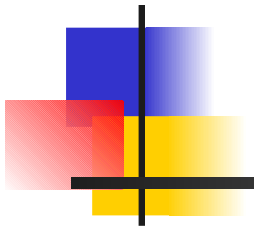


Bài 3

Mô hình hoá cấu trúc

1. Đối tượng và lớp
2. Bước 3: Mô hình hoá lĩnh vực ứng dụng
3. Bước4: Xác định các đối tượng và lớp tham gia các ca sử dụng

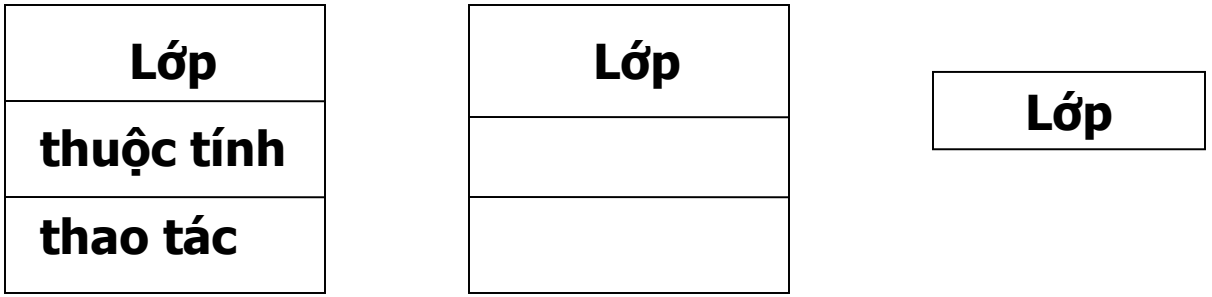
1. Đối tượng và lớp

1.1. Định nghĩa và biểu diễn của đối tượng và lớp

- ***Đối tượng*** **(tin học) là một biểu diễn trừu tượng của một thực thể (vật lý hay khái niệm) có căn cước và ranh giới rõ ràng trong thế giới thực, cho phép thâu tóm cả trạng thái và hành vi của thực thể đó, nhằm mục đích mô phỏng hay điều khiển thực thể đó.**
  - ***Trạng thái*** **của đối tượng thể hiện bởi một tập hợp các *thuộc tính*. ở mỗi thời điểm, mỗi thuộc tính của đối tượng có một giá trị nhất định.**
  - ***Hành vi*** **của đối tượng thể hiện bằng một tập hợp các *thao tác*, đó là các dịch vụ mà nó có thể thực hiện khi được một đối tượng khác yêu cầu.**
  - ***Căn cước*** **của đối tượng là cái để phân biệt nó với đối tượng khác.**
- ***Lớp*** **là một mô tả của một tập hợp các đối tượng cùng có chung các thuộc tính, các thao tác, các mối liên quan, các ràng buộc và ngữ nghĩa. Vậy lớp là một kiểu, và mỗi đối tượng thuộc lớp là một *cá thể*** **(instance).**

1. Đối tượng và lớp

- Biểu diễn lớp



- Biểu diễn đối tượng

1. Đối tượng và lớp

1.2 Các thuộc tính

- *Thuộc tính* là một tính chất có đặt tên của một lớp và nó nhận một giá trị cho mỗi đối tượng thuộc lớp đó tại mỗi thời điểm.

Cú pháp của thuộc tính:

- [tầm nhìn] [/] tên [: Kiểu] [cơ số] [= giá trị đầu][{xâu tính chất}]

- *Tầm nhìn* (visibility) cho biết thuộc tính đó được thấy và dùng từ các lớp khác như thế nào. Tầm nhìn có thể là:
  - *Công cộng* (public), ký hiệu bởi dấu '+', nếu thuộc tính đó là thấy và dùng được cả bên ngoài lớp;
  - *Riêng tư* (private), ký hiệu bởi dấu '-', nếu thuộc tính không thể truy cập từ các lớp khác;
  - *Bảo hộ* (protected), ký hiệu bởi dấu '#', nếu thuộc tính có thể truy cập từ các lớp thừa kế (xem mục 5).
  - *Gói* (package), ký hiệu bởi dấu '~', nếu thuộc tính có thể truy cập từ các phần tử thuộc cùng một gói (hẹp nhất) với lớp.

1. Đối tượng và lớp

- **Cú pháp của thuộc tính:**
- **[tầm nhìn] [/] tên [: Kiểu] [cơ số] [= giá trị đầu][{xâu tính chất}]**

  - ***Cơ số*** **(multiplicity), trỏ số các giá trị có thể nhận, chẳng hạn [0..1] để trỏ thuộc tính này là tuỳ chọn (không nhận giá trị nào, hay có nhận một giá trị).**
  - ***Kiểu*** **(type): đó là kiểu của các giá trị thuộc tính, thông thường thì đó là các kiểu nguyên thuỷ như Integer, Real, Boolean, nhưng cũng có thể là các kiểu cấu trúc như Point, Area, Enumeration, kể cả kiểu đó là một lớp khác.**
  - ***Giá trị đầu*** **(initial value) là giá trị ngầm định gán cho thuộc tính khi một đối tượng được tạo lập từ lớp đó.**
  - ***Xâu tính chất*** **(property-string) để trỏ các giá trị có thể gán cho thuộc tính, thường dùng đối với một kiểu liệt kê, chẳng hạn:**
  - **tình trạng: Tình trạng = chưa trả {chưa trả, đã trả}**

1. Đối tượng và lớp

- **1.2 Các thuộc tính** (tiếp theo)
- Cú pháp của thuộc tính:
- Ngoài ra, mỗi thuộc tính lại có thể có phạm vi lớp (class-scope) nếu nó phản ánh đặc điểm của lớp chứ không phải của riêng đối tượng nào. Chẳng hạn thuộc tính 'số các hoá đơn' trong lớp Hoá đơn. Thuộc tính thuộc phạm vi lớp phải được gạch dưới.
- Một thuộc tính là dẫn xuất, nếu giá trị của nó được tính từ giá trị của những thuộc tính khác của lớp. Thuộc tính dẫn xuất phải mang thêm dấu gạch chéo '/' ở đầu, chẳng hạn /tuổi (khi đã có ngày sinh).

1. Đối tượng và lớp

- **1.3. Các thao tác**
- ***Thao tác*** **là một dịch vụ mà đối tượng có thể đáp ứng được khi được yêu cầu (thông qua một thông điệp). Các thao tác được cài đặt thành các phương thức.**
- **Cú pháp đầy đủ của một thao tác là như sau:**
- **[tầm nhìn] tên[(danh sách tham số)][: Kiểu trả lại][{xâu tính chất}]**
- ***Tầm nhìn*** **hoàn toàn giống tầm nhìn của thuộc tính.**
- ***Danh sách tham số*** **là một danh sách gồm một số các tham số hình thức, cách nhau bằng dấu phẩy, mỗi tham số có dạng:**
- **[hướng] tên : Kiểu [= giá trị ngầm định]**
  - ***Hướng*** **có thể lấy các giá trị in, out, inout và return tuỳ thuộc tham số là input - không thể điều chỉnh, output - có thể điều chỉnh để đưa thông tin cho bên gọi, hoặc là input có thể điều chỉnh được, hay là để trả lại kết quả cho bên gọi.**
  - ***Giá trị ngầm định*** **là giá trị được sử dụng khi trong lời gọi khuyết tham số thực tương ứng.**
- ***Xâu tính chất*** **bao gồm các tiền đề, hậu đề, các tác động lên trạng thái đối tượng...**

