



# Lớp và đối tượng trong UML

Bởi:

duongkieuhoa

tonthathoan

## Lớp và đối tượng trong UML

UML thể hiện lớp bằng hình chữ nhật có 3 phần. Phần thứ nhất chứa tên lớp. Trong phần thứ hai là thuộc tính và các dữ liệu thành phần của lớp và trong phần thứ ba là các phương thức hay hàm thành phần của lớp.

### Tên lớp (lass name) :

Tên lớp được in đậm (bold) và căn giữa. Tên lớp phải được dẫn xuất từ phạm vi vấn đề và rõ ràng như có thể. Vì thế nó là danh từ, ví dụ như tài khoản, nhân viên, ....

### Thuộc tính (attribute):

Lớp có thuộc tính miêu tả những đặc điểm của đối tượng. Giá trị của thuộc tính thường là những dạng dữ liệu đơn giản được đa phần các ngôn ngữ lập trình hỗ trợ như Integer, Boolean, Floats, Char, ...

Thuộc tính có thể có nhiều mức độ trông thấy được (visibility) khác nhau, miêu tả liệu thuộc tính đó có thể được truy xuất từ các lớp khác, khác với lớp định nghĩa ra nó. Nếu thuộc tính có tính trông thấy là công cộng (public), thì nó có thể được nhìn thấy và sử dụng ngoài lớp đó. Nếu thuộc tính có tính trông thấy là riêng (private), bạn sẽ không thể truy cập nó từ bên ngoài lớp đó. Một tính trông thấy khác là bảo vệ (protected), được sử dụng chung với công cụ khái quát hóa và chuyên biệt hóa. Nó cũng giống như các thuộc tính riêng nhưng được thừa kế bởi các lớp dẫn xuất.

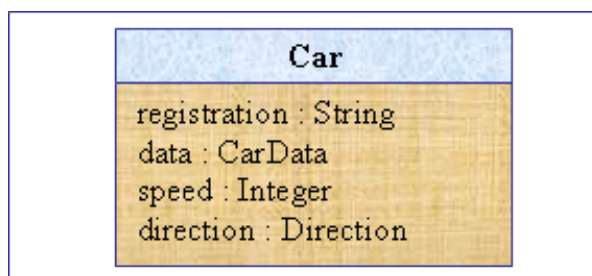
Trong UML, thuộc tính công cộng mang kí hiệu "+" và thuộc tính riêng mang dấu "-".

Giá trị được gán cho thuộc tính có thể là một cách để miêu tả trạng thái của đối tượng. Mỗi lần các giá trị này thay đổi là biểu hiện cho thấy có thể đã xảy ra một sự thay đổi trong trạng thái của đối tượng.

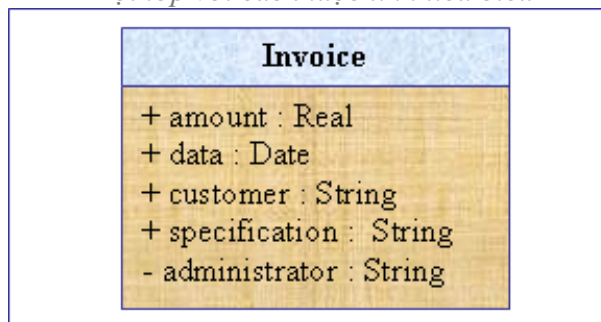
Lưu ý: Mọi đặc điểm của một thực thể là những thông tin cần lưu trữ đều có thể chuyển thành thuộc tính của lớp miêu tả loại thực thể đó.

### Phương thức (methods):

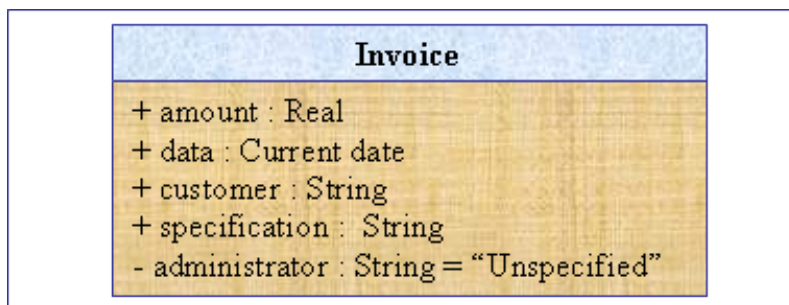
Phương thức định nghĩa các hoạt động mà lớp có thể thực hiện. Tất cả các đối tượng được tạo từ một lớp sẽ có chung thuộc tính và phương thức. Phương thức được sử dụng để xử lý thay đổi các thuộc tính cũng như thực hiện các công việc khác. Phương thức thường được gọi là các hàm (function), nhưng chúng nằm trong một lớp và chỉ có thể được áp dụng cho các đối tượng của lớp này. Một phương thức được miêu tả qua tên, giá trị trả về và danh sách của 0 cho tới nhiều tham số. Lúc thi hành, phương thức được gọi kèm theo một đối tượng của lớp. Vì nhóm các phương thức miêu tả những dịch vụ mà lớp có thể cung cấp nên chúng được coi là giao diện của lớp này. Giống như thuộc tính, phương thức cũng có tính trông thấy được như công cộng, riêng và bảo vệ.



