

## Chương 7 XÂY DỰNG SƠ ĐỒ ĐỐI TƯỢNG HỆ THỐNG

### **Mục tiêu:**

- Các khái niệm về sự phân loại
- Tìm các lớp đối tượng với các phương pháp: cụm danh từ, phân loại đối tượng và sử dụng sơ đồ use case
- Xác định liên kết giữa các lớp
- Xác định thuộc tính và phương thức của lớp

### **Giới thiệu**

Phân tích hướng đối tượng là một tiến trình mà qua đó chúng ta có thể định dạng được các lớp đóng một vai trò quan trọng nhằm đạt được mục tiêu và yêu cầu hệ thống. Mô hình hoá đối tượng là một tiến trình mà trong đó, các đối tượng trong một hệ thống thực được thể hiện bởi các đối tượng luận lý trong các sơ đồ và trong chương trình. Sự thể hiện trực quan này của các đối tượng và quan hệ giữa chúng cho phép dễ dàng hiểu về đối tượng của hệ thống. Tuy nhiên, việc xác định lớp là một công việc khó nhất bởi vì không có một cấu trúc lớp nào hoàn hảo cũng như không có một cấu trúc nào hoàn toàn đúng.

Trong phần dưới đây sẽ trình bày về cách để phát triển các mô hình đối tượng bằng cách xây dựng các sơ đồ lớp mô tả việc phân loại đối tượng hệ thống. Trước tiên, chúng ta sẽ ôn lại các khái niệm cơ bản của sơ đồ lớp. Sau đó, chúng ta sẽ được giới thiệu xây dựng sơ đồ lớp thông qua việc giới thiệu lần lượt về các cách tiếp cận để phân loại đối tượng và xác định lớp, cách xác định liên kết giữa các lớp cũng như thuộc tính và phương thức của lớp.

### **Sơ đồ lớp (Class diagram)**

#### **Các khái niệm**

##### ***Đối tượng***

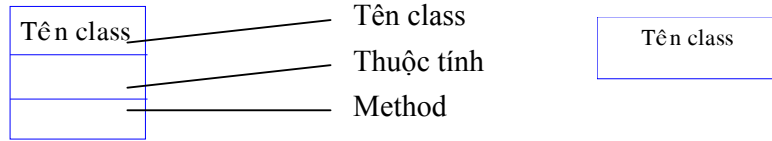
Trong tiếp cận hướng đối tượng, chúng ta mô hình hoá hệ thống bằng các đối tượng, nghĩa là nhìn hệ thống như là một đối tượng. Do đó, trước khi tiếp cận để mô hình hoá hệ thống. Chúng ta cần phải hiểu như thế nào là một đối tượng (object). Có nhiều nguồn mô tả hoặc định nghĩa về đối tượng, tuy nhiên trong tài liệu này chúng ta có thể tổng hợp lại như sau: một đối tượng là một thực thể có một vai trò xác định rõ ràng trong lãnh vực ứng dụng, có trạng thái, hành vi và định danh. Một đối tượng là một khái niệm, một sự trừu tượng hoá hoặc một sự vật có ý nghĩa trong phạm vi ngữ cảnh của hệ thống.

Đối tượng được có thể là một thực thể hữu hình, trực quan (như là: con người, vị trí, sự vật,...); có thể là một khái niệm, sự kiện (ví dụ: bộ phận, đăng ký, ...); có thể là một khái niệm trong tiến trình thiết kế (như là: User interface, Controller, Scheduler,...)

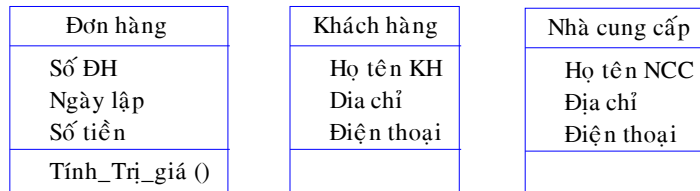
##### ***Lớp (class)***

Là một tập hợp các đối tượng chia sẻ chung một cấu trúc và hành vi (cùng thuộc tính, hoạt động, mối quan hệ và ngữ nghĩa). Cấu trúc được mô tả bởi các thuộc tính và các mối quan hệ, còn hành vi được mô tả bởi các hoạt động. Một lớp là một sự trừu tượng hoá của các đối tượng thể giới thực, và các đối tượng tồn tại trong thế giới thực được xem như là các thể hiện của lớp.

*Ký hiệu:* lớp được trình bày gồm 3 phần: tên lớp, danh sách các thuộc tính (attribute), danh sách các hoạt động (operation). Trong đó, phần thuộc tính và phần hoạt động có thể bị che dấu đi trong mức độ trình bày tổng quan.



Ví dụ: biểu diễn tập hợp các đơn hàng NGK, khách hàng mua NGK, nhà cung cấp NGK,... cùng chia sẻ chung thuộc tính, hoạt động, mối quan hệ và ngữ nghĩa thành các lớp:



### Mối kết hợp (association)

Mối kết hợp nhị phân: là quan hệ ngữ nghĩa được thiết lập giữa hai hay nhiều lớp, biểu diễn bởi những thành phần sau:

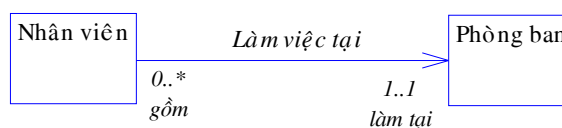
- Tên quan hệ: thường là cụm động từ phản ánh mục đích của mối kết hợp
- Vai trò quan hệ (role): là một phần của mối kết hợp dùng để mô tả ngữ nghĩa tham gia của một lớp vào mối kết hợp đó (không phải một phần của lớp). Mỗi quan hệ có thể có 2 vai trò (quan hệ nhị phân) hoặc nhiều hơn (quan hệ đa phân).
  - o Tên vai trò: dùng động từ hoặc danh từ (cụm danh từ) để biểu diễn vai trò của các đối tượng. Trong mối kết hợp *làm việc tại* có hai vai trò, *làm tại* và *gồm* cho biết: nhân viên làm việc tại phòng ban và phòng ban gồm có các nhân viên trực thuộc.
  - o Bản số: là cặp giá trị (mincard, maxcard) xác định khoảng giá trị cho phép một đối tượng của một lớp có thể tham gia bao nhiêu lần vào mối kết hợp với các đối tượng của các lớp khác.

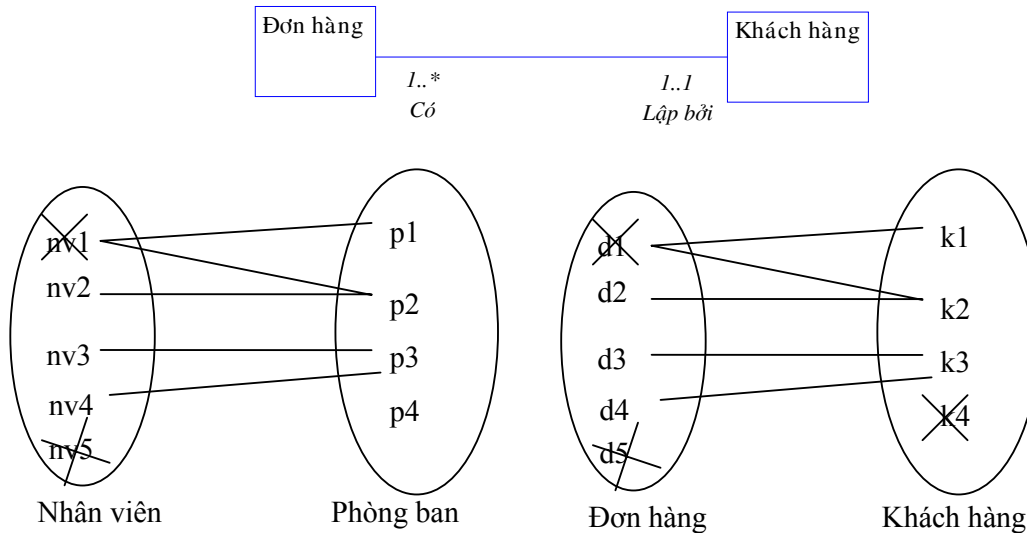
Giá trị mincard: qui định về ràng buộc tối thiểu của một đối tượng tồn tại trong lớp phải tham gia vào mối kết hợp với một số lượng lớn hơn hoặc bằng.

Giá trị maxcard: qui định số lượng tối đa mà một đối tượng của lớp nếu tồn tại trong lớp đó không được tham gia vào mối kết hợp vượt giá trị này.

Bản số mối kết hợp dưới cho biết một nhân viên phải thuộc ít nhất và nhiều nhất (duy nhất) một phòng ban, tuy nhiên mỗi đối tượng phòng ban có thể tồn tại mà không có nhân viên làm việc thuộc phòng. Các mẫu bản số thường là:

0..1, 1..1, 3..5, 0..\*, 1..\*, 2..\*



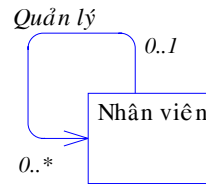


Tổng quát, cho hai lớp C1, C2 và mỗi kết hợp A giữa chúng. Tùy theo giá trị bản số tối thiểu chúng ta có những trường hợp sau:

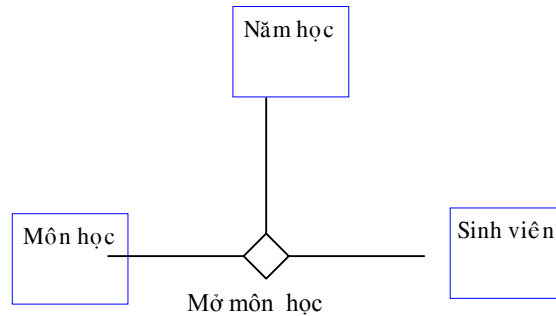
- Nếu mincard (C1,A) = 0 thì chúng ta nói rằng lớp C1 có sự tham gia tùy ý trong mỗi kết hợp bởi vì một đối tượng của lớp C1 có thể không tham gia kết hợp với đối tượng lớp C2 trong mỗi kết hợp A.
- Nếu mincard (C1,A) >0 thì chúng ta nói rằng lớp C1 có sự tham gia bắt buộc vào mỗi kết hợp bởi vì một đối tượng của lớp C1 phải bắt buộc tham gia kết hợp với ít nhất một phần tử của lớp C2 trong mỗi kết hợp A.

Tùy theo giá trị của bản số tối đa mà chúng ta có các trường hợp sau:

- Nếu maxcard(C1,A) = 1 và maxcard(C2,A) = 1 thì ta gọi là mỗi kết hợp một - một (one- to – one)
- Nếu maxcard(C1,A) = 1 và maxcard(C2,A) = n thì ta gọi là mỗi kết hợp một - nhiều (one- to – many)
- Nếu maxcard(C1,A) = n và maxcard(C2,A) = 1 thì ta gọi là mỗi kết hợp nhiều - một (many- to – one)
- Nếu maxcard(C1,A) = n và maxcard(C2,A) = n thì ta gọi là mỗi kết hợp nhiều - nhiều (many- to – many)
- Tính khả điều hướng (navigability): được mô tả bởi một mũi tên chỉ ra hướng truy xuất trong mỗi kết hợp từ một đối tượng của lớp đến một đối tượng của lớp còn lại. Tính khả điều hướng có thể là không có, hoặc chỉ một, hoặc cả hai. Ví dụ trên cho thấy chiều mũi tên trong mỗi kết hợp *làm việc tại* cho biết chúng ta có thể truy cập lớp *phòng ban* từ mỗi kết hợp, tuy nhiên, chúng ta không thể truy xuất tới lớp *nhân viên* từ mỗi kết hợp này. Hoặc trong mỗi kết hợp giữa lớp *Đơn hàng* và *Khách hàng*, hướng truy xuất là có thể cho cả hai lớp (không có chiều mũi tên)
- Mỗi kết hợp phản thân: một mỗi kết hợp có thể được thiết lập từ một lớp đến chính nó. Ví dụ mỗi kết hợp *quản lý* được thiết lập giữa lớp *Nhân viên* tới chính nó cho biết một nhân viên quản lý những nhân viên khác.