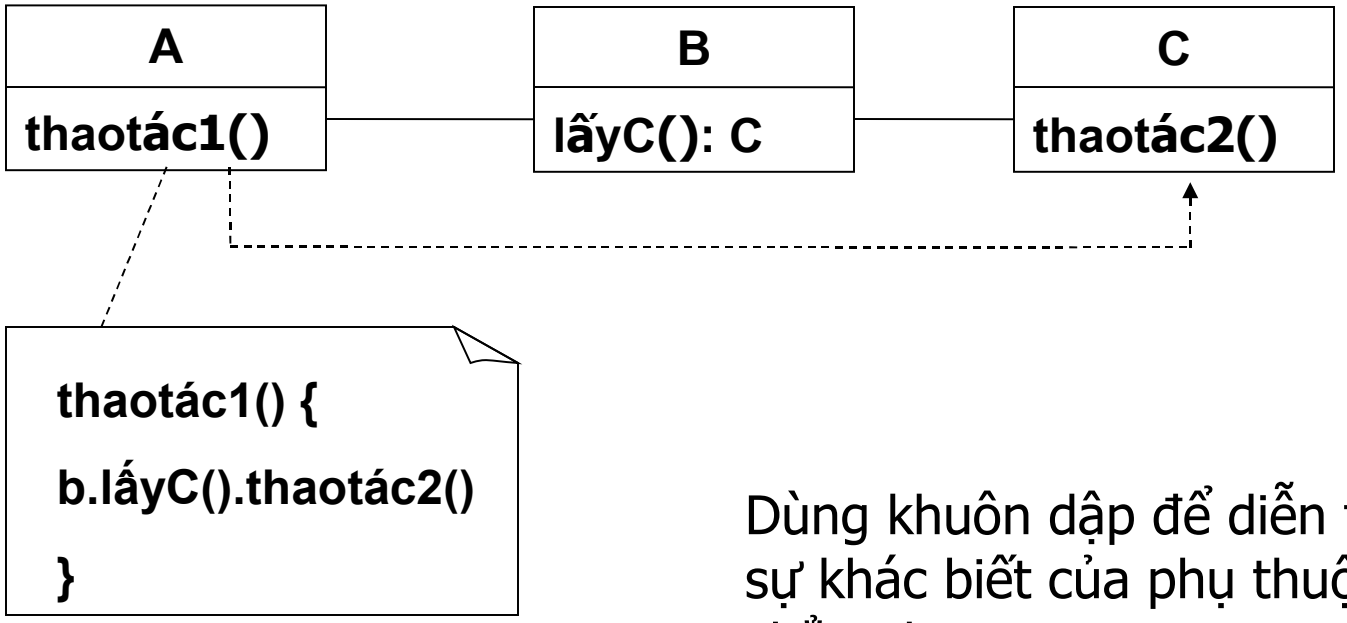
1. Đối tượng và lớp

- **1.4. Mối liên quan phụ thuộc**
- **Giữa các lớp có thể có ba mối liên quan:**
  - **mối liên quan phụ thuộc,**
  - **mối liên quan khái quát hóa,**
  - **mối liên quan liên kết.**

- ***Mối liên quan phụ thuộc*** **(dependency relationship) thường dùng để diễn đạt một lớp (bên phụ thuộc) chịu ảnh hưởng của mọi thay đổi trong một lớp khác (bên độc lập), mà ngược lại thì không nhất thiết. Thường thì bên phụ thuộc cần dùng bên độc lập để đặc tả hay cài đặt cho mình. UML biểu diễn mối liên quan phụ thuộc bằng một mũi tên đứt nét (từ bên phụ thuộc sang bên độc lập).**

1. Đối tượng và lớp

1.4. Mối liên quan phụ thuộc (tiếp)



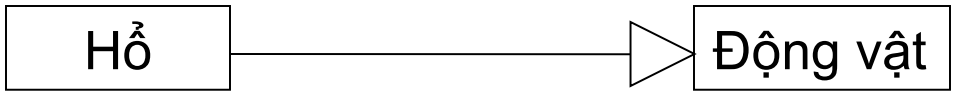
Dùng khuôn dập để diễn tả sự khác biết của phụ thuộc, chẳng hạn: <<use>>, <permit>>, <<refine>>

1. Đối tượng và lớp

- **1.5. Mối liên quan khái quát hoá**
- **Khái quát hoá (generalization) là sự rút ra các đặc điểm chung của nhiều lớp để tạo thành một lớp giản lược hơn gọi là *lớp trên* (hay cha).**
- **Quá trình ngược lại gọi là chuyên biệt hoá (specialization): từ một lớp đã cho ta tăng cường thêm một số đặc điểm mới, tạo thành một lớp chuyên hơn, gọi là *lớp dưới* (hay con).**
- **Một lớp có thể không có cha, có một hay nhiều cha. Một lớp không có cha và có một hay nhiều con gọi là lớp gốc hay lớp cơ sở. Một lớp chỉ có một cha gọi là lớp thừa kế đơn (simple inheritance). Một lớp có nhiều cha được gọi là lớp thừa kế bội (multiple inheritance).**

1. Đối tượng và lớp

- **1.5. Mối liên quan khái quát hoá (tiếp)**
- **Thuật ngữ thừa kế vốn được dùng nhiều trong các ngôn ngữ lập trình, nhằm diễn tả lớp con có mọi thuộc tính, thao tác và liên kết (sẽ nói ở dưới) được mô tả ở lớp cha. Hơn nữa lớp dưới có thể thêm thuộc tính, thao tác và liên kết mới và lại còn có thể định nghĩa lại (đè lấp) một thao tác của lớp trên (gọi đó là sự *đa hình* hay *đa xạ* (polymorphism)).**
- **Biểu diễn khái quát hoá:**