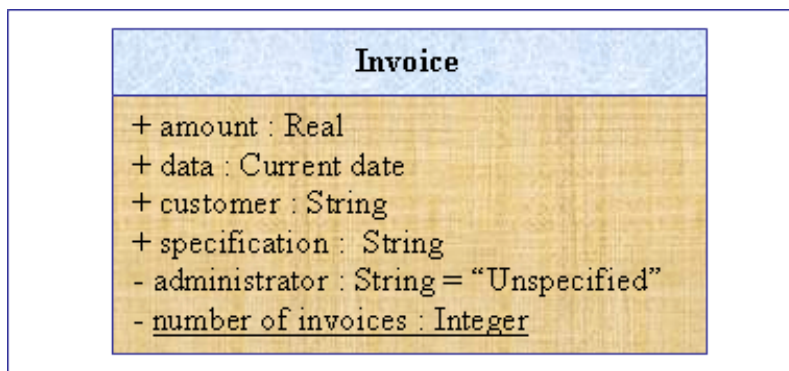
*Một lớp với các thuộc tính tiêu biểu*



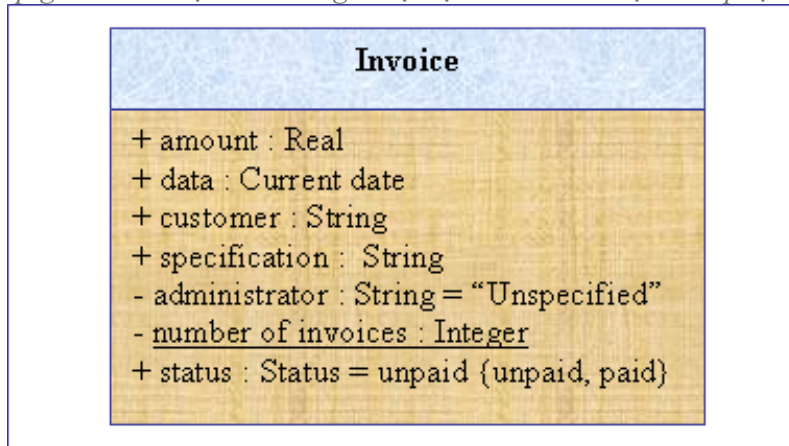
Một lớp với các thuộc tính chung và riêng



*Một lớp với các thuộc tính và giá trị mặc nhiên*

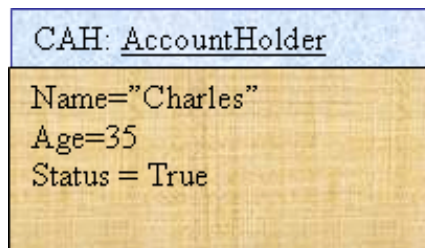


*Một lớp gồm các thuộc tính với giá trị mặc nhiên và thuộc tính phạm vi lớp*



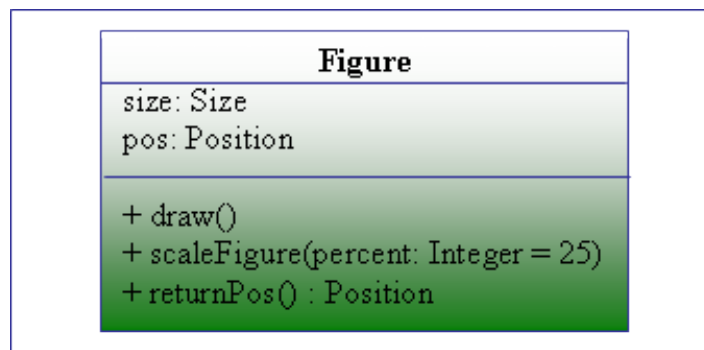
### Kí hiệu đối tượng:

Đối tượng là thực thể của các lớp nên kí hiệu dùng cho đối tượng cũng là kí hiệu dùng cho lớp.

