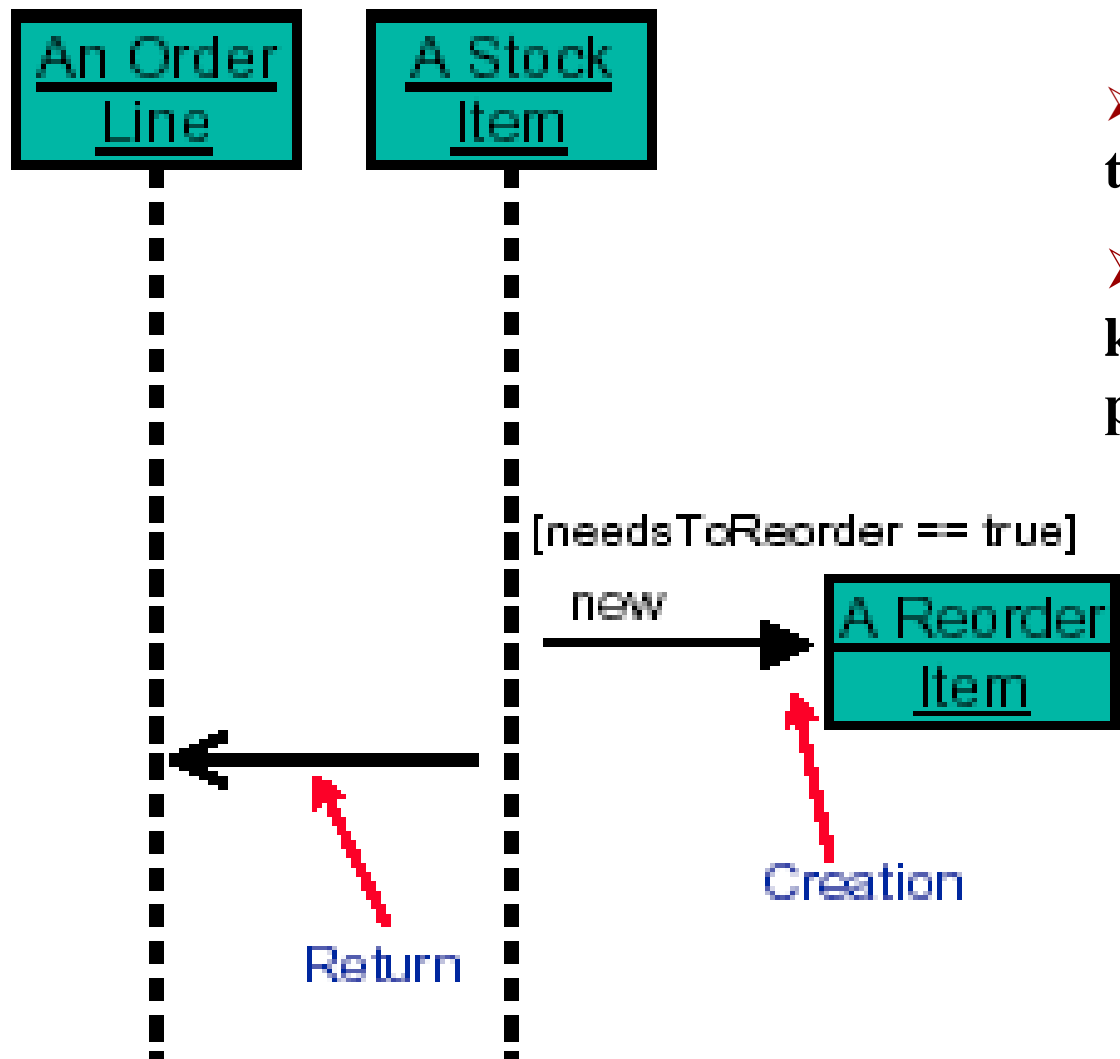
Biểu đồ trình tự (TT)



➤ Điều kiện (Condition) được kiểm tra xem message này có được gửi đi hay không

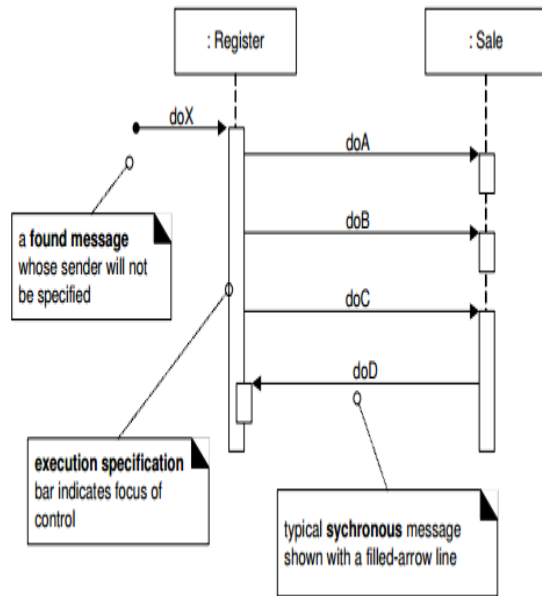
➤ Self-delegation được sử dụng để thể hiện một đối tượng có thể tự gửi message cho chính nó.

Biểu đồ trình tự (TT)

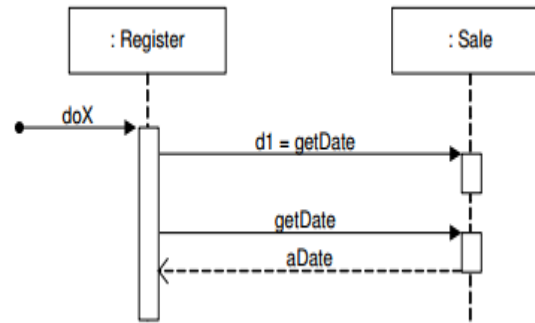


- Creation được sử dụng khi cần tạo ra một đối tượng mới
- Return để trả lại điều khiển sau khi thực hiện một message, không phải message mới.

Thông điệp



Reply or Returns



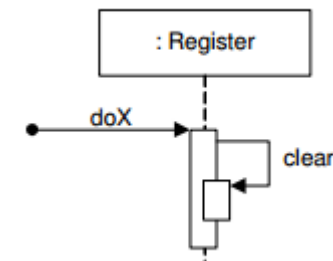
```

public class Register {
    private Sale sale = new Sale();

    public void doX() {
        Date d1 = sale.getDate();
    }
}

```

Messages to "self" or "this":