1. Đối tượng và lớp

1.5. Mối liên quan khái quát hoá (tiếp)

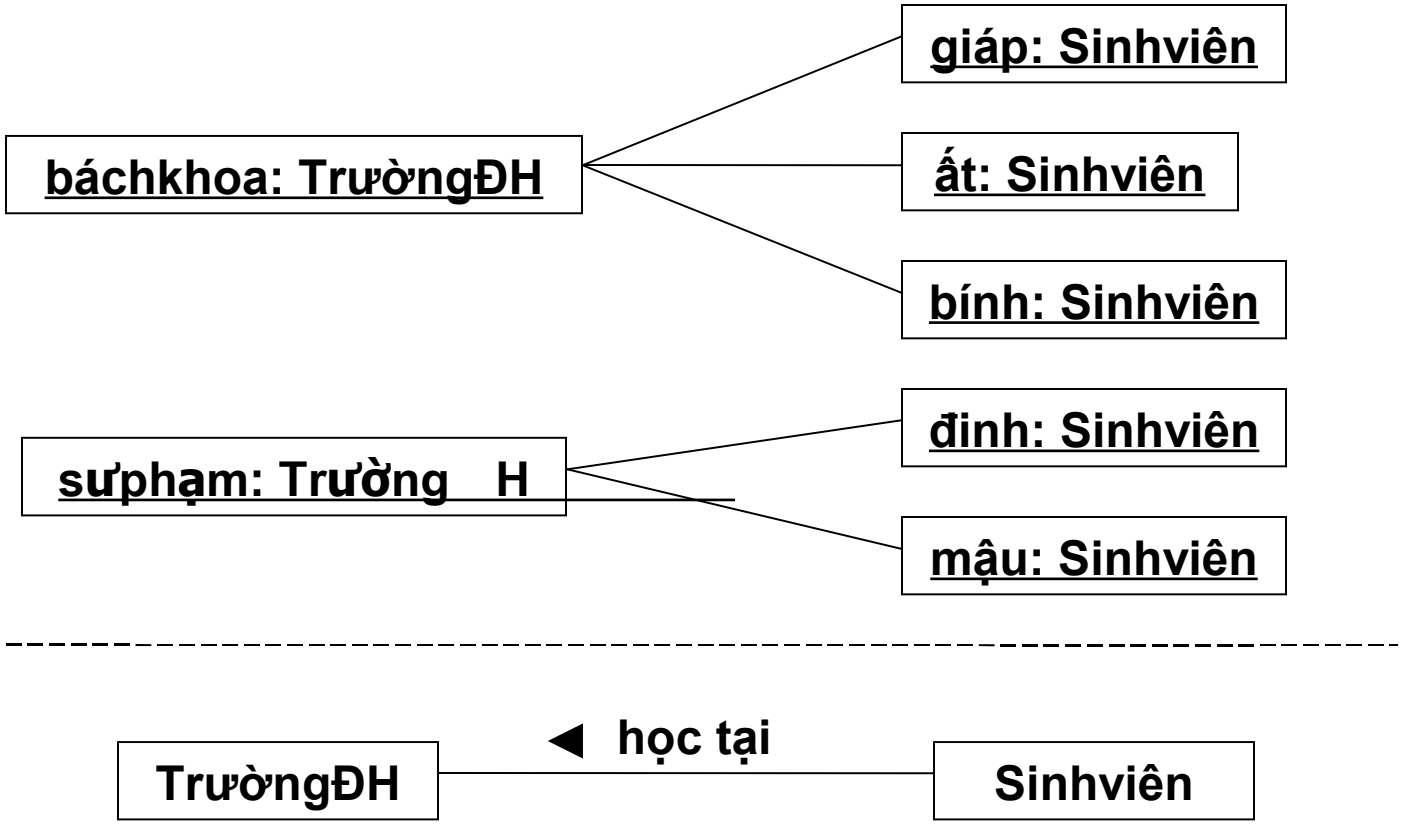
- Thông qua sự khái quát hoá ta có thể làm xuất hiện các lớp trừu tượng, nghĩa là các lớp không có cá thể, mà chỉ dùng cho việc mô tả các đặc điểm chung của những lớp dưới mà thôi. Một lớp trừu tượng thường có chứa các thao tác trừu tượng, là các thao tác chỉ có tiêu đề mà không có cài đặt. Thao tác trừu tượng phải được định nghĩa lại và kèm cài đặt ở các lớp dưới. Tên của lớp trừu tượng và tiêu đề của thao tác trừu tượng phải được viết xiên và có thể kèm thêm xâu tính chất {abstract}.
- Thí dụ: *Động vật* là lớp trừu tượng được khái quát hoá từ các lớp Ngựa, Hổ, Dơi. Nó có một thao tác trừu tượng là *ngủ*(). Thao tác trừu tượng này sẽ được cụ thể hoá trong các lớp chuyên biệt là Ngựa, Hổ, Dơi theo các cách thức khác nhau, nhưng vẫn giữ nguyên tên là ngủ() (ngủ đứng, ngủ nằm và ngủ treo).

1. Đối tượng và lớp

- **1.6. Liên kết**
- **Kết nối và liên kết**
  - **Giữa các cá thể của hai lớp có thể tồn tại những sự ghép cặp, phản ánh một mối liên hệ nào đó trên thực tế. Gọi đó là một kết nối (link). Chẳng hạn kết nối vợ - chồng, kết nối thầy - trò, kết nối xe máy - chủ xe, kết nối khách hàng - hóa đơn v.v...**
  - **Tập hợp những kết nối cùng loại (cùng ý nghĩa) giữa cá thể của hai lớp tạo thành một mối liên quan giữa hai lớp đó, gọi là một liên kết (association). Theo nghĩa đó thì đây chính là một quan hệ (hiểu theo nghĩa toán học) giữa hai tập hợp (là hai lớp).**

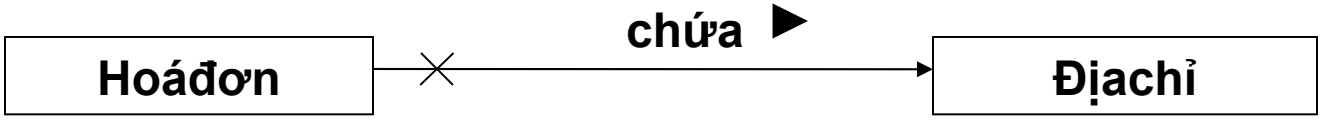
1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Biểu diễn kết nối và liên kết**



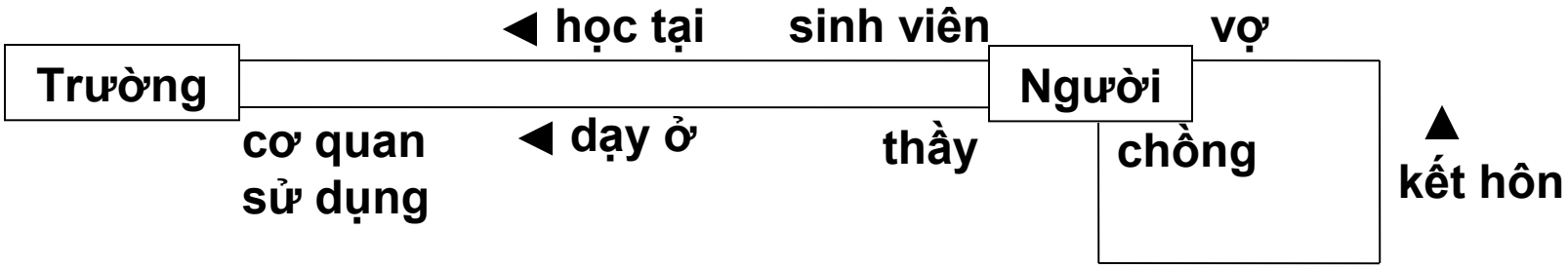
1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Sự lưu hành**
  - **Sự tồn tại một kết nối giữa hai đối tượng (của hai lớp trong một liên kết) có nghĩa là các đối tượng đó "biết nhau". Do đó từ một đối tượng này, nhờ có kết nối mà ta có thể tìm đến được đối tượng kia. Gọi đó là sự *lưu hành* (navigation) trên một liên kết.**
  - **Nói chung thì sự lưu hành có thể thực hiện theo cả hai chiều (tức là cả ở hai đầu) trên một liên kết. Song cũng có khi sự lưu hành trên liên kết chỉ hạn chế theo một chiều. Bấy giờ ta thêm một mũi tên vào đầu được lưu hành và dấu chéo vào đầu không được lưu hành. Khi không có mũi tên và dấu chéo thì ta xem là sự lưu hành không được chỉ rõ.**