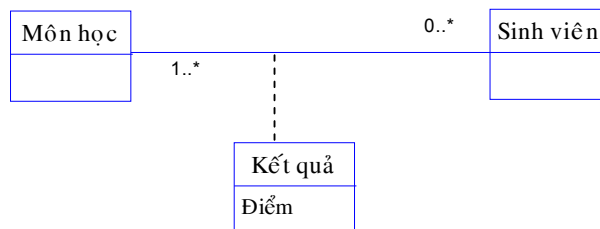


- **Mỗi kết hợp đa phân**: là mỗi kết hợp được thiết lập từ ba lớp trở lên. Ký hiệu mỗi kết hợp đa phân là một hình thoi với hơn ba vai trò nối tới các lớp tham gia. Đây là mỗi kết hợp được thừa hưởng từ cách tiếp cận truyền thống của mô hình thực thể - kết hợp (ER). Tuy nhiên, mỗi kết hợp đa phân không cho phép quan hệ thu nạp, bản số phức tạp. Do đó, trong cách sử dụng chúng ta thường thay thế nó bằng một lớp và đưa về mỗi kết hợp nhị phân.

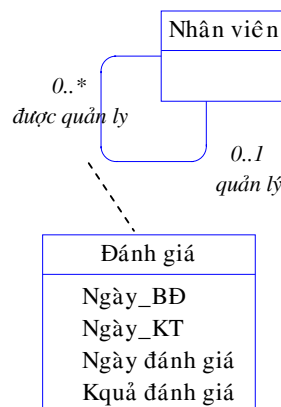


### Lớp kết hợp

Khi một mỗi kết hợp có các đặc trưng (thuộc tính, hoạt động, và các mỗi kết hợp), chúng ta tạo một lớp để chứa các thuộc tính đó và kết nối với mỗi kết hợp, lớp này được gọi là lớp kết hợp. Tên của lớp này chính là tên của mỗi kết hợp, trong trường hợp lớp này có thuộc tính nhưng không có hoạt động hoặc bất kỳ mỗi kết hợp nào khác, thì tên của mỗi kết hợp vẫn duy trì trên mỗi kết hợp và để trống phần tên của lớp này để duy trì tính tự nhiên của nó.

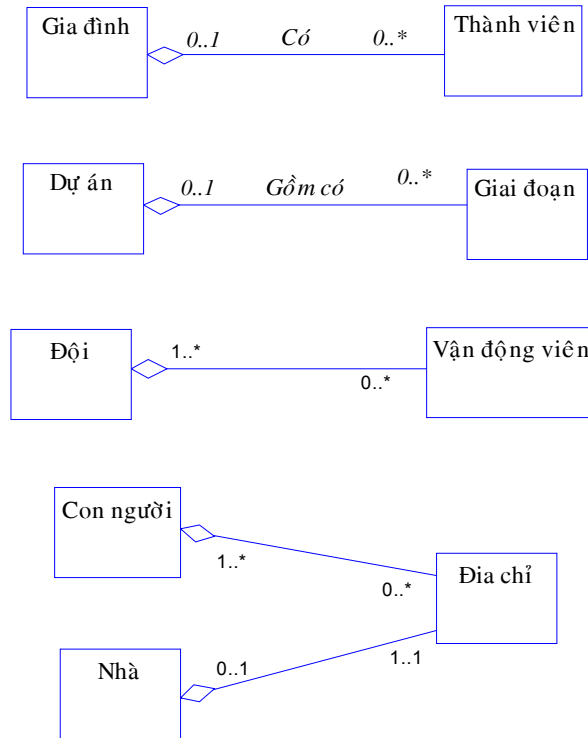


Trường hợp phổ biến nhất của lớp kết hợp là mỗi kết hợp nhiều - nhiều. Ví dụ trên cho thấy, sinh viên tham gia các môn học khác nhau và mỗi lần đăng ký học, sinh viên sẽ có một kết quả được ghi nhận bởi điểm thi. Vậy điểm thi là một thuộc tính được hình thành thông qua việc tham gia học tập của một sinh viên trên một môn học nên nó là thuộc tính của mỗi kết hợp.



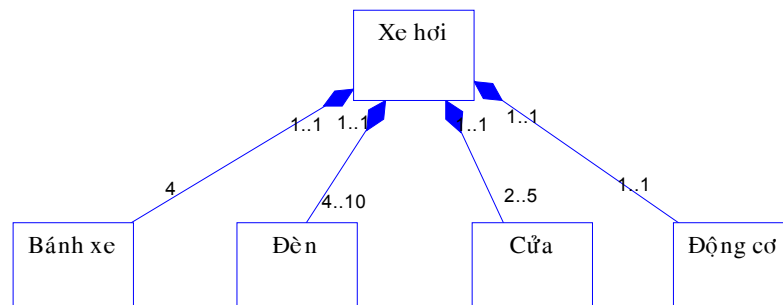
**Quan hệ thu nạp (aggregation) và quan hệ thành phần (composition, a-part-of)**

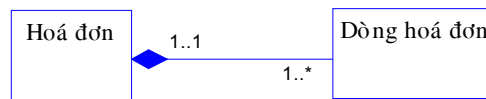
- **Quan hệ thu nạp (aggregation):** mô tả mối quan hệ giữa một đối tượng lớn hơn được tạo ra từ những đối tượng nhỏ hơn. Một loại quan hệ đặc biệt này là quan hệ “có”, nó có nghĩa là một đối tượng tổng thể có những đối tượng thành phần. Ví dụ dưới đây cho thấy, *Gia đình* là một đối tượng tổng thể có những *Thành viên* trong gia đình.



Một đối tượng thành phần cũng có thể tham gia kết hợp với nhiều đối tượng tổng thể khác nhau, trường hợp này gọi là chia sẻ. Ví dụ một vận động viên có quan hệ tới một đội với ý nghĩa là một phần tử của đội, tuy nhiên vận động viên này cũng có thể thành viên của một đội khác, trường hợp này gọi là sự chia sẻ. Do đó, nếu một đội bị hủy bỏ, thì không nhất thiết phải hủy bỏ vận động viên này.

- **Quan hệ thành phần (composition)** là một loại đặc biệt của quan hệ thu nạp, nó có một sự liên hệ mạnh mẽ hơn để trình bày thành phần của một đối tượng phức hợp. Quan hệ thành phần cũng được xem như là quan hệ thành phần - tổng thể (part-whole), và đối tượng tổng hợp sẽ quản lý việc tạo lập và hủy bỏ của những đối tượng thành phần của nó.



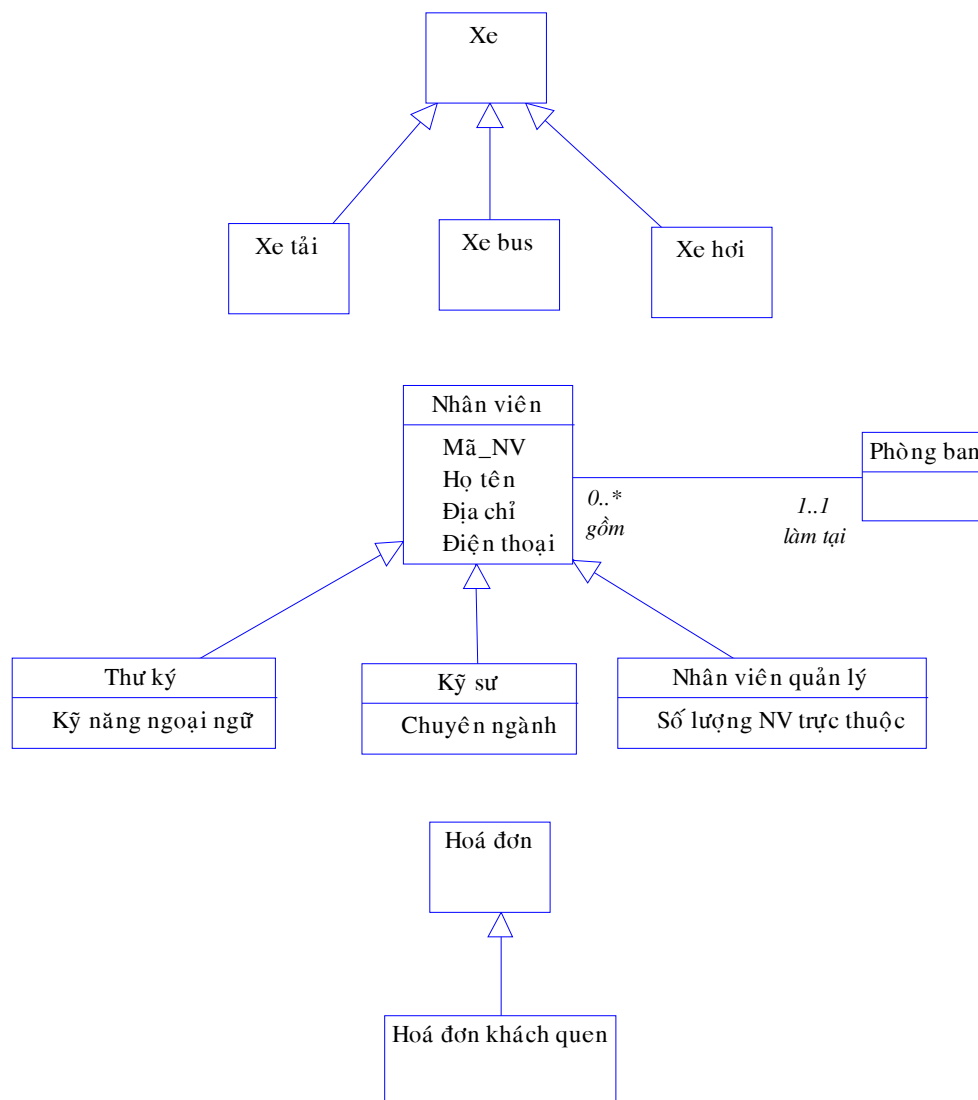


Như vậy, quan hệ thành phần mô tả sự phụ thuộc rất chặt chẽ giữa lớp tổng thể đến lớp thành phần về sự phụ thuộc. Nghĩa là các lớp thành phần là một bộ phận cấu tạo nên lớp tổng thể và thể hiện vật lý của nó là nằm trong lớp tổng thể.

Ví dụ trên cho thấy, một chiếc xe hơi được làm nên bởi những bánh xe, đèn, cửa, động cơ, ... việc tạo thành một chiếc xe hơi là việc lắp ráp các thành phần này. Cũng như hoá đơn chứa các dòng hoá đơn trong đó, một hoá đơn bị huỷ nghĩa là các dòng của hóa đơn đó cũng sẽ bị huỷ theo.

### Quan hệ tổng quát hóa

Là quan hệ được thiết lập giữa một lớp tổng quát hơn đến một lớp chuyên biệt. Quan hệ này dùng để phân loại một tập hợp đối tượng thành những loại xác định hơn mà hệ thống cần làm rõ ngữ nghĩa.



Ví dụ trên đây chỉ ra rằng, tất cả các lớp chuyên biệt Thư ký, Kỹ sư, Nhân viên quản lý đều có thể kế thừa các thuộc tính (Mã\_NV, Họ tên, Địa chỉ, Điện thoại) của lớp tổng quát Nhân viên và mối kết hợp giữa Nhân viên với Phòng ban.

Trong mỗi kết hợp tổng quát hoá, một thể hiện của lớp chuyên biệt cũng là một thể hiện của lớp tổng quát. Ví dụ trên cho thấy một đối tượng Kỹ sư, hoặc Thư ký, hoặc Nhân viên quản lý đều là một đối tượng của lớp Nhân viên. Vì lý do đó, đặc trưng của loại kết hợp này là tính **kế thừa**, một lớp chuyên biệt có thể kế thừa tất cả các đặc trưng (thuộc tính, mỗi kết hợp, hoạt động) của lớp tổng quát.