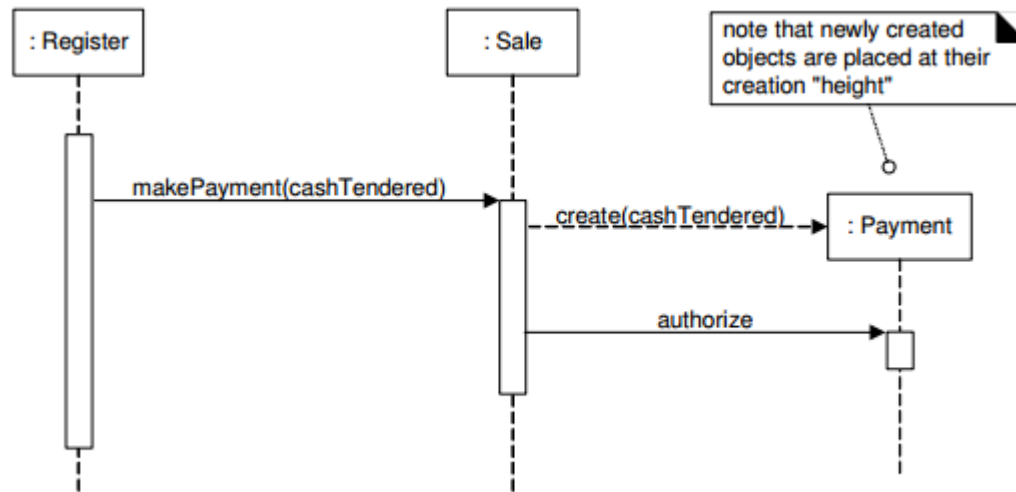


```

public class Register {
    public void doX(){
        this.clear();
        ...
    }
    private void clear(){
        ...
    }
}

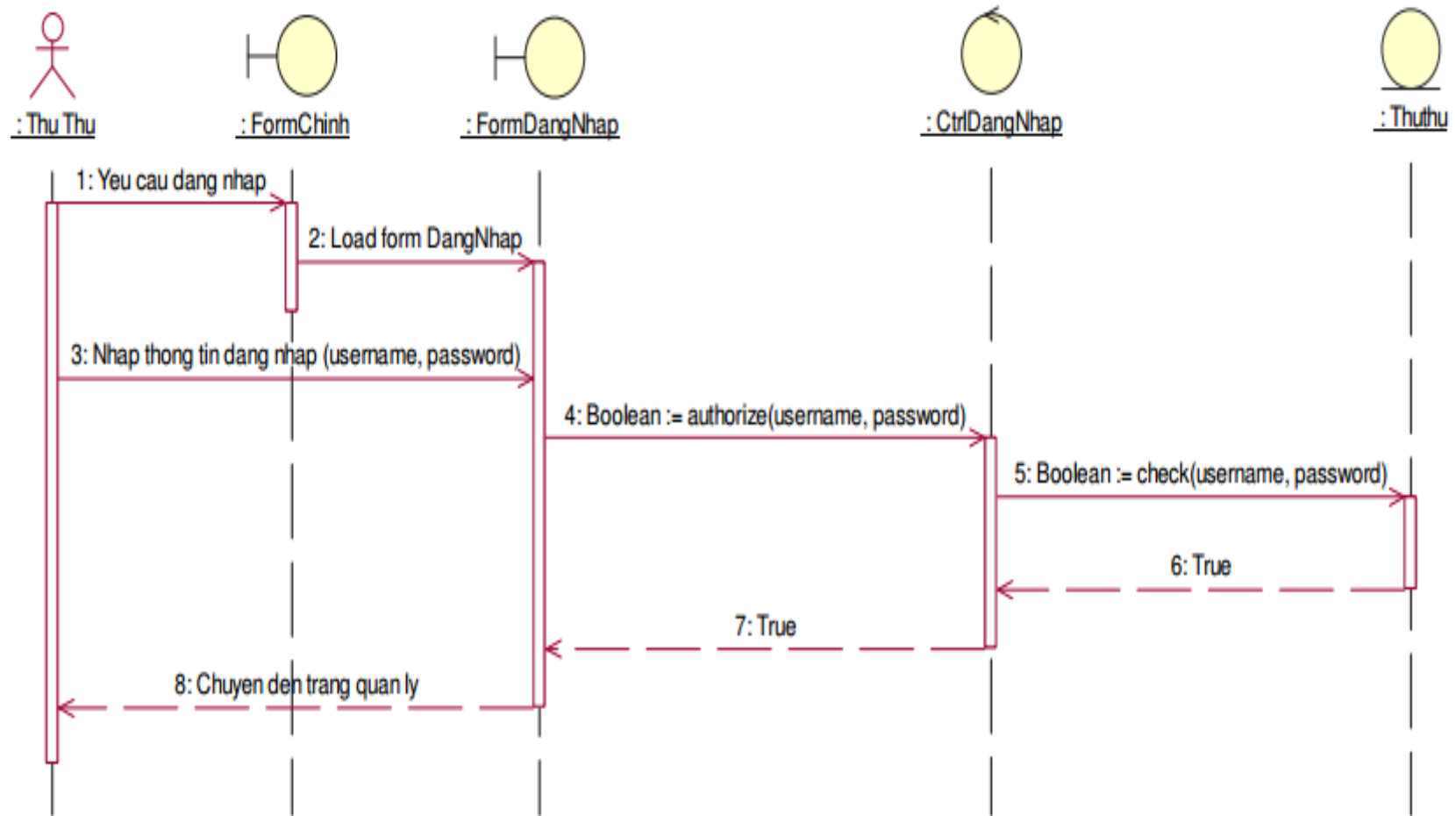
```


Creation of Instances:

