



CHƯƠNG 9

TỔNG QUAN VỀ UML

(Unified Modeling Language)

Giảng viên: Phạm Văn Tiệp

Mục tiêu bài học

- Hiểu khái niệm về UML.
- Nắm vững các loại biểu đồ chính trong UML.
- Ứng dụng UML trong phân tích và thiết kế phần mềm

Tổng quan về UML

- UML (Unified Model Language) là một ngôn ngữ dùng cho phân tích thiết kế hướng đối tượng (OOAD – Object Oriented Analysis and Design)
- Được duy trì và phát triển bởi OMG (Object Management Group), do **Jacobson, Booch, Rumbaugh** sáng lập. Ngoài ra còn có hàng trăm các tập đoàn lớn khác bảo trợ phát triển.
- UML 2.x có 13 loại biểu đồ để thể hiện các khung nhìn khác nhau (View) về hệ thống.
- Các biểu đồ UML cho ta cái nhìn rõ hơn về hệ thống (cả cái nhìn tĩnh và động)

Tổng quan về UML

- Hiện nay UML được sử dụng rất phổ biến trong các dự án phần mềm.
- UML thể hiện phương pháp phân tích hướng đối tượng nên không lệ thuộc ngôn ngữ lập trình.
- Có rất nhiều công cụ phần mềm hỗ trợ phân tích thiết kế dùng UML.
- Nhiều công cụ có thể sinh ra mã từ UML và ngược lại (từ mã thành UML-Reverse Eng)
- UML không phải là ngôn ngữ lập trình!
- Phiên bản mới nhất của UML là 2.5.1 (www.omg.org)

UML dùng để làm gì ?

UML là một ngôn ngữ dùng để:

1. Trực quan hóa (Visualizing)
2. Đặc tả (Specifying)
3. Xây dựng (Constructing)
4. Viết tài liệu (Documenting)

Trực quan hóa - Visualizing

- Dùng tập các ký hiệu đồ họa phong phú để biểu diễn hệ thống đang được nghiên cứu.
- Hệ thống ký hiệu đều có ngữ nghĩa chặt chẽ, có thể hiểu bởi nhiều công cụ khác nhau.
- Giúp cho các nhà thiết kế, nhà lập trình khác biệt về ngôn ngữ đều có thể hiểu được.

UML là ngôn ngữ cho đặc tả - specifying

- UML giúp xây dựng các mô hình chính xác, đầy đủ và không nhập nhằng.
- Tất cả các công đoạn từ phân tích, thiết kế cho đến triển khai đều có các biểu đồ UML biểu diễn.
- Use case (dùng cho phân tích); Class, Sequence, Activity... (cho thiết kế); Component, Deployment (cho triển khai).

Xây dựng - Constructing

- Các mô hình UML có thể kết nối với nhiều ngôn ngữ lập trình. Tức là có thể ánh xạ các mô hình UML về một NNLT như C++, Java, C#...
- Việc chuyển các mô hình trong UML thành Code trong ngôn ngữ lập trình → Forward engineering
- Việc chuyển ngược trở lại code trong một ngôn ngữ lập trình thành UML → Reverse Engineering.
- Cần công cụ để chuyển đổi “xuôi” & “ngược”

UML là ngôn ngữ giúp viết tài liệu

- Giúp xây dựng tài liệu đặc tả - requirements
- Tài liệu kiến trúc (architecture)
- Tài liệu thiết kế
- Source code
- Tài liệu để kiểm thử - Test
- Tài liệu mẫu - Prototype
- Tài liệu triển khai – Deployment
-

Một số Tool hỗ trợ UML

- Rational Rose
- Visual Paradigm
- Microsoft Visio
- Power designer
- Visual Case
- Pacestar UML Diagrammer
- Astah
- **LucidChart**

Các loại biểu đồ chính trong UML

UML gồm 2 loại biểu đồ chính:

❖ **Biểu đồ cấu trúc (Structural Diagrams):**

- Biểu đồ lớp (Class diagram)
- Biểu đồ đối tượng (Object diagram)
- Biểu đồ gói (Package diagram)
- Biểu đồ thành phần (Component diagram)
- Biểu đồ triển khai (Deployment diagram)

❖ **Biểu đồ hành vi (Behavioral Diagrams):**

- Biểu đồ ca sử dụng (Use-case diagram)
- Biểu đồ hoạt động (Activity diagram)
- Biểu đồ máy trạng thái (State machine diagram)
- Biểu đồ trình tự (Sequence diagram)
- Biểu đồ giao tiếp (Communication diagram)

Biểu đồ lớp (Class Diagram)

- **Mục đích:** Biểu diễn cấu trúc tĩnh của hệ thống qua các lớp và mối quan hệ giữa chúng.
- **Thành phần chính:** Lớp (Class), thuộc tính (Attribute), phương thức (Method), quan hệ kế thừa (Inheritance), quan hệ kết hợp (Association).

Biểu đồ lớp (Class Diagram)

❖ Biểu diễn lớp:

Tên lớp
DS thuộc tính
DS phương thức

Tên lớp

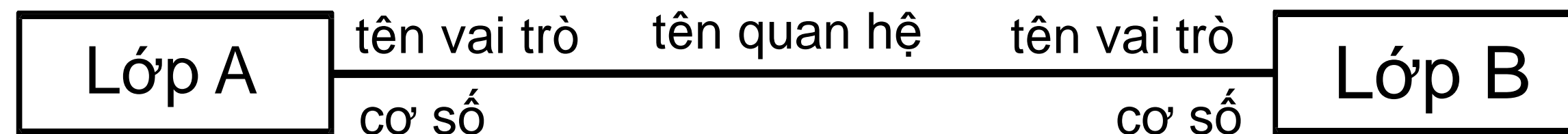
❖ Biểu diễn chỉ định truy cập:

