

# Phân Tích và Thiết Kế Hệ Thống Thông Tin (*IT3620*)

**Nguyễn Nhật Quang**

*quang.nguyennhat@hust.edu.vn*

---

Trường Đại học Bách Khoa Hà Nội  
Viện Công nghệ thông tin và truyền thông  
Năm học 2015-2016

# Nội dung học phần:

- Giới thiệu về Phân tích và thiết kế hệ thống hướng đối tượng
- Giới thiệu về Ngôn ngữ mô hình hóa UML
- Giới thiệu về Quy trình phát triển phần mềm
- Phân tích môi trường và nhu cầu
- Phân tích chức năng
- Phân tích cấu trúc
- **Phân tích hành vi**
- Thiết kế tổng thể
- Thiết kế đối tượng, lớp
- Thiết kế giao diện
- Thiết kế dữ liệu
- Mẫu thiết kế, các mô hình kiến trúc

# Phân tích hành vi

- MHH sự tương tác với biểu đồ trình tự
- MHH sự tương tác với biểu đồ giao tiếp
- Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
- Bài tập tổng hợp

# Khái niệm

- **Hành vi** (hay động thái) là sự hoạt động của các đối tượng nhằm tạo ra các kịch bản
- Hành vi bao gồm **tương tác** (trao đổi thông điệp) và **ứng xử** (phản ứng với các sự kiện)

# MHH sự tương tác với biểu đồ trình tự

- Mục đích MHH tương tác
- Các thông điệp
- Biểu đồ trình tự
- MHH tương tác trong ca sử dụng với biểu đồ trình tự

# Mục đích MHH tương tác (1)

- Mục đích của bước mô hình hoá tương tác là dùng các biểu đồ tương tác để diễn tả sự tương tác giữa các đối tượng nhằm tạo ra các kịch bản của mỗi ca sử dụng của hệ thống
- Hình thức tương tác duy nhất có thể có giữa các đối tượng là chuyển giao thông điệp và có hai biểu đồ chính được sử dụng để diễn tả sự tương tác (một cách tương đương với nhau) là **Biểu đồ trình tự** và **Biểu đồ giao tiếp**

# Mục đích MHH tương tác (2)

- Dù hai biểu đồ này là khác nhau về hình thức, song khi thành lập chúng, ta có thể áp dụng chung các nguyên tắc sau đây:
  - Các tác nhân chỉ có thể tương tác (gửi thông điệp) tới các đối tượng biên
  - Các đối tượng biên chỉ có thể tương tác tới các đối tượng điều khiển hay đối tượng biên khác
  - Các đối tượng điều khiển có thể tương tác tới các đối tượng biên, các đối tượng thực thể hay các đối tượng điều khiển khác
  - Các đối tượng thực thể chỉ có thể tương tác với các đối tượng thực thể mà thôi

# Các thông điệp (1)

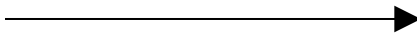
- **Thông điệp (message)** là một đặc tả cho sự giao lưu giữa hai đối tượng, bao gồm sự truyền đạt một số thông tin và/hoặc sự yêu cầu thực hiện một hoạt động nào đó thuộc khả năng của bên nhận
- Hành động tạo nên bởi một thông điệp có thể là các hành động sau:
  - **Gọi (Call):** Yêu cầu thực hiện một thao tác của đối tượng nhận. Một đối tượng có thể gửi một thông điệp cho chính nó mà kết quả là hủy động một thao tác riêng tư của nó (gọi cục bộ).
  - **Trả lại (Return):** Trả lại một giá trị cho bên gọi
  - **Gửi (Send):** Gửi một tín hiệu tới một đối tượng
  - **Tạo lập (Create):** Tạo lập một đối tượng mới
  - **Hủy bỏ (Destroy):** Hủy một đối tượng. Một đối tượng cũng có thể hủy bỏ chính nó (terminate)



# Các thông điệp (2)

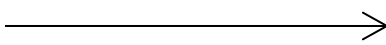
- Khi một đối tượng gửi một thông điệp cho một đối tượng khác, thì đối tượng này trong hoạt động đáp ứng thông điệp trên lại có thể gửi thông điệp cho đối tượng khác, ..., cứ thế tạo thành một luồng kích hoạt lan dần
  - Đó là một **luồng điều khiển** (thread of control)
- Luồng điều khiển có thể **phẳng (tuyến tính)** hay **lồng (do có sự trả lại)**. Bởi thế về hình thức tiếp nối thông điệp, ta phân biệt các loại thông điệp sau:

# Các thông điệp (3)

- **Thông điệp đồng bộ**, biểu diễn bằng mũi tên đầu tam giác đặc: 
  - Đó là một chuyển giao điều khiển lòng, tức là một lời gọi thao tác: bên gọi chuyển điều khiển cho bên bị gọi, **rồi tạm ngưng để chờ** bên bị gọi trả lại điều khiển
  - Bên bị gọi thực hiện thao tác được yêu cầu, nếu cần có thể chuyển điều khiển cho một đối tượng khác; và khi thao tác hoàn thành trả điều khiển về bên gọi, có thể kèm theo kết quả trả lời
  - Thông điệp trả về có thể biểu diễn tường minh bởi mũi tên đứt nét, hoặc có thể bỏ qua vì nó là mặc định ở thời điểm kết thúc thao tác

<-----

# Các thông điệp (4)

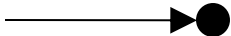
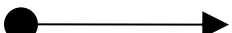
- **Thông điệp không đồng bộ**, biểu diễn bằng mũi tên thường: 

UML 1.3 trở về trước dùng mũi tên nửa: 

- Đó là một chuyển giao điều khiển phẳng, thông qua sự gửi đi một tín hiệu. Thông điệp đi vào hàng đợi của bên nhận.
- Bên gửi không cần biết thông điệp đã được nhận chưa, mà tiếp tục đi vào làm việc ngay (tức là làm việc đồng thời)
- Bên nhận thực hiện một thao tác và cũng có thể trả về một thông tin cho bên gửi. Nhưng nếu có sự trả lại, thì phải biểu diễn tường minh.

# Các thông điệp (5)

Ngoài ra, còn tùy thuộc vào nơi phát và nơi nhận thông điệp là có được biết rõ không mà UML 2.0 còn đưa thêm hai loại thông điệp nữa là:

- **Thông điệp mất hút (lost message):** Là thông điệp mà nơi phát thì biết rõ, song nơi nhận thì không được biết (vì ở ngoài phạm vi mô tả, hoặc vì đó là một sự phát tán). Thông điệp mất hút được biểu diễn bằng một mũi tên có hình tròn đen ở cuối: 
- **Thông điệp kiểm được (found message):** Là thông điệp mà nơi nhận thì biết rõ, song nơi phát thì không biết là đâu (vì ở ngoài phạm vi mô tả). Thông điệp kiểm được được biểu diễn bằng một mũi tên có hình tròn đen ở gốc: 

# Biểu đồ trình tự (1)

- **Biểu đồ trình tự (Sequence diagram)** là một trong hai biểu đồ tương tác chính, với chủ ý làm nổi bật trình tự theo thời gian của các thông điệp
- Nó trình bày một tập hợp các đối tượng cùng với những thông điệp chuyển giao giữa chúng với nhau. Các đối tượng ở đây thường là các cá thể có tên hay khuyết danh của các lớp, song thế vào chỗ các đối tượng, cũng còn có thể là các tác nhân, các hợp tác, các thành phần, các nút

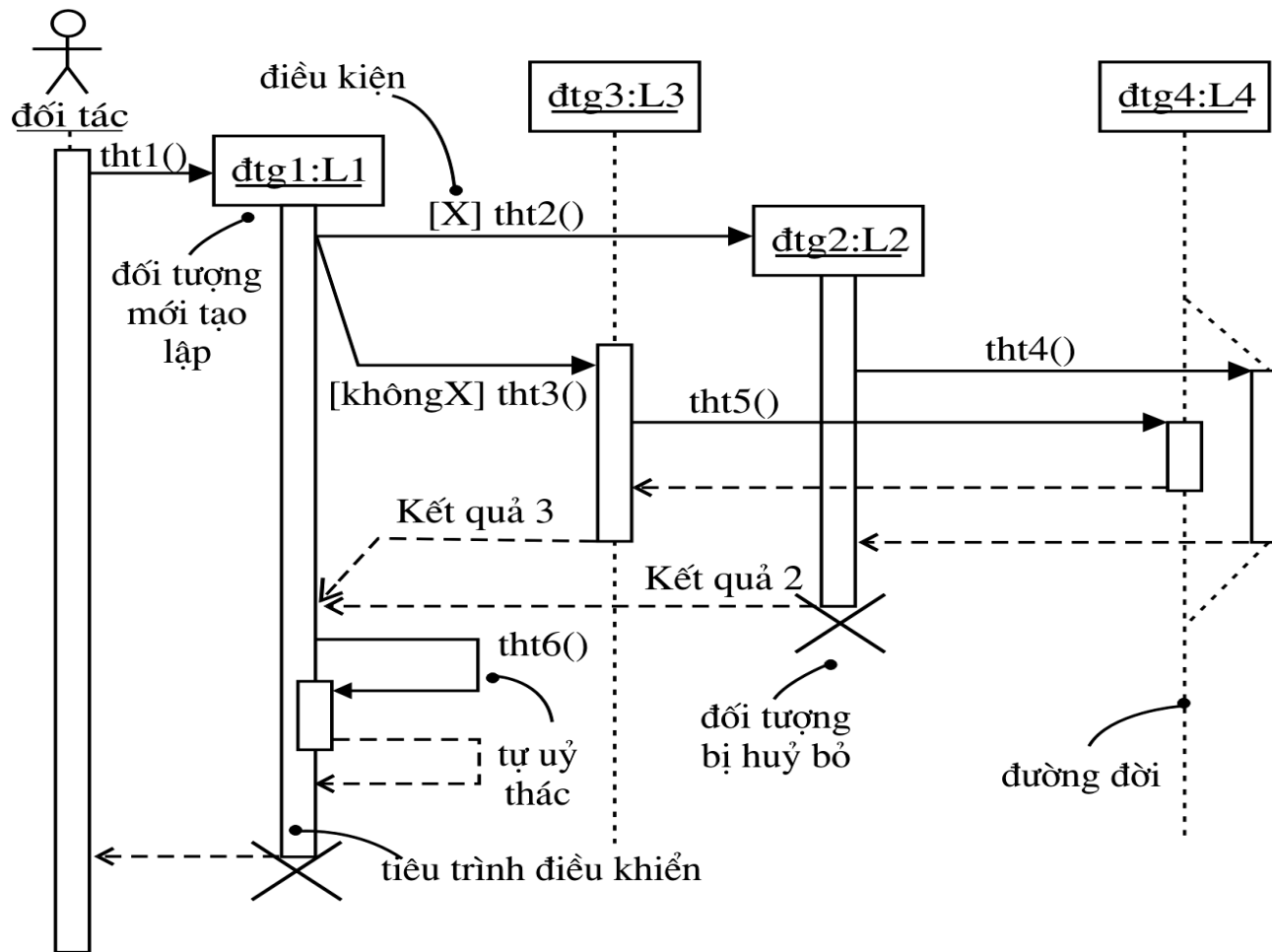
# Biểu đồ trình tự (2)

- Biểu đồ trình tự được trình bày theo hai chiều:
  - Chiều ngang bố trí các đối tượng. Các đối tượng được vẽ theo dạng hình chữ nhật hoặc bằng biểu tượng, dàn thành một hàng ngang trên đỉnh biểu đồ. Trật tự các đối tượng là không quan trọng, song các đối tượng khởi phát thông điệp nên vẽ ở phía trái. Các đối tượng mới được tạo lập thì vẽ thấp xuống, ngang với thông điệp tạo lập chúng.
  - Chiều dọc là trục thời gian (hướng xuống dưới). Mỗi đối tượng có mang một trục đứng (vẽ đứt nét), gọi là **vòng đời (lifecycle)**. Vòng đời của đối tượng sẽ kết thúc bằng một dấu gạch chéo, khi đối tượng bị huỷ bỏ.