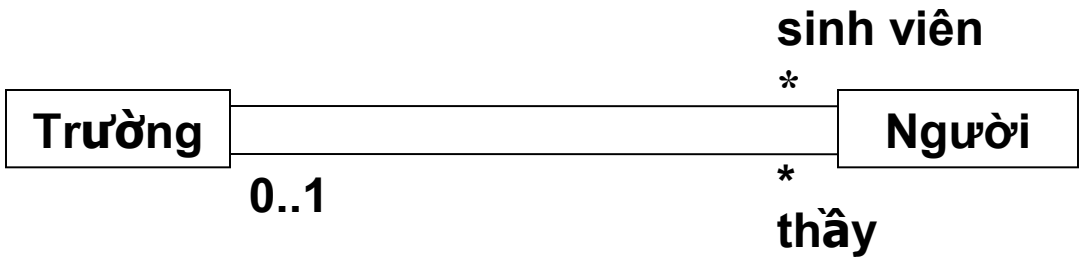
1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Vai trò**
  - **Trong một liên kết giữa hai lớp, thì mỗi lớp đóng một *vai trò* (role) khác nhau.**
  - **Tên vai trò, với chữ cái đầu tiên viết thường, có thể được viết thêm vào mỗi đầu của liên kết (vì vậy mà vai trò cũng được gọi là *tên của một đầu liên kết*).**
  - **Về ý nghĩa, thì một vai trò biểu diễn cho một tập con các đối tượng của lớp tương ứng.**



1. Đối tượng và lớp

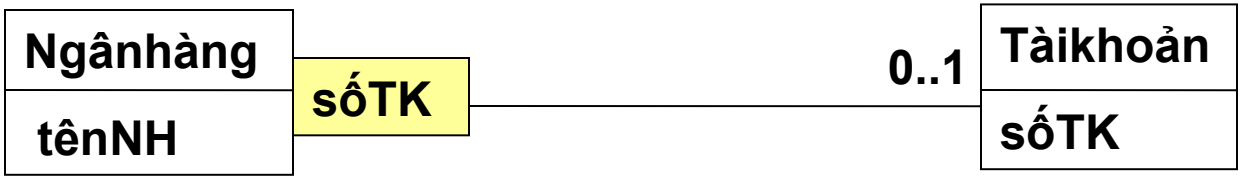
- **1.6. Liên kết (tiếp)**
- **Cơ số**
- **Mỗi đầu của liên kết còn có thể chua thêm một *cơ số* (multiplicity), cho biết số cá thể (tối thiểu và tối đa) của đầu đó tham gia liên kết với một cá thể ở đầu kia. Các giá trị của cơ số thường dùng là:**
- **1 một và chỉ một**
- **0..1 không hay một**
- **m..n từ m tới n (m và n là các số tự nhiên)**
- **0..* hay * từ 0 tới nhiều**
- **1..* một tới nhiều**



1. Đối tượng và lớp

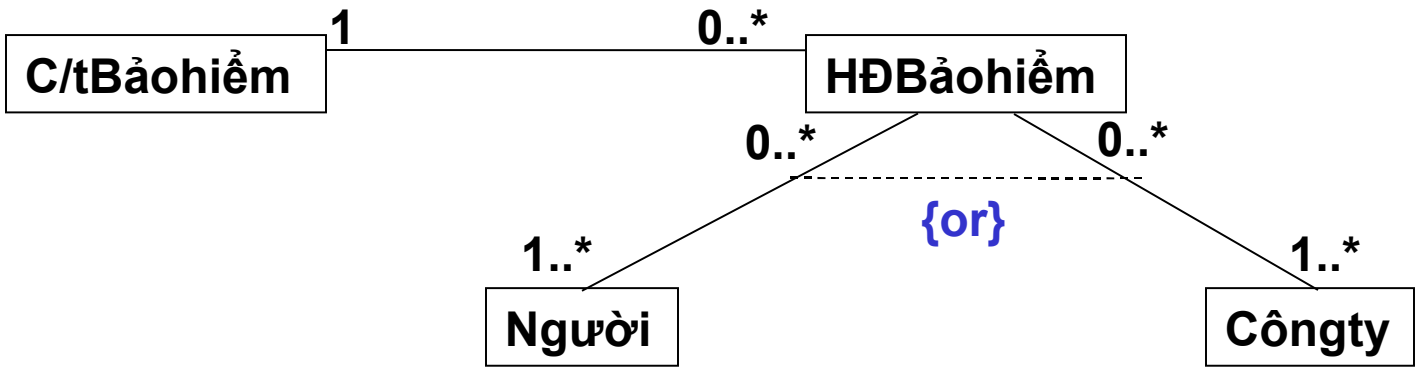
1.6. Liên kết (tiếp)

- Hạn định
  - Vấn đề luôn luôn cần giải quyết khi mô hình hóa các liên kết, đó là vấn đề **tìm kiếm**: cho trước một đối tượng ở một đầu của liên kết, hãy tìm một đối tượng hay tập hợp các đối tượng kết nối với nó ở đầu kia.
  - Để giảm bớt số lượng các đối tượng tìm được, ta có thể hạn chế khu vực tìm kiếm theo (giá trị của) một thuộc tính nào đó. Thuộc tính này (hay cũng có thể là một số thuộc tính) được gọi là một **hạn định** (qualifier) được ghi trong một hộp nhỏ gắn vào đầu mút của liên kết, phía lớp xuất phát của sự lưu hành.
  - Như vậy hạn định được áp dụng cho các liên kết 1-nhiều hay nhiều-nhiều để giảm từ nhiều xuống 1 hay 0..1, hoặc cũng có thể giảm từ nhiều xuống nhiều, nhưng ít hơn.



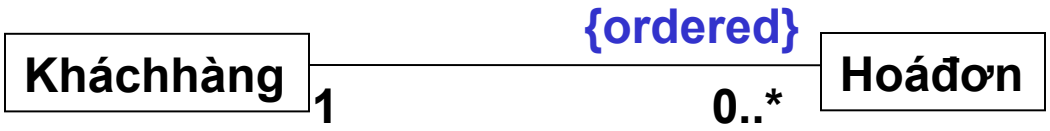
1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Liên kết Hoặc**
- **Trong một số mô hình không phải mọi tổ hợp các kết nối của các liên kết đều đúng đắn. Chẳng hạn một cá nhân và một công ty không được cùng kết nối với một hợp đồng bảo hiểm. Bấy giờ ta dùng *liên kết hoặc* (or-association) để diễn tả, thể hiện bằng một đường đứt nét với xâu tính chất {or} vẽ đi ngang qua các liên kết, với nghĩa là: một đối tượng (ở đây là hợp đồng bảo hiểm) chỉ được phép tham gia nhiều nhất vào một trong những liên kết được nối, ở một thời điểm.**