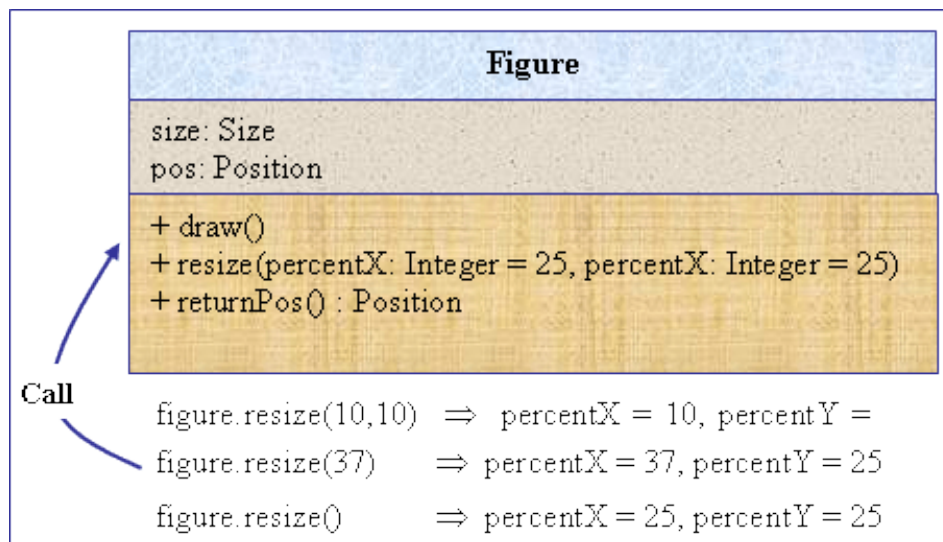


### Ký hiệu đối tượng

Hình trên được đọc như sau: CAH là đối tượng của lớp AccountHolder. Các thuộc tính được gán giá trị, đây là các giá trị khi lớp được thực thể hóa. Chú ý rằng kí hiệu đối tượng không chứa phần phương thức.



### Các dấu hiệu hành động



### Các giá trị mặc nhiên của tham số

### Quan hệ giữa các lớp

Biểu đồ lớp thể hiện các lớp và các mối quan hệ giữa chúng. Quan hệ giữa các lớp gồm có bốn loại:

- Liên hệ (Association)
- Khái quát hóa (Generalization)
- Phụ thuộc (Dependency)
- Nâng cấp (Refinement)

Một liên hệ là một sự nối kết giữa các lớp, cũng có nghĩa là sự nối kết giữa các đối tượng của các lớp này. Trong UML, một liên hệ được định nghĩa là một mối quan hệ miêu tả

một tập hợp các nối kết (links), trong khi nối kết được định nghĩa là một sự liên quan về ngữ nghĩa (semantic connection) giữa một nhóm các đối tượng.

Khái quát hóa là mối quan hệ giữa một yếu tố mang tính khái quát cao hơn và một yếu tố mang tính chuyên biệt hơn. Yếu tố mang tính chuyên biệt hơn có thể chứa chỉ các thông tin bổ sung. Một thực thể (một đối tượng là một thực thể của một lớp) của yếu tố mang tính chuyên biệt hơn có thể được sử dụng ở bất cứ nơi nào mà đối tượng mang tính khái quát hóa hơn được phép.

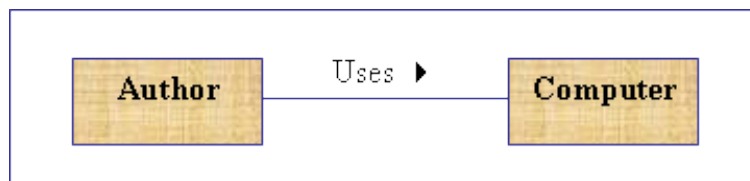
Sự phụ thuộc là một mối quan hệ giữa các yếu tố, gồm một yếu tố mang tính độc lập và một yếu tố mang tính phụ thuộc. Một sự thay đổi trong yếu tố độc lập sẽ ảnh hưởng đến yếu tố phụ thuộc.

Một sự nâng cấp là mối quan hệ giữa hai lời miêu tả của cùng một sự vật, nhưng ở những mức độ trừu tượng hóa khác nhau.

### Liên hệ (Association)

Một 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. Một mối liên hệ biểu thị bằng các đối tượng của hai lớp có nối kết với nhau, ví dụ rằng "chúng biết về nhau", "được nối với nhau", "cứ mỗi X lại có một Y", .... Lớp và liên hệ giữa các lớp là những công cụ rất mạnh mẽ cho việc mô hình hóa các hệ thống phức tạp, ví dụ như cấu trúc sản phẩm, cấu trúc văn bản và tất cả các cấu trúc thông tin khác.

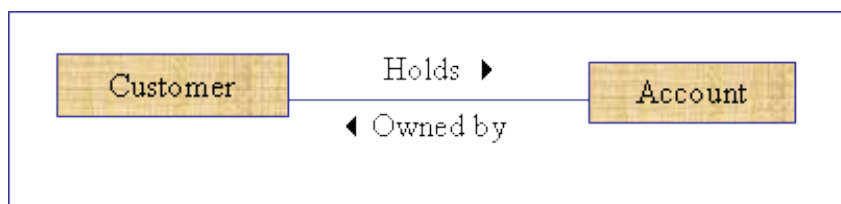
Mỗi liên kết được thể hiện trong biểu đồ UML bằng một đường thẳng nối hai lớp.



Một lớp Author kết hợp với lớp Computer

### Vai trò trong liên hệ:

Một liên hệ có thể có các vai trò (**Roles**). Các vai trò được nối với mỗi lớp bao chứa trong quan hệ. Vai trò của một lớp là chức năng mà nó đảm nhận nhìn từ góc nhìn của lớp kia. Tên vai trò được viết kèm với một mũi tên chỉ từ hướng lớp chủ nhân ra, thể hiện lớp này đóng vai trò như thế nào đối với lớp mà mũi tên chỉ đến.