# Biểu đồ trình tự (3)

- Các thông điệp (đồng bộ, không đồng bộ hay trả lời) là những mũi tên nằm ngang nối vòng đời của hai đối tượng và được vẽ lần lượt từ trên xuống dưới theo thứ tự thời gian
- Nếu muốn làm rõ thời kỳ hoạt động (tức là lúc đối tượng nắm giữ điều khiển) và làm rõ sự lồng nhau của các thông điệp, ta vẽ thêm trên vòng đời một hay một số dải hẹp hình chữ nhật, gọi là **tiêu trình điều khiển (focus of control)**.
- Lề phải và lề trái của biểu đồ có thể dùng để ghi các giải thích, các ràng buộc

# Biểu đồ trình tự (4)





# Biểu đồ trình tự (5)

Mỗi thông điệp mang theo tên thông điệp dưới dạng:

**thôngđiệp(danh sách tham số)**

Tên thông điệp có thể gắn thêm các tiền tố với các ý nghĩa như sau:

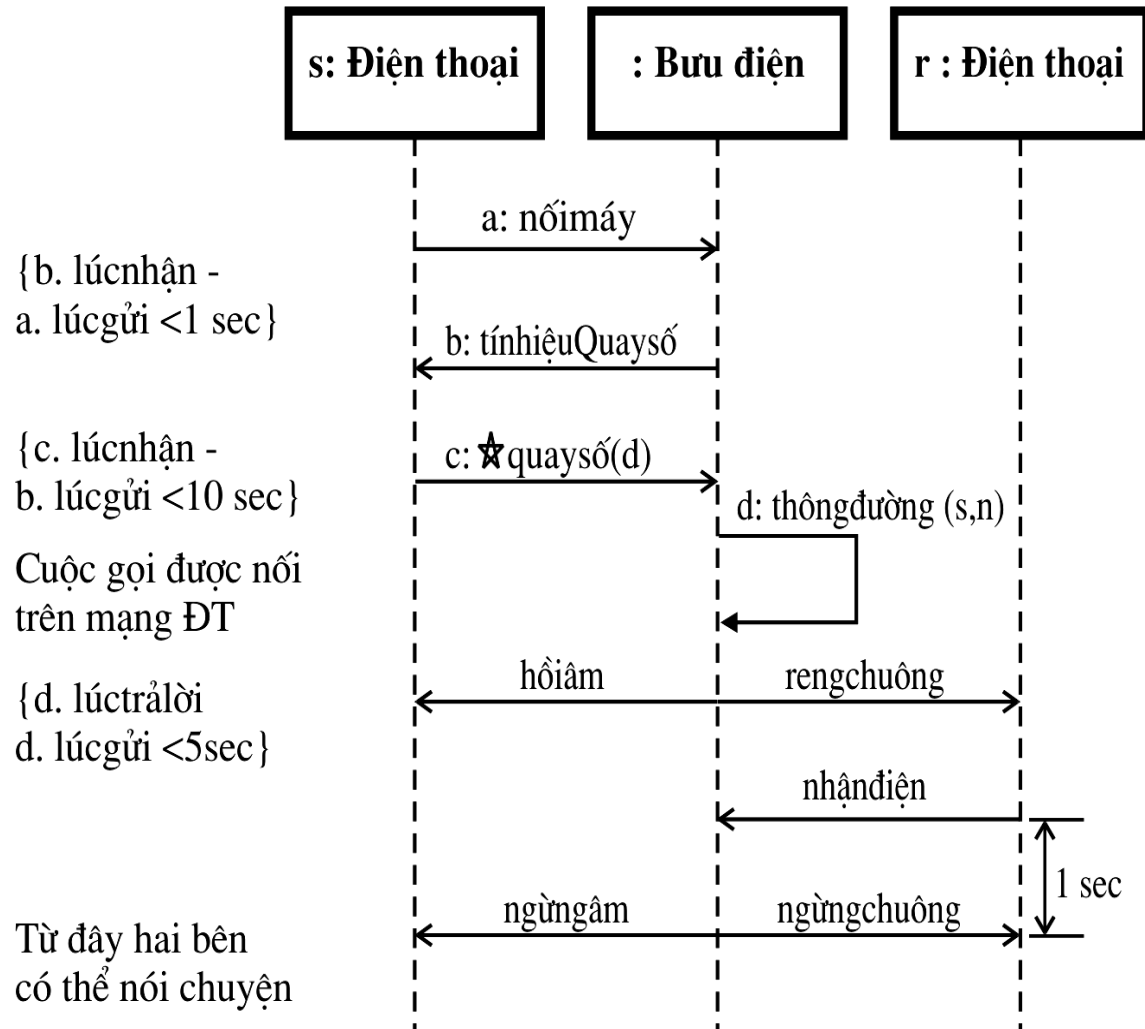
- Một **biểu thức trình tự** có dạng **a:** , thường thì a là số thứ tự của thông điệp, nhưng cũng có thể là một nhãn (ký tự). Vì trật tự thông điệp đã biểu hiện rõ, nên ở biểu đồ trình tự các biểu thức trình tự thường ít dùng.
- Một **điều kiện chọn**, ở dạng **[điều kiện]**, với nghĩa là thông điệp chỉ được gửi đi khi điều kiện này thoả mãn. Nếu vẽ nhiều thông điệp cùng xuất phát ở một điểm, mỗi thông điệp mang một điều kiện riêng thì ta có thể diễn tả:
  - Một rẽ nhánh chọn, nếu các điều kiện là loại trừ lẫn nhau.
  - Một rẽ nhánh song song, nếu các điều kiện đó không loại trừ lẫn nhau.

# Biểu đồ trình tự (6)

- Một **ký hiệu lặp** ở dạng \*, với nghĩa là thông điệp được lặp lại nhiều lần (thường thì sự lặp thực hiện trên nhiều đối tượng, do đối tượng nhận là một đối tượng bội)

*Thí dụ:* Một biểu đồ trình tự diễn tả một cuộc liên lạc bằng điện thoại. Các đối tượng đều là các đối tượng tương tranh (làm việc song song), do đó đều được vẽ với viền đậm. Các thông điệp đều là không đồng bộ (trừ d).

# Biểu đồ trình tự (7)



# MHH tương tác trong ca sử dụng với biểu đồ trình tự (1)

- Một ca sử dụng có thể gồm nhiều kịch bản tương ứng với nhiều luồng điều khiển khác nhau
- Một biểu đồ trình tự chỉ có thể diễn tả một luồng điều khiển, mặc dầu trong biểu đồ cũng có thể diễn tả sự rẽ nhánh hay lặp đơn giản
- Vì vậy, thông thường ta lập một số biểu đồ trình tự cho một ca sử dụng
  - Một vài biểu đồ trong số đó là chính; các biểu đồ còn lại diễn tả các lỗi rẽ khả dĩ hay các trường hợp mắc lỗi

# MHH tương tác trong ca sử dụng với biểu đồ trình tự (2)

- Để lập một biểu đồ trình tự cho một kịch bản của ca sử dụng, ta tiến hành các bước như sau:
  - **Xem lại biểu đồ các lớp tham gia của ca sử dụng** (đã lập ở bước trước), để xác định các cá thể nào của những lớp trong biểu đồ đó tham gia thực sự vào kịch bản đang xét. Muốn thế, cần phải kiểm tra từng bước trong kịch bản để xem các đối tượng đóng vai trò gì trong bước đó (biên, điều khiển, thực thể).
  - Dàn các đối tượng thành hàng ngang trên đỉnh biểu đồ trình tự. Bố trí các đối tượng quan trọng ở bên trái và các đối tượng phụ trợ ở bên phải. Tác nhân, nếu có, thì đặt ở lề trái.

# MHH tương tác trong ca sử dụng với biểu đồ trình tự (3)

- ❑ Vẽ vòng đời cho mỗi đối tượng (và tác nhân). Trong đa số các trường hợp thì đối tượng tồn tại suốt thời gian tương tác. Tuy nhiên cũng có đối tượng được tạo lập và bị huỷ bỏ trong thời gian tương tác, cần được vẽ đường đời ngắn, từ lúc sinh tới lúc chết; với các khuôn dập thích hợp cho các thông điệp đã tạo lập hay huỷ bỏ chúng.
- ❑ Xuất phát với thông điệp đã khởi đầu tương tác. Bố trí các thông điệp tiếp theo lần lượt từ trên xuống dưới, giữa các vòng đời. Chỉ rõ đặc điểm của mỗi thông điệp (vd: các tham số của nó). Nếu cần thì cho thêm giải thích về ngữ nghĩa của tương tác.

# MHH tương tác trong ca sử dụng với biểu đồ trình tự (4)

- ❑ Nếu thấy cần làm rõ thời kỳ hoạt động của các đối tượng và làm rõ sự lồng nhau của các thông điệp, thì vẽ thêm các tiêu trình điều khiển trên mỗi vòng đời của đối tượng.
- ❑ Nếu cần làm rõ các ràng buộc phải có về thời gian và không gian, thì cho thêm các ký hiệu thời gian (như biểu thức trình tự, ký hiệu lặp \*) và các ràng buộc không gian/thời gian thích hợp.
- ❑ Nếu thấy cần diễn tả luồng điều khiển này một cách hình thức hơn, hãy đưa thêm các tiền đề và hậu đề cho mỗi thông điệp.

# MHH sự tương tác với biểu đồ giao tiếp

- Biểu đồ giao tiếp
- MHH tương tác trong ca sử dụng với biểu đồ giao tiếp



# Biểu đồ giao tiếp (1)

- **Biểu đồ giao tiếp (Communication diagram)** là một trong hai biểu đồ tương tác chính, nhằm làm nổi bật khung cảnh tổ chức của sự tương tác. Tên cũ trong UML 1.x là biểu đồ hợp tác (Collaboration diagram).
- Biểu đồ giao tiếp trình bày một tập hợp các đối tượng, các kết nối giữa các đối tượng đó cùng với những thông điệp chuyển giao giữa chúng với nhau. Các đối tượng ở đây thường là các cá thể có tên hay khuyết danh của các lớp, song thế vào chỗ các đối tượng, cũng còn có thể là các tác nhân (người hoặc hệ thống bên ngoài), các thành phần, các nút.
- Thông điệp được biểu diễn bằng một mũi tên nhỏ, vẽ dọc theo một kết nối giữa hai đối tượng, với hàm ý rằng nhờ có kết nối đó, mà bên gửi biết bên nhận để có thể gửi thông điệp. Nói cách khác, thông điệp không thể gửi đi nếu bên nhận không ở trong "**tầm nhìn**" của bên gửi.

# Biểu đồ giao tiếp (2)

- Có kiểu kết nối thể hiện tầm nhìn.
  - Tồn tại **liên kết** giữa hai lớp của hai đối tượng: Kết nối được ghi thêm từ khoá <<**association**>>
  - Đối tượng nhận là **toàn cục** đối với đối tượng gửi: Kết nối được ghi thêm từ khoá <<**global**>>
  - Đối tượng nhận là **cục bộ** trong thao tác (của bên gửi) đã gửi thông điệp đi: Kết nối được ghi thêm từ khoá <<**local**>>
  - Đối tượng nhận là **tham số** của thao tác (của bên gửi) đã gửi thông điệp đi: Kết nối được ghi thêm từ khoá <<**parameter**>>
  - Đối tượng nhận cũng **chính là đối tượng gửi**: Kết nối được ghi thêm từ khoá <<**self**>>

# Biểu đồ giao tiếp (3)

- Các đối tượng và các kết nối có thể được **tạo lập** hay bị **huỷ bỏ** trong thời gian thực hiện. Để phân biệt, ta ghi thêm các xâu tính chất:
  - **{new}**: cho đối tượng hay kết nối mới được tạo lập;
  - **{destroyed}**: cho đối tượng hay kết nối bị huỷ bỏ;
  - **{transient}**: cho đối tượng hay kết nối được thành lập trong thời gian thực hiện, rồi bị huỷ bỏ.
- Mũi tên thông điệp (ở dạng  $\longrightarrow$  hay  $\Rightarrow$ ) phải kèm theo tiêu đề của thông điệp với cú pháp đầy đủ là như sau:

**[Bước trước] [Biểu thức trình tự] [Trả lời :=] Tên Thông điệp (D/s Tham số)**

trong đó trừ Tên Thông điệp là bắt buộc, còn các phần khác đều là tùy ý

# Biểu đồ giao tiếp (4)

- **Bước trước:** Một danh sách các số thứ tự, cách nhau bằng dấu phẩy và kết thúc bằng gạch chéo ('/').
- Thông điệp chỉ được chuyển giao khi các thông điệp tương ứng với các số thứ tự cho trong danh sách này đã chuyển giao rồi
- Đây là một yêu cầu đồng bộ hoá của lộ trình điều khiển
- Ví dụ: 1.1, 2.3/

# Biểu đồ giao tiếp (5)

- **Biểu thức trình tự:** Là dãy số và ký tự kết thúc bởi dấu hai chấm (':') diễn tả trình tự tiếp nối các thông điệp.
  - Trường hợp đơn giản, thì đó chỉ là một số thứ tự, viết theo dạng ký pháp chấm (.)
  - Khi một thông điệp kích hoạt một thao tác mà bên trong thao tác này có một loạt thông điệp được gửi đi, thì những thông điệp này mang số thứ tự là số thứ tự của thông điệp trên kèm một dấu chấm rồi các số 1, 2, 3... lần lượt. Cách đánh số này cho thấy sự lồng nhau của các thông điệp.
  - Biểu thức trình tự nhằm mục đích:
    - Diễn tả **lộ trình điều khiển**, như là A3, B4/ C3.1.2:
    - Diễn tả **điều kiện chọn**, như là 4.2. [x<0]:
    - Diễn tả **thông điệp lặp**, như là 1.1.\*: hay 2.1.\* [i:=1..n]:

# Biểu đồ giao tiếp (6)

- **Trả lời:** Là tên của kết quả trả về bên gửi
  - Tên này sau đó có thể dùng làm tham số cho thông điệp khác
  - Tên này có thể lấy là một biến cục bộ trong thao tác gửi thông điệp, hoặc là một thuộc tính của đối tượng gửi
- **Tên Thông điệp (D/sTham số):** Là tên của thông điệp, kèm theo danh sách tham số (có thể rỗng)
  - Tên này thường lấy trùng với tên thao tác cần được thực hiện (được gọi) ở bên nhận

# Biểu đồ giao tiếp – Ví dụ (1)

Chương trình sau đây thực hiện biểu đồ giao tiếp nêu trên, cho phép ta hiểu rõ hơn về các ký pháp vận dụng trong đó:

```
dànhsẵn(d : Đơnhàng)
```

```
{
```

```
    DòngĐH dòng; Hàng hàng; int sốlượng;
```

```
    for (int i=1; i < d.sốdòng(); i++) {
```

```
        dòng = d.choDòng(i);
```

```
        hàng = dòng.choHàng();
```