- public: +
- protected: #
- không xác định: ~
- private: -

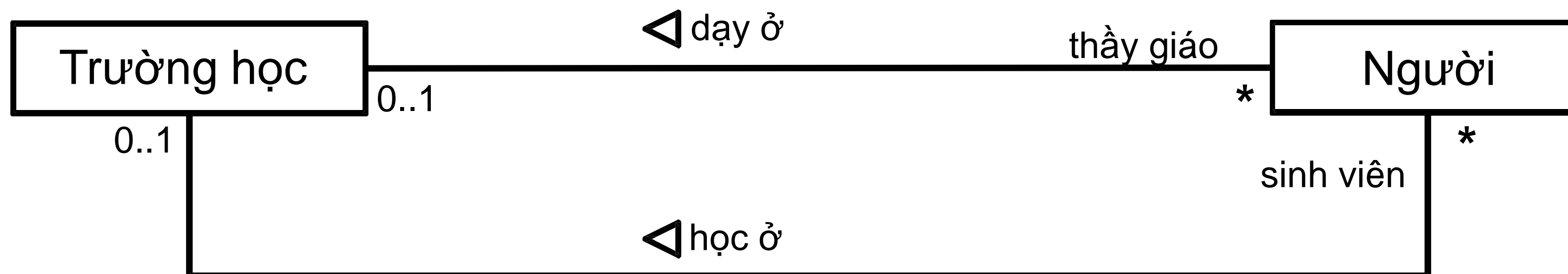
❖ Biểu diễn giao diện:

<<interface>> Tên giao diện

Biểu diễn quan hệ giữa các lớp

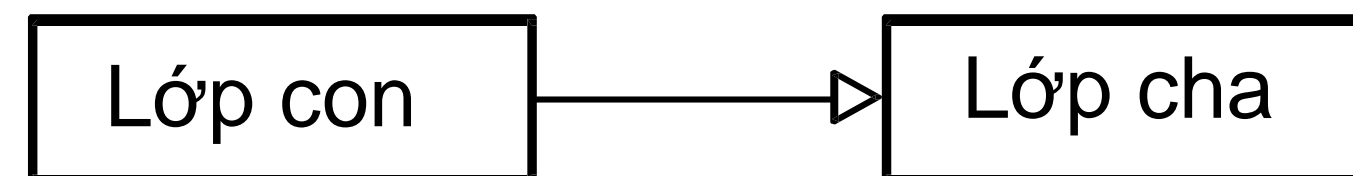


- **Tên liên kết:** chỉ ra mối quan hệ giữa các lớp trên liên kết
- **Tên vai trò:** chỉ ra vai trò của mỗi lớp trên quan hệ đó
- **Cơ số:** chỉ ra số lượng các đối tượng của mỗi lớp tham gia vào liên kết:
 - ✓ Một số nguyên dương (1, 2, 3..., *)
 - ✓ Một dải giá trị: 0..1, 2..4, 1..*
 - ✓ Không ghi: mặc định là 1