### Vai trò trong liên hệ giữa Customer và Account

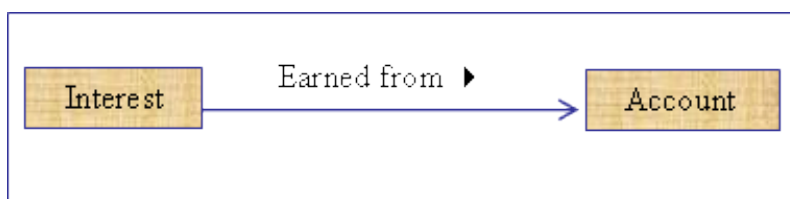
Trong ví dụ trên: một khách hàng có thể là chủ nhân của một tài khoản và tài khoản được chiếm giữ bởi khách hàng. Đường thẳng thể hiện liên hệ giữa hai lớp.

Một số điểm cần chú ý khi đặt tên vai trò :

- Tên vai trò có thể bỏ đi nếu trùng với tên lớp
- Tên vai trò phải là duy nhất.
- Tên vai trò phải khác với các thuộc tính của lớp.
- Tên vai trò phải miêu tả được chức năng mà lớp này đảm nhận trong quan hệ, tức cần phải là các khái niệm lấy ra từ phạm vi vấn đề, giống như tên các lớp.

### Liên hệ một chiều (Uni-Directional Association):

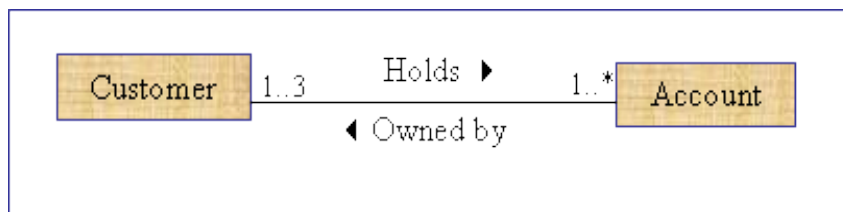
Ta cũng có thể sử dụng mối liên hệ một chiều bằng cách thêm một mũi tên và một đầu của đường thẳng nối kết. Mũi tên chỉ ra rằng sự nối kết chỉ có thể được sử dụng duy nhất theo chiều của mũi tên.



### Liên hệ một chiều giữa Interest và Account

Biểu đồ phần 5.15 thể hiện rằng giữa hai lớp có liên hệ, nhưng không hề có thông tin về số lượng các đối tượng trong quan hệ. Ta không thể biết một khách hàng có thể có bao nhiêu tài khoản và một tài khoản có thể là của chung cho bao nhiêu khách hàng. Trong UML, loại thông tin như thế được gọi là số lượng phân tử (Cardinality) trong quan hệ.

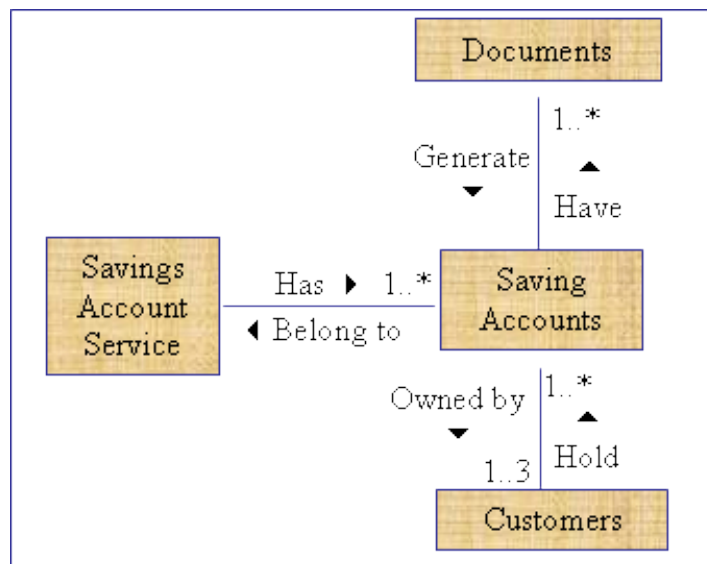
### Số lượng (Cardinality) trong liên hệ:



Số lượng trong liên hệ giữa Customer và Account

Biểu đồ trên nói rõ một khách hàng có thể mở một hoặc nhiều tài khoản và một tài khoản có thể thuộc về một cho tới ba khách hàng.

Số lượng được ghi ở phía đầu đường thẳng thể hiện liên hệ, sát vào lớp là miền áp dụng của nó. Phạm vi của số lượng phần tử trong liên hệ có thể từ 0-tới-1 (0..1), 0-tới-nhiều (0..\* hay ), một-tới-nhiều (1..), hai (2), năm-tới-mười một (5..11). Cũng có thể miêu tả một dãy số ví dụ (1,4,6, 8..12). Giá trị mặc định là 1.