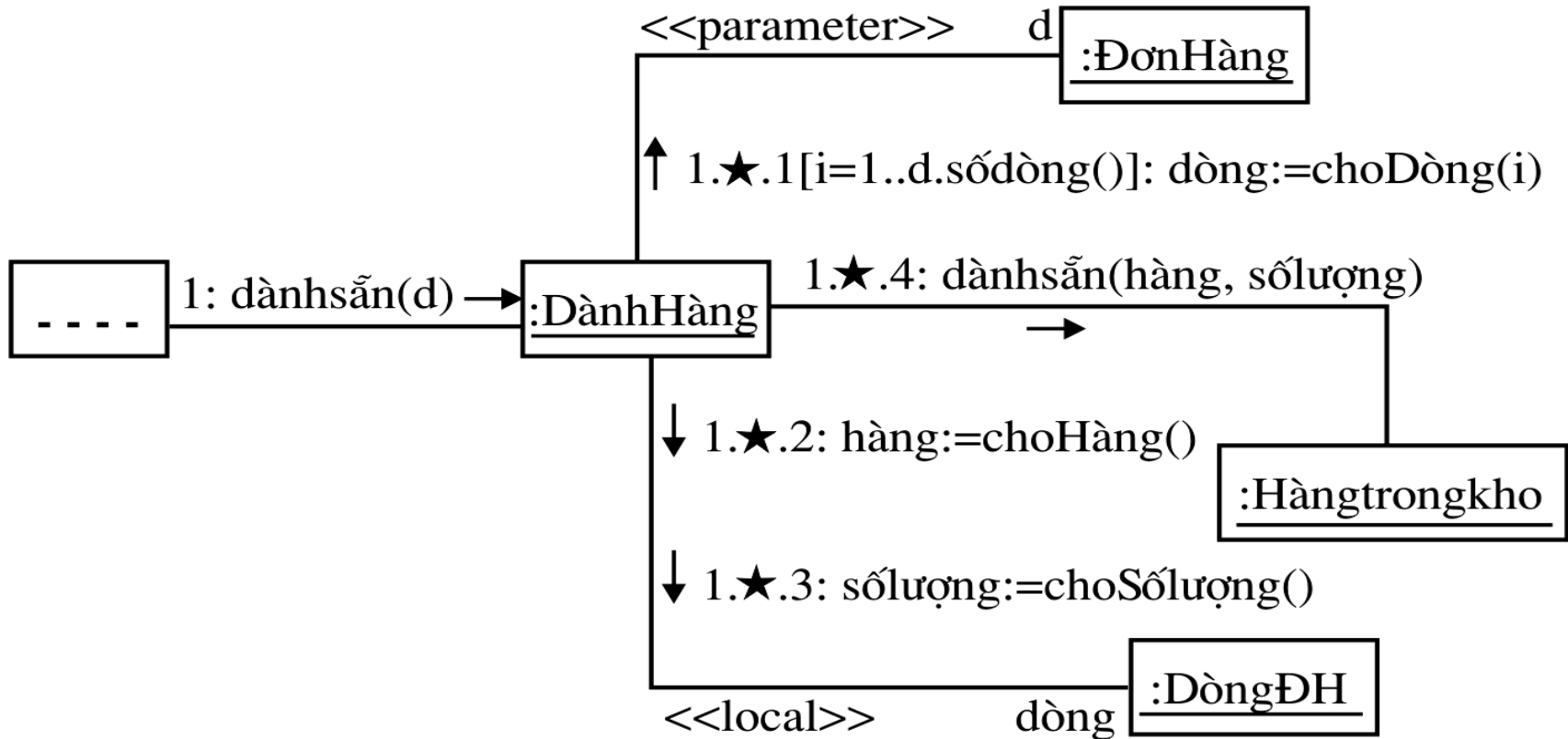
```
        sốlượng = dòng.choSốlượng();
```

```
        hàngtrongKho.dànhsẵn(hàng, sốlượng);
```

```
    }
```

```
}
```

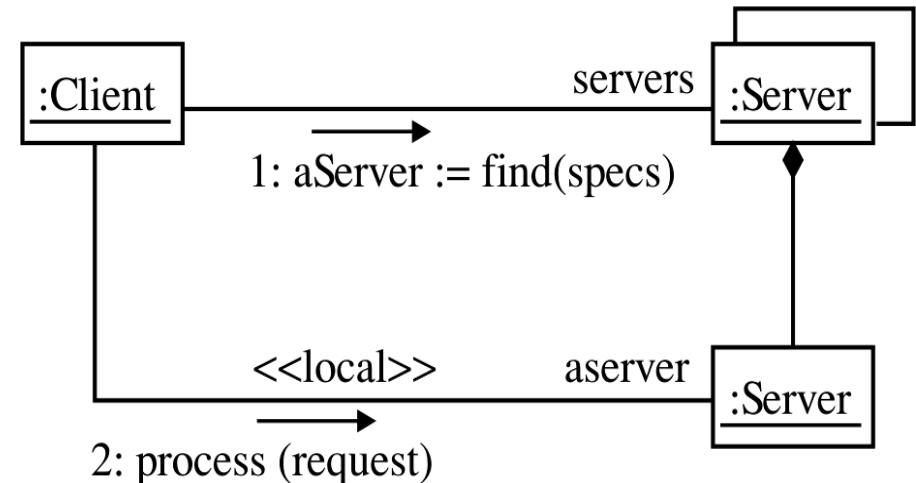
# Biểu đồ giao tiếp – Ví dụ (2)





# Biểu đồ giao tiếp (7)

- Nhiều khi một thông điệp gửi đi không hẳn là đến một đối tượng, mà đến nhiều đối tượng một lúc. Chẳng hạn thông điệp tìm một đối tượng thích hợp tại đầu "nhiều" của một liên kết, hoặc tìm một cuốn sách trong catalô. Bấy giờ ta nên diễn tả bằng **đối tượng bội**.
- Đối tượng bội (multiobject) là một ký pháp UML dùng để biểu diễn nhiều đối tượng của cùng một lớp. Nó được vẽ thành hai đối tượng chồng lên nhau. Dùng đối tượng bội, ta không phải đề cập quá sớm vào cách thức cài đặt (chẳng hạn không cần biết đây sẽ là lớp Vector hay lớp ArrayList, hay là gì).



# MHH tương tác trong ca sử dụng với biểu đồ giao tiếp (1)

- Để lập biểu đồ giao tiếp cho một kịch bản của ca sử dụng, ta tiến hành các bước sau:
  - Xem lại biểu đồ các lớp tham gia của ca sử dụng (đã lập ở bài trước) để xác định các cá thể nào của những lớp trong biểu đồ đó tham gia thực sự vào kịch bản đang xét. Muốn thế, kiểm tra từng bước trong kịch bản để xem đối tượng đóng vai trò gì trong bước đó.
  - Vẽ các đối tượng như là các đỉnh của đồ thị. Bố trí các đối tượng quan trọng vào giữa, các đối tượng khác ở xung quanh.

# MHH tương tác trong ca sử dụng với biểu đồ giao tiếp (2)

- Gán những tính chất đầu tiên cho mỗi đối tượng
- Nếu có đối tượng nào đó mà tính chất của nó (giá trị thuộc tính, giá trị gắn nhãn, trạng thái, vai trò) thay đổi một cách đáng kể trong thời gian tương tác, hãy vẽ thêm bản sao cho các đối tượng đó, cập nhật chúng với các giá trị mới và nối với đối tượng gốc bằng một phụ thuộc với khuôn dạng như là <<become>> hay <<copy>>

# MHH tương tác trong ca sử dụng với biểu đồ giao tiếp (3)

- Xác định các kết nối giữa các đối tượng, cùng với các thông điệp có thể có trên đó
  - Trước hết, hãy vẽ các kết nối liên kết (<<associate>>); đó là những kết nối quan trọng nhất vì chúng diễn tả cấu trúc tĩnh
  - Tiếp theo, hãy vẽ các kết nối khác và ghi chú với các khuôn dập thích hợp (như là <<global>>, <<local>>) để làm rõ các đối tượng liên hệ với nhau theo kiểu gì

# MHH tương tác trong ca sử dụng với biểu đồ giao tiếp (4)

- Xuất phát với thông điệp đã khởi đầu tương tác. Bố trí lần lượt các thông điệp tiếp theo trên các kết nối thích hợp, thêm các số thứ tự thích hợp. Dùng ký pháp chấm để làm rõ sự lồng nhau.
- Nếu thấy cần các ràng buộc về không gian và thời gian, hãy thêm vào thông điệp các dấu hiệu thời gian và đính kèm các ràng buộc về không gian hay thời gian
- Nếu thấy cần diễn tả luồng điều khiển này một cách hình thức hơn, hãy đưa thêm tiền đề và hậu đề cho mỗi thông điệp

# Đối chiếu và chỉnh sửa các mô hình

- Đến đây, ta đã lập các biểu đồ cấu trúc (biểu đồ các lớp lĩnh vực, biểu đồ các lớp tham gia) và đã lập các biểu đồ tương tác (biểu đồ trình tự, biểu đồ giao tiếp). Ta cần phải đối chiếu các biểu đồ đó với nhau, để chỉnh sửa lại các chỗ không phù hợp giữa chúng.
- Các việc chỉnh sửa gồm:
  - Thêm bớt các lớp
  - Thêm và chỉnh sửa các thao tác trong các lớp
  - Thêm các kiểu cho các thuộc tính, các tham số và các trả lời
  - Thêm và chỉnh sửa các mối liên quan giữa các lớp

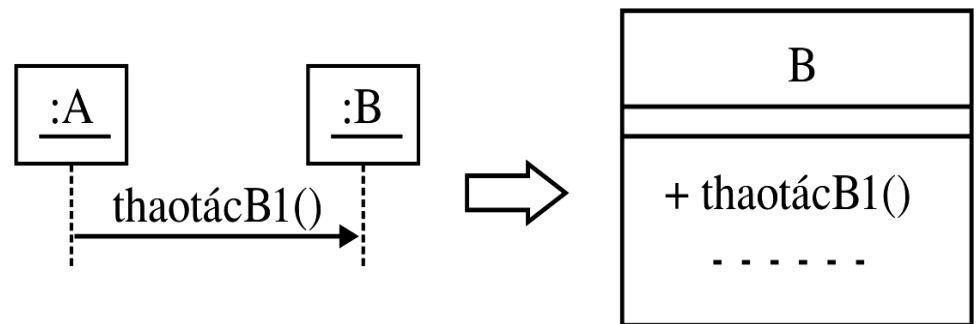
# Thêm bớt các lớp

- Một số lớp đã được phát hiện thêm khi ta lập các biểu đồ tương tác; cần bổ sung chúng vào các biểu đồ cấu trúc
- Ngược lại, có những lớp trong biểu đồ cấu trúc không hề có đối tượng tham gia vào một biểu đồ tương tác nào; cần phải loại bỏ chúng đi

# Thêm và chỉnh sửa các thao tác trong các lớp (1)

Khi có một thông điệp được gửi từ một đối tượng của lớp A sang một đối tượng của lớp B, thì:

- Trong lớp B (bên nhận) phải có một thao tác công cộng tương ứng với thông điệp đó. Tên thông điệp và thao tác phải phù hợp ngữ nghĩa với nhau, thông thường thì thông điệp nên lấy trùng tên với thao tác. Danh sách các tham số của thao tác phải phù hợp các yêu cầu của thông điệp.
- Trong lớp A (bên gửi) phải có một thao tác có trách nhiệm gửi đi thông điệp trên





# Thêm và chỉnh sửa các thao tác trong các lớp (2)

Tuy nhiên quy luật trên cũng có những ngoại lệ

- Thông điệp đến một đối tác:
  - Nếu tác nhân là người, thì thông điệp chỉ yêu cầu người làm một việc gì đó ghi trong bản hướng dẫn sử dụng, chứ không phải là một thao tác (tức là một chương trình máy tính)
  - Nếu tác nhân là một thiết bị, thì thông điệp chỉ kích hoạt một cảm biến hay một động cơ ở thiết bị đó
  - Nếu tác nhân là một hệ thống ngoài, thì nó phải được đại diện bởi một lớp chứa các thao tác thích hợp. Khi đó, thông điệp sẽ kích hoạt một thao tác trong lớp này.