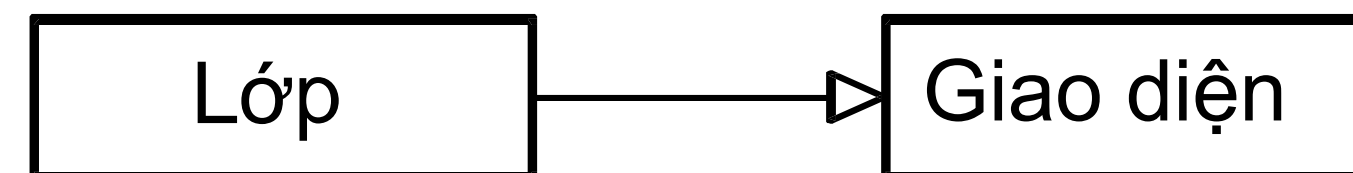


Biểu đồ lớp – Quan hệ

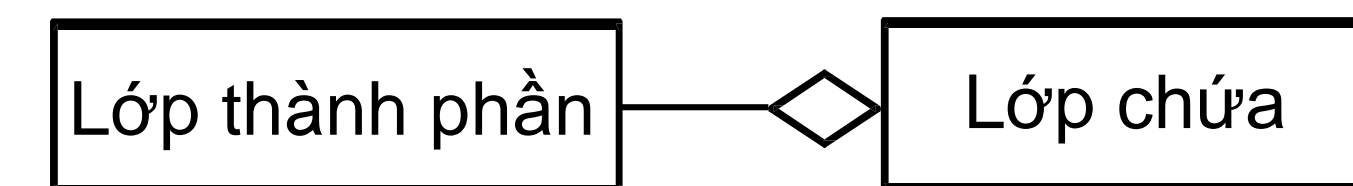
- **Quan hệ kế thừa (Inheritance):**



- **Quan hệ triển khai (Realization):**

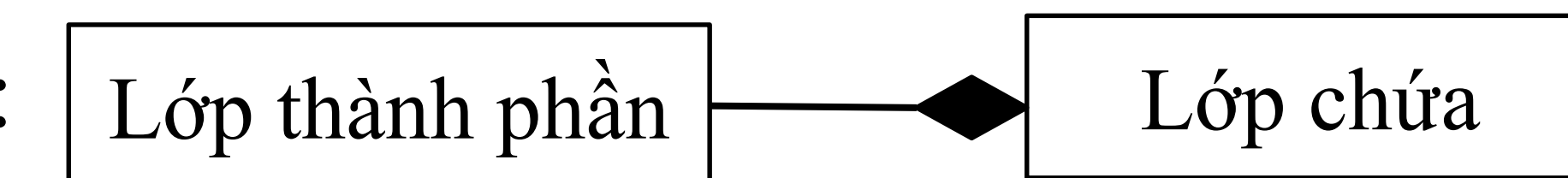


- **Quan hệ kết tập (Aggregation):**



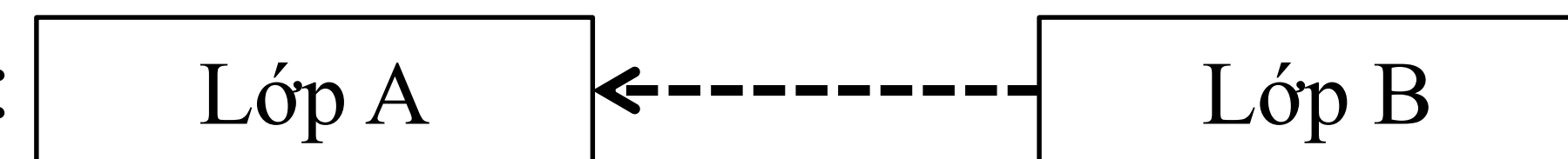
✓ Đối tượng lớp thành phần có thể tồn tại độc lập với lớp chứa

- **Quan hệ cấu thành (Composition):**



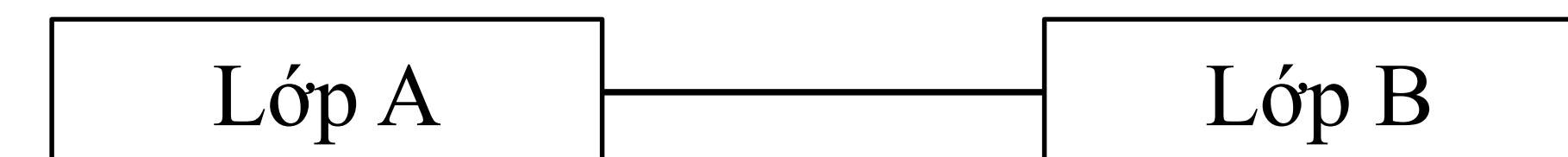
✓ Đối tượng lớp thành phần không thể tồn tại ngoài một đối tượng lớp chứa