Một sơ đồ lớp tiêu biểu

Hình trên là ví dụ cho một biểu đồ lớp tiêu biểu. Biểu đồ giải thích rằng bộ phận dịch vụ tài khoản tiết kiệm của một nhà băng có thể có nhiều tài khoản tiết kiệm nhưng tất cả những tài khoản này đều thuộc về bộ phận đó. Một tài khoản tiết kiệm về phần nó lại có thể có nhiều tài liệu, nhưng những tài liệu này chỉ thuộc về một tài khoản tiết kiệm mà thôi. Một tài khoản tiết kiệm có thể thuộc về từ 1 cho tới nhiều nhất là 3 khách hàng. Mỗi khách hàng có thể có nhiều hơn một tài khoản.

### **Phát hiện liên hệ:**

Thường sẽ có nhiều mối liên hệ giữa các đối tượng trong một hệ thống. Quyết định liên hệ nào cần phải được thực thi là công việc thuộc giai đoạn thiết kế. Có thể tìm các mối liên hệ qua việc nghiên cứu các lời phát biểu vấn đề, các yêu cầu. Giống như danh từ đã giúp chúng ta tìm lớp, các **động từ** ở đây sẽ giúp ta tìm ra các mối quan hệ.

Một vài lời mách bảo khi tìm liên hệ :

- Vị trí về mặt vật lý hoặc sự thay thế, đại diện: Mỗi cụm động từ xác định hay biểu lộ một vị trí đều là một biểu hiện chắc chắn cho liên hệ. Ví dụ: tại địa điểm, ngồi trong, ...
- Sự bao chứa: Cụm động từ biểu lộ sự bao chứa, ví dụ như : là thành phần của....
- Giao tiếp: Có nhiều cụm động từ biểu lộ sự giao tiếp, ví dụ truyền thông điệp, nói chuyện với, ...
- Quyền sở hữu: Ví dụ : thuộc về, của, ...
- Thoả mãn một điều kiện: Những cụm từ như : làm việc cho, là chồng/vợ của, quản trị, ....

### **Xử lý các liên hệ không cần thiết:**

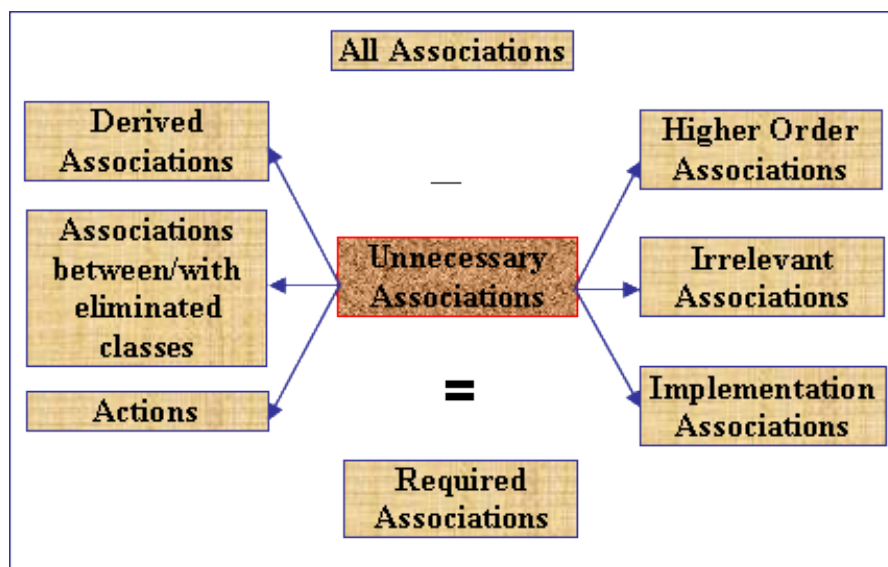
Sau khi tìm các mối liên hệ, bước tiếp theo đó là phân biệt các liên hệ cần thiết ra khỏi các liên hệ không cần thiết. Liên hệ không cần thiết có thể bao gồm những liên hệ bao chứa các lớp ứng cử viên đã bị loại trừ hoặc các liên hệ không liên quan đến hệ thống. Có những liên hệ được tạo ra nhằm mục đích tăng hiệu quả. Những liên hệ như thế là ví dụ tiêu biểu của các chi tiết thực thi và không liên quan tới giai đoạn này.

Cần chú ý phân biệt giữa hành động và mối liên hệ. Người ta thường có xu hướng miêu tả hành động như là liên hệ, bởi cả liên hệ lẫn hành động đều được dẫn xuất từ những cụm từ mang tính động từ trong bản miêu tả yêu cầu. Các hành động đã được thể hiện sai thành liên hệ cũng cần phải được loại bỏ. Khi làm việc này, có thể áp dụng một nguyên tắc: liên hệ là nối kết mang tính tĩnh giữa các đối tượng, trong khi hành động chỉ là thao tác xảy ra một lần. Hành động vì vậy nên được coi là Phương thức đối với một đối tượng chứ không phải quan hệ giữa các lớp.