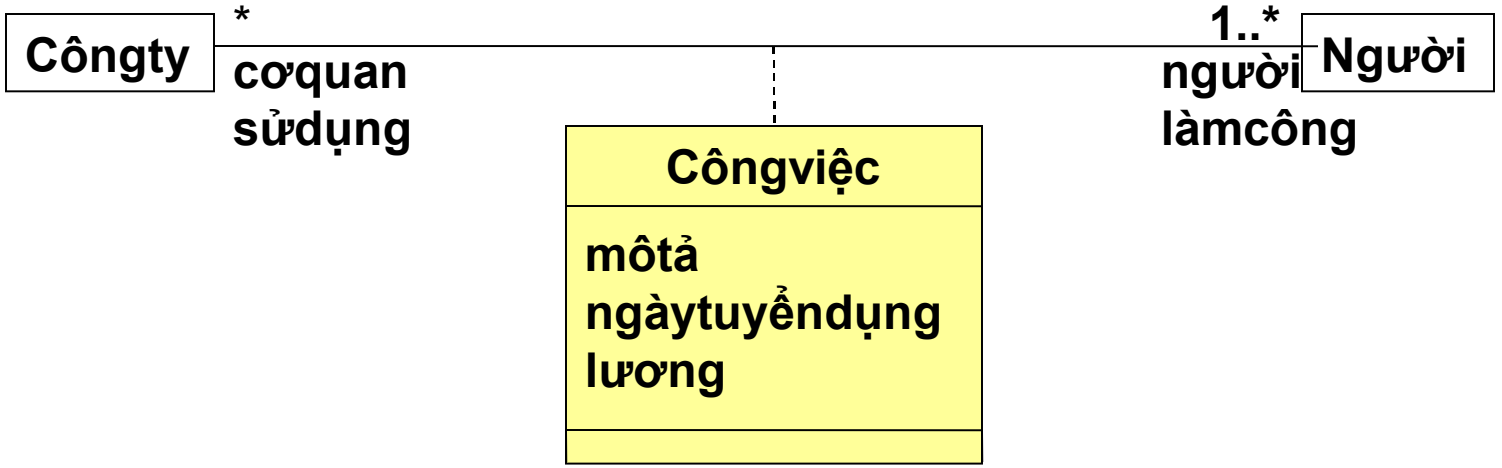
1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Liên kết có thứ tự**
- **Một cách mặc định, thì các kết nối (với một đối tượng) trong cùng một liên kết là không có thứ tự. Song cũng có khi ta muốn các kết nối được sắp theo một thứ tự nào đó. Gọi đó là một *liên kết có thứ tự* (ordered association). Bấy giờ ta gắn thêm xâu tính chất {ordered} bên cạnh liên kết về phía lớp các đối tượng được sắp. Còn muốn rõ sắp như thế nào, thì có thể thêm một xâu tính chất nữa về cách sắp, chẳng hạn {sắp theo thứ tự thời gian}.**



1. Đối tượng và lớp

- **1.6. Liên kết (tiếp)**
- **Lớp liên kết**
- **Bản thân các liên kết cũng có thể cần có các tính chất đặc trưng cho nó. UML thể hiện điều này bằng các *lớp liên kết* (association class). Lớp liên kết là một lớp như bình thường, nghĩa là có thể có các thuộc tính, các thao tác và tham gia liên kết với những lớp khác, song ngăn tên có thể có tên hay để trống tùy ý, và nó được gắn với liên kết mô tả bởi một đường đứt nét**

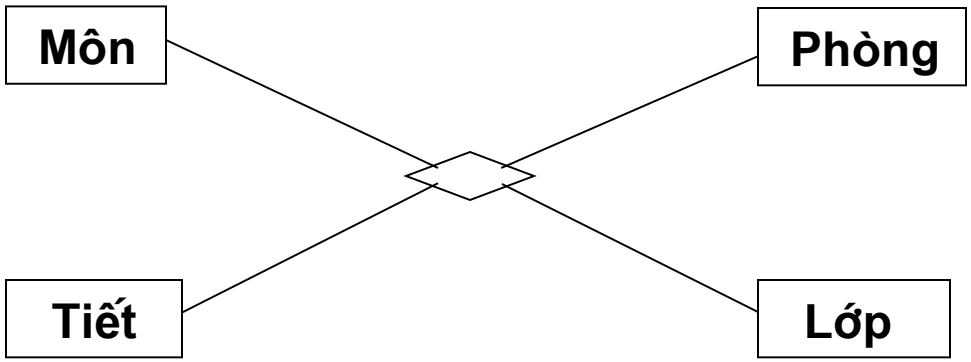


1. Đối tượng và lớp

1.6. Liên kết (tiếp)

Liên kết nhiều bên

Có thể có liên kết giữa nhiều hơn hai lớp, theo nghĩa của quan hệ nhiều ngôi (một kết nối ở đây là một bộ - n). Lúc đó liên kết được diễn tả bởi một hình thoi nhỏ, nối bằng các đường liền nét với các lớp tham gia và cũng có thể có một lớp liên kết cho nó.

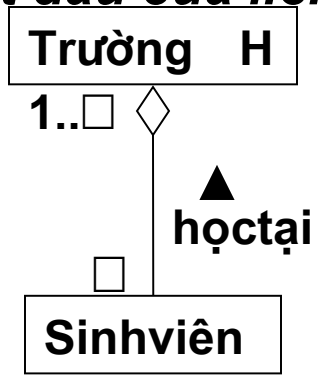


1. Đối tượng và lớp

1.6. Liên kết (tiếp)

- **Kết nhập**

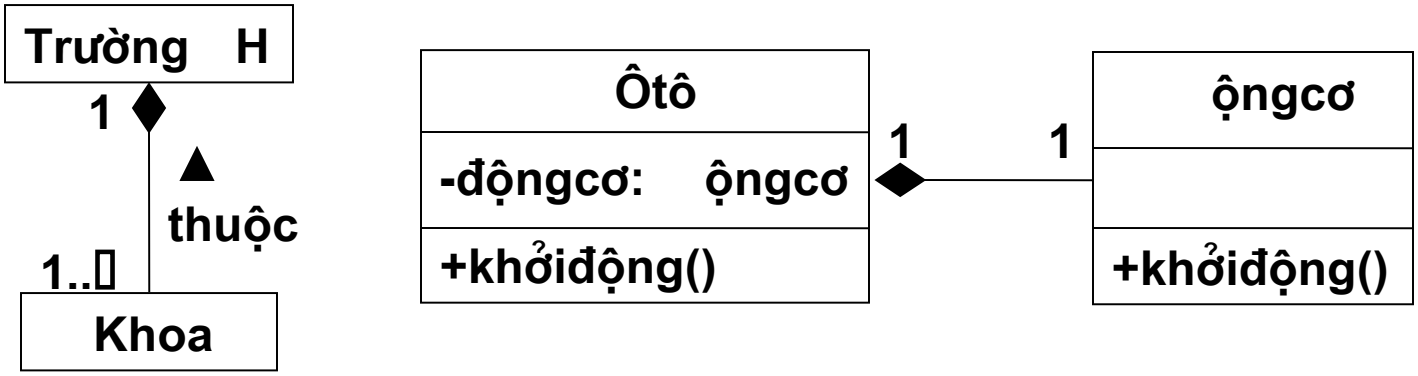
  Trong một liên kết, hai bên tham gia được xem là bình đẳng, không bên nào được nhấn mạnh hơn bên nào. Tuy nhiên cũng có lúc ta muốn mô hình hóa mối quan hệ "toàn thể/bộ phận" giữa một lớp các vật thể lớn (cái "toàn thể") với một lớp các vật thể bé (các "bộ phận") bao gồm trong chúng. Đó là một loại liên kết đặc biệt, được gọi là *kết nhập (aggregation) và được biểu diễn bằng cách gắn thêm một hình thoi nhỏ vào một đầu của liên kết, phía cái toàn thể*