- **Quan hệ phụ thuộc (Dependency):**

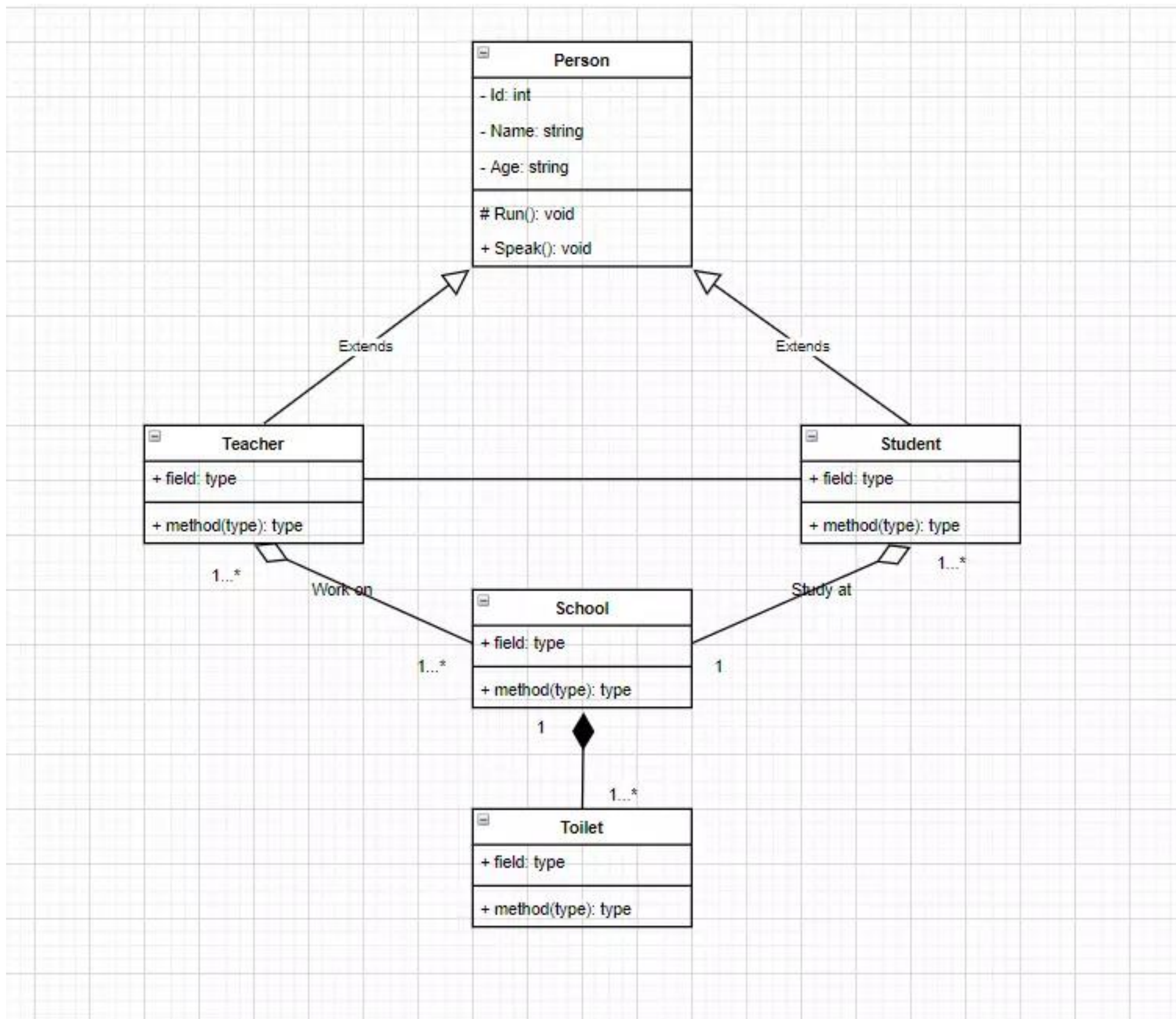


✓ Sự thay đổi ở đối tượng lớp A có thể dẫn đến sự thay đổi trên đối tượng lớp B

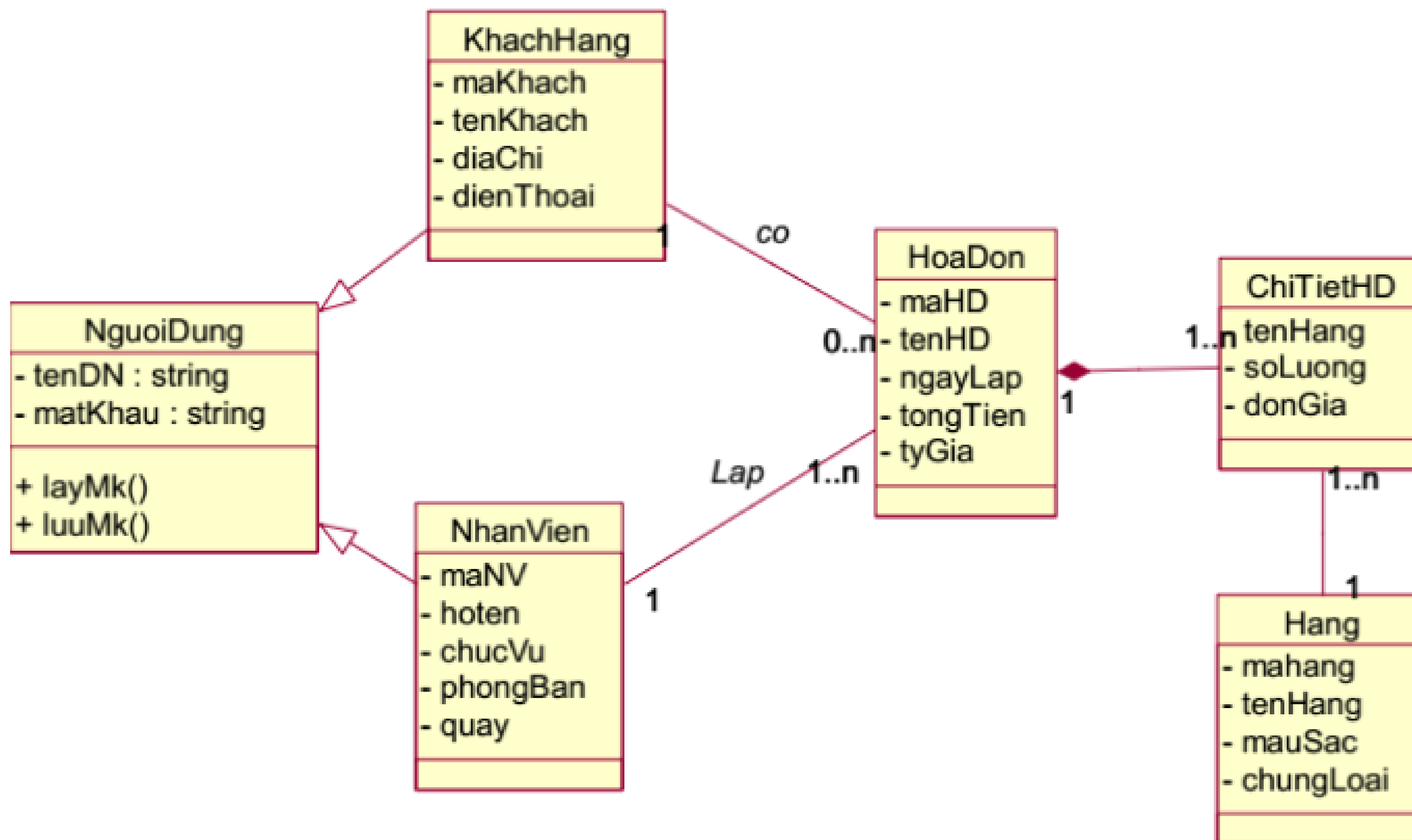
- **Quan hệ kết hợp (Association):**



Biểu đồ lớp – Ví dụ



Biểu đồ lớp hệ thống quản lý bán hàng



Biểu đồ ca sử dụng (Use Case Diagram)

- **Mục đích:** Mô tả các chức năng của hệ thống từ góc nhìn người dùng.
- **Thành phần chính:** Tác nhân (Actor), ca sử dụng (Use Case), mối quan hệ (Relationship).

Biểu đồ Ca sử dụng (Use-case diagram)

- **Biểu đồ Use Case** được sử dụng để mô tả các chức năng chính mà hệ thống phần mềm cung cấp, từ góc nhìn của người dùng (hoặc tác nhân bên ngoài).
- Nó thể hiện sự tương tác giữa các tác nhân và hệ thống thông qua các **ca sử dụng (use cases)**, là các hành vi hoặc chức năng cụ thể mà hệ thống thực hiện.

Thành phần chính của biểu đồ Use Case

❖ Tác nhân (Actor):

- Đại diện cho người hoặc hệ thống bên ngoài tương tác với hệ thống.
- Có thể là tác nhân chính (Primary Actor) hoặc tác nhân phụ (Secondary Actor).

❖ Ca sử dụng (Use Case): Là các chức năng hoặc nhiệm vụ mà hệ thống thực hiện cho tác nhân.

❖ Hệ thống (System): là phần được mô hình hóa để thực hiện các ca sử dụng. Được biểu diễn bằng một hình chữ nhật bao quanh các ca sử dụng và thể hiện giới hạn của hệ thống phần mềm.