Ví dụ với "Ban quản trị nhà băng đuổi việc một nhân viên", động từ "đuổi việc" thể hiện hành động. Trong khi đó với "Một nhân viên làm việc cho hãng" thì động từ "làm việc" miêu tả liên hệ giữa hai lớp nhân viên và hãng.



Trong khi cố gắng loại bỏ các liên hệ dư thừa, bạn sẽ thấy có một số liên hệ dư thừa đã "lèn vào" mô hình của chúng ta trong giai đoạn thiết kế. Hình sau chỉ ra một số loại liên hệ dư thừa cần đặc biệt chú trọng.



Loại bỏ các liên hệ không cần thiết

### Nâng cấp các mối liên hệ:

Một khi các mối liên hệ cần thiết đã được nhận dạng, bước tiếp theo là nghiên cứu kỹ mô hình và nâng cấp các mối liên hệ đó.

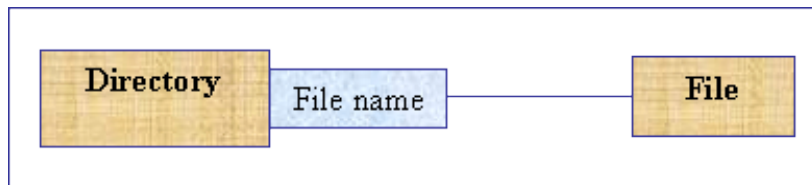
Động tác nâng cấp đầu tiên là xem xét lại tên liên hệ, tên vai trò, đặt lại cho đúng với bản chất quan hệ mà chúng thể hiện. Mỗi liên hệ cần phải được suy xét kỹ về phương diện số lượng thành phần tham gia (Cardinality). Sự hạn định (Qualification) cho liên hệ đóng một vai trò quan trọng ở đây, bổ sung yếu tố hạn định có thể giúp làm giảm số lượng. Nếu cần thiết, hãy bổ sung các liên hệ còn thiếu. Nghiên cứu kỹ các thuộc tính, xem liệu trong số chúng có thuộc tính nào thật ra thể hiện liên hệ. Nếu có, hãy chuyển chúng thành liên hệ. Bổ sung các thông tin và điều kiện cần thiết cũng như xem xét các mối liên hệ trong mô hình tổng thể để xác định các dạng quan hệ giữa chúng với nhau.

### Liên hệ và yếu tố hạn định (Qualifier):

Một liên hệ được hạn định liên hệ hai lớp và một yếu tố hạn định (Qualifier) với nhau. Yếu tố hạn định là một thuộc tính hạn chế số lượng thành phần tham gia trong một mối liên hệ. Có thể hạn định các mối liên hệ một-tới-nhiều và nhiều-tới-nhiều. Yếu tố hạn định giúp phân biệt trong nhóm đối tượng của đầu nhiều của liên hệ.

Ví dụ một thư mục có nhiều tập tin. Một tập tin chỉ thuộc về một thư mục mà thôi. Trong một thư mục xác định, tên của tập tin sẽ xác định duy nhất tập tin mang tên đó. Thư mục

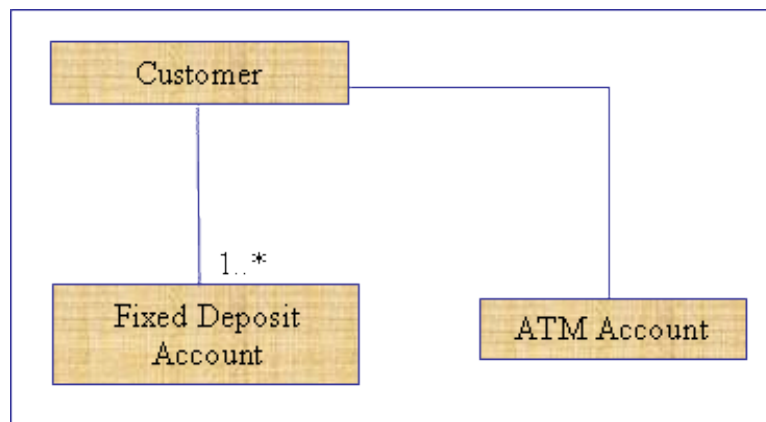
và Tập tin là hai lớp, và tên tập tin ở đây đóng vai trò yếu tố hạn định. Một thư mục và một tên tập tin xác định một tập tin. Yếu tố hạn định ở đây đã chuyển một mối liên hệ một-tới-nhiều thành liên hệ một-tới-một.