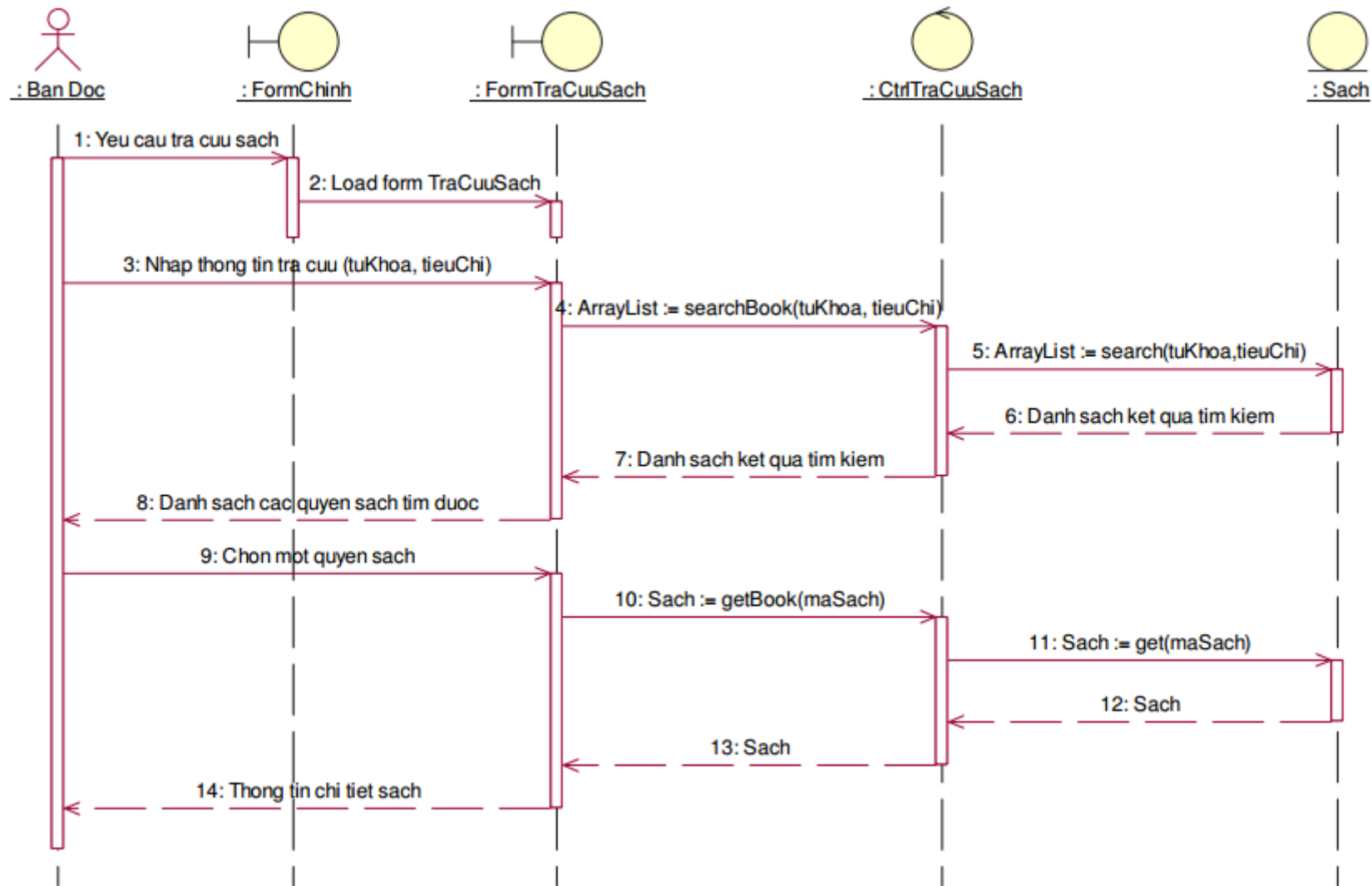


```
public class Sale {  
    private Payment payment;  
  
    public void makePayment(Money cashTendered) {  
        payment = new Payment(cashTendered);  
        payment.authorize();  
    }  
}
```

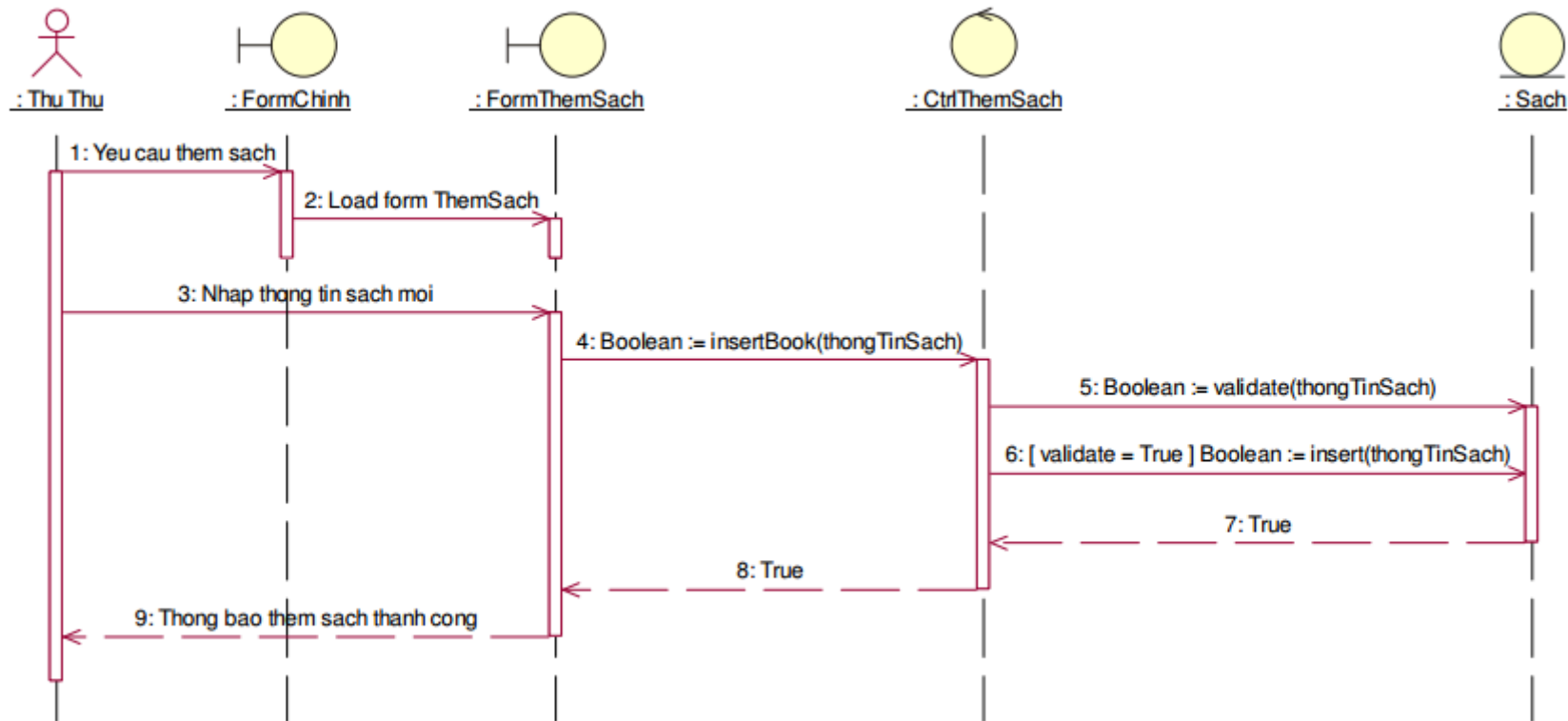
Ví dụ: Biểu đồ tuần tự của use case đăng nhập

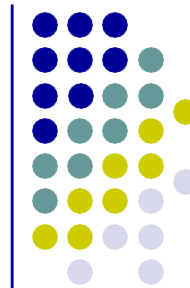


Vd: biểu đồ tuần tự của use case tra cứu sách



Biểu đồ tuần tự của use case thêm sách





Bước 4.4: BIỂU ĐỒ HOẠT ĐỘNG

Biểu đồ hoạt động



Biểu đồ hoạt động (activity diagram) là biểu đồ mô tả một nội dung hoạt động, theo các luồng đi từ việc này sang việc khác. Thường được dùng để biểu diễn logic của 1 ca sử dụng, 1 kịch bản, 1 nhóm ca sử dụng, 1 quy tắc hay 1 thao tác phức tạp và được dùng theo nhiều cách khác nhau.

Như một ***công cụ phân tích***, nó được dùng để:

Mô tả các dòng nghiệp vụ (business flow).

Mô tả các dòng trong use-case hoặc giữa các use-case.