❖ Mối quan hệ (Relationships): Liên kết (Association), Bao gồm (Include), Mở rộng (Extend), Kế thừa (Generalization).

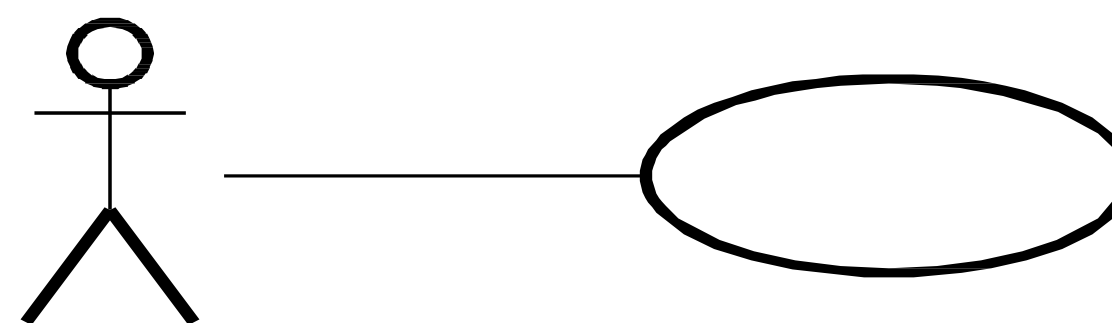
Biểu đồ Ca sử dụng

- Đặc điểm của ca sử dụng:
 - Phải liên kết với một hoặc một số tác nhân
 - Phải dẫn tới một kết quả cụ thể
 - Phải là tập hợp của nhiều chuỗi hành động
- Đặc tả ca sử dụng:
 - Sử dụng biểu đồ Ca sử dụng
 - Sử dụng ngôn ngữ tự nhiên:
 - Mã và tên ca sử dụng
 - Tác nhân
 - Điều kiện đầu vào
 - Luồng chính
 - Luồng phụ (xử lý ngoại lệ)

Các liên kết trong ca sử dụng

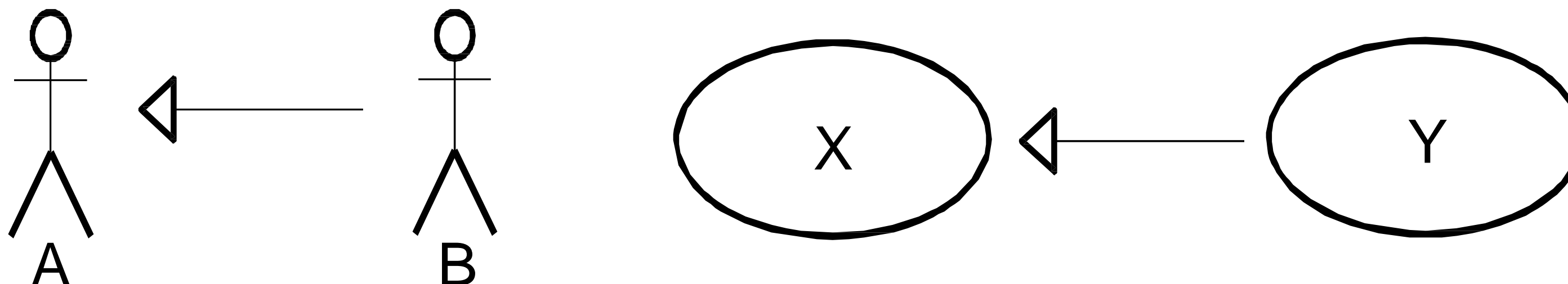
❖ Liên kết tác nhân và ca sử dụng: _____

- Tác nhân và ca sử dụng có trao đổi dữ liệu với nhau
- Nếu trao đổi là 1 chiều: _____→



❖ Khái quát hóa: _____▷

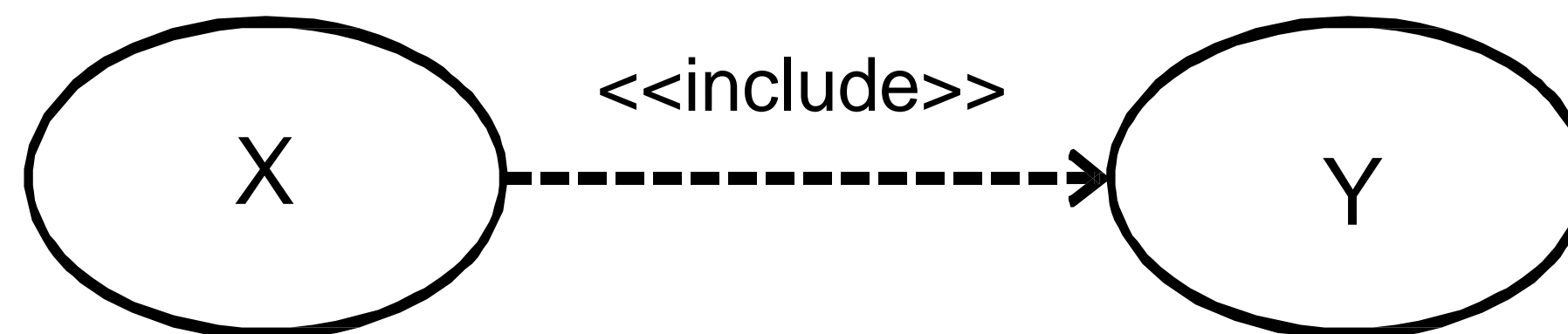
- Đối tác A là khái quát hóa của B nếu B kế thừa mọi đặc điểm của A
- Ca sử dụng X là khái quát hóa của Y nếu Y kế thừa mọi đặc điểm của X