# Thêm và chỉnh sửa các thao tác trong các lớp (3)

- Thông điệp đến một lớp giao diện GUI
  - Yêu cầu của thông điệp được cài đặt thành các yếu tố đồ hoạ của GUI (như là trường, hay nút, ...), chứ không là một thao tác
- Các thao tác riêng tư
  - Đó là những thao tác dùng trong nội bộ, chỉ tương ứng với các thông điệp đệ quy, chứ không tương ứng với thông điệp từ đối tượng khác gửi đến

# Thêm các kiểu cho các thuộc tính, các tham số và các trả lời

- Các kiểu có thể là: các kiểu nguyên thủy, hay các kiểu do người dùng đưa vào, hay các lớp
- Tuy nhiên, chưa nên vội dùng tên các kiểu của một ngôn ngữ lập trình cụ thể nào, mà vẫn dùng các tên chung, như vậy linh hoạt hơn

# Thêm và chỉnh sửa các mối liên quan giữa các lớp

- Khi đối tượng của một lớp A gửi một thông điệp cho một đối tượng của một lớp B, thì giữa hai đối tượng đó có một kết nối với ý nghĩa là bên nhận thuộc phạm vi hiểu biết (ở trong tầm nhìn) của bên gửi
  - Nếu sự hiểu biết đó là bản chất (vốn có từ lâu), thì đây là một biểu hiện của sự liên kết (association) giữa A và B
  - Nếu sự hiểu biết đó là mới được cung cấp (chẳng hạn thông qua việc truyền tham số hay việc nhận kết quả trả lời), thì đây là một biểu hiện của sự phụ thuộc (dependency) của A vào B

# Bài tập tổng hợp (1)

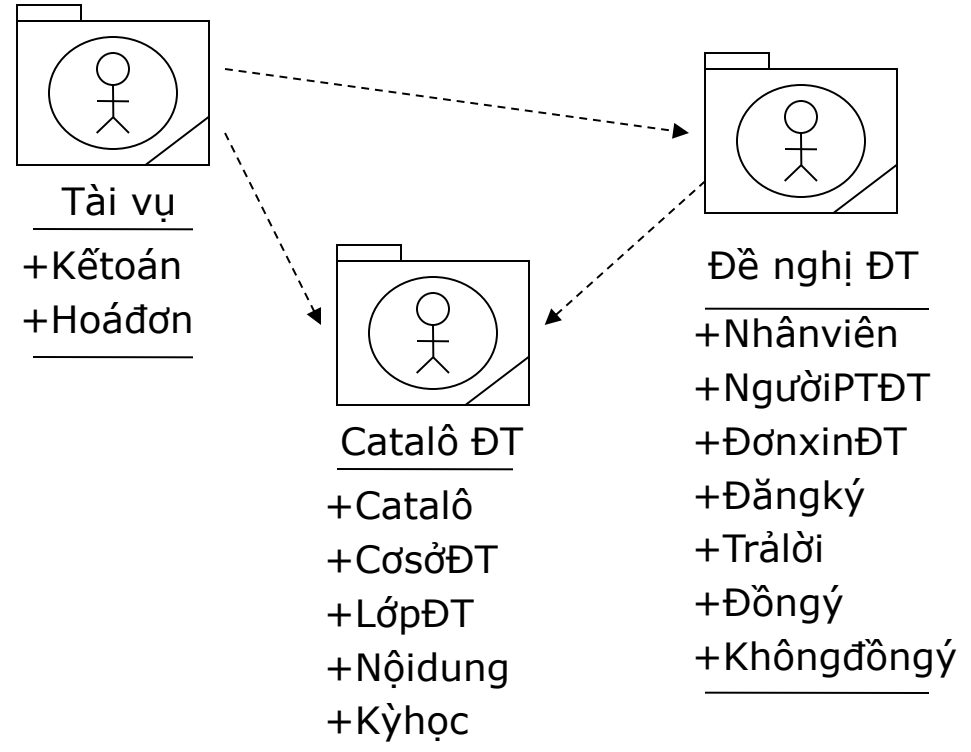
- Để MHH tương tác, Larman và Jacobson đề nghị các bước sau:
  - 1) Xác định các **vòng lặp**: Mỗi vòng lặp thường là một ca sử dụng, nhưng cũng có thể là một gói ca sử dụng
  - 2) Xác định các **thao tác hệ thống**: Đó là các yêu cầu mà một tác nhân đưa ra đối với hệ thống (tức là một kịch bản mà tác nhân khởi động)
  - 3) Lập một biểu đồ tương tác cho mỗi thao tác hệ thống

# Bài tập tổng hợp (2)

**Câu hỏi 17:** *Xác định các vòng lặp cho việc MHH tương tác.*

■ Nhìn các phụ thuộc trong biểu đồ gói, ta thấy là nên bắt đầu với gói Catalô DT, rồi đến gói Đề nghị DT

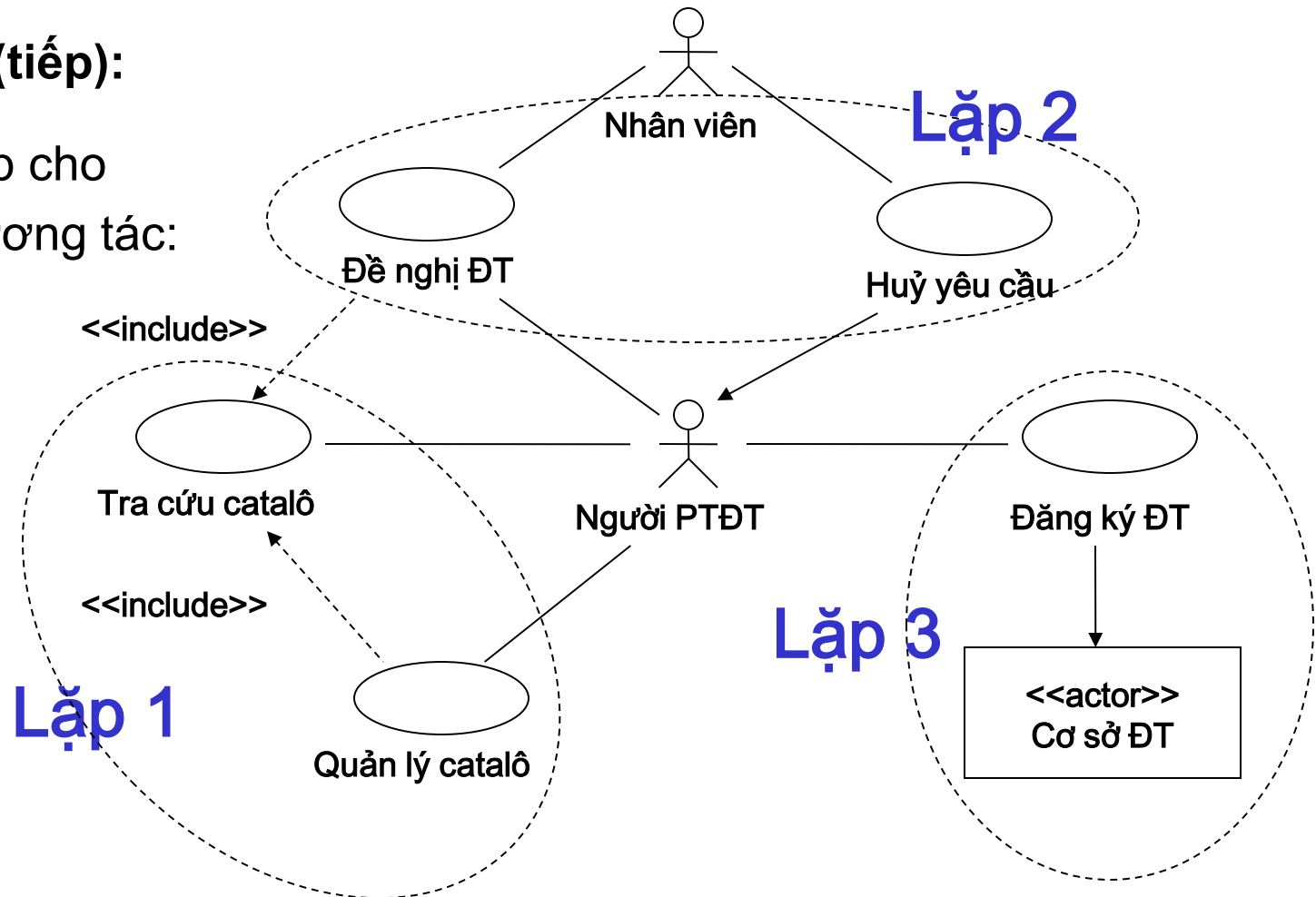
■ Vậy trên biểu đồ ca sử dụng ta chia cắt các vòng lặp như ở trang sau



# Bài tập tổng hợp (3)

**Câu hỏi 17 (tiếp):**

Các vòng lặp cho  
việc MHH tương tác:



# Bài tập tổng hợp (4)

**Câu hỏi 18:** Định nghĩa các thao tác hệ thống cho Vòng lặp 1.

- Với mỗi thao tác hệ thống, Larman đề nghị chỉ ra một mô tả của thao tác gồm các điểm sau:

- 1) Tên
- 2) Trách nhiệm
- 3) Tham chiếu
- 4) Điều kiện vào
- 5) Điều kiện ra
- 6) Ngoại lệ (tùy chọn)
- 7) Chú thích (tùy chọn)

Hệ thống
tạoLớpĐT() điều chỉnhLớpĐT() tạoCơ sởĐT() điều chỉnhCơ sởĐT() tạoChủ đề() điều chỉnhChủ đề() tạoKỳ học() điều chỉnhKỳ học()

Nghiên cứu ca sử dụng 'Quản lý catalô', ta đưa ra được các thao tác hệ thống cho vòng lặp 1, như ở Hình bên trên

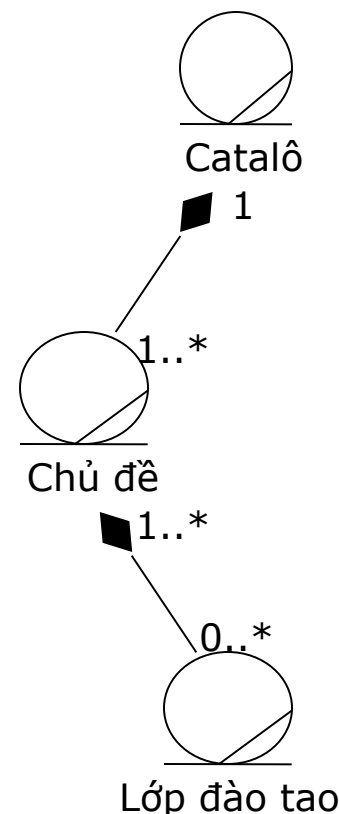


# Bài tập tổng hợp (5)

## Câu hỏi 18 (tiếp):

*Lập mô tả của thao tác hệ thống:*  
"tạoLậpLớpĐT".

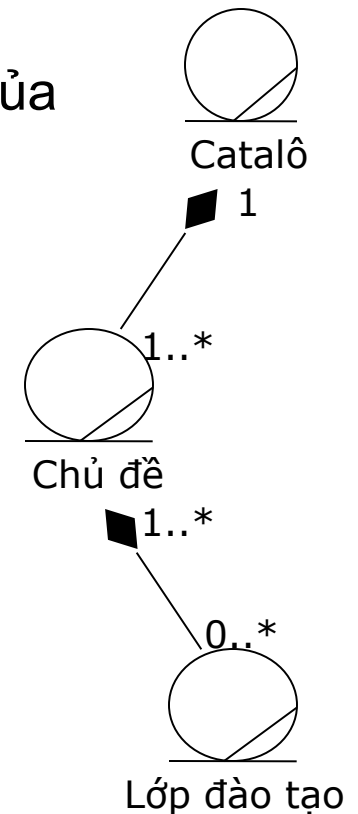
- Trước hết ta cần lưu ý các lớp thuộc gói Catalô ĐT, vì thao tác hệ thống "tạoLậpLớpĐT" sẽ đề cập các lớp này
- Tuy nhiên, ta thấy thiếu lớp Chủ đề, mà ta sẽ phải bổ sung vào gói đó như ở hình bên.
- Bây giờ, có thể lập mô tả của thao tác hệ thống như sau:



# Bài tập tổng hợp (6)

## Câu hỏi 18 (tiếp):

- 1) Tên: tạoLớpĐT
- 2) Trách nhiệm: Tạo lập một Lớp đào tạo mới theo mô tả của CSĐT và đặt lớp đó vào ít nhất một Chủ đề
- 3) Tham chiếu: Ca sử dụng “Quản lý Catalô”
- 4) Các điều kiện vào:
  - ❑ Catalô ĐT đã có từ trước
  - ❑ Có ít nhất một Chủ đề trong catalô
  - ❑ Cơ sở ĐT cung cấp thông tin đã có trong catalô
  - ❑ Người PTĐT đang kết nối với Internet

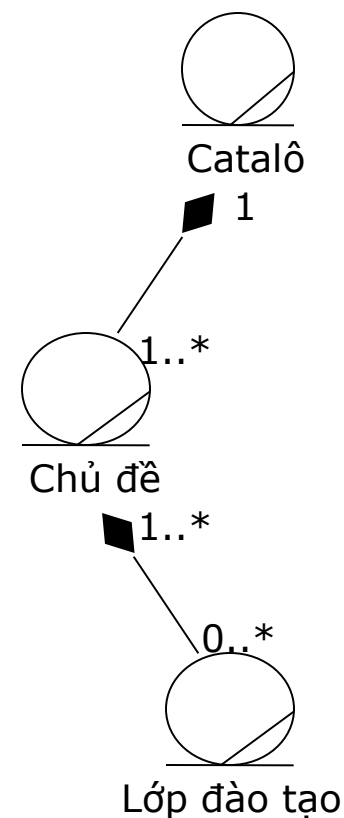


# Bài tập tổng hợp (7)

## Câu hỏi 18 (tiếp):

### 3) Các điều kiện ra:

- ❑ Một Lớp đào tạo d đã được tạo lập cùng các thuộc tính
- ❑ Một đối tượng nội dung n đã được tạo lập cùng các thuộc tính
- ❑ n đã được kết nối với d
- ❑ d đã được kết nối với Cơ sở DT tương ứng
- ❑ Các kỳ học nếu đã có phải được tạo lập cùng các thuộc tính
- ❑ d phải được kết nối với ít nhất một Chủ đề