Biểu đồ hoạt động



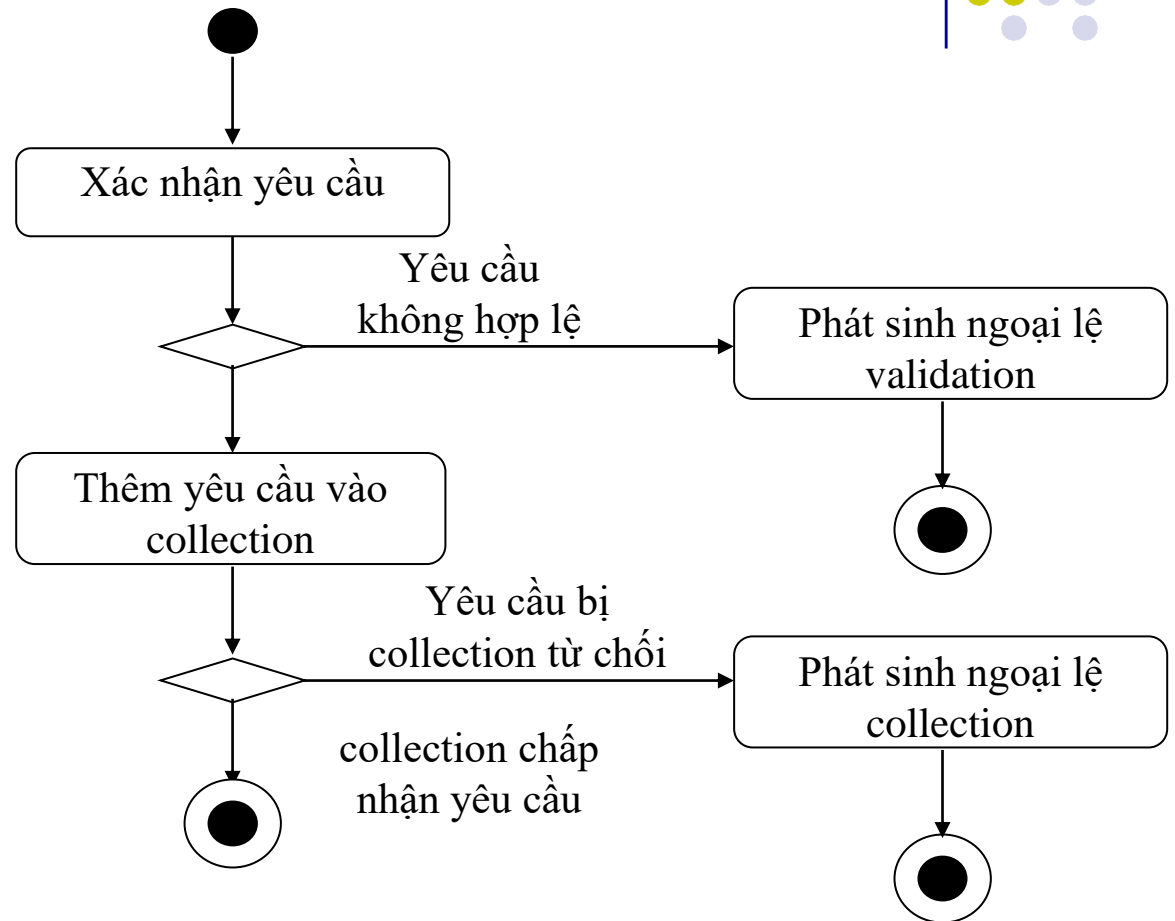
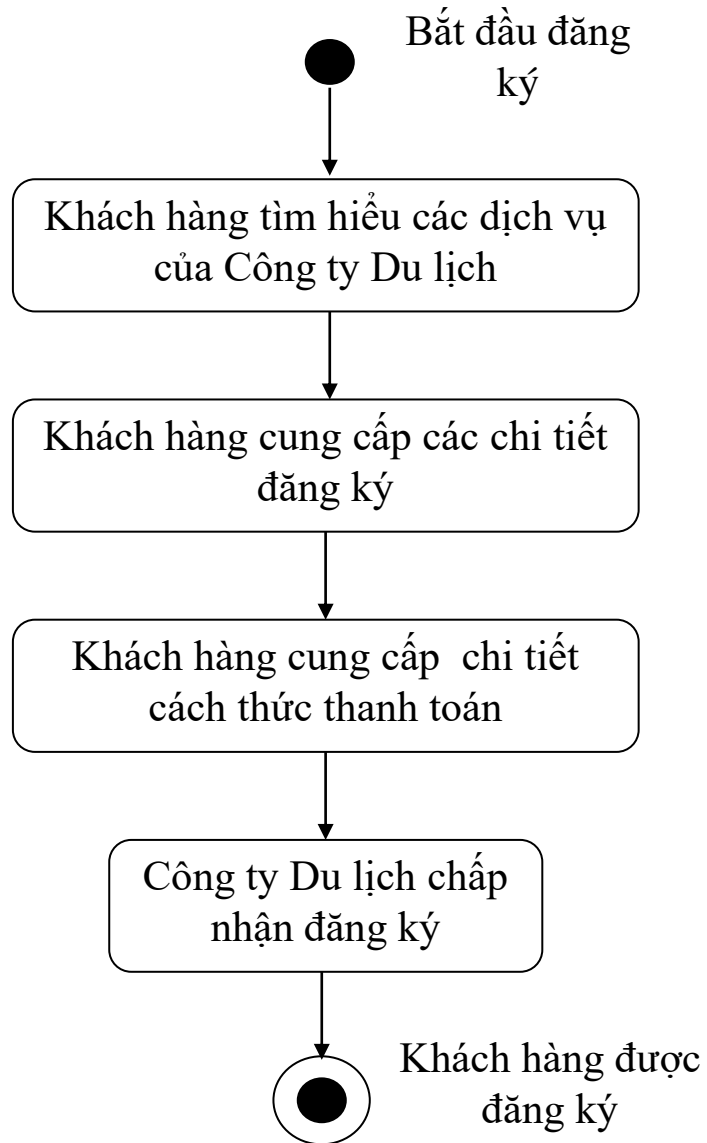
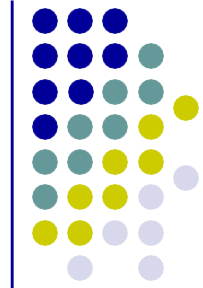
Biểu đồ hoạt động gồm:

- + *Hoạt động* (activity)
- + *Trạng thái* (state)
- + *Chuyển tiếp* (transition).

Nếu các hoạt động là chủ yếu thì ta gọi là **biểu đồ hoạt động**

Nếu trạng thái là chủ yếu thì ta gọi là **biểu đồ trạng thái**.