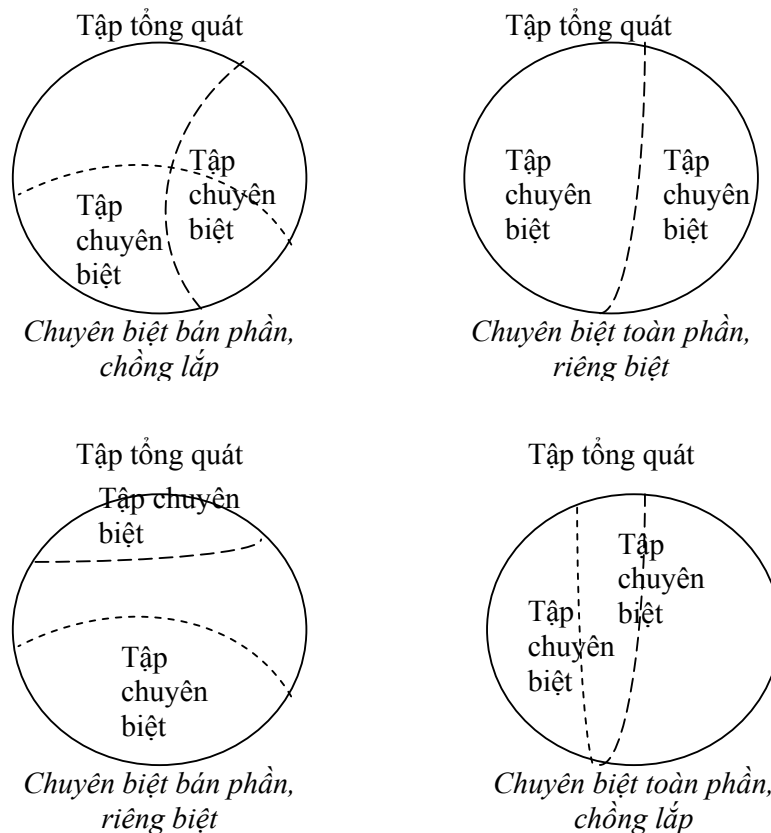
### Sự tương quan của các lớp trong quan hệ tổng quát hoá

*Sự tương quan giữa các lớp chuyên biệt với lớp tổng quát:*

- Tập hợp các đối tượng của tất cả các lớp chuyên biệt phủ toàn bộ tập đối tượng của lớp tổng quát thì gọi là toàn phần (complete).
- Tập hợp các đối tượng của tất cả các lớp chuyên biệt không phủ toàn bộ tập đối tượng của lớp tổng quát thì gọi là bán phần (incomplete).

*Sự tương quan giữa các lớp chuyên biệt:*

- Không tồn tại một đối tượng của lớp tổng quát thuộc hai lớp chuyên biệt trở lên thì gọi là riêng biệt (disjoint).
- Tồn tại một đối tượng của lớp tổng quát thuộc hai lớp chuyên biệt trở lên thì gọi là chồng lấp (overlapping).



Hình 3. Sự tương quan giữa các lớp trong quan hệ tổng quát hoá

Như vậy, trong quan hệ tổng quát hoá, sự tương quan giữa các lớp được biểu diễn qua bốn trường hợp (bán phần - chồng lấp, bán phần - riêng biệt, toàn phần - chồng lấp, toàn phần - riêng biệt). Sự tương quan này phản ánh ràng buộc ngữ nghĩa trong tập hợp các đối tượng của quan hệ: một đối tượng của lớp chuyên biệt này có thể là đối tượng trong lớp chuyên biệt khác hay không? Và một đối tượng trong lớp tổng quát có thể không thuộc một lớp chuyên biệt nào hay không?.

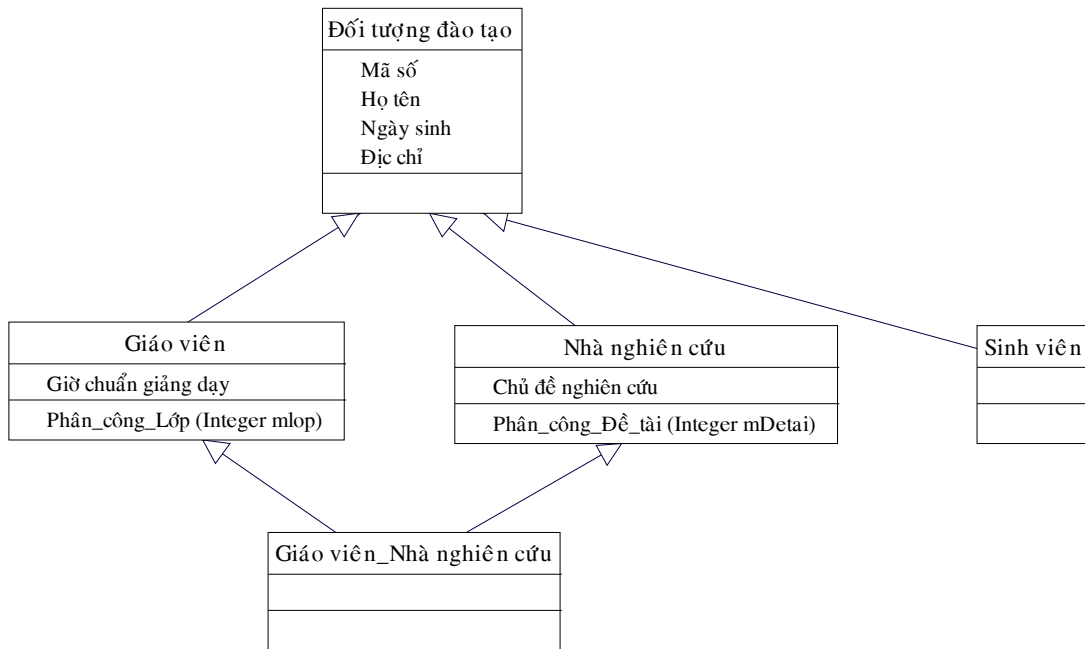
Ví dụ, quan hệ tổng quát hoá giữa Xe – Xe tải, Xe bus, Xe hơi có sự tương quan là bán phần – riêng biệt (incomplete, disjoint). Quan hệ giữa Nhân viên – Thư ký, Kỹ sư, Nhân viên quản lý có sự tương quan là bán phần - chồng lấp (incomplete, overlapping).

### Đa kế thừa

Đa số các trường hợp trong quan hệ tổng quát hoá là đơn kế thừa, nơi mà một lớp là chuyên biệt duy nhất cho một lớp tổng quát. Trong một số trường hợp đặc biệt chúng ta cũng thấy một lớp chuyên biệt có thể kế thừa từ hai hoặc nhiều lớp tổng quát. Trường hợp này gọi là đa kế thừa.

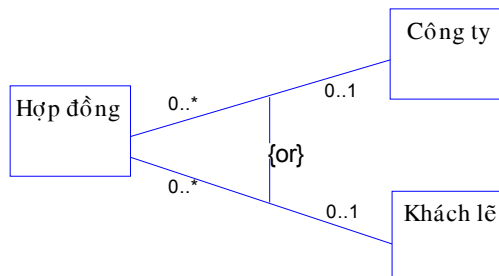
Ví dụ sau cho thấy, lớp Giáo viên\_Nhà nghiên cứu là lớp đa kế thừa từ hai lớp Giáo viên và lớp Nhà nghiên cứu. Lớp này sẽ thừa kế tất cả các đặc trưng như: Giờ chuẩn giảng dạy, Chủ đề nghiên cứu, Phân\_công\_Lớp và Phân\_công\_Đề\_tài của hai lớp trên.

Tuy nhiên, theo lời khuyên của các chuyên gia thì không nên sử dụng đa kế thừa vì tính chất phức tạp của nó. Do đó, đa kế thừa không được đưa vào ngôn ngữ UML gốc và một số ngôn ngữ hướng đối tượng khác.

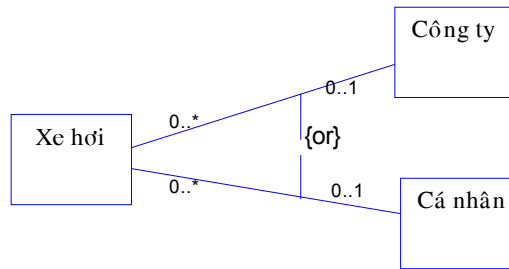


### Quan hệ hoặc (OR)

Là mối quan hệ xác định một tình huống mà trong đó hai (hoặc nhiều) lớp tham gia mỗi kết hợp với một lớp thứ ba với ràng buộc loại trừ. Một thể hiện của lớp thứ ba sẽ tham gia kết hợp loại trừ với các đối tượng của hai lớp kia (hoặc là không kết hợp, hoặc kết hợp chỉ các đối tượng của một trong hai lớp) tại một thời điểm. Ví dụ dưới đây cho thấy, một hợp đồng có thể được lập bởi một công ty hoặc bởi một khách hàng lẻ. Hoặc một chiếc xe hơi thì được sở hữu bởi một cá nhân hoặc bởi một công ty, không sở hữu một lúc bởi cả hai.







Thực chất của mỗi kết hợp OR cũng chính là một ràng buộc ngữ nghĩa giữa các lớp tham gia kết hợp với một lớp thứ ba: sự hiện diện tham gia của đối tượng này sẽ không cho phép sự tham gia của đối tượng kia và ngược lại.

### Thuộc tính (attribute)

Thuộc tính dùng để mô tả đặc trưng của đối tượng, người ta có thể chia thuộc tính thành ba loại sau:

- Thuộc tính đơn trị: là thuộc tính chỉ có một giá trị duy nhất cho một đối tượng, đây là thuộc tính phổ biến nhất. Ví dụ: họ tên, ngày sinh, lương,...
- Thuộc tính đa trị: là thuộc tính có thể có nhiều giá trị cho một đối tượng. Ví dụ: nếu chúng ta muốn lưu nhiều số điện thoại của một nhân viên, chúng ta có thể đặt thuộc tính số điện thoại trong lớp nhân viên là đa trị.
- Thuộc tính tham chiếu.

### Biểu diễn một thuộc tính

Thuộc tính được biểu diễn gồm những thành phần như sau:

<phạm vi> <tên thuộc tính> : <biểu thức kiểu> = <giá trị khởi tạo>

<phạm vi>: nhận một trong những giá trị sau:

+ toàn cục (có thể truy cập bởi tất cả lớp)

# bảo vệ (có thể truy cập bởi lớp và lớp chuyên biệt của nó)

- riêng (chỉ được truy cập bởi lớp)

**Bản số:** là một cặp (số tối thiểu, số tối đa) mà thuộc tính có thể có giá trị.

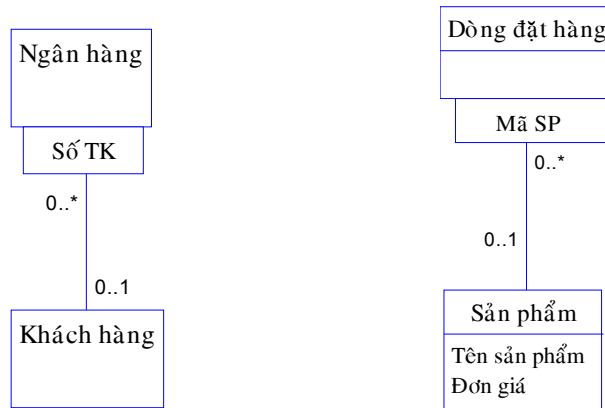
- số tối thiểu = 0 → thuộc tính được gọi là không bắt buộc.
- số tối thiểu = 1 → thuộc tính được gọi là bắt buộc.
- số tối đa = 1 → thuộc tính đơn trị.
- số tối đa > 1 → thuộc tính đa trị

Ví dụ: *Số điện thoại*[0..\*]: string, *Địa chỉ*[0..1]: string,...

Trong giai đoạn phân tích việc xác định thuộc tính thường chỉ bao gồm xác định tên thuộc tính (có thể thêm kiểu dữ liệu), các đặc điểm khác của thuộc tính sẽ được xác định lại trong giai đoạn thiết kế.

### Thuộc tính quan hệ (Qualifier)

Là một thuộc tính quan hệ (không phải của lớp). Nghĩa là thuộc tính này sẽ hình thành từ một quan hệ giữa các lớp. Nhằm thực hiện việc thiết lập mối liên kết giữa một tập thể hiện với một thể hiện khác. Một đối tượng và một giá trị của thuộc tính qualifier sẽ xác lập một định danh duy nhất, hình thành nên khoá phức hợp.