# Bài tập tổng hợp (8)

**Câu hỏi 19:** *Lập một biểu đồ tương tác (biểu đồ trình tự hay biểu đồ giao tiếp) thực hiện thao tác hệ thống “tạoLớpLớpĐT”.*

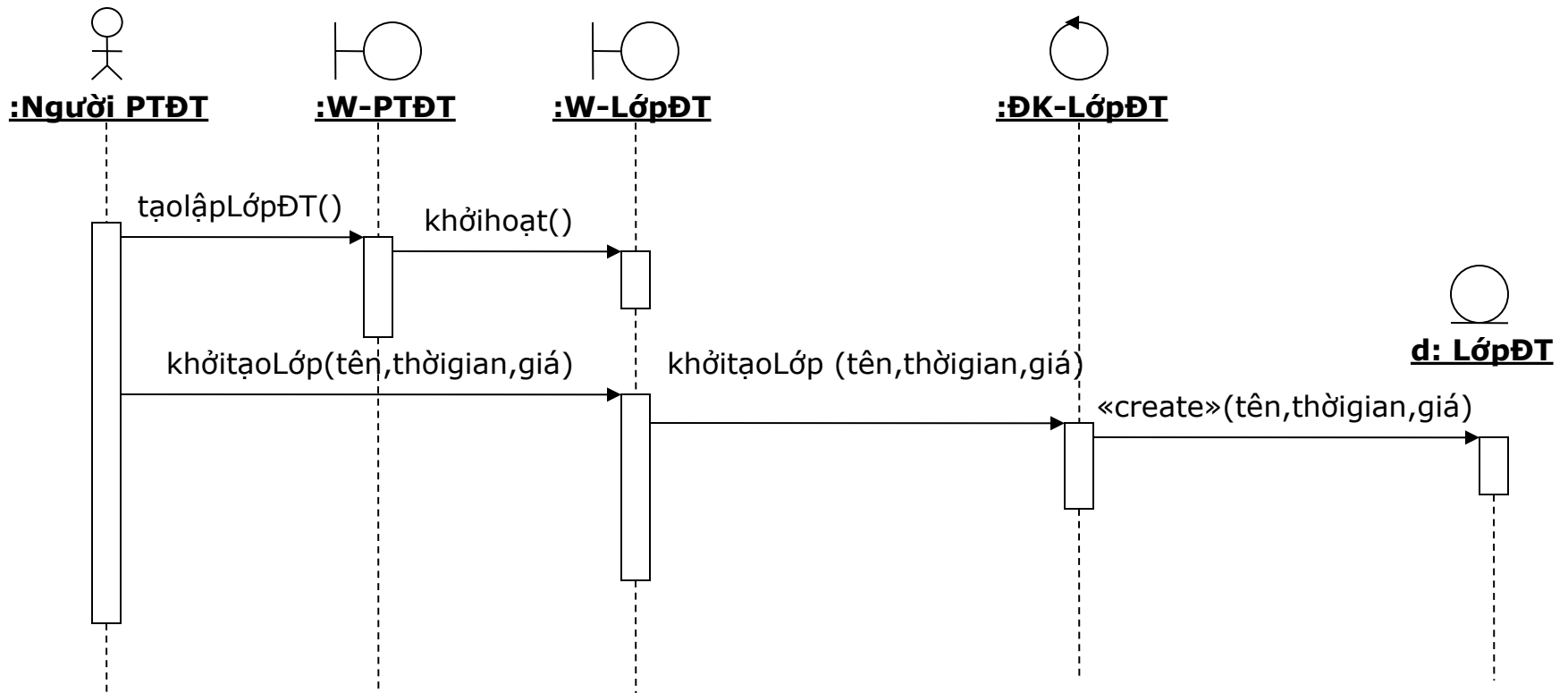
- Các điều kiện ra đã nêu ở trên (Câu hỏi 18) cho ta thấy cái đích cần đạt được:
  - ❑ Một Lớp đào tạo d đã được tạo lập cùng các thuộc tính
  - ❑ Một đối tượng nội dung n đã được tạo lập cùng các thuộc tính
  - ❑ n đã được kết nối với d
  - ❑ d đã được kết nối với cơ sở ĐT tương ứng
  - ❑ Các kỳ học nếu đã có phải được tạo lập cùng các thuộc tính
  - ❑ d phải được kết nối với ít nhất một Chủ đề

# Bài tập tổng hợp (9)

- Tuy nhiên, các điều kiện này là không có thứ tự.
- Ta cần quyết định một trình tự thực hiện, chẳng hạn:
  - 1) Khởi tạo đối tượng  $d$  và các thuộc tính của nó
  - 2) Tạo lập nội dung  $n$  của nó
  - 3) Tạo lập các kỳ học
  - 4) Ghép  $d$  vào một (vài) chủ đề và với cơ sở đào tạo

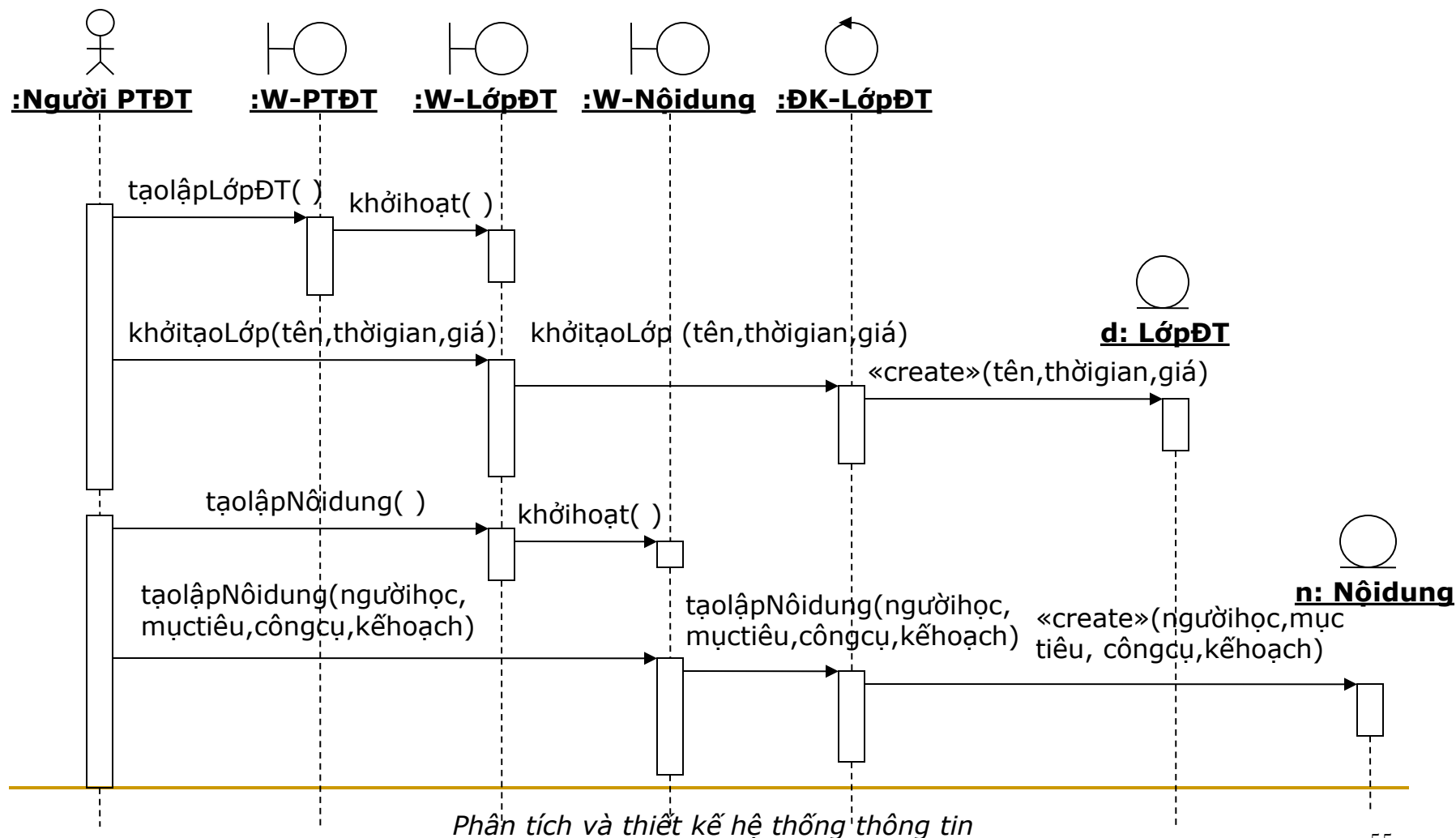
# Bài tập tổng hợp (10)

- Thực hiện bước 1 (nêu trong trang trước) bằng một biểu đồ trình tự:



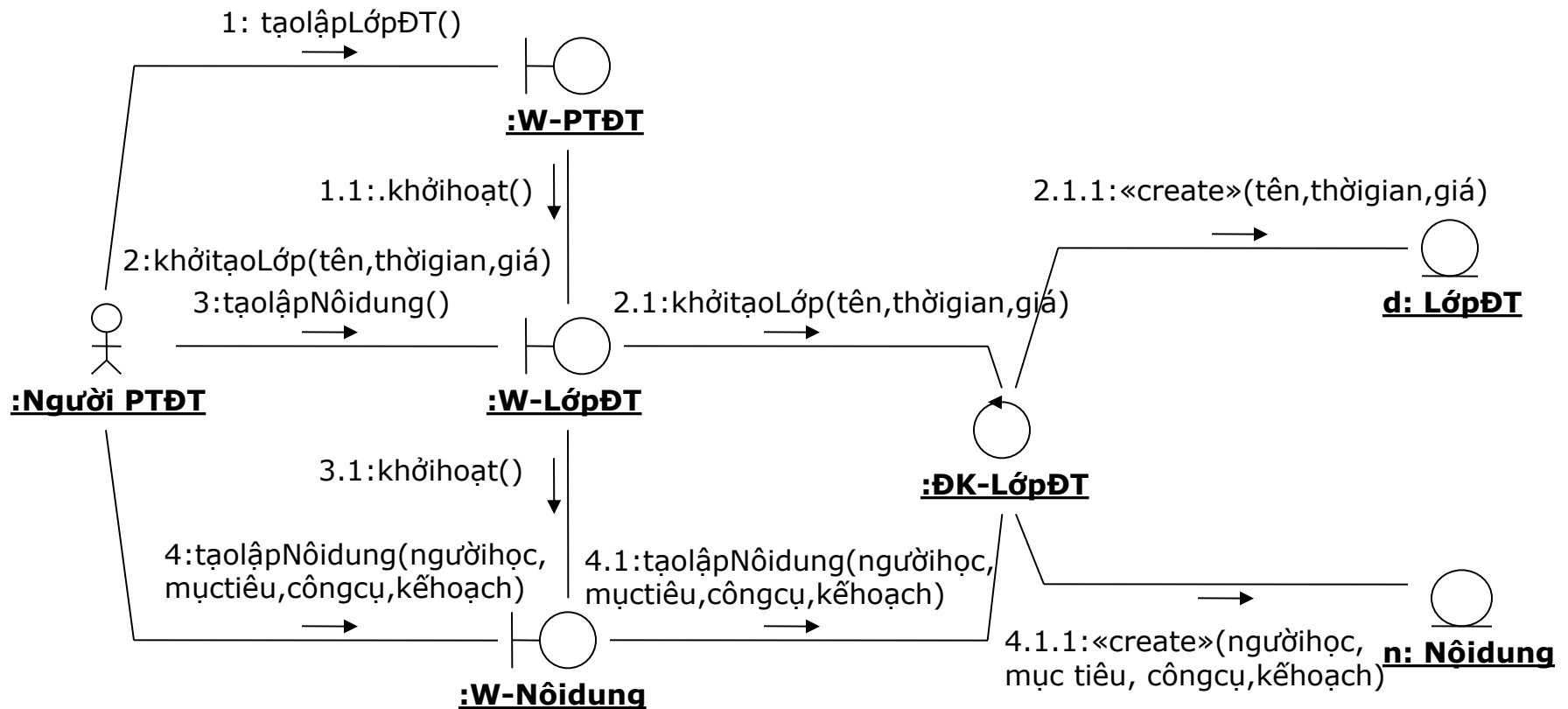
# Bài tập tổng hợp (11)

- Thực hiện các bước 1 và 2 (ở slide 53) bằng một biểu đồ trình tự:



# Bài tập tổng hợp (12)

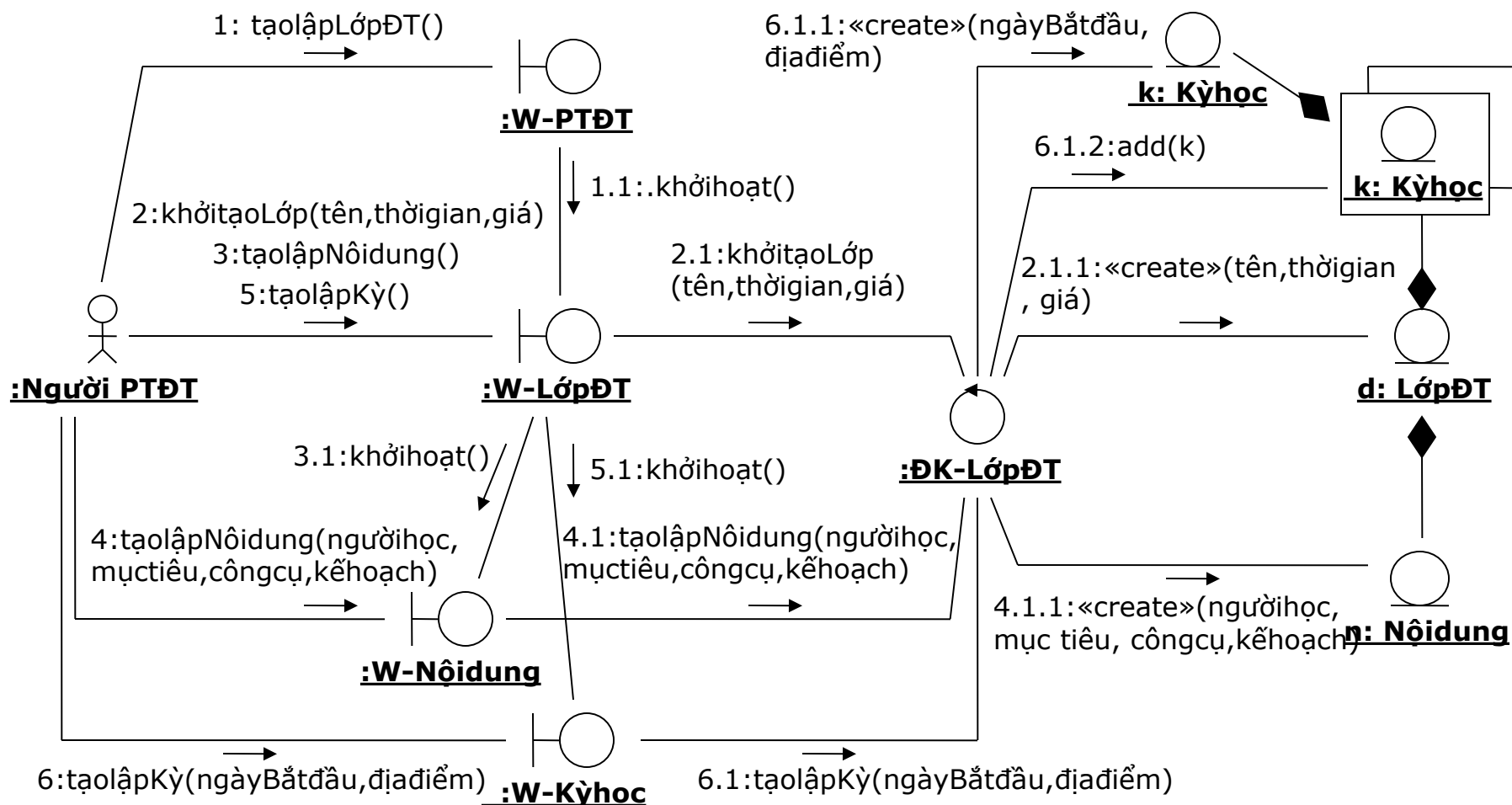
- Số đối tượng càng nhiều thì biểu đồ trình tự càng khó đọc, khó trình bày. Ta thay biểu đồ trình tự trên bằng biểu đồ giao tiếp (tương đương) sau và dùng nó để triển khai các bước tiếp:





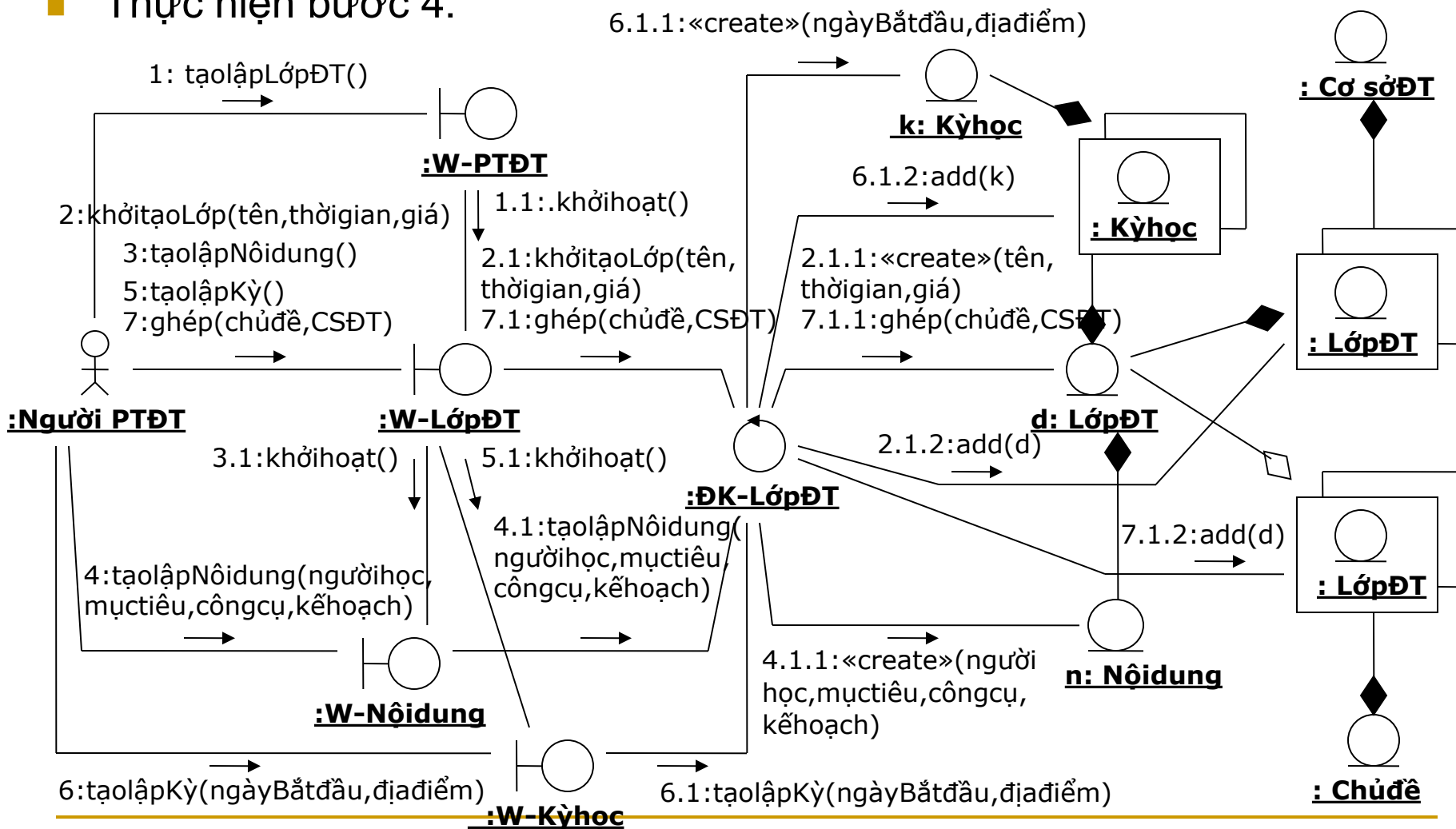
# Bài tập tổng hợp (13)

- Thực hiện bước 3 (ở slide 53) bằng cách dùng đối tượng bội:



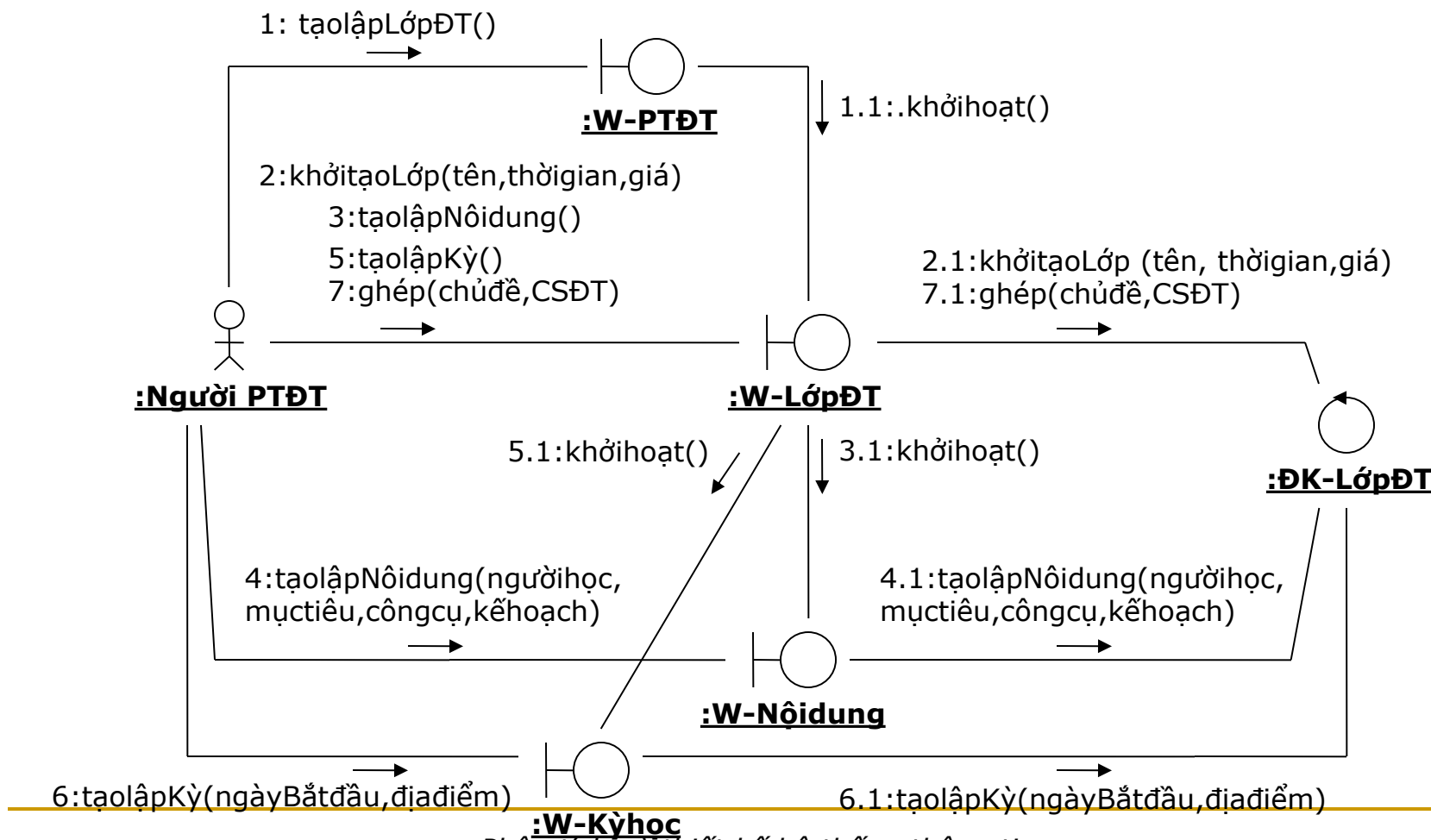
# Bài tập tổng hợp (14)

## ■ Thực hiện bước 4:

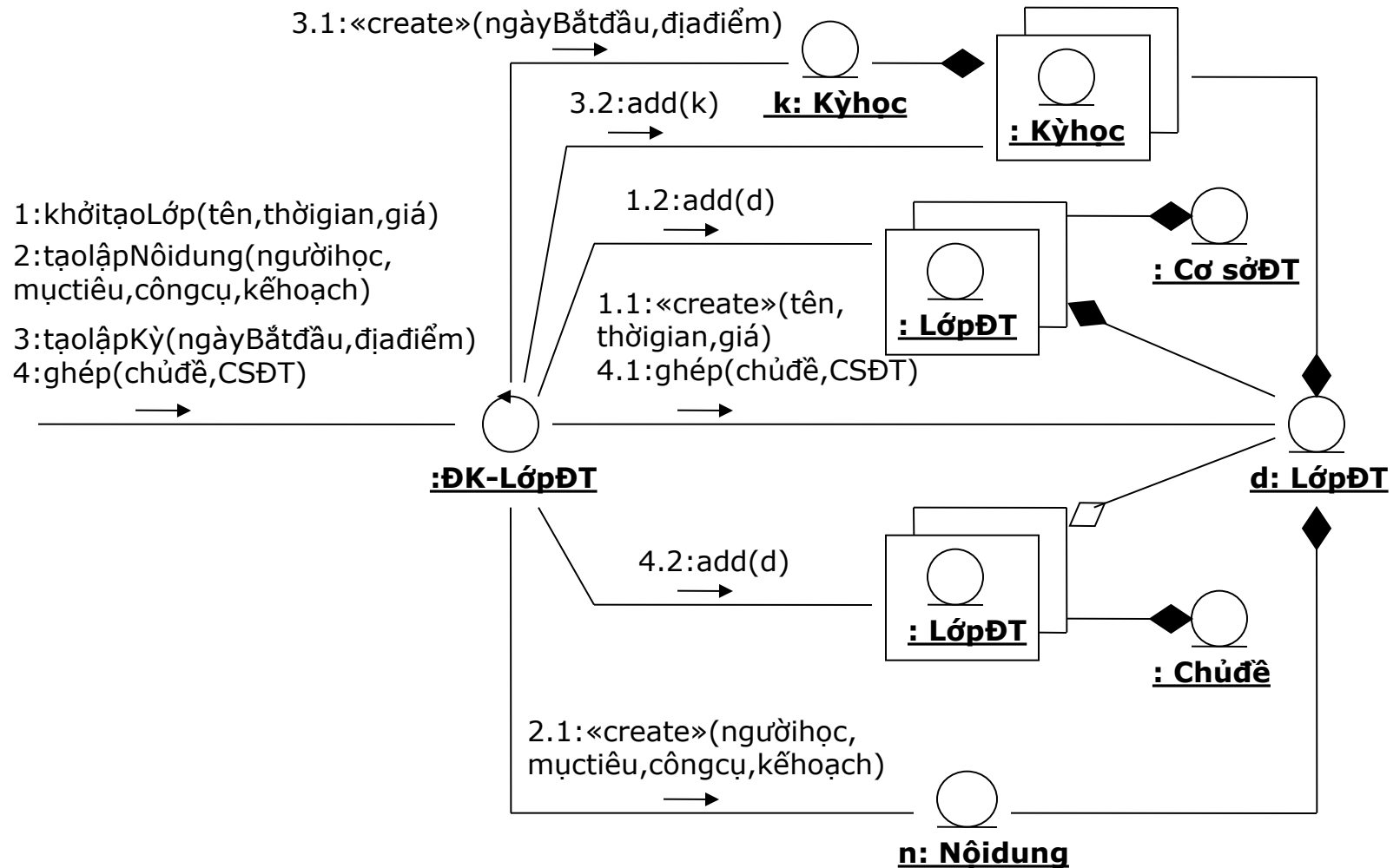


# Bài tập tổng hợp (15)

- Biểu đồ ở trang trước là quá lớn. Để lập trên các biểu đồ nhỏ hơn, ta cắt nó ra làm đôi như ở trang này và trang sau.