Liên hệ được hạn định

### Liên hệ VÀ (AND Association)

Nhà băng nợ đưa ra quy định: khách hàng khi muốn mở một tài khoản ATM phải là chủ nhân của ít nhất một tài khoản đầu tư. Trong một trường hợp như thế, mối liên hệ VÀ (AND) sẽ được thể hiện như sau:

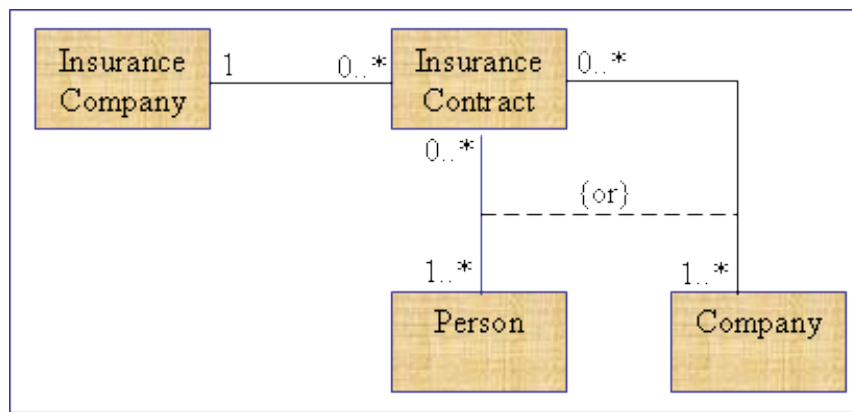


Liên hệ VÀ (AND Association)

Biểu đồ trên cho thấy một khách hàng có thể có nhiều hơn một tài khoản đầu tư có thời hạn và chỉ một tài khoản ATM. Trong biểu đồ có một mối liên hệ VÀ ngầm được áp dụng giữa nhóm tài khoản đầu tư và tài khoản ATM mà một khách hàng có thể có.

### Liên hệ HOẶC (OR Association)

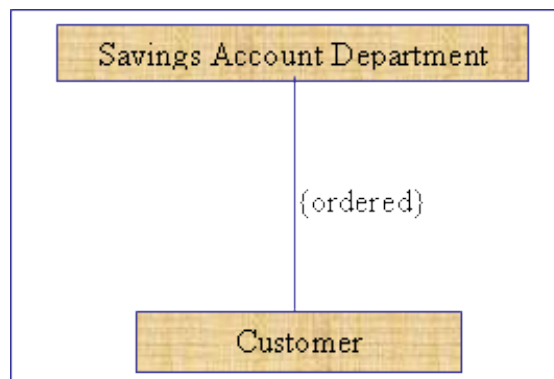
Ví dụ tại một hãng bảo hiểm nợ, cá nhân cũng công ty đều có thể ký hợp đồng bảo hiểm, nhưng cá nhân và công ty không được phép có cùng loại hợp đồng bảo hiểm như nhau. Trong một trường hợp như thế, giải pháp là sử dụng liên hệ HOẶC (OR Association). Một liên hệ HOẶC là một sự hạn chế đối với một nhóm hai hay nhiều liên hệ, xác định rằng đối tượng của một lớp này tại một thời điểm chỉ có thể tham gia vào nhiều nhất một trong các mối liên hệ đó.



Một liên hệ OR mà biểu thị chỉ một liên hệ là hợp lệ tại mỗi thời điểm

### Liên hệ được sắp xếp (Ordered Association)

Các mối nối kết (link) giữa các đối tượng có một trật tự ngầm định. Giá trị mặc định của trật tự này là ngẫu nhiên. Một liên hệ có trật tự rõ ràng có thể được hiểu là một liên hệ với **trật tự sắp xếp** (sort order) trong nhóm các nối kết, nó sẽ được thể hiện như sau:

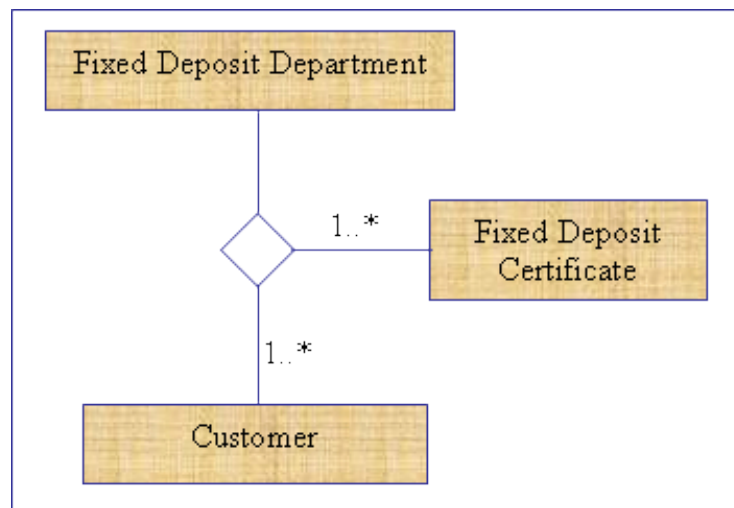


Tài khoản tiết kiệm được sắp xếp theo khách hàng

Nhãn **{ordered}** được ghi gần lớp có đối tượng được sắp xếp. Biểu đồ trên được đọc là các tài khoản tiết kiệm được sắp xếp theo khách hàng.

### Liên hệ tam nguyên (Ternary Association)

Có thể có nhiều hơn hai lớp nối kết với nhau trong một liên hệ tam nguyên.