Xem xét mối kết hợp giữa lớp Sản phẩm và lớp Dòng đặt hàng của đơn đặt hàng. Một dòng trong đơn hàng sẽ liên quan đến một sản phẩm sẽ được đặt, mỗi dòng đơn hàng tham chiếu đến duy nhất một sản phẩm, trong khi đó mỗi sản phẩm có thể được đặt trong nhiều dòng đơn hàng của những đơn hàng khác nhau. Mối liên kết qualifier có thuộc tính Mã SP cho biết: mỗi sản phẩm có duy nhất một Mã SP và Dòng đơn hàng liên kết với Sản phẩm sử dụng Mã SP.

### ***Định danh (identifier)***

Định danh là một hoặc nhiều thuộc tính của lớp có giá trị xác định duy nhất cho một đối tượng của lớp.

### ***Các cách tiếp cận xác định lớp đối tượng***

Gần như không có một phương thức chung để xác định các lớp của một hệ thống. Thông thường đây là một quá trình sáng tạo và lặp lại qua nhiều vòng lặp và cần phải có sự thống nhất với các chuyên gia trong lãnh vực ứng dụng nghiệp vụ. Có nhiều phương pháp tiếp cận để xác định lớp. Có phương pháp đề nghị tiến hành mô hình hoá nghiệp vụ, chỉ ra phạm vi bài toán nghiệp vụ sẽ được tự động hoá mà kết quả của nó sẽ cung cấp các lớp cho hệ thống tương ứng với việc tự động hoá việc quản lý, lưu trữ các đối tượng thông tin (thực thể) đã được thống nhất trên các yêu cầu với người dùng xử lý nghiệp vụ. Có phương pháp đề xuất xác định tất cả các lớp thuộc phạm vi bài toán, mối quan hệ của chúng. Sau đó, sẽ phân tích use case và phân bổ trách nhiệm các lớp theo use case. Có phương pháp đề xuất lấy mô hình use case làm nền tảng để tìm lớp (use case - driven), và trong quá trình xác định trách nhiệm thực hiện của use case thì các lớp sẽ được xác định. Vì quá trình xác định lớp trong giai đoạn này là một quá trình lặp lại mà kết quả của bước sau có thể làm thay đổi các kết quả của bước trước, cho nên các lớp được tìm thấy thường được gọi là lớp ứng viên (candidate class).

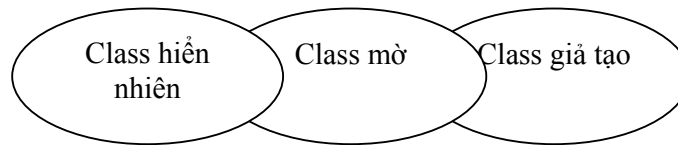
Dưới đây chúng ta đề cập đến một số kỹ thuật tiếp cận để xác định lớp: tiếp cận theo cụm danh từ; tiếp cận theo mẫu chung; tiếp cận theo hướng gia tăng; và tiếp cận theo use case.

### ***Tiếp cận theo cụm danh từ (noun phrase)***

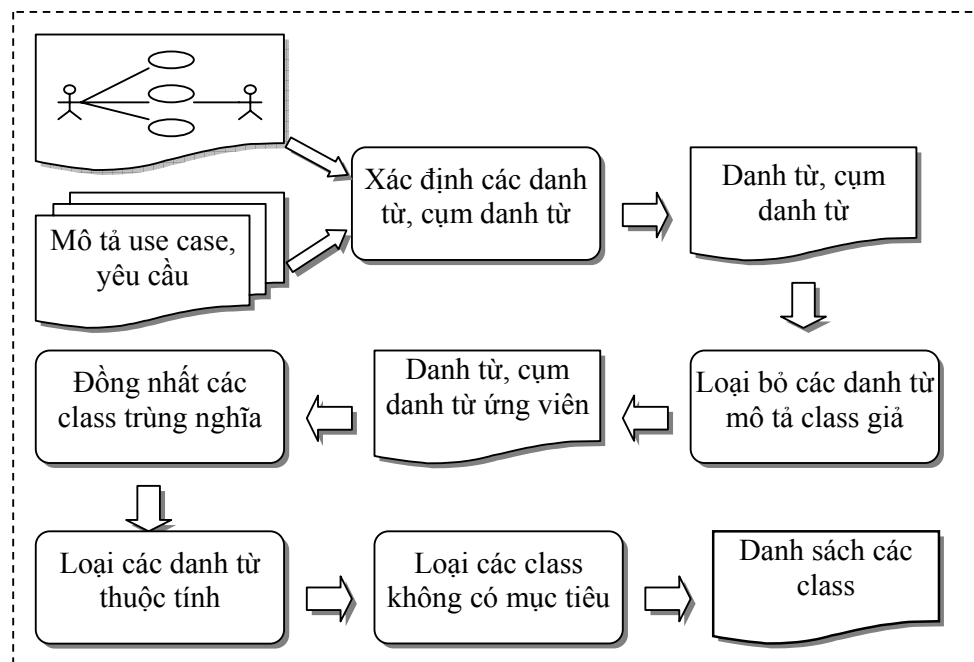
Phương pháp tiếp cận theo cụm danh từ được đề xuất bởi Rebecca Wirfs-Brock, Brian Wilkerson, và Lauren Wiener. Phương pháp đề xuất việc xác định các lớp thông qua việc đọc trong các văn bản mô tả use case hoặc các mô tả yêu cầu để tìm kiếm và trích lọc các cụm danh từ. Các cụm danh từ có thể được xem là các ứng viên của các lớp và các động từ là các ứng viên của phương thức (method) của lớp. Tất cả danh từ hoặc cụm danh từ tìm được sẽ được phân thành ba loại:

- Các lớp hiển nhiên

- Các lớp mờ
- Các lớp giả tạo



Đầu tiên tất cả lớp thuộc loại lớp giả sẽ bị loại bỏ, vì nó không có mục đích hoặc không cần thiết để sử dụng. Các lớp thuộc hai loại còn lại sẽ trở thành các ứng viên. Quy trình xác định như sau:



#### **Khởi tạo danh sách các lớp ứng viên**

- Tìm các danh từ hoặc các cụm danh từ trong các mô tả use case, yêu cầu
- Tất cả các lớp phải có ý nghĩa trong lĩnh vực ứng dụng, tránh đưa vào các lớp cài đặt được mô tả trong giai đoạn thiết kế.
- Đặt tên cho lớp

Trích lục trong use case và mô tả use case của hệ thống ATM, chúng ta có những danh từ và cụm danh từ sau:

Tài khoản	Bao thư
Số dư tài khoản	Bốn ký số
Số tiền	Ngân quỹ
Tiến trình đăng nhập	Tiền
Thẻ ATM	PIN
Máy ATM	PIN không hợp lệ

Ngân hàng	Thông điệp
Khách hàng ngân hàng	Mật khẩu
Thẻ	Mã PIN
Tiền mặt	Mẫu tin
Khách hàng	Bước
Tài khoản khách hàng	Hệ thống
VND	Giao dịch
	Lịch sử giao dịch

### ***Loại bỏ các lớp giả***

Các lớp ứng viên phải thuộc loại lớp hiển nhiên và lớp mờ. Các lớp giả sau đây sẽ bị loại khỏi danh sách: Bao thư, Bồn ký số, Bước.

Tài khoản	<del>Bao thư</del>
Số dư tài khoản	<del>Bồn ký số</del>
Số tiền	Ngân quỹ
Tiến trình đăng nhập	Tiền
Thẻ ATM	PIN
Máy ATM	PIN không hợp lệ
Ngân hàng	Thông điệp
Khách hàng ngân hàng	Mật khẩu
Thẻ	Mã PIN
Tiền mặt	Mẫu tin
Khách hàng	<del>Bước</del>
Tài khoản khách hàng	Hệ thống
VND	Giao dịch
	Lịch sử giao dịch

### ***Đồng nhất các lớp ứng viên trùng lặp***

Cần rà soát lại danh sách để tìm kiếm các danh từ, cụm danh từ trùng lặp về ý nghĩa mặc dù cách dùng từ có khác nhau. Chúng ta chọn lựa danh từ, hoặc cụm danh từ chứa đầy ngữ nghĩa nhất và loại những danh từ, cụm danh từ khác.

Khách hàng, Khách hàng ngân hàng	= Khách hàng
Tài khoản, Tài khoản khách hàng	= Tài khoản
PIN, Mã PIN	= PIN
Tiền, Ngân quỹ	= Ngân quỹ
Thẻ ATM, Thẻ	= Thẻ ATM
Tài khoản	<del>Bao thư</del>

Số dư tài khoản	<del>Bên ký số</del>
Số tiền	Ngân quỹ
Tiến trình đăng nhập	<del>Tiền</del>
Thẻ ATM	PIN
Máy ATM	PIN không hợp lệ
Ngân hàng	Thông điệp
<del>Khách hàng ngân hàng</del>	Mật khẩu
Thẻ	<del>Mã PIN</del>
Tiền mặt	Mẫu tin
Khách hàng	<del>Được</del>
<del>Tài khoản khách hàng</del>	Hệ thống
VND	Giao dịch
	Lịch sử giao dịch

***Xác định các danh từ, cụm danh từ có thể là các thuộc tính***

Các danh từ hoặc cụm danh từ là các thuộc tính khi:

- Chỉ được sử dụng như là giá trị
- Không có nhiều hơn một đặc trưng riêng, hoặc chỉ mô tả một đặc trưng của đối tượng khác.

Xem xét các danh từ, cụm danh từ có thể là thuộc tính của danh sách trên ta có:

Số tiền: một giá trị, không phải một lớp

Số dư tài khoản: thuộc tính của lớp Tài khoản

PIN không hợp lệ: một giá trị, không phải một lớp

Mật khẩu: một thuộc tính (có thể của lớp Khách hàng)

Lịch sử giao dịch: một thuộc tính (có thể của lớp Giao dịch)

PIN: một thuộc tính (có thể của lớp Khách hàng)

Sau đây là danh sách các ứng viên còn lại:

Tài khoản	<del>Bao thư</del>
<del>Số dư tài khoản</del>	<del>Bên ký số</del>
<del>Số tiền</del>	Ngân quỹ
Tiến trình đăng nhập	<del>Tiền</del>
Thẻ ATM	<del>PIN</del>
Máy ATM	<del>PIN không hợp lệ</del>
Ngân hàng	Thông điệp
<del>Khách hàng ngân hàng</del>	Mật khẩu
Thẻ	<del>Mã PIN</del>

Tiền mặt	Mẫu tin
Khách hàng	<del>Được</del>
<del>Tài khoản khách hàng</del>	Hệ thống
VND	Giao dịch
	<del>Lịch sử giao dịch</del>

***Loại bỏ các lớp ứng viên không có mục tiêu hoặc không thuộc phạm vi hệ thống***

Mỗi lớp phải có một mục tiêu khi thuộc hệ thống, mục tiêu này phải thật rõ ràng trong ngữ cảnh mục tiêu chung hệ thống. Nếu chúng ta không thể diễn đạt mục tiêu của lớp trong hệ thống thì loại ra khỏi danh sách. Hoặc các lớp mặc dù có tham gia vào hoạt động của hệ thống, tuy nhiên nó không thuộc phạm vi quản lý của hệ thống sẽ bị loại ra. Các lớp ứng viên là:

**Máy ATM:** cung cấp một giao diện tới ngân hàng

**Thẻ ATM:** cung cấp một khách hàng với một khoá tới một tài khoản

**Khách hàng:** một khách hàng là một cá nhân sử dụng máy ATM, có một tài khoản.

**Ngân hàng:** các khách hàng phụ thuộc vào ngân hàng. Nó là một nơi tập trung các tài khoản và xử lý các giao dịch tài khoản.

**Tài khoản:** nó mô hình hoá một tài khoản của khách hàng và cung cấp các dịch vụ về tài khoản cho khách hàng

**Giao dịch:** mô tả một giao tác của khách hàng khi sử dụng thẻ ATM. Một giao tác được lưu trữ với thời gian, ngày, loại, số tiền, và số dư.