# Bài tập tổng hợp (16)

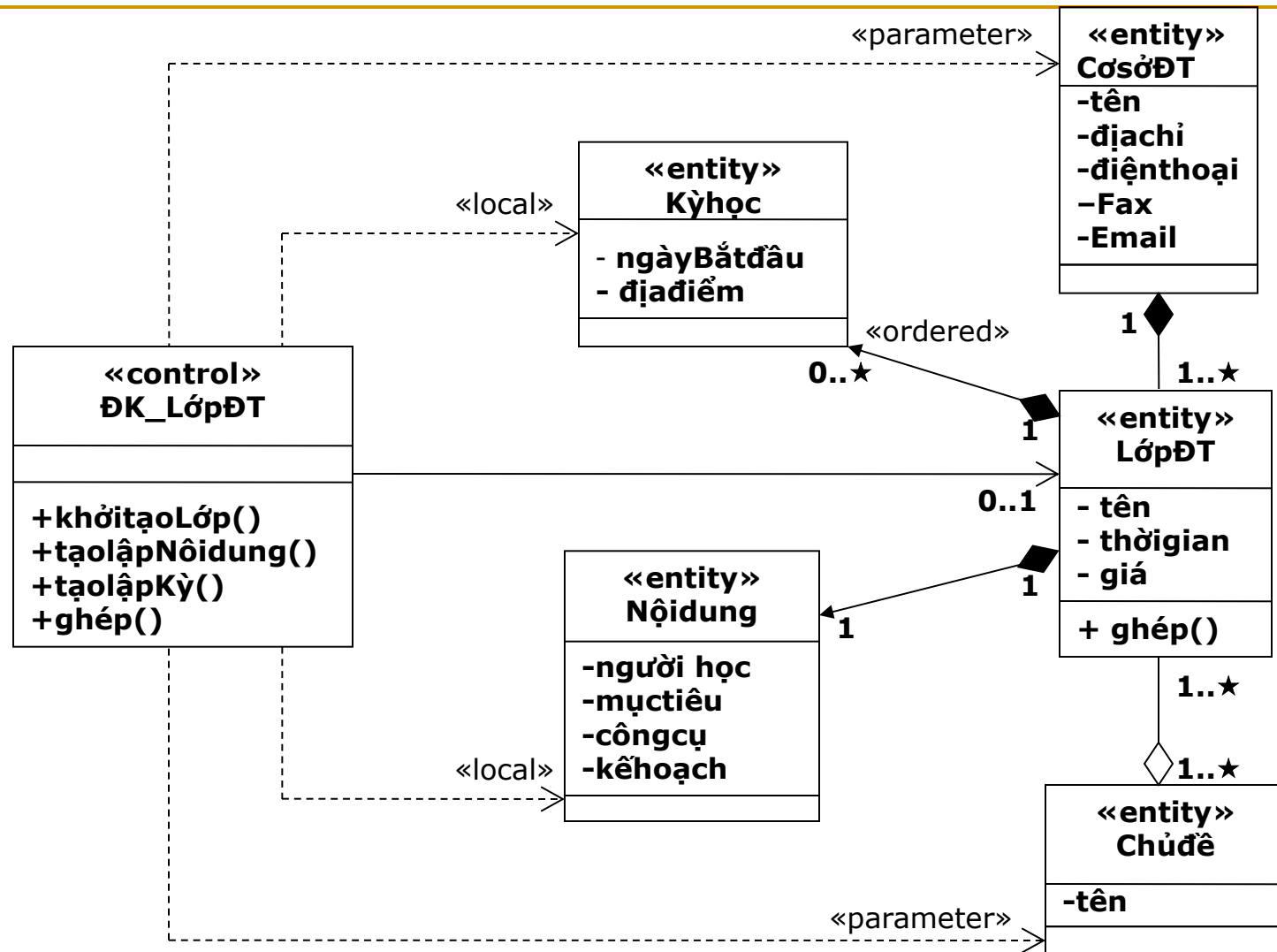


# Bài tập tổng hợp (17)

**Câu hỏi 20:** *Căn cứ trên biểu đồ giao tiếp vừa mới thành lập, hãy bổ sung nội dung cho các lớp tham gia.*

- Nội dung bổ sung gồm:
  - Các thao tác: Một thông điệp gửi tới một đối tượng phải tương ứng với một thao tác có sẵn trong lớp của đối tượng đó
  - Lưu hành: Tùy theo các kết nối giữa các đối tượng trong biểu đồ giao tiếp (giá đỡ cho các thông điệp) là «association», «self», hay «parameter», «global», «local» mà ta thiết lập liên kết hay phụ thuộc giữa hai lớp (theo chiều gửi thông điệp)
- Kết quả áp dụng 2 quy tắc trên vào biểu đồ giao tiếp đang xét là biểu đồ các lớp tham gia sau đây, trong đó để bớt rườm rà, ta lược bỏ thao tác «create» và các thao tác chung trên các lớp chứa đựng (như là add(...)):

# Bài tập tổng hợp (18)



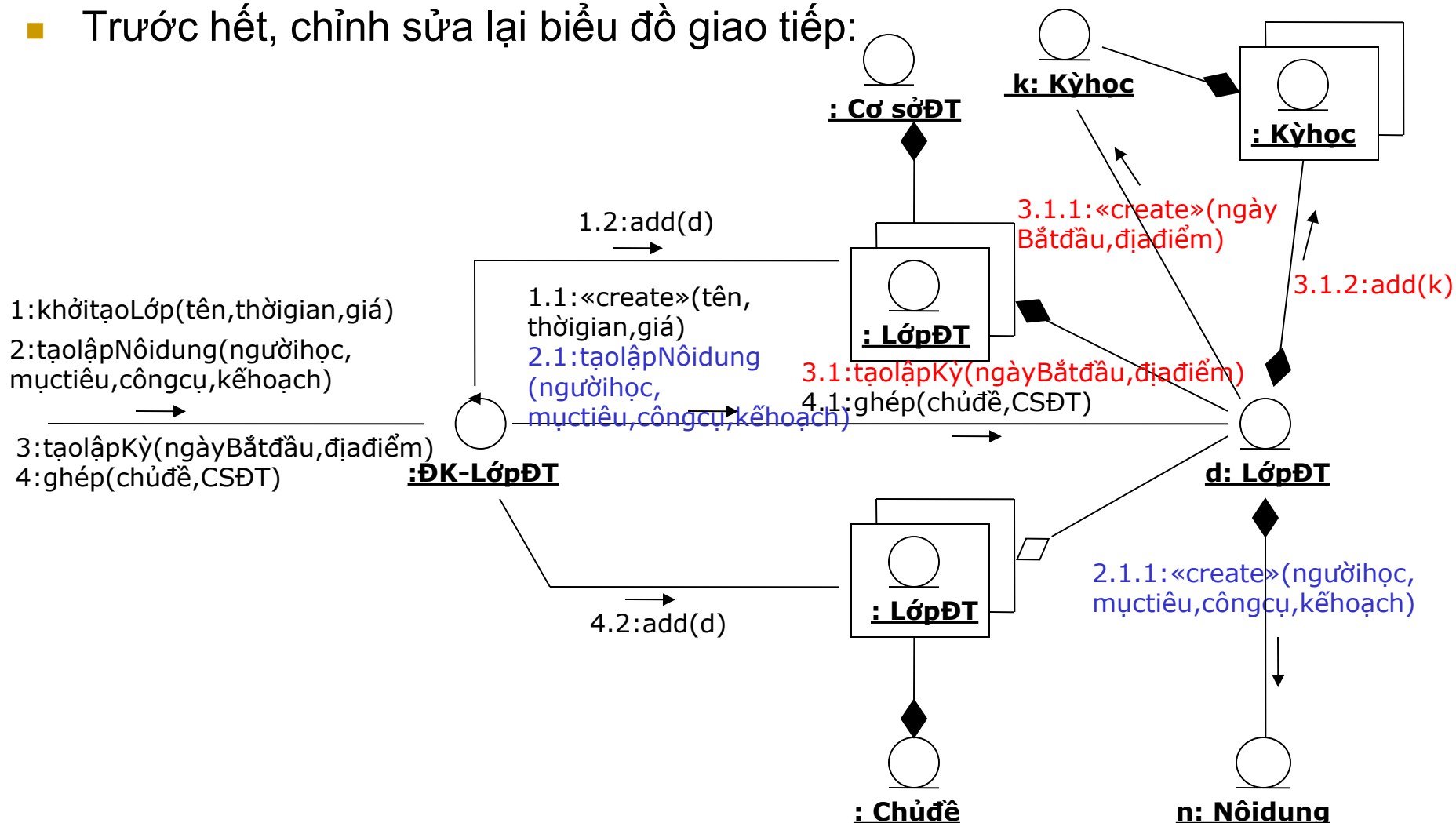
# Bài tập tổng hợp (19)

## Nhận xét và cải tiến mô hình:

- Các lớp tham gia vừa lập (đặc biệt là các lớp thực thể) đương nhiên là còn tham gia nhiều biểu đồ tương tác khác mà ta chưa xét tới; do vậy, những quyết định về chúng ở đây chỉ mới là tạm thời:
  - Sự chọn lựa hướng lưu hành trên các liên kết còn phải tham khảo thêm các biểu đồ tương tác khác.
  - Các phụ thuộc có thể sẽ trở thành các liên kết nếu các đối tượng đòi hỏi một kết nối lâu bền thay vì một kết nối nhất thời.
- Trong biểu đồ lớp vừa lập ta thấy lớp ĐK\_LớpĐT liên kết với mọi lớp còn lại
  - Đây là một hiện tượng nên tránh
  - Cần san sẻ bớt trách nhiệm (vd: các việc tạo lập các đối tượng Nội dung và Kỳ học) cho lớp khác (vd: cho lớp LớpĐT) như sau:

# Bài tập tổng hợp (20)

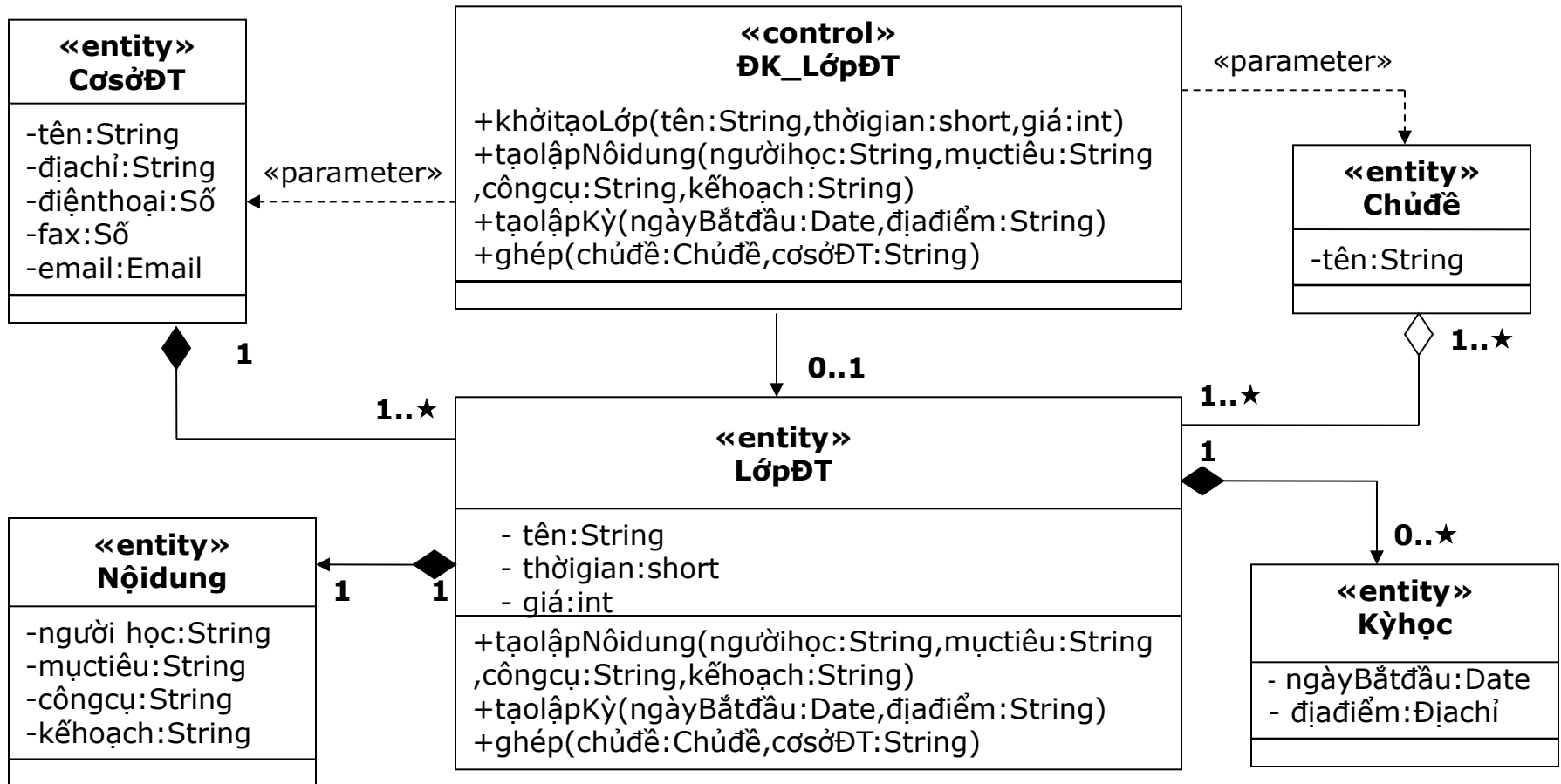
- Trước hết, chỉnh sửa lại biểu đồ giao tiếp:





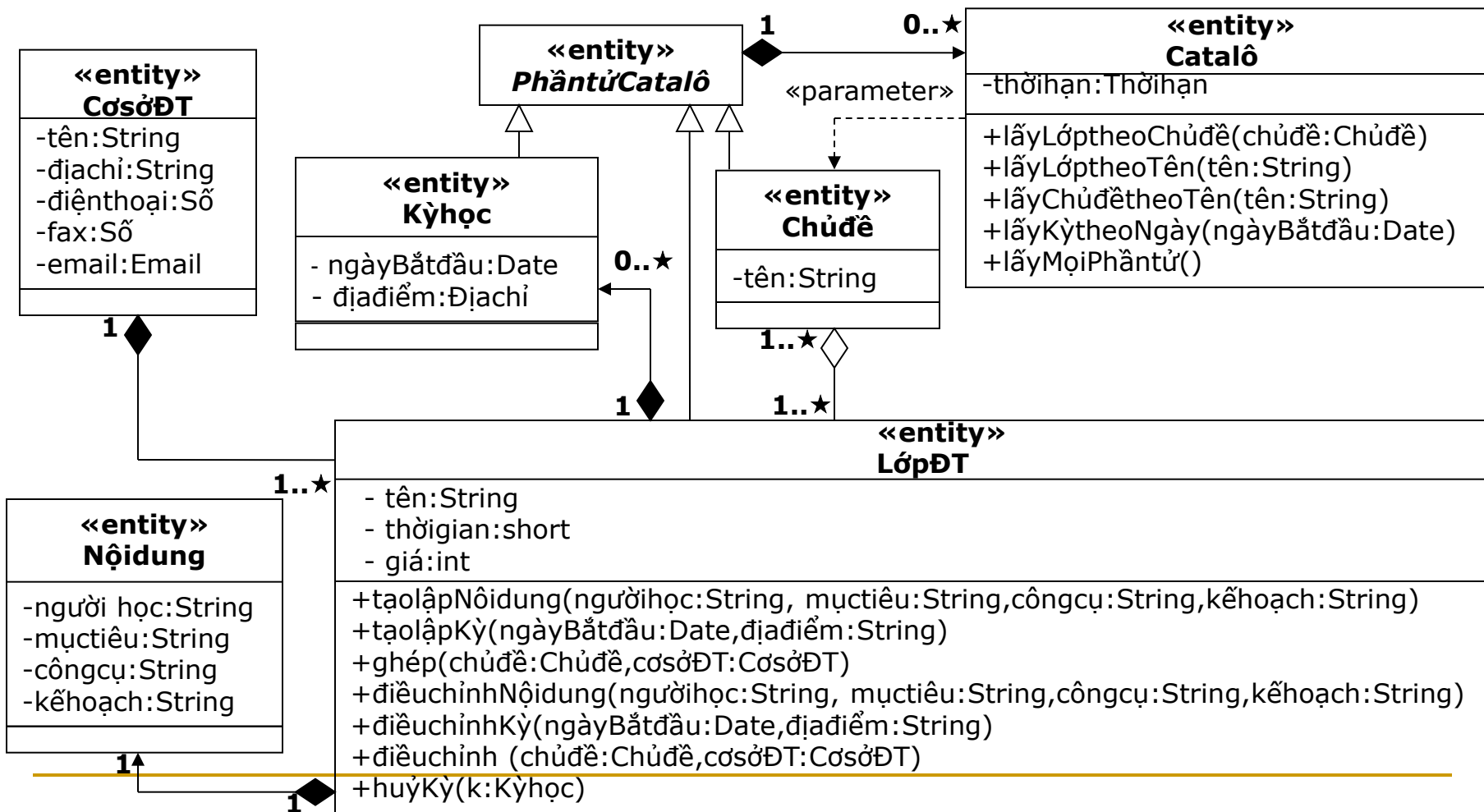
# Bài tập tổng hợp (21)

- Tiếp theo, từ biểu đồ giao tiếp đã chỉnh sửa, ta lập lại biểu đồ các lớp tham gia:



# Bài tập tổng hợp (22)

- Phân tích tiếp các phần còn lại của Vòng lặp 1, ta bổ sung thêm nhiều thông tin cho các lớp thực thể trong gói Catalô DT như sau:



# Bài tập tổng hợp (23)

**Câu hỏi 21:** *Định nghĩa các thao tác hệ thống cho Vòng lặp 2.*

- Đọc lại các bản mô tả của ca sử dụng 'Đề nghị đào tạo' và ca sử dụng 'Huỷ đào tạo' (hoặc xem biểu đồ trình tự hệ thống tương ứng), và lưu ý rằng các ca sử dụng này có 2 đối tác là Nhân viên và NgườiPTĐT, ta tìm ra các thao tác hệ thống cho Vòng lặp 2 như sau:

Hệ thống
lậpđơnxinĐT() chấpnhậnĐơn() từchốiĐơn() huỷĐơn() đềnghicácKỳ() chọnKỳ()

# Bài tập tổng hợp (24)

**Câu hỏi 22:** *Lập mô tả của các thao tác hệ thống: “lậpĐơnxinĐT” và “từ chốiĐơn”.*

- Lập mô tả cho thao tác hệ thống ‘lậpĐơnxinĐT’
  - 1) Tên: LậpĐơnxinĐT.
  - 2) Trách nhiệm: Lập một đơn xin đào tạo, dựa trên kết quả tra cứu từ catalô và gửi đơn đó cho người PTĐT để được duyệt.
  - 3) Tham chiếu: Ca sử dụng ‘Đề nghị ĐT’.
  - 4) Các điều kiện vào:
    - Catalô ĐT đã có trước.
    - Nhân viên đang kết nối với Internet.
    - Một đối tượng nv biểu diễn cho nhân viên đó đã có trong hệ thống.
  - 5) Các điều kiện ra:
    - Một đối tượng đơn xin đào tạo dx đã tạo lập.
    - Các thuộc tính ‘ngàygửi’ và ‘ngàyxinĐT’ của dx đã được khởi gán.
    - dx đã được kết nối với nv.
    - dx đã được kết nối với một phần tử trong catalô (điều có vẻ không phù hợp lắm với biểu đồ MHH nghiệp vụ).
  - 6) Ngoại lệ: Nhân viên có thể xin huỷ đơn xin của mình trước khi đơn được duyệt.

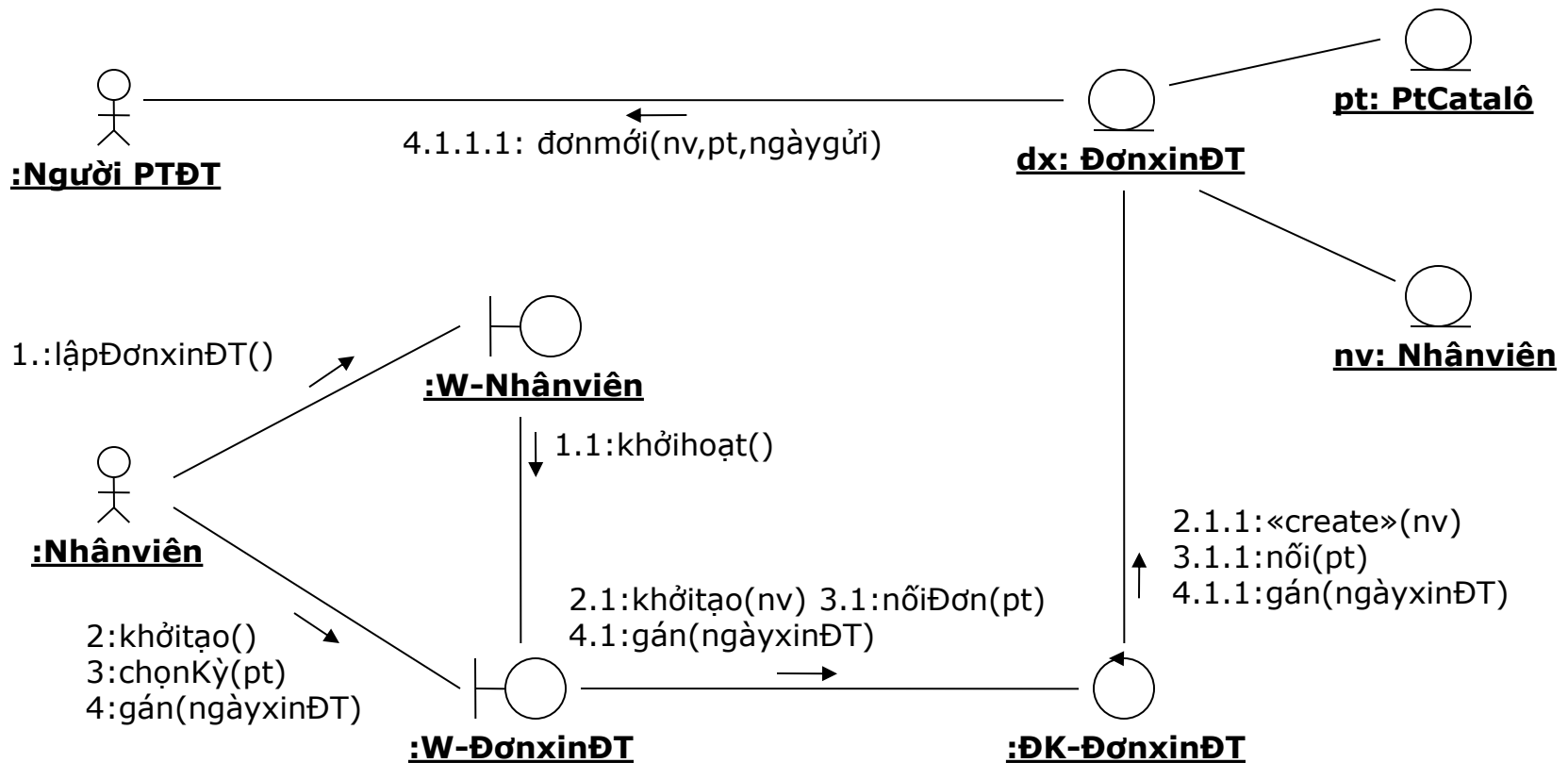
# Bài tập tổng hợp (25)

## Câu hỏi 22 (tiếp):

- Lập mô tả của thao tác HT: “từ chối Đơn”.
  - 1) Tên: từ chối Đơn.
  - 2) Trách nhiệm: Từ chối một đơn xin đào tạo của một nhân viên, có đưa ra lý do từ chối.
  - 3) Tham chiếu: Ca sử dụng ‘Đề nghị DT’.
  - 4) Các điều kiện vào:
    - Một đối tượng đơn xin đào tạo dx đã có trước.
    - Người PTĐT đang kết nối với Internet.
    - Một đối tượng nv biểu diễn cho nhân viên đã có trong hệ thống và đã kết nối với dx.
  - 5) Các điều kiện ra:
    - Đối tượng đơn xin đào tạo dx đã bị huỷ bỏ.
    - Một đối tượng từ chối t đã được tạo lập.
    - Các thuộc tính ‘ngày’ và ‘lý do’ của t đã được khởi gán.
    - Một e-mail chứa t đã được gửi cho nhân viên nv.
  - 6) Ngoại lệ: Không có.

# Bài tập tổng hợp (26)

- Câu hỏi 23:** Lập biểu đồ giao tiếp thực hiện thao tác hệ thống 'lậpĐơnxinĐT'



# Bài tập tổng hợp (27)

- Câu hỏi 24: Lập biểu đồ các lớp thực thể cho gói Đề nghị DT.