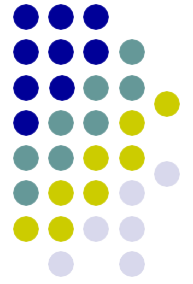
Biểu đồ hoạt động



Biểu đồ hoạt động

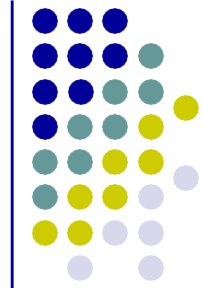
Biểu đồ hoạt động là 1 đồ thị có hướng, các nút là các hoạt động, các cung là các chuyển dịch

+ Hoạt động (Activity) là một công việc. Trong biểu đồ hoạt động, hoạt động được biểu diễn bằng 1 hình chữ nhật tròn góc, có mang tên của các hoạt động



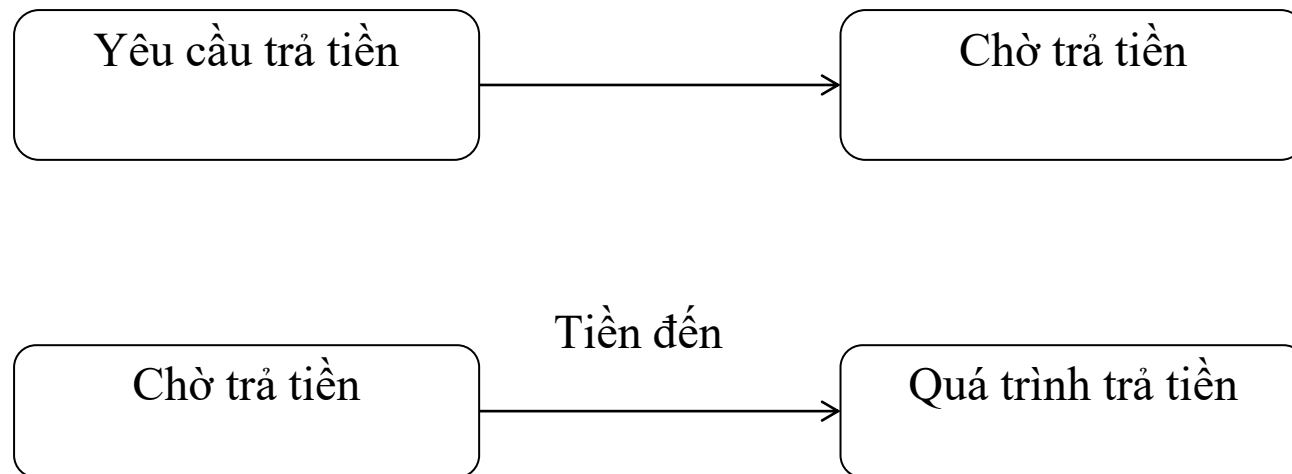
Xác nhận
yêu cầu

Biểu đồ hoạt động



Dịch chuyển (Transition)

Chỉ ra luồng điều khiển từ hoạt động này đến hoạt động khác, được biểu diễn bằng hình mũi tên.



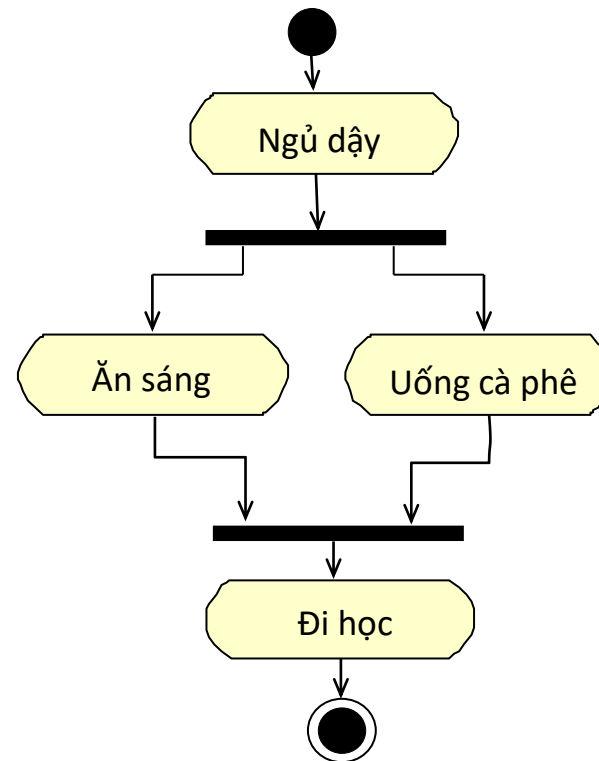
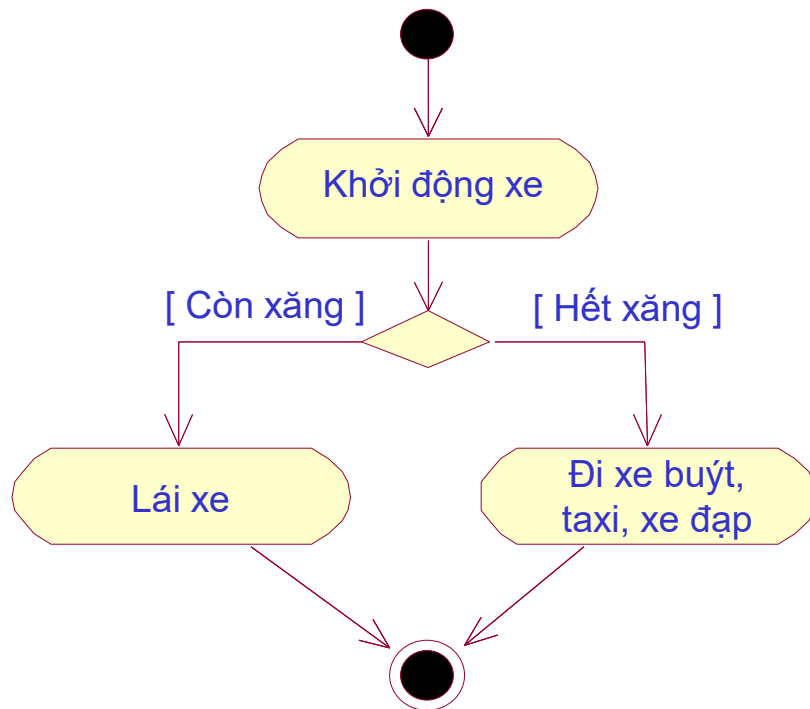
Biểu đồ hoạt động



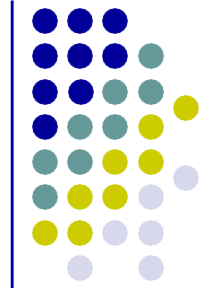
Điểm quyết định (decision point) là một điểm trên dòng công việc (được biểu diễn bằng hình thoi) mà tại đó thường là có một điều kiện quyết định cho việc rẽ nhánh của dòng công việc.