1. Đối tượng và lớp

1.6. Liên kết (tiếp)

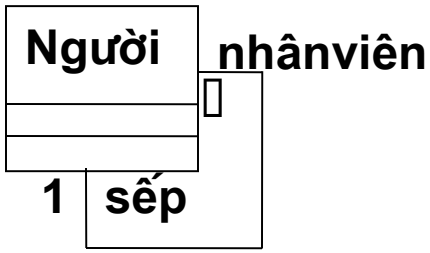
- **Hợp thành**
- **Một *hợp thành (composition) là một loại kết nhập đặc biệt với quan hệ sở hữu mạnh hơn, trong đó một bộ phận chỉ thuộc vào một cái toàn thể duy nhất và cái toàn thể có trách nhiệm tạo lập và hủy bỏ cái bộ phận. Như vậy khi cái toàn thể bị hủy bỏ thì cái bộ phận cũng buộc phải hủy bỏ theo. Hợp thành được biểu diễn bằng cách thay hình thoi rỗng trong kết nhập bởi hình thoi đặc***

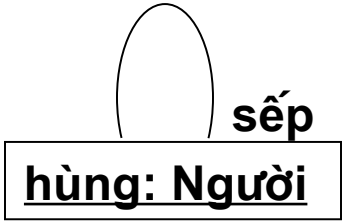


1. Đối tượng và lớp

1.6. Liên kết (tiếp)

- Biểu đồ lớp và biểu đồ đối tượng
- Biểu đồ lớp là một biểu đồ phô bày một tập hợp các lớp, các giao diện cùng với các mối liên quan có thể có giữa chúng, như là liên kết, kết nhập, hợp thành, khái quát hoá, phụ thuộc và thực hiện.
- Biểu đồ đối tượng diễn tả lại cấu trúc tĩnh cho trong biểu đồ lớp, song một cách cụ thể: các đối tượng thay cho các lớp, các kết nối thay cho các liên kết.
- Biểu đồ lớp được dùng để mô hình hoá cấu trúc tĩnh và nó bao gồm mọi phần tử khai báo, cho nên nó là cái nền (cái nâng đỡ) cho các hoạt động chức năng của hệ thống.





2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.1. Mục đích

Xuất phát từ các khái niệm về các sự vật trong lĩnh vực ứng dụng, ta trừu tượng hoá chúng thành các lớp gọi là các lớp lĩnh vực, hay lớp nghề nghiệp. Các lớp này thường chỉ dùng để phản ánh và mô phỏng các sự vật trong thế giới thực, cho nên trách nhiệm của chúng cũng thường chỉ là lưu gi€ v ƒấcung cấp các thông tin về các sự vật đó.

2.2. Trình tự tiến hành

Bước 3 được tiến hành qua ba bước nhỏ sau:

- Nhận định các khái niệm của lĩnh vực.
- Thêm các liên kết và cthuộc tính.
- Khái quát hoá các khái niệm.

2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.3. Nhận định các khái niệm của lĩnh vực

Nguồn tìm kiếm

Các khái niệm của lĩnh vực là những khái niệm về các sự vật (cụ thể hay trừu tượng) mà các người dùng, các chuyên gia nghiệp vụ sử dụng khi nói đến nghề nghiệp và công việc của mình. Bởi vậy để tìm kiếm các khái niệm này, ta dựa vào:

- Các kiến thức về lĩnh vực nghiệp vụ.
- Các cuộc phỏng vấn trao đổi với các người dùng và chuyên gia.
- Bản tổng quan về hệ thống và nhu cầu.
- Các tài liệu miêu tả các ca sử dụng đã lập ở bước trước.

2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.3. Nhận định các khái niệm của lĩnh vực

Cách nắm bắt các khái niệm

Cứ đọc văn bản miêu tả hệ thống (tức bản phát biểu nhu cầu):

- các danh từ sẽ là đối tượng hay thuộc tính,
- các động từ có thể là các thao tác.

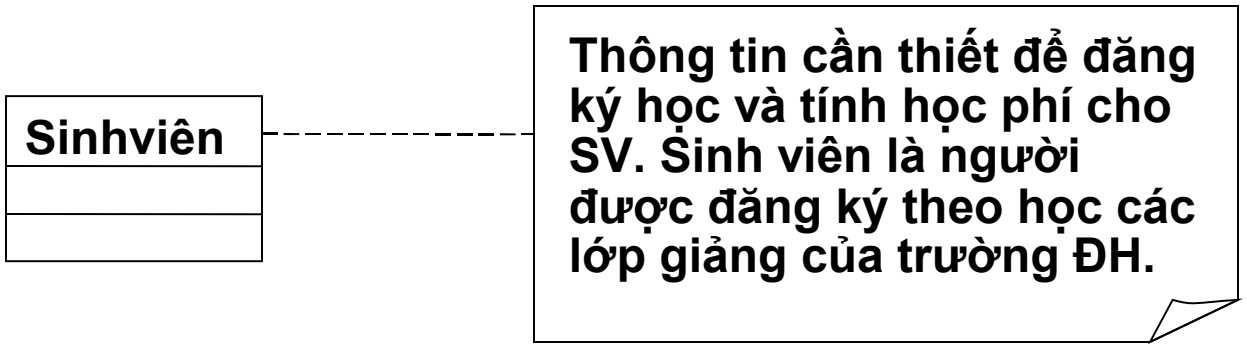
Từ đó đề xuất các đối tượng theo các thể loại:

- các *thực thể vật chất, như xe đạp, máy bay, cảm biến...*
- các *vai trò như mẹ, giáo viên, cảnh sát...*
- các *sự kiện như hạ cánh, ngắt, đăng ký xe máy...*
- các *tương tác như cho vay, hội thảo...*
- các *tổ chức như công ty, khoa, lớp,...*

v.v...

2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

- **2.3. Nhận định các khái niệm của lĩnh vực**
- **Đặt tên và gán trách nhiệm**
- Tiếp đến là đặt tên và gán trách nhiệm cho mỗi lớp vừa mới thành lập.
- Trách nhiệm mô tả nghĩa vụ và mục đích của lớp, chứ không phải là cấu trúc của nó, dù rằng sau này trách nhiệm sẽ cho phép ta định ra cấu trúc (thuộc tính và liên kết) cùng với hành vi (các thao tác) của lớp.



2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.3. Nhận định các khái niệm của lĩnh vực

Đặt tên và gán trách nhiệm

Việc gán tên và trách nhiệm cho một lớp đề cử cũng sẽ cho ta hay là việc chọn lựa lớp là có hợp lý không:

- Nếu chọn được tên và gán được trách nhiệm rõ ràng, chặt chẽ thì lớp đề cử là tốt.
- Nếu chọn được tên, song trách nhiệm lại giống trách nhiệm của một lớp khác, thì nên gộp hai lớp đó làm một.
- Nếu chọn được tên, song trách nhiệm lại quá dông dài, thì nên tách nó ra nhiều lớp.
- Khó chọn được tên hợp lý hay khó mô tả trách nhiệm, thì nên phân tích sâu thêm vào nó để chọn những biểu diễn thích hợp.

2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.4. Thêm các liên kết và các thuộc tính

Nhiều thuộc tính và liên kết của các lớp lĩnh vực đã có thể phát hiện trực tiếp

- **từ bản miêu tả hệ thống và nhu cầu,**
- **từ ý kiến của các chuyên gia lĩnh vực và người dùng và**
- **từ các trách nhiệm của các lớp mà ta vừa gán ở trên.**

Đương nhiên như thế là chưa đủ, sau này sẽ bổ sung dần các liên kết và các thuộc tính, cũng như bổ sung các thao tác cho các lớp, khi ta nghiên cứu sâu vào hành vi (tương tác và ứng xử) của hệ thống.