Các danh từ, cụm danh từ không có mục đích hoặc không thuộc phạm vi quản lý của hệ thống:

Thông điệp

Hệ thống

Mẫu tin

Ngân quỹ

VND

Tiền mặt

Tiến trình đăng nhập

Kết quả của quá trình chọn lựa gồm các lớp ứng viên sau hệ thống ATM:

MáyATM

ThẻATM

KháchHàng

NgânHàng

TàiKhoản

GiaoDịch

Nhận xét: một hạn chế chính của cách tiếp cận cụm danh từ là nó phụ thuộc vào tính đúng và đầy đủ của các tài liệu mô tả. Điều này trên thực tế để có được những tài liệu này thì quá là

khó. Hoặc chẳng một văn bản lớn của hệ thống có thể dẫn đến quá nhiều lớp ứng viên! Dầu vậy, cách tiếp cận này rất có tính sư phạm và hữu dụng khi kết hợp với các cách tiếp cận khác.

### **Tiếp cận phân loại**

Phương pháp thứ hai được gọi là phương pháp sử dụng mẫu lớp chung, phương pháp này dựa trên một cơ sở tri thức về việc phân loại lớp theo những mẫu chung. Các mẫu chung đó là:

#### ***Lớp khái niệm (concept)***

Một khái niệm là một quan niệm hoặc sự hiểu biết riêng biệt về thế giới. Lớp khái niệm bao gồm các nguyên lý được dùng để tổ chức hoặc để lưu trữ các hoạt động và các trao đổi về mặt quản lý. Thông thường các khái niệm là các ý tưởng, sự hiểu biết được chia sẻ trong cộng đồng và dùng để trao đổi.

Ví dụ: phương pháp, phương pháp luận, mô hình,... là ví dụ của đối tượng lớp khái niệm.

#### ***Lớp sự kiện (event)***

Lớp sự kiện là các điểm thời gian cần được lưu trữ. Các sự việc xảy ra tại một thời điểm, hoặc một bước trong một dãy tuần tự các bước. Liên quan tới các sự việc được lưu trữ là các thuộc tính (và các đối tượng chứa thuộc tính) như là: ai, cái gì, khi nào, ở đâu, như thế nào, hoặc tại sao.

Ví dụ: đăng ký, kết quả, hoá đơn,...

#### ***Lớp tổ chức (organization)***

Một lớp tổ chức là một tập hợp con người, tài nguyên, phương tiện, hoặc những nhóm xác định chức năng người dùng,....

Ví dụ: đơn vị, bộ phận, phòng ban, chức danh,...

#### ***Lớp con người (people)***

Lớp con người thể hiện các vai trò khác nhau của người dùng trong việc tương tác với ứng dụng. Những đối tượng này thường là người dùng hệ thống hoặc những người không sử dụng hệ thống nhưng thông tin về họ được lưu trữ bởi hệ thống (đa số là những đối tượng mà hệ thống có trao đổi thông tin nhưng không sử dụng hệ thống)

Ví dụ: sinh viên, khách hàng, giáo viên, nhân viên,...

#### ***Lớp vị trí (place)***

Các vị trí vật lý mà hệ thống cần lưu trữ thông tin về nó.

Ví dụ: toà nhà, kho, văn phòng, chi nhánh, đại lý, ...

#### ***Sự vật hữu hình và lớp thiết bị***

Các đối tượng vật lý hoặc các nhóm của đối tượng hữu hình mà có thể cảm nhận trực quan và các thiết bị mà hệ thống tương tác.

Ví dụ: xe hơi, máy bay, ... là các sự vật hữu hình; thiết bị cảm ứng nhiệt là một lớp thiết bị.

Ví dụ: chúng ta cố gắng xác định lại các lớp trong hệ thống ATM dùng phương pháp tiếp cận:

- Các lớp khái niệm:

TàiKhoản
----------

- Các lớp sự kiện:

GiaoDich

- Các lớp tổ chức:

NgânHàng

- Các lớp con người:

KháchHàng

- Các lớp sự vật hữu hình và thiết bị:

MáyATM

ThẻATM

### Cách tiếp cận theo use case

Như chúng ta đã được giới thiệu, use case được dùng để mô hình hoá các kịch bản trong hệ thống và xác định cách thức các tác nhân tương tác với kịch bản. Kịch bản có thể được mô tả bằng văn bản hoặc thông qua một thứ tự các bước. Một khi hệ thống được mô tả trong ngữ nghĩa các kịch bản. Chúng ta có thể kiểm tra đoạn mô tả văn bản hoặc các bước của mỗi kịch bản để xác định các đối tượng nào cần thiết để cho kịch bản được thực hiện. Chúng ta có thể mô hình hoá các kịch bản của use case sử dụng sơ đồ tuần tự (sequence diagram) hoặc sơ đồ hợp tác (collaboration diagram). Các mô hình này cho phép chúng ta mô hình hoá một cách xác định hơn ở giai đoạn phân tích và trợ giúp trong việc thiết kế hệ thống qua việc mô hình hoá sự tương tác giữa các đối tượng trong hệ thống.

Tuy nhiên, việc mô hình hoá kịch bản của use case một cách quá cụ thể sẽ dễ dẫn đến mô tả hoạt động phần mềm hệ thống nơi mà các đối tượng phần mềm có thể sẽ được xác định (mà đúng ra nó phải được xác định ở giai đoạn thiết kế). Do đó, cách tiếp cận này nên kết hợp với cách tiếp cận phân tích cụm danh từ hoặc cách tiếp cận phân loại để xác định đúng các đối tượng trong giai đoạn phân tích.

Trước tiên, chúng ta xác định các dòng tương tác của tác nhân với hệ thống trong một use case. Sau đó, chúng ta sẽ đặt câu hỏi “đối tượng nào của hệ thống sẽ chịu trách nhiệm tiếp nhận sự tương tác này?”. Trả lời câu hỏi này giúp chúng ta tìm ra đối tượng đầu tiên của use case. Nếu đối tượng này chuyển giao toàn bộ hoặc một phần trách nhiệm xử lý cho đối tượng khác nào đó, thì chúng ta tiếp tục tiếp tục xác định đối tượng đó. Quá trình này cứ tiếp tục cho đến khi hết tất cả các dòng tương tác đã được kiểm tra.

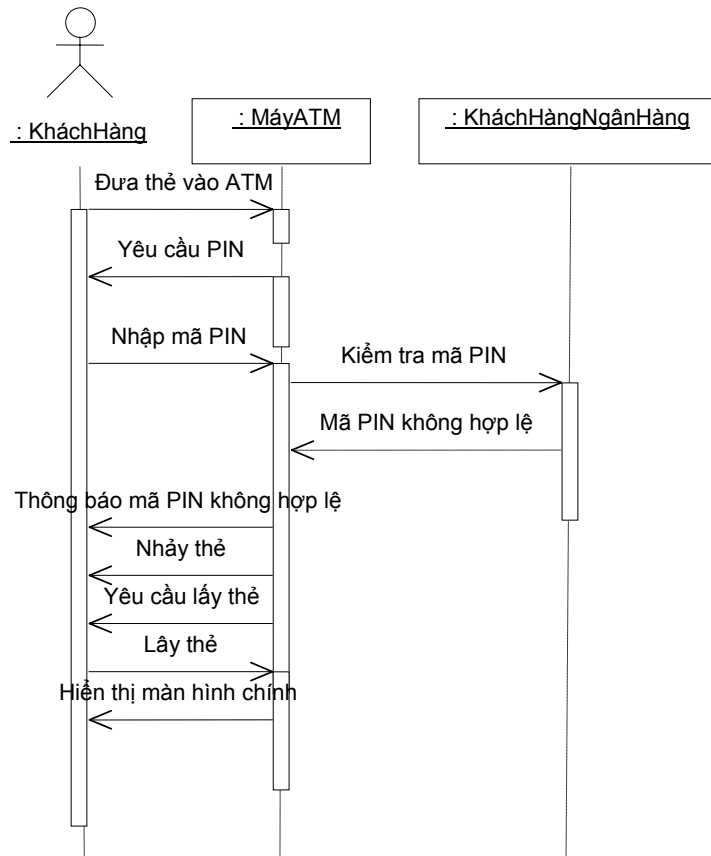
Ví dụ: trong hệ thống ATM chúng ta xem hoạt động của use case “Giải quyết PIN không hợp lệ”. Ở đây chúng ta cần nghĩ về tuần tự các hoạt động mà một khách hàng có thể thực hiện:

- Đưa vào thẻ ATM
- Nhập mã PIN
- Rút thẻ ATM

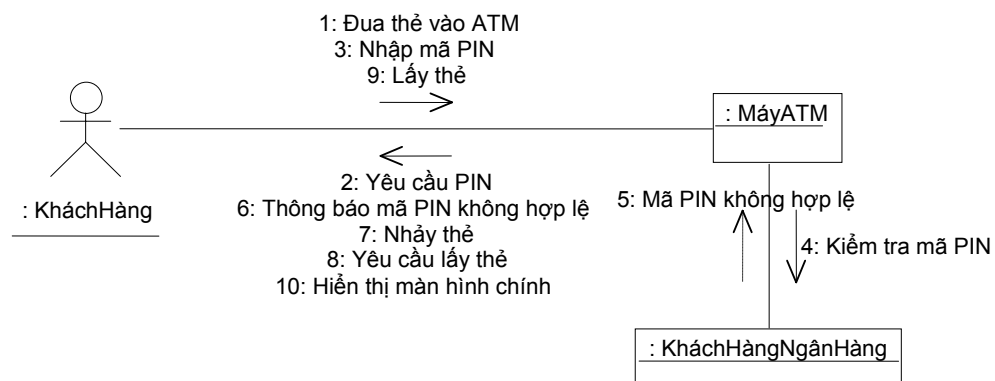


Dựa trên các hoạt động này, phản ứng của hệ thống hoặc chấp nhận quyền truy cập của tài khoản tương ứng hoặc từ chối. Kế tiếp chúng ta cần xác định một cách tường minh hơn về hệ thống: Chúng ta đang tương tác với cái gì (của hệ thống)? Máy ATM. Tiếp tục với kịch bản tiếp theo: máy ATM sẽ sử dụng đối tượng nào để kiểm tra mã PIN? Khách hàng ngân hàng. Một khách hàng trong trường hợp này là bất kỳ người nào muốn truy cập đến một tài khoản thông qua máy ATM, và có thể có hoặc có thể không có tài khoản. Ngược lại, một khách hàng ngân hàng có một tài khoản.