Đồng bộ (Synchronization)

Đồng bộ là cách mô tả hai hay nhiều nhánh flows xảy ra đồng thời

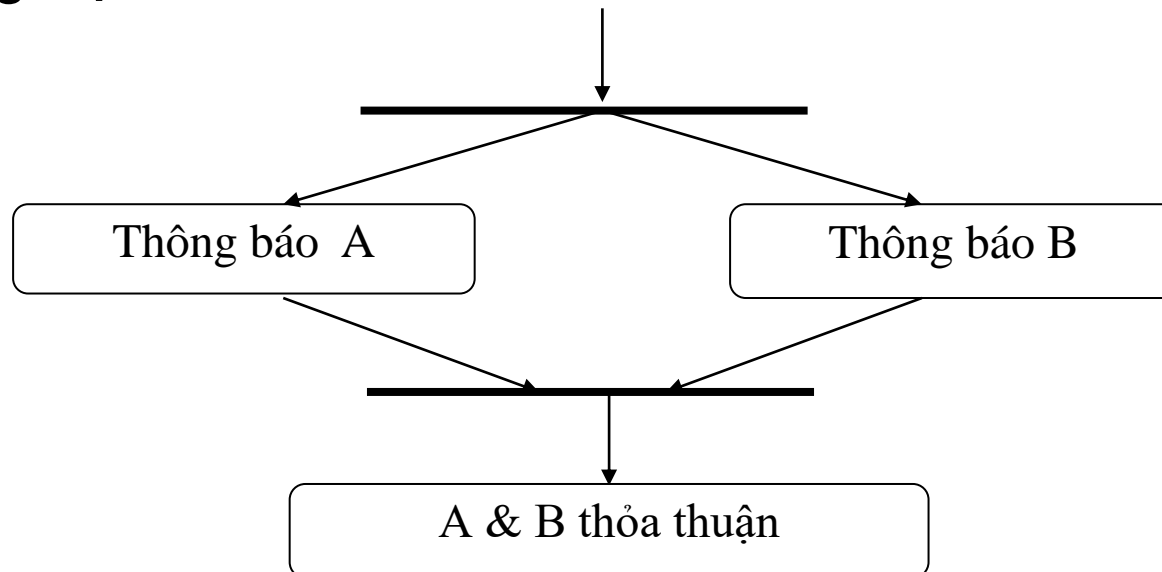


Biểu đồ hoạt động



Fork và Join (phân nhánh và kết hợp)

Đôi khi có một số hoạt động được tiến hành song song. Một chuyển dịch có thể được chia thành nhiều nhánh (điểm chia được gọi là fork) và ngược lại các nhánh có thể được kết hợp thành một chuyển dịch (điểm kết hợp được gọi là Join) bằng cách dùng một thanh đồng bộ hóa.



Biểu đồ hoạt động



Làn bơi (Swimlanes)