Chẳng hạn, từ văn bản trách nhiệm của lớp Sinh viên, ta có:

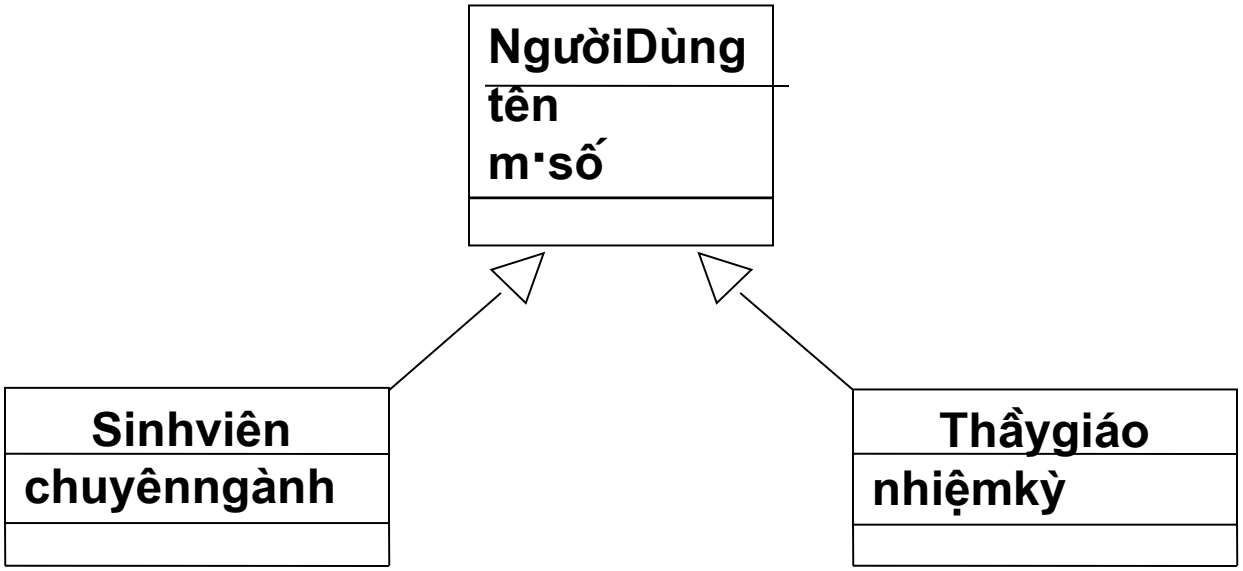
- **Câu "Thông tin cần thiết để đăng ký học và tính học phí" cho ta suy ra một số thuộc tính như tên, mã SV, địachỉ ... cho lớp Sinhviên.**
- **Câu "Sinh viên là người được đăng ký theo học các lớp giảng của trường ĐH" cho ta suy ra là có liên kết giữa lớp Sinhviên và Lớpgiảng, với tên liên kết có thể là "đăng ký".**

2. Bước 3: Mô hình hoá lĩnh vực ứng dụng

2.5. Khái quát hoá các lớp

- Để cho mô hình thêm tốt, ta tìm cách rút ra các phần chung giữa các lớp để lập thành lớp khái quát.

Chẳng hạn trong hệ ĐKMH (thí dụ xuyên suốt) thì Sinh viên và Thầy giáo có những thuộc tính chung, như tên, mã số, có thể rút ra để lập một lớp khái quát là NgườiDùng.



3. Bước 4: Xác định các lớp tham gia ca SD

3.1. Mục đích

- Trong bước 3 (bước mô hình hoá lĩnh vực) ta đã tiến hành nghiên cứu lĩnh vực, mà không xem xét tới ứng dụng. Các lớp được phát hiện ở bước 3 chỉ là các lớp lĩnh vực.
- Đặc thù của mỗi ứng dụng nằm ở các ca sử dụng. Để nghiên cứu ứng dụng, ta phải đi sâu nghiên cứu cấu trúc và hành vi của các ca sử dụng.
- Ca sử dụng được hình dung như là một hợp tác của một số đối tượng, trong đó có một số lớp lĩnh vực và còn thêm một số lớp phù trợ khác nữa.
- Mục đich của bước 4 này chính là phân tích cấu trúc tĩnh của các ca sử dụng. Ta sẽ phải lần lượt thực hiện các việc sau:
    1. Phát hiện các đối tượng tham gia ca sử dụng.
    2. Xác định vai trò của mỗi đối tượng trong ca sử dụng.
    3. Nghiên cứu các liên kết giữa các lớp tham gia ca sử dụng.

3. Bước 4: Xác định các lớp tham gia ca SD

3.2. Phát hiện các đối tượng/lớp tham gia ca SD

- Các ca sử dụng được đem ra nghiên cứu để phát hiện các đối tượng/lớp tham gia từng ca sử dụng đó. Các lớp tham gia ca sử dụng được gọi chung là các lớp phân tích, gồm 3 loại: lớp biên, lớp điều khiển, lớp lĩnh vực.
- Các lớp biên *(boundary) hay lớp đối thoại (dialog):*
  - **Là các lớp nhằm chuyển đổi thông tin giao tiếp giữa các đối tác và hệ thống. Điển hình là các màn hình giao lưu với các người dùng, cho phép thu thập thông tin hay xuất các kết quả.**
  - **Cũng có thể là các giao diện (cứng và mềm) chuyển đổi tương tự/số giữa hệ thống và các thiết bị mà nó điều khiển hay thu thập thông tin.**
  - **Cứ mỗi cặp đối tác - ca sử dụng thì phải có ít nhất một lớp biên. Lớp biên chính này lại có thể cần đến các lớp biên phù trợ để nó uỷ thác một phần nào đó trong các trách nhiệm quá lớn của nó.**
  - **Nói chung thì các đối tượng biên có đời sống kéo dài cùng với ca sử dụng liên quan.**

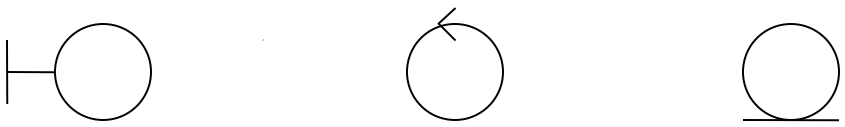
3. Bước 4: Xác định các lớp tham gia ca SD

3.2. Phát hiện các đối tượng/lớp tham gia ca SD

- **Các lớp điều khiển (control):**
  - **Là các lớp điều hành sự diễn biến trong một ca sử dụng; có thể nói đó là cái "động cơ" làm cho ca sử dụng chuyển vận được.**
  - **Các lớp này chứa các quy tắc nghiệp vụ và đứng trung gian giữa các lớp biên với các lớp thực thể, cho phép từ màn hình có thể thao tác được các thông tin chứa đựng trong các thực thể.**
  - **Cứ mỗi ca sử dụng ta lập ít nhất một lớp điều khiển.**
  - **Các lớp điều khiển khi bước qua giai đoạn thiết kế không nhất thiết là sẽ còn tồn tại như một lớp thực sự, vì nhiệm vụ của nó có thể bị phân tán vào các lớp khác, song trong giai đoạn phân tích, nhất thiết phải có chúng để bảo đảm không bỏ sót các chức năng hay hành vi các ca sử dụng.**