Sơ đồ tuần tự

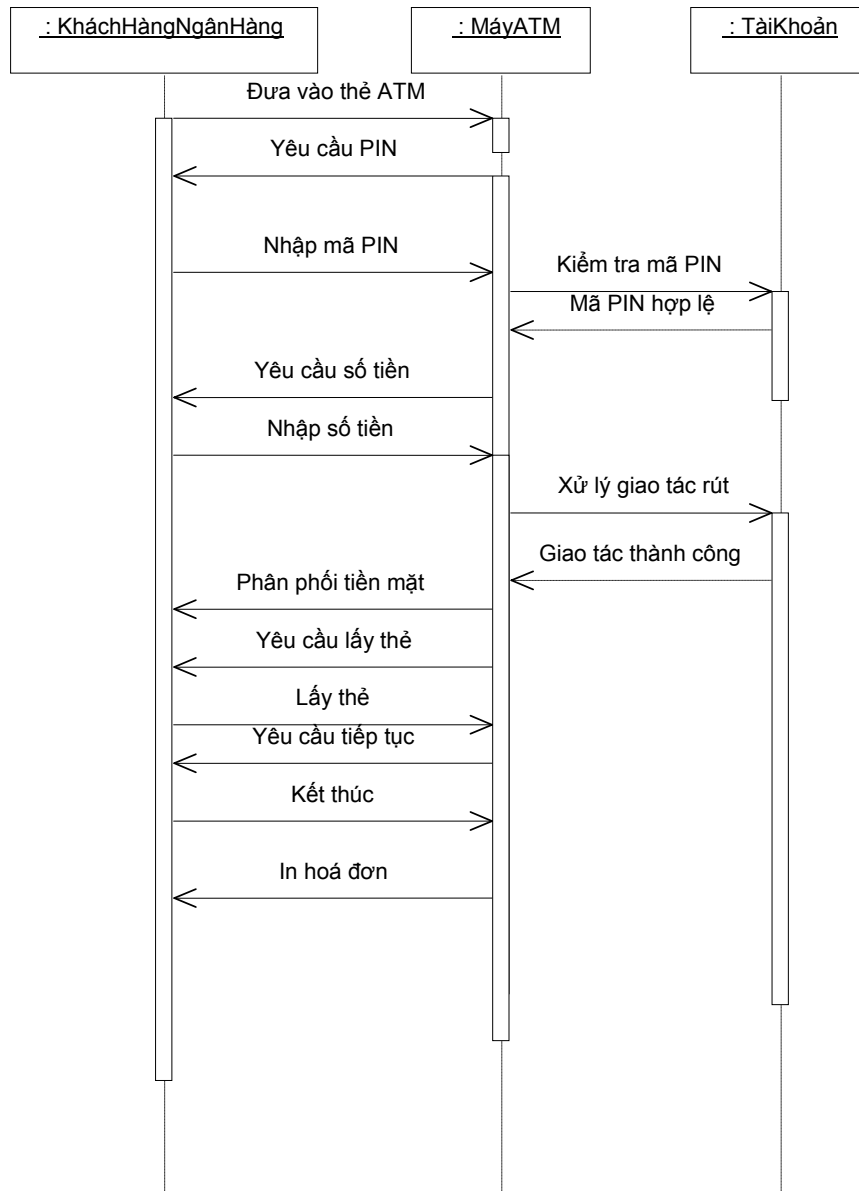


Sơ đồ hợp tác

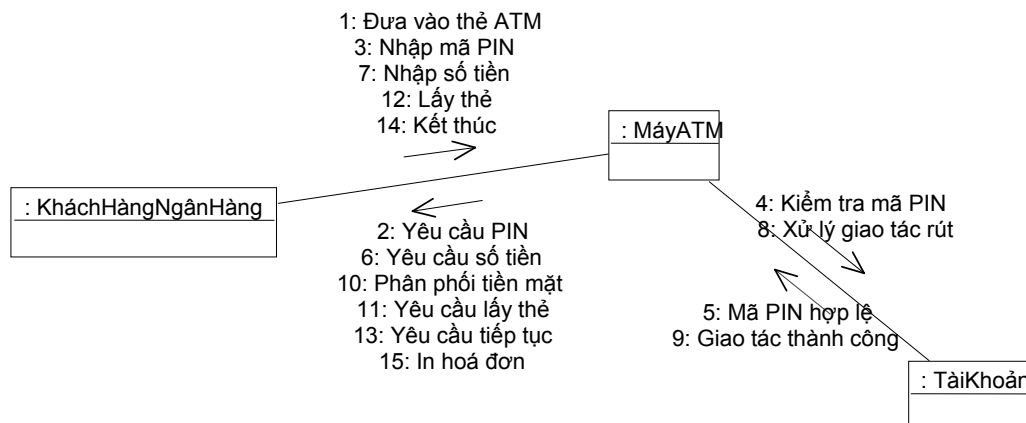


Hoặc xét use case “Rút tiền”

Sơ đồ tuần tự



Sơ đồ hợp tác



Như vậy, dựa vào hai sơ đồ tuần tự của hai use case chúng ta đã xác định được các lớp: MáyATM, KháchHàngNgânHàng (KháchHàng), TàiKhoản. Tiếp tục mô hình hoá với sơ đồ tuần tự hoặc hợp tác với các use case còn lại của hệ thống ATM, chúng ta sẽ xác định được các lớp còn lại.

### **Xác định mối quan hệ giữa các lớp**

Trong môi trường hướng đối tượng, đối tượng đảm nhiệm một vai trò chủ động trong một hệ thống. Đối tượng không tồn tại một cách độc lập mà luôn tương tác với những đối tượng khác. Sự tương tác này thể hiện thông qua mỗi kết hợp bao gồm luôn các hoạt động và phương thức của các đối tượng. Sau đây chúng ta sẽ được giới thiệu các hướng dẫn giúp xác định ba loại liên kết: mỗi kết hợp, mỗi kết hợp thu nạp (thành phần) và mỗi kết hợp tổng quát hoá.

### **Xác định mỗi kết hợp (association)**

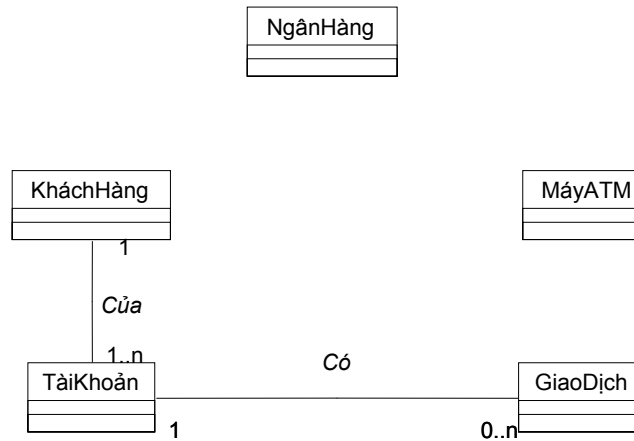
Xác định mỗi kết hợp bắt đầu bằng việc phân tích sự tương tác giữa các lớp. Thông thường thì bất kỳ có một liên kết phụ thuộc nào giữa hai hay nhiều lớp đều có thể thiết lập mỗi kết hợp. Một cách xác định chính là kiểm tra trách nhiệm của đối tượng để thiết lập mỗi kết hợp, nói cách khác, nếu một đối tượng được thiết lập để thi hành một nhiệm vụ và lại thiếu thông tin để thi hành nhiệm vụ đó, thì đối tượng này phải ủy quyền thực hiện lại cho đối tượng khác sở hữu thông tin đó. Có nhiều cách tiếp cận để xác định mỗi kết hợp, đầu tiên trích lọc các mỗi kết hợp ứng viên từ việc tham khảo mô tả hệ thống và yêu cầu hệ thống và những tài liệu khác liên quan đến hệ thống. Sau đó, tinh chế dần để chọn ra mỗi kết hợp có ý nghĩa nhất. Chú ý rằng mỗi kết hợp và mỗi kết hợp thành phần có ngữ nghĩa rất gần nhau, do đó có những trường hợp khó phân biệt, mỗi kết hợp thành phần chỉ là một trường đặc biệt của mỗi kết hợp. Tùy theo lãnh vực ứng dụng nếu chúng ta tìm được một cách tự nhiên để biểu diễn mỗi kết hợp thành phần thì hãy chọn nó, ngược lại hãy chọn mỗi kết hợp để biểu diễn. Sau đây là các hướng dẫn và các mẫu chung cho phép xác định mỗi kết hợp:

#### **Hướng dẫn xác định mỗi kết hợp**

- Một sự phụ thuộc giữa hai hay nhiều lớp có thể thiết lập thành mỗi kết hợp. Mỗi 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 mỗi kết hợp.

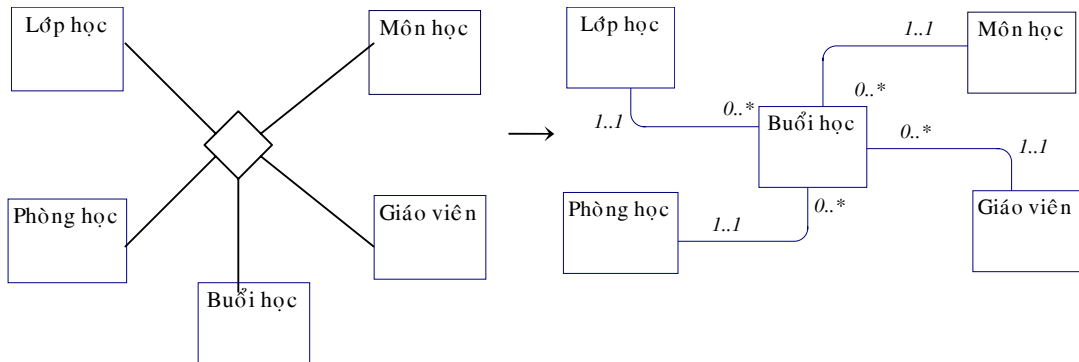
#### **Mẫu chung xác định mỗi kết hợp**

- Mỗi kết hợp vị trí (location): *liên kết tới*, *thành phần của*, *chứa trong*, *làm việc tại*, ....
- Mỗi kết hợp sở hữu: *của*, *có*, *thuộc*, ... Mỗi kết hợp *có* giữa lớp TàiKhoản và lớp GiaoDich, mỗi kết hợp *của* giữa lớp TàiKhoản và lớp KháchHàng
- Mỗi kết hợp truyền thông, liên lạc (communication): *đặt tới*, *trao đổi với*, *gởi cho*, *tiếp nhận từ*, ...

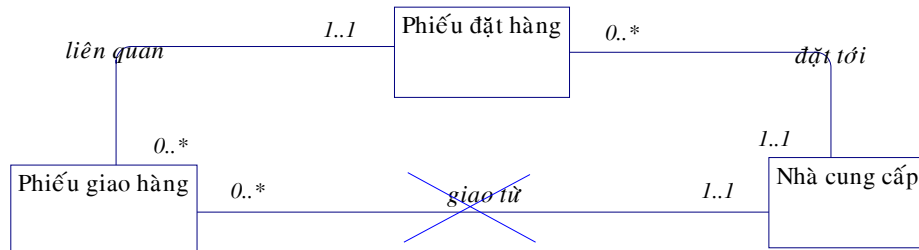


### Loại bỏ những mối kết hợp không cần thiết

- **Mối kết hợp cài đặt:** mối kết hợp cài đặt nên được đưa vào trong quá trình thiết kế và cài đặt hệ thống. Mối kết hợp cài đặt là mối kết hợp mô tả sự liên quan giữa các lớp trong giai đoạn thiết kế cài đặt hệ thống bên trong môi trường phát triển hoặc ngôn ngữ lập trình cụ thể và không phải là mối liên kết giữa các đối tượng mô tả nghiệp vụ.
- **Mối kết hợp đa phân:** là mối kết hợp giữa ba lớp trở lên, mối kết hợp này phức tạp trong cách thể hiện. Nếu có thể, phát biểu lại nó dùng mối kết hợp nhị phân.



- **Mối kết hợp trực tiếp dư thừa (directed action):** là các mối kết hợp được định nghĩa trong ngữ nghĩa của những mối kết hợp khác (còn gọi là mối kết hợp suy diễn hoặc bắc cầu). Những mối kết hợp loại này là dư thừa do đó cần loại bỏ nó khỏi mô hình. Trong các mối kết hợp dưới đây *liên quan*, *đặt tới*, *giao từ* thì mối kết hợp *giao từ* có thể được suy ra từ mối kết hợp *liên quan* và *đặt tới*. Vậy mối kết hợp *giao từ* là dư thừa và có thể loại bỏ nó khỏi mô hình.

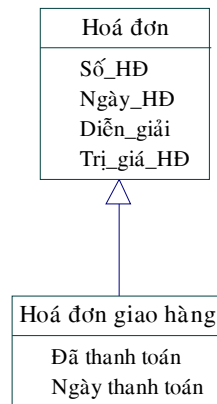


### Xác định mối kết hợp tổng quát – chuyên biệt (generalization)

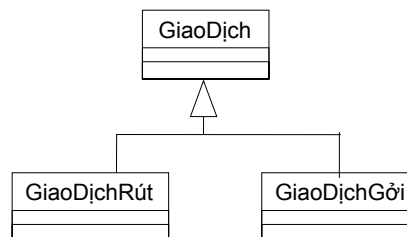
Mối kết hợp tổng quát hoá – chuyên biệt thể hiện quan hệ kế thừa giữa các lớp và một cấu trúc phân cấp xác định những dòng kế thừa này. Quan hệ này cho phép các đối tượng có thể xây dựng từ những đối tượng khác. Một thuận lợi chính khi sử dụng mối kết hợp này là chúng ta có thể xây dựng trên những gì đang có và quan trọng hơn là sử dụng lại cái đã có. Sau đây là các tiếp cận hướng dẫn xác định mối kết hợp tổng quát hoá:

**Tiếp cận trên xuống (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 này mà hệ thống cần chỉ ra hay không. Nếu quyết định là một lớp thì nó là một loại chuyên biệt của lớp ban đầu.