Các liên kết trong ca sử dụng

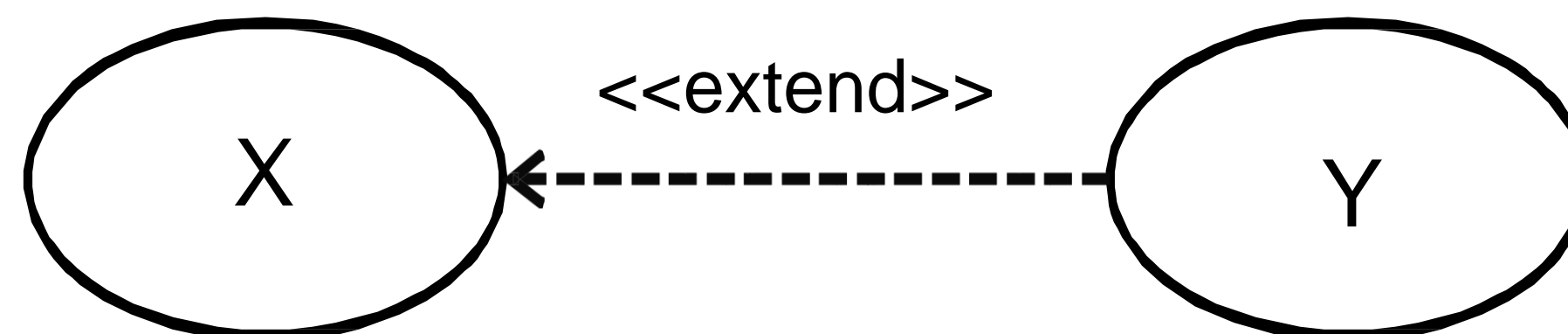
- **Bao hàm** (include): Ca sử dụng X bao hàm Y nếu trong quá trình thực hiện X phải thực hiện đầy đủ Y

✓ X: ca sử dụng cơ sở



- **Mở rộng** (extend): Ca sử dụng X thêm vào các bước thực hiện và kết quả của ca sử dụng Y trong điều kiện nào đó

✓ X: ca sử dụng cơ sở



Lưu ý: Ca sử dụng không chỉ ra trình tự các bước thực hiện

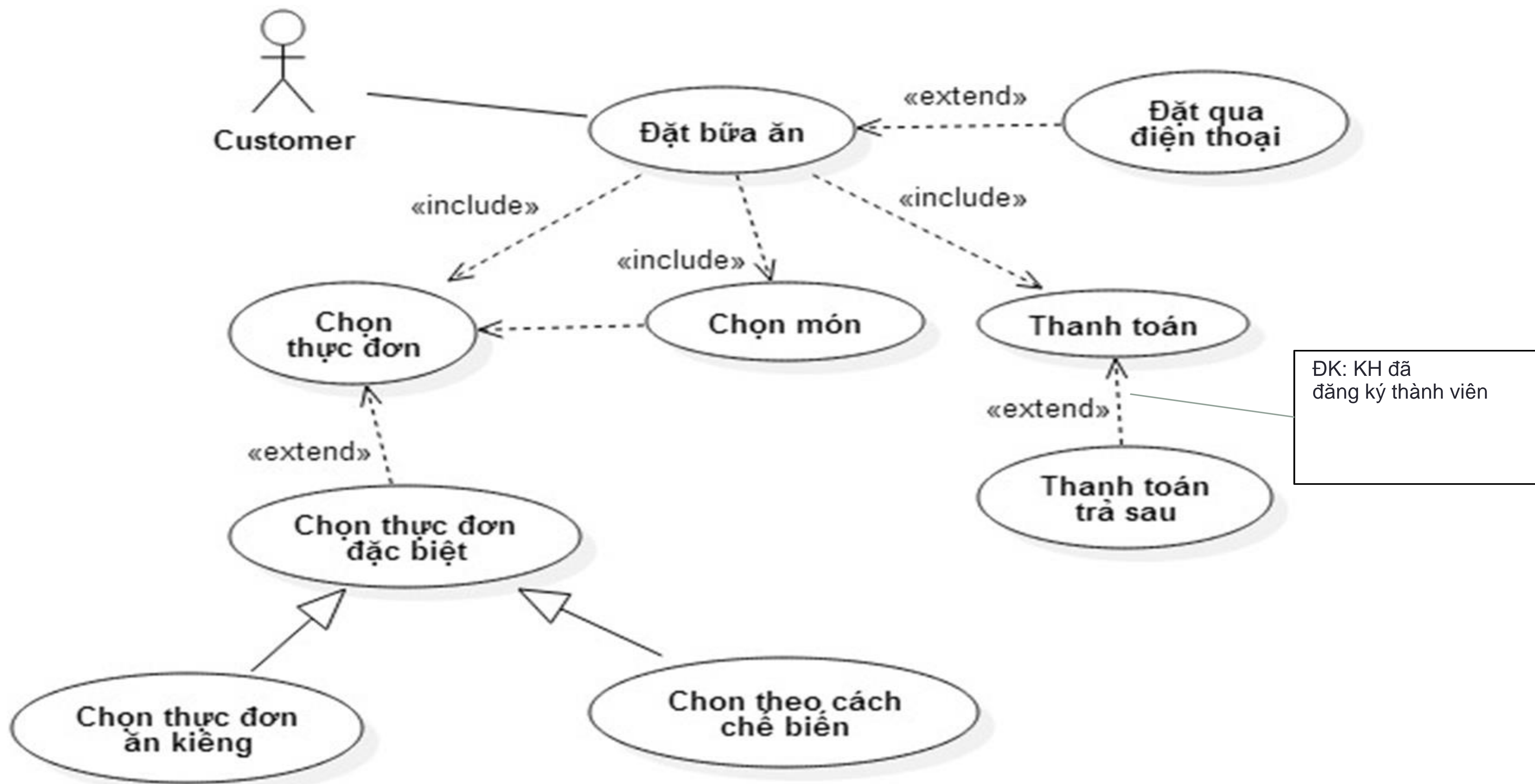
Các liên kết trong ca sử dụng

- **Phụ thuộc (dependence):** Ca sử dụng Y phụ thuộc X khi quá trình thực hiện Y phụ thuộc vào kết quả đã thực hiện X