Sử dụng để mô hình hóa luồng công việc trong tiến trình nghiệp vụ

Chỉ ra ai có trách nhiệm thực hiện từng hoạt động

Để phân hoạch các trạng thái hoạt động vào nhóm

Phân tách nhóm trên biểu đồ bằng các làn bơi

Mỗi hoạt động thuộc về một làn bơi

Chuyển dịch có thể được vẽ từ làn bơi này đến làn bơi khác

Mỗi làn bơi có thể được cài đặt bởi một hay nhiều lớp

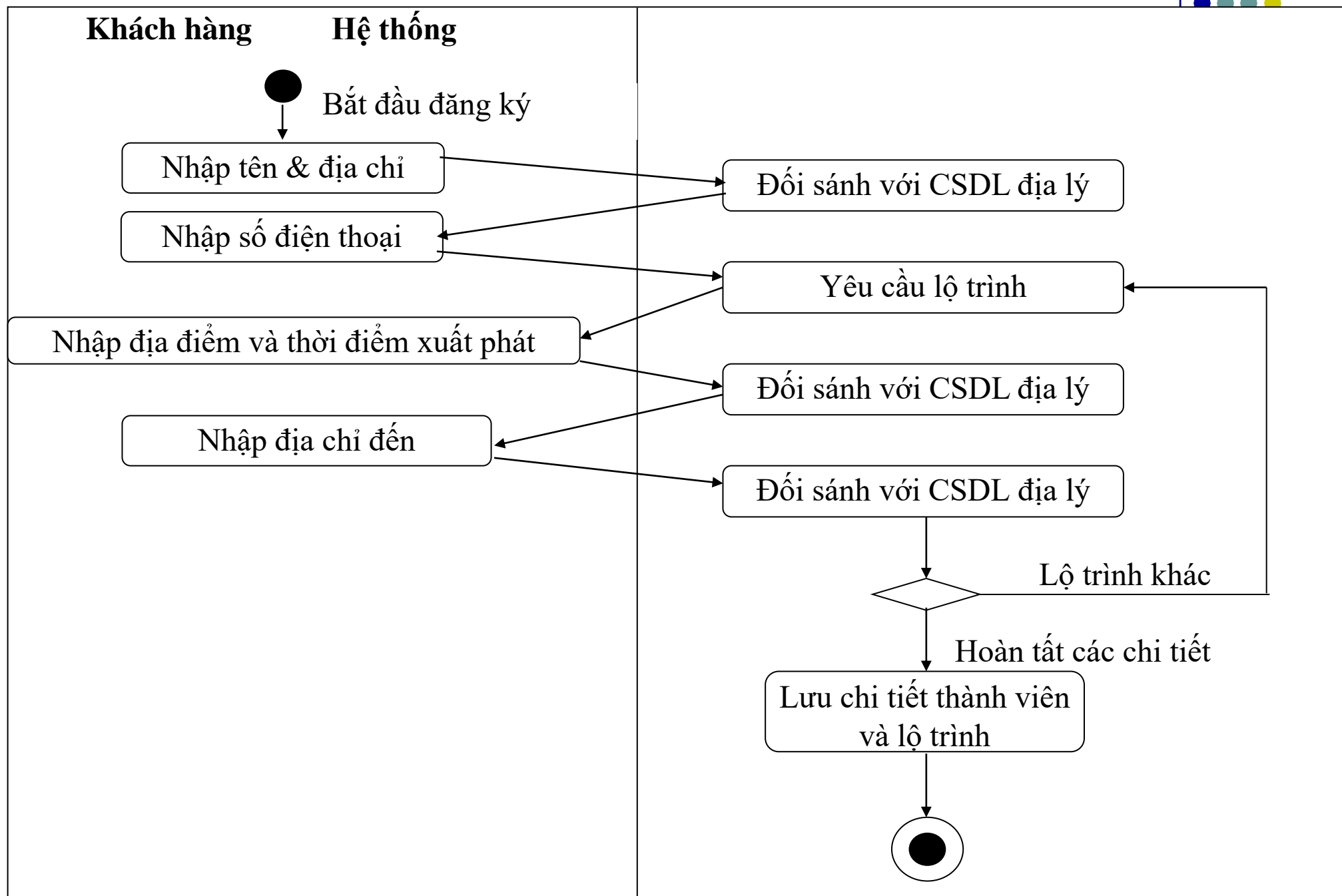
Biểu đồ hoạt động

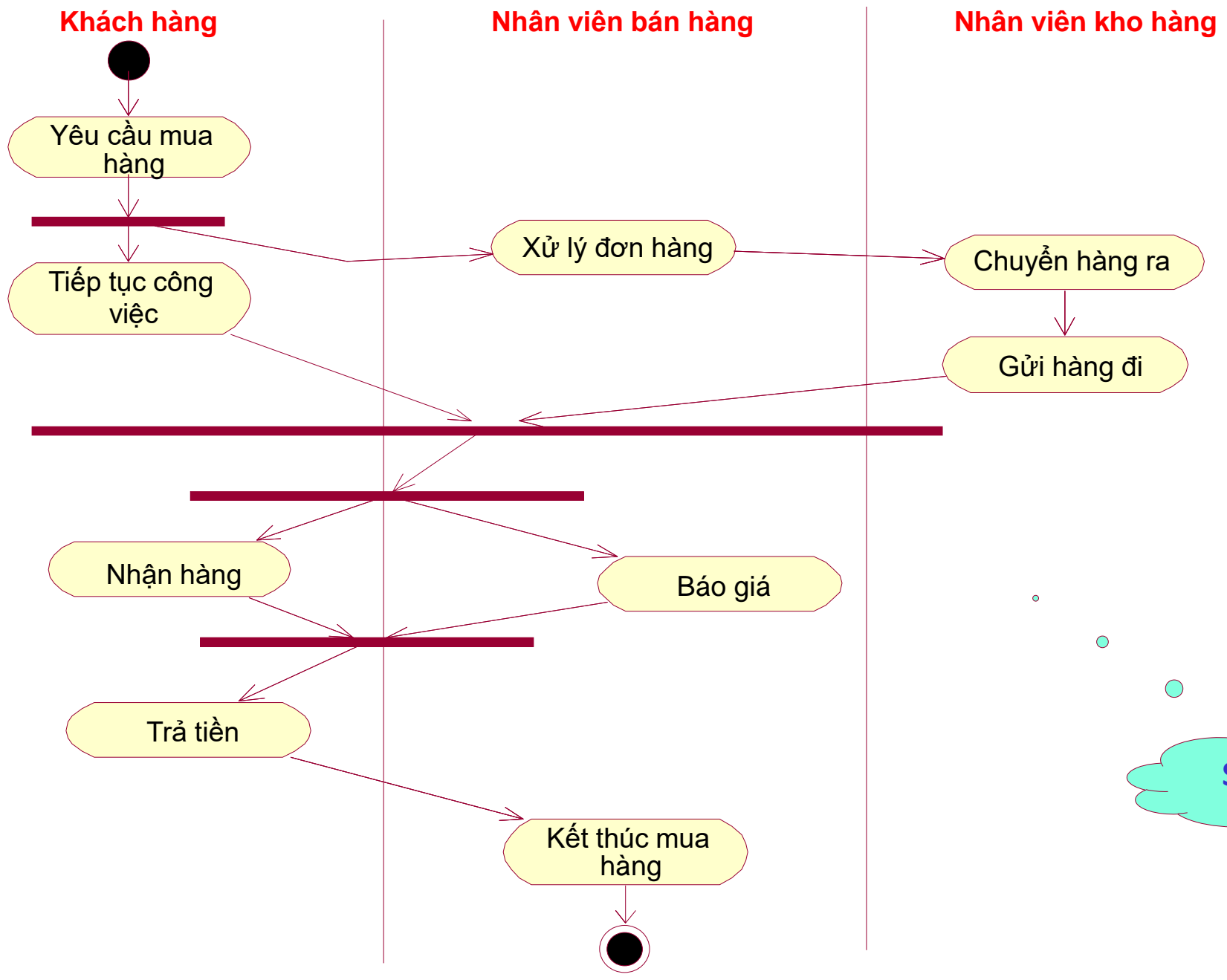
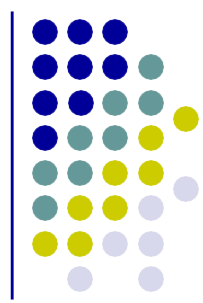


Swimlane (đường phân dòng nghiệp vụ): trong một mô hình phức tạp, người ta nhóm các hoạt động thành các swimlane và bố trí thành các cột, mỗi cột tương ứng với một swimlane.

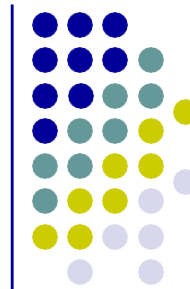
Các hoạt động trên biểu đồ được sắp xếp vào trong các cột tương ứng và cho ta cách nhìn rõ ràng hơn là hành động hoặc trạng thái đang ở dòng nghiệp vụ nào

Dòng nghiệp vụ chính trong use-case "Đăng ký dịch vụ dùng chung xe"





Swimlanes



Bước 4.5: BIỂU ĐỒ TRẠNG THÁI

Biểu đồ trạng thái



Mô tả chu kỳ tồn tại của đối tượng từ khi nó sinh ra đến khi nó bị phá hủy

Sử dụng để mô hình hóa khía cạnh động của lớp

Biểu đồ bao gồm các thông tin sau

Các trạng thái của đối tượng

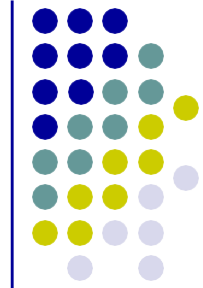
Hành vi của đối tượng

Sự kiện tác động làm thay đổi trạng thái

Thông thường

Xây dựng biểu đồ chuyển trạng thái cho một vài đối tượng của lớp có nhiều hành vi động trong dự án

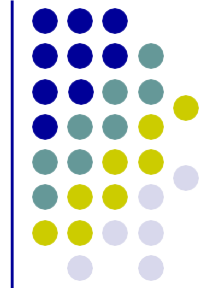
Không phải mọi dự án sử dụng biểu đồ loại này



Biểu đồ trạng thái chủ yếu được dùng để **mô tả hành vi của các lớp**

Khi nào sử dụng biểu đồ trạng thái (when to use State Diagram)?

- Biểu đồ trạng thái dùng để biểu diễn sự thay đổi **trạng thái của một lớp** tương ứng với các thông điệp mà lớp đó gửi đi hoặc nhận được.
- Biểu đồ trạng thái được sử dụng để mô tả ứng xử của một đối tượng trải qua nhiều use-case.
- Hãy vẽ biểu đồ trạng thái cho những lớp mà các ứng xử của chúng không dễ hiểu và do đó cần mô tả chi tiết hơn.



Khi nào sử dụng biểu đồ trạng thái (when to use State Diagram)?

- Nếu bạn cần mô tả nhiều đối tượng cùng tham gia vào một use-case thì nên sử dụng biểu đồ tương tác.
- Để diễn tả các ứng xử của nhiều đối tượng qua nhiều use-case thì sử dụng biểu đồ hoạt động

(Biểu đồ trạng thái chủ yếu được dùng để ***mô tả các trạng thái*** và các sự kiện, hoạt động ***trong vòng đời của một đối tượng***)

Trạng thái đối tượng?