Ví dụ: từ lớp Hoá đơn chúng ta tìm thấy một cụm danh từ Hoá đơn giao hàng có chứa từ Hoá đơn, và trường hợp dưới đây quyết định Hoá đơn giao hàng là một lớp chuyên biệt của lớp Hóa đơn.



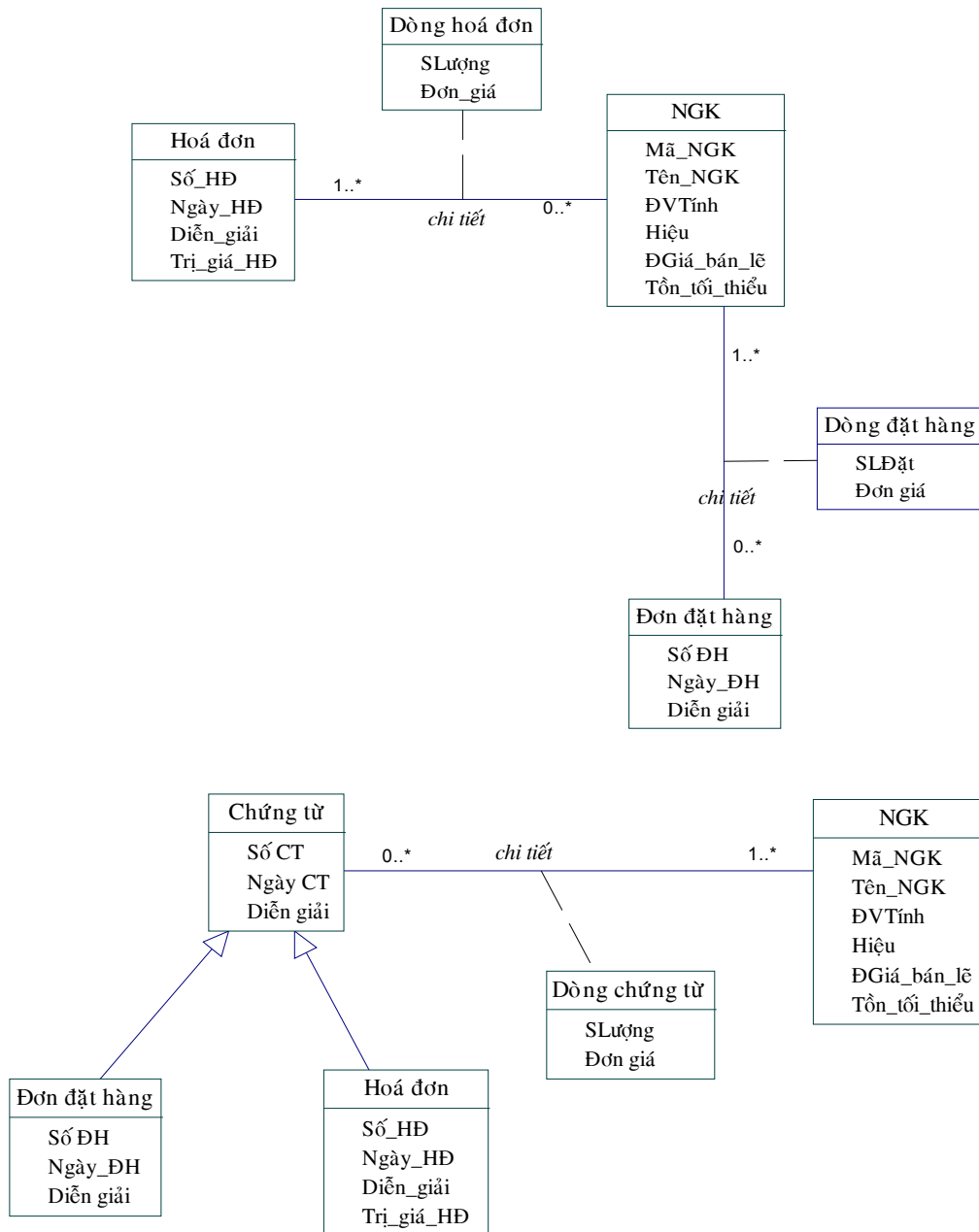
Hoặc trong hệ thống ATM, từ lớp GiaoDich chúng ta có thể chuyên biệt hoá xuống hai lớp con GiaoDichRút và GiaoDichGửi.



Trong quá trình tạo cấu trúc phân cấp tổng quát – chuyên biệt, chúng ta không nên quá lạm dụng việc chuyên biệt hóa mà đưa vào tất cả các lớp chuyên biệt không cần thiết. Các lớp chuyên biệt cần thiết là các lớp có những đặc trưng (thuộc tính, hành vi và mối kết hợp) riêng được biểu diễn trong hệ thống. Ví dụ, có rất nhiều loại hoá đơn có thể tạo lớp chuyên biệt như là: hoá đơn bán lẻ, hoá đơn bán sỉ, hoá đơn giao hàng. Trong khi đó, chỉ có Hoá đơn giao hàng là có những thuộc tính riêng nên nó được biểu diễn trong hệ thống, còn hai lớp còn lại không cần thiết vì không tìm ra đặc trưng riêng của nó.

**Tiếp cận dưới lên (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 mối kết hợp tổng quát hoá từ các lớp này đến lớp tổng quát mới xác định.

Mô hình dưới đây cho thấy, các lớp Hoá đơn và Đơn đặt hàng có ba thuộc tính giống nhau (Hoá đơn: Số\_HĐ, Ngày\_HĐ, Diễn\_giải. Đơn đặt hàng: Số\_ĐH, Ngày\_ĐH, Diễn\_giải) và mỗi kết hợp với lớp NGK giống nhau. Do đó, chúng ta tạo ra một lớp tổng quát Chứng từ và di chuyển cả ba thuộc tính và mỗi kết hợp lên lớp này và tạo ra mỗi kết hợp tổng quát hoá từ lớp Hoá đơn và Đơn đặt hàng đến lớp Chứng từ.



**Tiếp cận tái sử dụng:** di chuyển các thuộc tính và phương thức lên càng cao càng tốt trong cấu trúc phân cấp.

**Đa kế thừa:** tránh sử dụng quá mức đa kế thừa trong mô hình vì đa kế thừa sẽ phức tạp trong việc xác định đường dẫn kế thừa một hành vi từ những lớp nào.

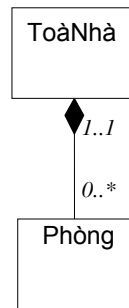
**Xác định mối quan hệ thành phần (a-part-of, aggregation)**

Mối quan hệ thành phần được sử dụng trong tình huống một lớp hình thành bao gồm những lớp thành phần. Hai đặc trưng chính của mối quan hệ thành phần là tính bắc cầu và tính phản đối xứng.

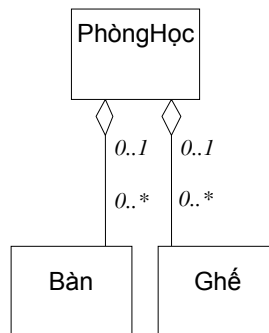
- *Tính bắc cầu*: nếu lớp A là một thành phần của lớp B và lớp B là thành phần của lớp C, thì lớp A là thành phần của lớp C.
- *Tính phản đối xứng*: nếu lớp A là thành phần của lớp B thì lớp B không phải là thành phần của lớp A.

Việc xác định mối quan hệ thành phần có thể dựa trên các mẫu chỉ dẫn sau:

- **Tập hợp**: một đối tượng vật lý được hình thành từ các đối tượng vật lý thành phần khác.

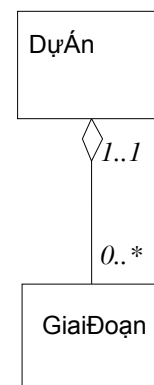
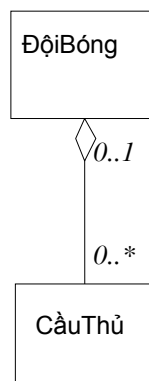
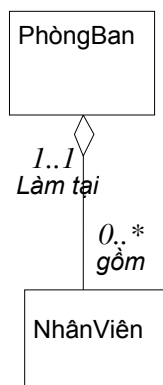


- **Vật chứa**: một đối tượng vật lý chứa đựng các thành phần nhưng không được cấu tạo bởi các thành phần.



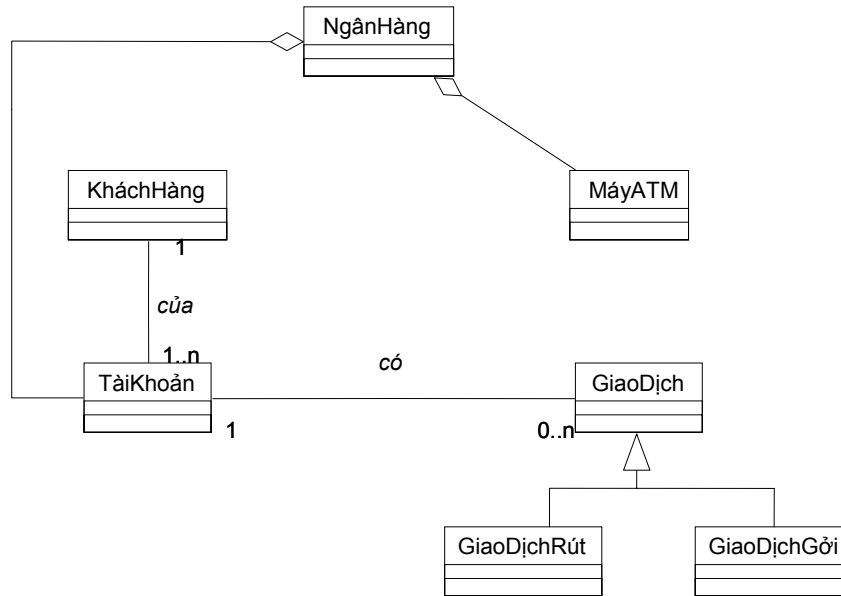
Một phòng học chứa đựng tất cả các đối tượng bàn ghế nhưng không được cấu tạo từ những đối tượng này.

- **Tập hợp – thành viên**: một đối tượng quan niệm chứa các thành phần có thể vật lý hoặc quan niệm.



Các lớp PhòngBan, ĐộiBóng, DữAn, GiaiĐoạn là các lớp quan niệm; các lớp NhânViên, CầuThủ là các lớp vật lý.

Trong hệ thống ATM, một ngân hàng bao gồm các máy ATM, các tài khoản, các toà nhà, các nhân viên,... Tuy nhiên, các đối tượng toà nhà, nhân viên,... không thuộc phạm vi hệ thống đang xét. Do đó, chúng ta định nghĩa mối kết hợp thành phần giữa lớp NgânHàng và các lớp: MáyATM, TàiKhoản.



### **Xác định thuộc tính (attribute) và phương thức (method) của lớp**

Xác định thuộc tính và phương thức cũng là một công việc khó như là xác định lớp, và đây cũng là một tiến trình lặp. Thông thường người ta cũng dựa vào use case và các sơ đồ UML khác để xác định các thuộc tính và phương thức của lớp.

Thuộc tính là các thành phần mà một đối tượng phải ghi nhớ như là màu sắc, trị giá, hàng sản xuất,... Xác định thuộc tính của một lớp hệ thống bắt đầu với việc tìm hiểu về các trách nhiệm của hệ thống. Và chúng ta đã xác định rằng, trách nhiệm của hệ thống có thể được nhận định qua việc phát triển các use case và các đặc điểm mong muốn của ứng dụng như là xác định thông tin gì mà người dùng cần cho hệ thống. Các câu hỏi sau đây có thể giúp xác định nhiệm vụ của lớp và các thành phần dữ liệu mà hệ thống muốn lưu trữ.