- **Lưu ý:** Biểu đồ ca sử dụng không chỉ ra trình tự các bước thực hiện. Nội dung này sẽ được đặc tả bằng ngôn ngữ tự nhiên

Ví dụ - Ca sử dụng đặt bữa tại nhà hàng



Ví dụ - Đặc tả use-case

UC-01	Khách hàng đặt bữa
Đối tác chính	Khách hàng
Điều kiện	Không có
Luồng chính	<ol style="list-style-type: none">1. Khách hàng cung cấp các thông tin: Họ tên, Số điện thoại, Thời gian, Số người ăn...2. Hệ thống cung cấp các thực đơn của nhà hàng3. Khách hàng chọn thực đơn. Nếu cần, khách hàng có thể chọn các thực đơn đặc biệt dành cho người ăn kiêng, hoặc theo các chế biến (Nướng, Lẩu...)4. Hệ thống cung cấp các món ăn theo thực đơn khách đã chọn5. Khách hàng hoàn tất việc đặt món6. Hệ thống cung cấp hóa đơn đặt hàng7. Khách hàng xác nhận và thanh toán hóa đơn. Nếu khách hàng đã đăng ký thành viên trên hệ thống có thể chọn tùy chọn thanh toán trả sau8....
Luồng phụ	Nếu khách hàng chưa đăng nhập khi chọn thanh toán trả sau, yêu cầu đăng nhập.

Biểu đồ hoạt động

- ❖ Biểu đồ hoạt động mô tả luồng xử lý của hệ thống khi thực hiện một chức năng nào đó
- ❖ Hoạt động: một công việc trong luồng xử lý
- ❖ **Các ký pháp:**
 - Hoạt động: Tên hoạt động
 - Luồng điều khiển: →
 - Chuyển từ hoạt động này sang hoạt động khác: →
 - Nút khởi tạo:
 - Nút kết thúc:

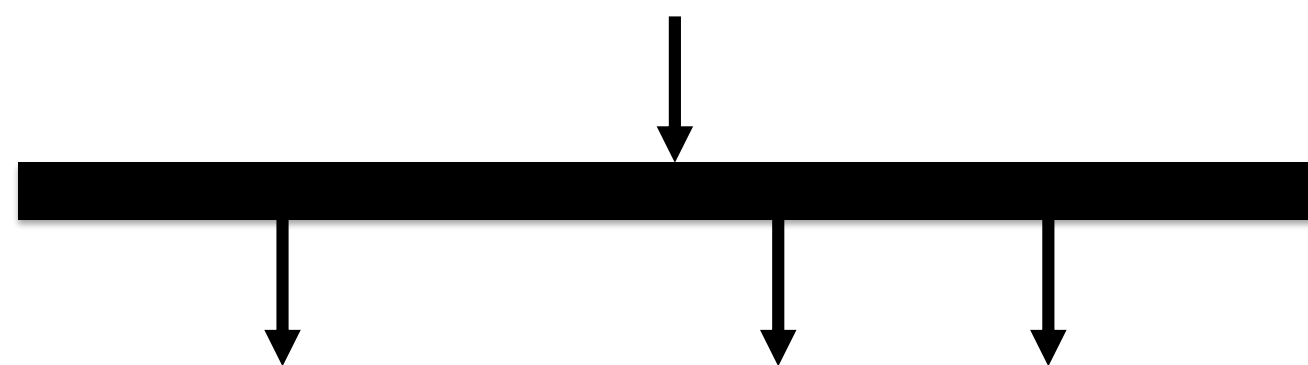
Các ký pháp

❖ Rẽ nhánh:

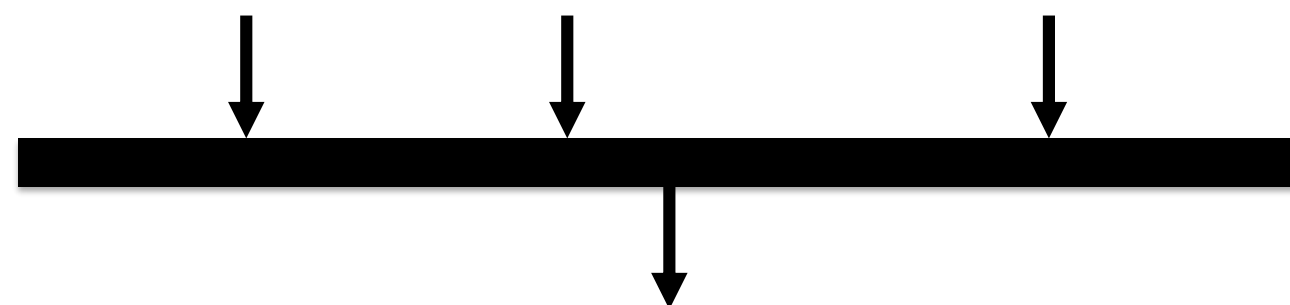
- Quyết định: một luồng vào, nhiều luồng ra, các luồng ra phải loại trừ nhau
- Hòa nhập: nhiều luồng vào, một luồng ra, nếu có một luồng vào nào đó xuất hiện, điểm hòa nhập sẽ được vượt qua