Trạng thái đối tượng là kết quả của các hoạt động trước đó của đối tượng

Đối tượng luôn ở trong một trạng thái xác định tại một thời điểm

Trạng thái được xác định bởi giá trị của thuộc tính và liên kết với đối tượng khác

Thí dụ

Con người cụ thể của lớp Person có các trạng thái: Người lao động, Thất nghiệp, Về hưu

Hóa đơn mua hàng: Đã thanh toán, chưa thanh toán

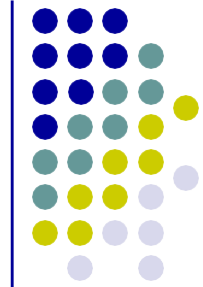
Xe ô tô: Đang chạy, Đang đứng

Thay đổi trạng thái đối tượng

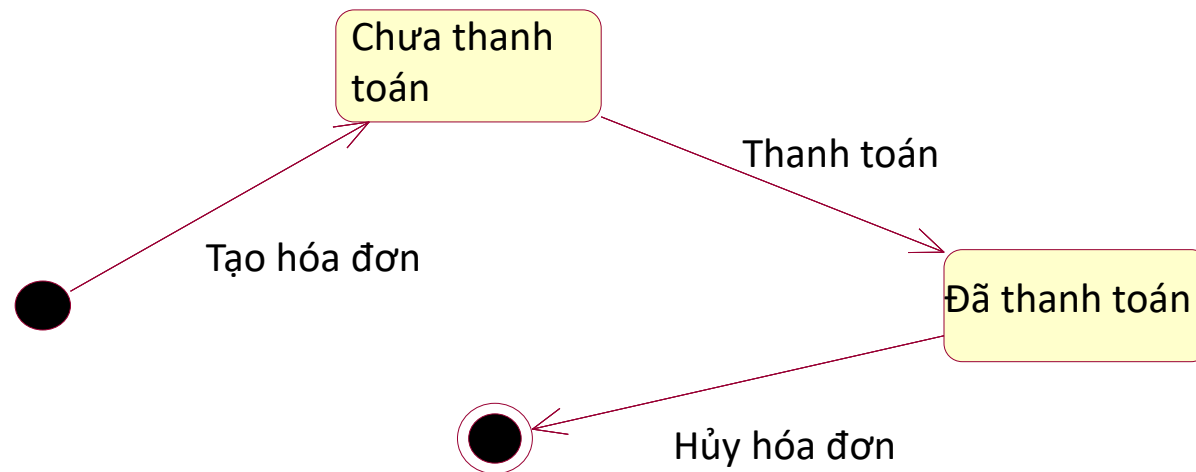
Có sự kiện xảy ra

Thí dụ: ai đó thanh toán hóa đơn hàng

Biểu đồ trạng thái



Thí dụ biểu đồ trạng thái

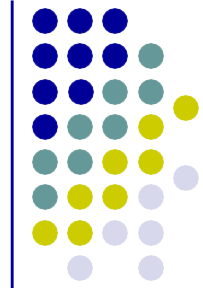


Sử dụng biểu đồ trạng thái để làm gì?

Phân tích viên, người thiết kế và người sử dụng hiểu hành vi đối tượng

Người phát triển hiểu hành vi đối tượng để cài đặt nó

Các phần tử đồ họa



Trạng thái khởi đầu: Khi đối tượng được tạo ra



Trạng thái dừng: Khi đối tượng bị phá hủy



Trạng thái (State)

Hoạt động

Hành động vào

Hành động ra

Chuyển dịch(Transition)

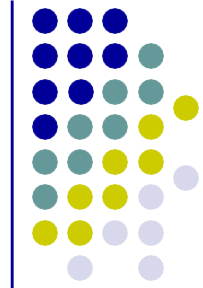
Sự kiện

Điều kiện canh

Hành động

Trạng thái ẩn

Các phần tử đồ họa



Trạng thái (State)

Trạng thái được xác định từ khảo sát thuộc tính lớp và quan hệ giữa các lớp

Ký pháp đồ họa

Tên trạng thái

Khi đối tượng trong trạng thái nào đó nó thực hiện vài hoạt động (Activity)

Phát sinh báo cáo, Thực hiện tính toán và Gửi thông điệp đến đối tượng khác

Có năm loại thông tin có thể gộp trong trạng thái

Hoạt động, Hành động vào, Hành động ra, Sự kiện, Lịch sử trạng thái.

Các phần tử đồ họa

Chuyển dịch(Transition)

Chuyển dịch là chuyển động từ trạng thái này sang trạng thái khác

Chuyển dịch phản thân

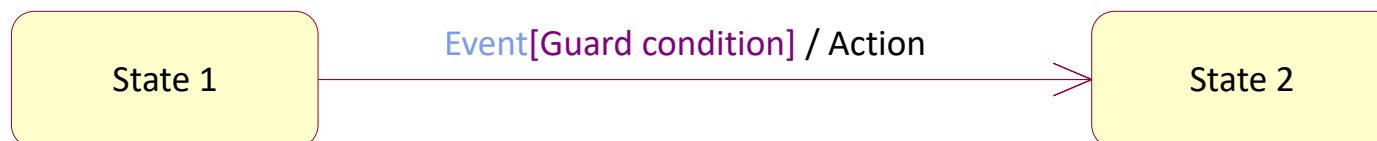
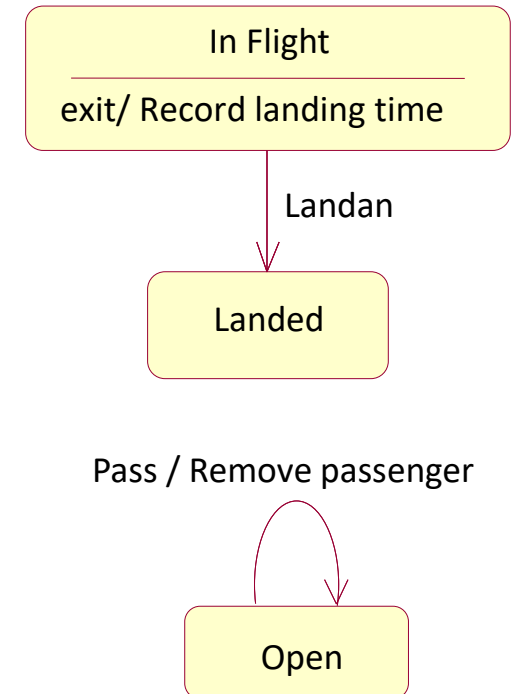
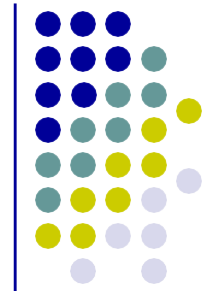
Đặc tả chuyển dịch

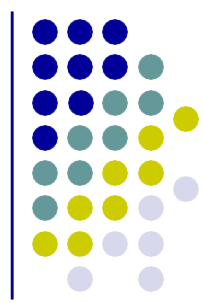
Sự kiện (**Event**): cái gì đó là nguyên nhân chuyển từ trạng thái này sang trạng thái khác

Hầu hết chuyển dịch đều có sự kiện. Sự kiện có thể có đối số, thí dụ, Remove passenger(name)

Điều kiện canh (**Guard**): xác định khi nào sự kiện xảy ra, thí dụ, Trạng thái máy bay từ Open sang Full khi chỗ cuối cùng đã có người mua vé

Hành động (**Action**): hành vi không ngắt được, xảy ra như một phần của chuyển tiếp.





Mô tả quan hệ giữa biểu đồ trạng thái và lớp

Thí dụ Biểu đồ trạng thái của lớp **Digital watch**