- Thông tin gì về đối tượng sẽ được lưu trữ?
- Dịch vụ gì mà một lớp phải cung cấp?

Trả lời câu hỏi thứ nhất giúp chúng ta xác định các thuộc tính của một lớp. Trả lời câu hỏi thứ hai giúp chúng ta xác định các phương thức của lớp.

### **Xác định thuộc tính**

Bằng việc phân tích các use case, các yêu cầu, các mô tả và các sơ đồ chúng ta có thể bắt đầu hiểu trách nhiệm của lớp và cách thức mà các lớp tương tác để thi hành công việc. Mục tiêu chính ở đây là để hiểu những gì mà một lớp có trách nhiệm về tri thức.

Sau đây là một số hướng dẫn giúp xác định lớp trong các use case:



- Thuộc tính thường tương ứng tới các danh từ đi theo bởi các cụm phó từ như là: *chi phí của* sản phẩm. Các thuộc tính cũng có thể tương ứng tới các tính từ hoặc các phó từ.
- Giữ cho lớp đơn giản: chỉ dùng đủ thuộc tính để diễn đạt trạng thái đối tượng
- Các thuộc tính ít có thể được mô tả đầy đủ trong mô tả vấn đề. Do đó, chúng ta phải sử dụng tri thức về lãnh vực ứng dụng và thực tế để tìm chúng.
- Không nên quan tâm quá về việc phải khám phá hết thuộc tính. Chúng ta có thể bổ sung thêm các thuộc tính trong các vòng lặp tiếp theo.

Ví dụ: xác định các thuộc tính cho các lớp của hệ thống ATM.

### **Xác định thuộc tính cho lớp KháchHàng**

Bằng việc phân tích use case, các sơ đồ tuần tự và hợp tác và sơ đồ hoạt động, rõ ràng rằng, với lớp KháchHàng, lãnh vực bài toán và hệ thống đưa ra một vài thuộc tính. Tìm kiếm trong sơ đồ tuần tự của use case “Xử lý PIN không hợp lệ” chúng ta tìm thấy rằng lớp KháchHàng phải có một mã PIN (hay password) và số thẻ. Do đó, *mãPIN* và *sốThẻ* là hai thuộc tính thích hợp của lớp KháchHàng. Các thuộc tính khác của KháchHàng là các biểu diễn tri thức chung về khách hàng, do đó các thuộc tính của lớp KháchHàng là:

tênKháchHàng

họKháchHàng

mãPIN

sốThẻ

trong thời điểm này chúng ta chỉ quan tâm đến chức năng của đối tượng khách hàng mà không quan tâm đến các thuộc tính cài đặt

### **Xác định thuộc tính cho lớp TàiKhoản**

Tương tự các thuộc tính của lớp TàiKhoản được xác định là:

sốTàiKhoản

loạiTàiKhoản

sốDư

### **Xác định thuộc tính cho lớp GiaoDịch**

giaoDịchID

ngàyGiaoDịch

thờiGianGiaoDịch

loạiGiaoDịch

sốTiền

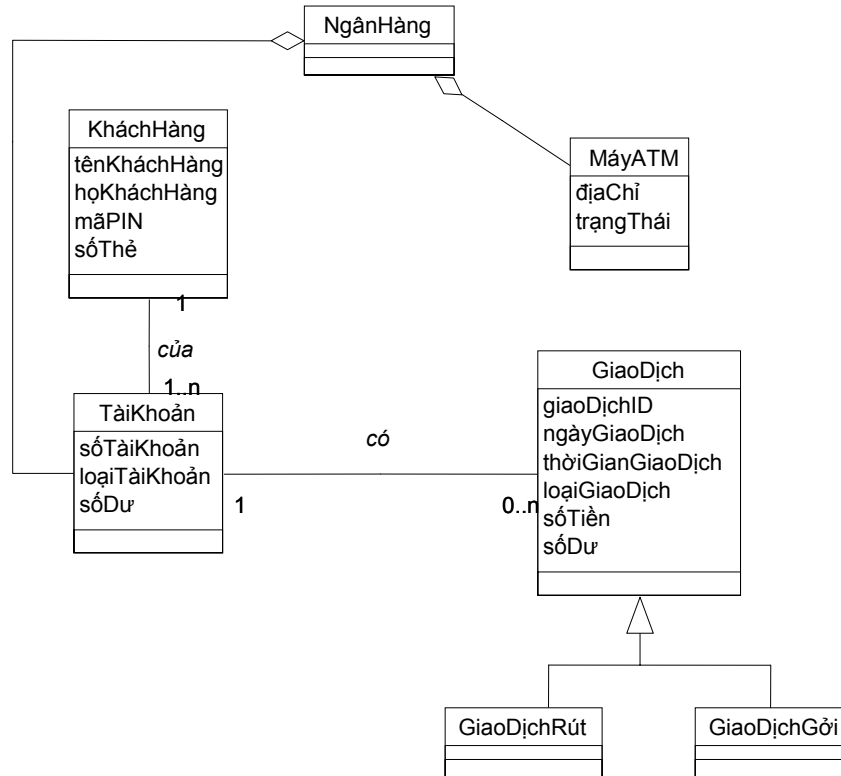
sốDư

### **Xác định thuộc tính cho lớp MáyATM**

Máy ATM là một đối tượng vật lý và hữu hình, do đó thuộc tính của nó dùng để mô tả vị trí và trạng thái của máy.

địaChi

trạngThái



### Xác định phương thức

Phương thức và thông điệp (message) là các thành phần gánh vác công việc xử lý hệ thống hướng đối tượng. Trong một môi trường hướng đối tượng, mỗi phần dữ liệu hoặc đối tượng được bao quanh bởi một tập hợp các thường trình gọi là các phương thức. Những phương thức này chính là các dịch vụ và các toán tử của một lớp định nghĩa để cài đặt hành vi của các đối tượng thành viên của lớp. Các phương thức chính là cách thức mà đối tượng thực hiện tương tác với các đối tượng khác trong hệ thống. Phát hiện các phương thức để cài đặt các hành vi đối tượng là một hoạt động quan trọng trong giai đoạn phân tích.

Như là thuộc tính, một lớp sẽ có những phương thức nội bộ (private) và toàn cục (public). Trong giai đoạn phân tích, chúng ta chỉ quan tâm phát hiện các phương thức toàn cục mà không quan tâm đến các phương thức nội bộ của đối tượng. Các phương thức thường tương ứng với các truy vấn về các thuộc tính của đối tượng. Nói cách khác, các phương thức chịu trách nhiệm quản lý các trị của thuộc tính như là truy vấn, cập nhật, đọc và ghi. Chú ý rằng trong giai đoạn phân tích, chúng ta đang làm việc ở mức cao của sự trừu tượng hoá. Do đó, các phương thức như là tạo (constructor) hoặc các phương thức mô tả chi tiết việc cài đặt sẽ được phát hiện trong giai đoạn thiết kế.

Trong phần này chúng ta xem xét cách thức xác định phương thức sử dụng các sơ đồ UML như là: sơ đồ trạng thái, sơ đồ hoạt động, sơ đồ tuần tự/hợp tác và sơ đồ use case.

### Xác định phương thức bằng việc phân tích các sơ đồ UML và use case

Trong sơ đồ tuần tự, các đối tượng được đặt trong sơ đồ với các đường đứt quãng thẳng đứng. Do đó, các sự kiện xảy ra giữa các đối tượng được đặt theo dòng nằm ngang. Một sự kiện được xem như là một hành động để chuyển thông tin. Mặt khác, những hành động này là các toán tử mà đối tượng phải thực thi.

Ví dụ, để xác định các phương thức của lớp TàiKhoản, chúng ta xem xét các sơ đồ tuần tự ứng với các use case:

Rút tiền

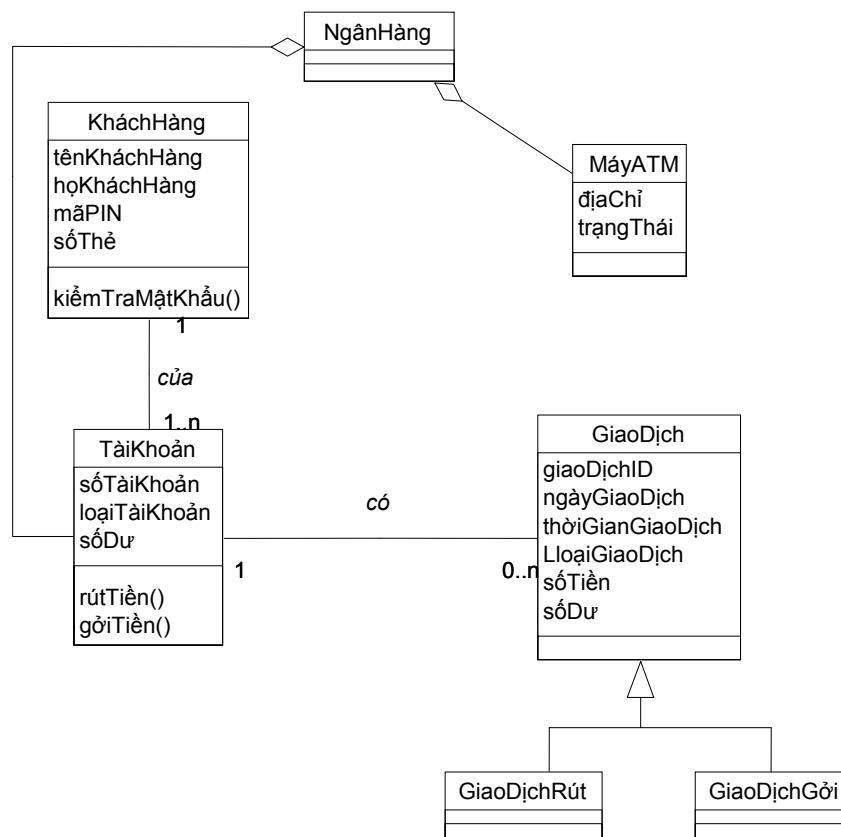
Gửi tiền

Xem thông tin tài khoản

Sơ đồ tuần tự có thể trợ giúp chúng ta xác định các dịch vụ mà các đối tượng phải cung cấp. Ví dụ, qua việc nghiên cứu sơ đồ tuần tự của use case *Rút tiền*, chúng ta thấy rằng lớp TàiKhoản phải cung cấp dịch vụ *rútTiền*. Qua việc nghiên cứu use case *Gửi tiền*, lớp TàiKhoản phải cung cấp dịch vụ *gửiTiền*.

Tương tự, các dịch vụ lớp KháchHàng:

kiểmTraMậtKhẩu (kiểm tra mật khẩu của khách hàng)



Chú ý, các dịch vụ được đưa vào lớp tổng quát sẽ được thừa kế trong các lớp chuyên biệt.

## Câu hỏi và bài tập

### Câu hỏi

1. Các đối tượng của hệ thống trong giai đoạn phân tích có thể xác định từ đâu?
2. Mô tả chiến lược “cụm danh từ” trong việc xác định các lớp ứng viên trong một lãnh vực vấn đề?
3. Chiến lược phân loại là gì?
4. Các lớp khái niệm là gì?

5. Các lớp sự kiện là gì?
6. Các lớp tổ chức là gì?
7. Các lớp con người là gì?
8. Các lớp vị trí là gì?
9. Các lớp sự vật hữu hình và thiết bị là gì?
10. Tại sao xây dựng các sơ đồ tuần tự/hợp tác là một hoạt động hữu dụng cho việc xác định các lớp?
11. Tại sao việc xác định lớp là một quá trình gia tăng?
12. Tại sao việc xác định sự phân cấp của các lớp là quan trọng trong phân tích hướng đối tượng?
13. Liên kết kết hợp (association), tổng quát hoá (generalization) là gì?
14. Thuộc tính và phương thức của lớp được xác định bằng cách nào?

## **Bài tập**

1. Hãy xây dựng sơ đồ lớp của hệ thống “Diễn đàn trao đổi học tập của khoa Công Nghệ Thông Tin”.