❖ Đồng bộ hóa: biểu diễn các luồng thực hiện song song

- Nút chạc (fork): mở các nhánh song song



- Nút chụm (join): đóng các nhánh song song

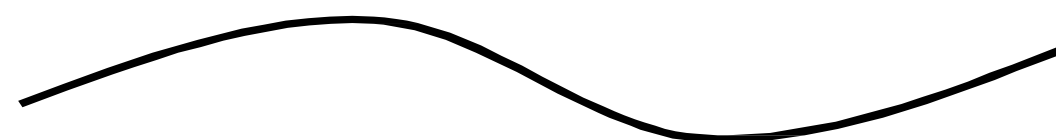


Các ký pháp

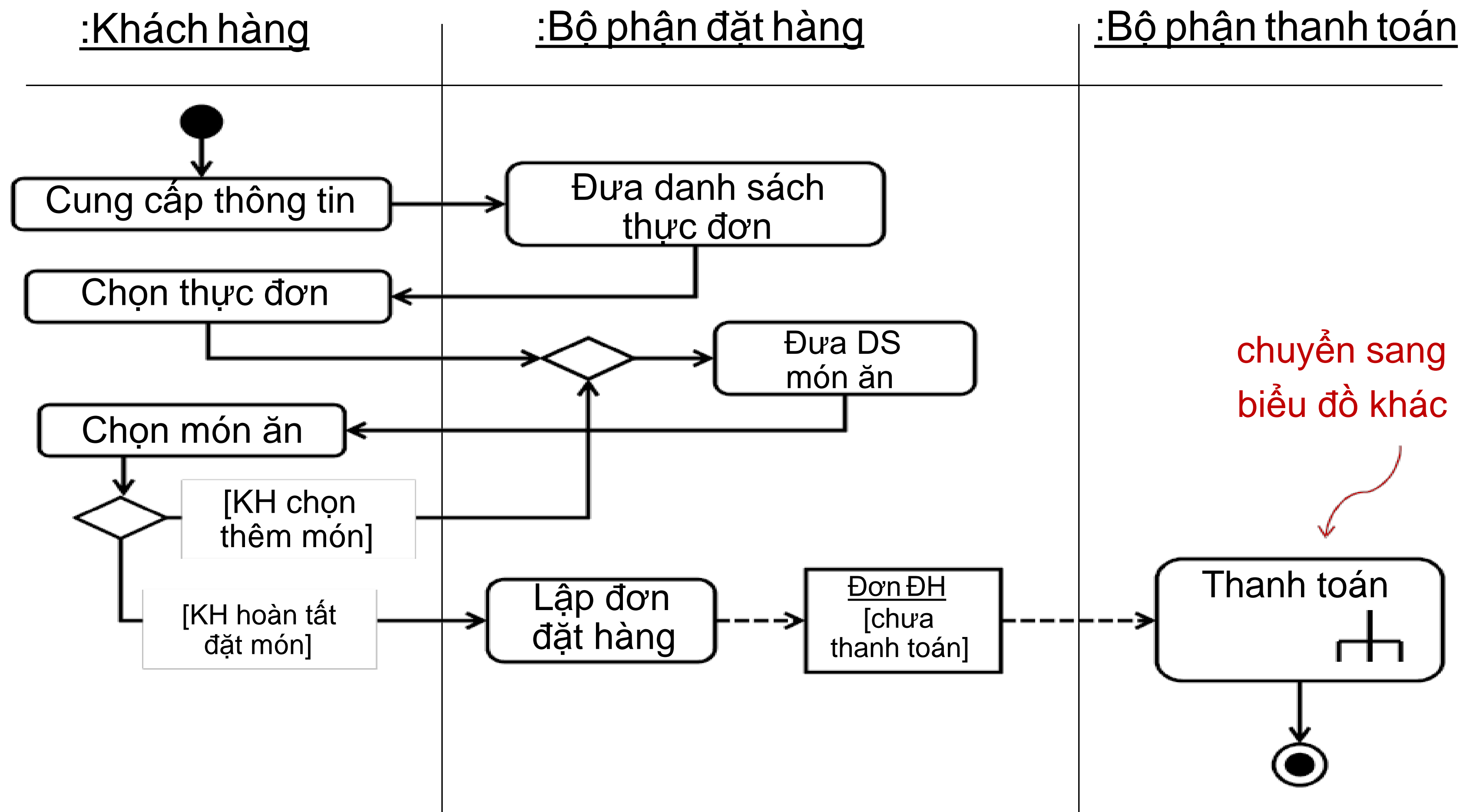
- Điều kiện canh giữ: [Điều kiện]
 - ✓ Luồng chỉ được phép đi qua nếu điều kiện xảy ra
- Phân tuyến: sử dụng các đường swim-line để phân định hoạt động cho một (một nhóm) đối tượng thực hiện

<u>:ĐỐI TƯỢNG 1</u>	<u>:ĐỐI TƯỢNG 2</u>	<u>:ĐỐI TƯỢNG 3</u>


- Phân vùng:





Ví dụ



Biểu đồ tuần tự (Sequence diagram)

- ❖ **Biểu đồ tuần tự:** biểu diễn trình tự thực hiện theo thời gian của các thông điệp trao đổi giữa các đối tượng
- ❖ Thông điệp gọi: 

 - Thông điệp kiểm được (found message): không rõ nguồn 
 - Thông điệp mất hút (lost message): không rõ đích 

- ❖ Thông điệp được lặp nhiều lần: *
- ❖ Biểu đồ trình tự biểu diễn theo 2 chiều:
 - **Chiều ngang:** bố trí các đối tượng, thứ tự các đối tượng không quan trọng, song đối tượng khởi tạo quá trình nên nằm bên trái nhất
 - **Chiều dọc:** trục thời gian hướng xuống dưới, biểu diễn đường đời của đối tượng. Khi đối tượng bị hủy, kết thúc đường đời bằng dấu gạch chéo (X)

Biểu đồ tuần tự - Ví dụ