3. Bước 4: Xác định các lớp tham gia ca SD

- **3.2. Phát hiện các đối tượng/lớp tham gia ca SD**
- **Các lớp lĩnh vực *(domain) hay lớp thực thể (entity):***
- **Là các lớp nghiệp vụ, đến trực tiếp từ biểu đồ lĩnh vực, song chúng sẽ được khẳng định vị thế khi được xuất hiện trong các ca sử dụng.**
- **Nói chung thì đấy là các lớp trường cữu, nghĩa là các lớp mà các dữ liệu và các mối liên quan của chúng còn được lưu lại (trong cơ sở dữ liệu hay trên các tệp) sau khi ca sử dụng của chúng đã kết thúc.**
- **Lớp thực thể được chọn tham gia ca sử dụng khi thông tin chứa đựng trong nó là được đề cập đến trong ca sử dụng.**
- **Biểu diễn: <<boundary>> <<control>> <<entity>>**



Thí dụ: Lớp tham gia ca SD trong hệ ĐKMH

Xét ca sử dụng 'Chọn môn học để giảng dạy'

Từ kịch bản, ta phát hiện ra các lớp tham gia như sau:

- Các lớp thực thể, lấy từ biểu đồ lĩnh vực sang và có mặt ở đây bởi thông tin về chúng được đề cập trong kịch bản, là: Lớp giảng, Môn học, Thầy giáo.
- Các lớp biên gồm:
  - Lớp W_Thầy là màn hình chính giao tiếp với đối tác Thầy giáo.
  - Lớp W_Lớpgiảng và lớp W_Lịchgiảng là các màn hình phụ, dùng tương ứng cho các trường hợp Thêm/Bớt lớp giảng và Xem/In lịch giảng.
  - Lớp điều khiển chỉ cần có một, lấy tên là QLLớpThầy.

3. Bước 4: Xác định các lớp tham gia ca SD

3.3. Diễn tả cấu trúc tĩnh của ca sử dụng bằng một biểu đồ lớp

- **Mục đích cuối cùng của bước 4 là lập một biểu đồ lớp cho mỗi ca sử dụng, phản ánh cấu trúc tĩnh của sự hợp tác. Chính biểu đồ lớp này, mà ta thường gọi là biểu đồ các lớp tham gia ca sử dụng, sẽ là cái nền trên đó diễn ra các hoạt động tương tác giữa các lớp, mà ta sẽ đi sâu tìm hiểu trong bước 5.**
- **Biểu đồ các lớp tham gia một ca sử dụng phải bao gồm đủ các yếu tố cấu trúc (thuộc tính, thao tác, liên kết) cần thiết cho sự hợp tác giữa các lớp.**
  - **Các thuộc tính phải lưu giữ những thông tin về các đối tượng, cần dùng trong hợp tác.**
  - **Các thao tác cung cấp các khả năng dịch vụ cần thiết của mỗi lớp khi tham gia vào hợp tác.**
  - **Các liên kết tạo ra các cầu nối giữa các lớp, cho phép chúng biết nhau và trao đổi với nhau khi hợp tác.**

3. Bước 4: Xác định các lớp tham gia ca SD

3.3. Diễn tả cấu trúc tĩnh của ca sử dụng bằng một biểu đồ lớp

Trước hết bổ sung thuộc tính và thao tác:

- Các lớp thực thể tạm thời chỉ có các thuộc tính. Các thuộc tính này diễn tả các thông tin trường cữu của hệ thống.
- Các lớp điều khiển lại chỉ có các thao tác. Các thao tác này diễn tả lôgic của ứng dụng, các quy tắc nghiệp vụ, các hành vi của hệ thống tin học.
- Các lớp biên có cả thuộc tính và thao tác:
  - Các thuộc tính diễn tả các trường thu thập thông tin hay xuất kết quả. Các kết quả được phân biệt bằng ký hiệu thuộc tính dẫn xuất.
  - Các thao tác biểu diễn những hành động mà người dùng thực hiện trên màn hình giao diện.

3. Bước 4: Xác định các lớp tham gia ca SD

3.3. Diễn tả cấu trúc tĩnh của ca sử dụng bằng một biểu đồ lớp

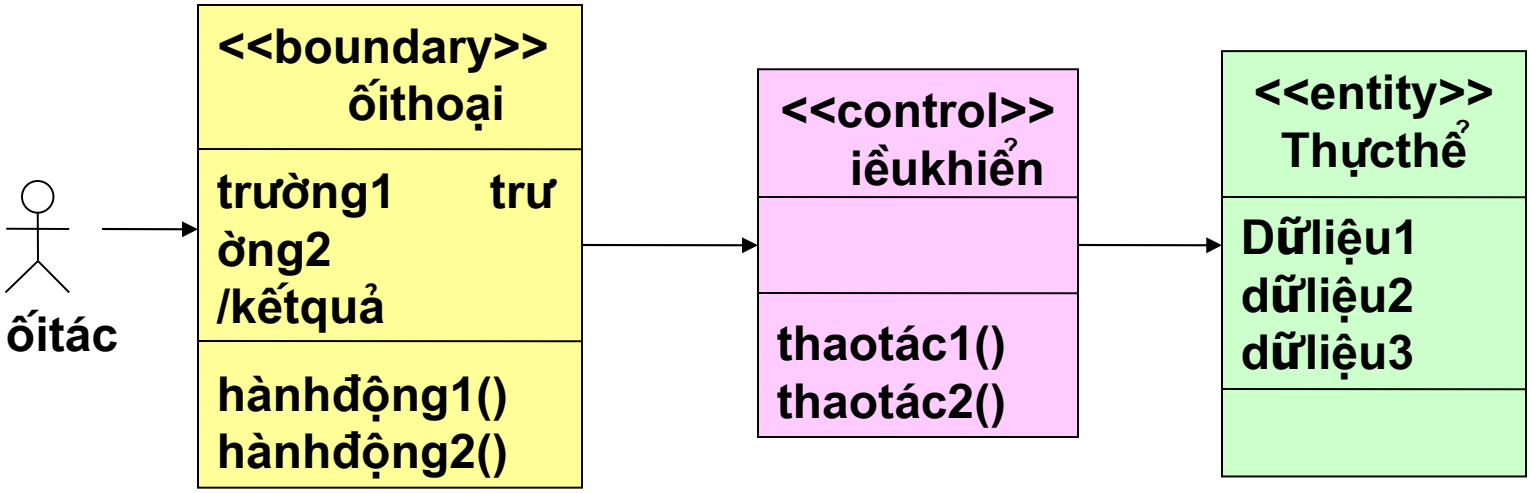
Sau đó bổ sung các liên kết:

- Các lớp biên chỉ được nối với các lớp điều khiển hay với các lớp biên khác. Nói chung th□ các liên kết là một chiều từ lớp biên đến lớp điều khiển, trừ khi lớp điều khiển lại tạo lập một đối thoại mới (chẳng hạn một trang các kết quả).
- Các lớp thực thể chỉ được nối với các lớp điều khiển hay lớp thực thể khác. Liên kết với các lớp điều khiển luôn luôn là một chiều (từ lớp điều khiển tới lớp thực thể).
- Các lớp điều khiển được phép truy cập tới mọi loại lớp, bao gồm các lớp điều khiển khác.

3. Bước 4: Xác định các lớp tham gia ca SD

3.3. Diễn tả cấu trúc tĩnh của ca sử dụng bằng một biểu đồ lớp

- **Cuối cùng thì ta thêm các đối tác vào biểu đồ: một đối tác chỉ được nối với một (hay một số) lớp biên.**



Thí dụ: BĐ lớp của 1 ca SD trong hệ ĐKMH

B các lớp tham gia cho ca sử dụng 'Chọn môn học để giảng dạy':