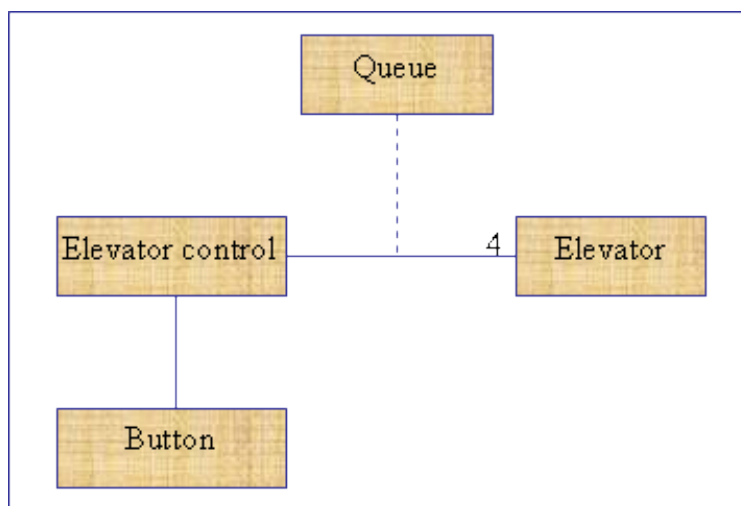
### Liên hệ Tam nguyên

Biểu đồ trên được đọc như sau: Một khách hàng có thể quan hệ với bộ phận đầu tư và một bộ phận đầu tư có thể có một hoặc nhiều khách hàng. Một giấy chứng nhận tài khoản đầu tư sẽ xuất hiện qua quan hệ giữa khách hàng và bộ phận đầu tư.

### 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.

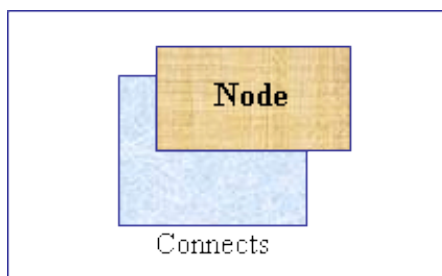
Ví dụ sau miêu tả một hệ thống thang máy. Bộ phận điều khiển chỉ huy bốn thang máy. Cho mỗi nối kết giữa nhóm thang máy và bộ phận điều khiển có một hàng xếp (queue). Mỗi hàng lưu trữ những yêu cầu kể cả từ phía bộ phận điều khiển lẫn từ phía thang máy (những nút bấm bên trong thang). Khi bộ phận điều khiển chọn một thang máy để thực hiện một lời yêu cầu đến từ một hành khách đứng ngoài thang máy (một hành khách trên hành lang), nó sẽ đọc các hàng và chọn thang máy nào có hàng yêu cầu ngắn nhất.



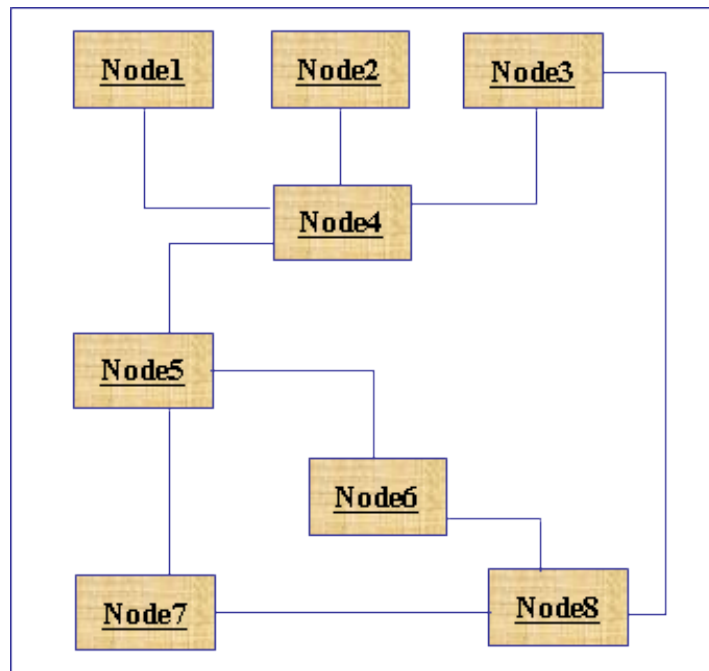
Lớp liên hệ (Association class)

### Liên hệ đệ quy (Recursive Association)

Có thể liên kết một lớp với bản thân nó trong một mối liên hệ. Mỗi liên hệ ở đây vẫn thể hiện một sự liên quan ngữ nghĩa, nhưng các đối tượng được nối kết đều thuộc chung một lớp. Một liên hệ của một lớp với chính bản thân nó được gọi là một liên hệ đệ quy, và là nền tảng cho rất nhiều mô hình phức tạp, sử dụng ví dụ để miêu tả các cấu trúc sản phẩm. Hình dưới 5.25 chỉ ra một ví dụ của liên hệ đệ quy và hình 5.26 là một biểu đồ đối tượng cho biểu đồ lớp trong hình



Một mạng gồm nhiều nút nối với nhau.



Một biểu đồ đối tượng của hình 5.25, với tên của các đối tượng.