

## Chương 5: Phân tích hành vi - Mô hình hóa sự tương tác

---

Nguyễn Hồng Hạnh, MSc  
Bộ môn Công nghệ Phần mềm  
Khoa Công nghệ thông tin - ĐHXD Hà Nội  
Email: hanhnh@huce.edu.vn

1

1

### Nội dung trình bày

---

1. Tổng quan
2. MHH sự tương tác với biểu đồ trình tự
3. MHH sự tương tác với biểu đồ giao tiếp
4. Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
5. Bài tập tổng hợp



2

2

# 1. Tổng quan

## 1.1 Hành vi là gì?

- **Hành vi** là sự hoạt động của các đối tượng nhằm tạo ra các kịch bản, bao gồm **tương tác** (trao đổi thông điệp) và **ứng xử** (phản ứng với các sự kiện).
- Các đối tượng sẽ trở nên vô nghĩa nếu chúng không tương tác với nhau để giải quyết vấn đề.
  - Mỗi đối tượng có trách nhiệm quản lý hành vi và trạng thái của nó.
  - Không một ai, không một đối tượng nào lại tự mình làm được mọi việc.

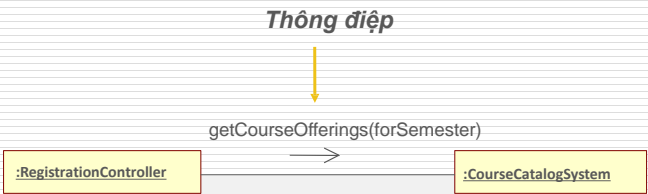


# 1. Tổng quan

- Các đối tượng tương tác với nhau như thế nào?
  - Chúng tương tác với nhau thông qua các thông điệp.
  - Cho biết làm thế nào mà một đối tượng yêu cầu một đối tượng khác thực hiện hành động.

Ví dụ:

Tương tác giữa *đối tượng đăng ký khóa học* với *hệ thống thông tin khóa học*



# 1. Tổng quan

## 1.2 Biểu đồ tương tác là gì?

- ❑ Biểu đồ tương tác (interaction diagram) được sử dụng trong UML để mô hình hóa khía cạnh động của hệ thống (hệ thống đang chạy).
- ❑ Biểu đồ tương tác bao gồm tập đối tượng, quan hệ và các thông điệp trao đổi giữa chúng.
- ❑ Mỗi biểu đồ tương tác chỉ ra từng bước của một luồng điều khiển cụ thể trong một yêu cầu chức năng (UC - usecase).
- ❑ Nếu một UC có nhiều luồng sự kiện thì ta sẽ sử dụng nhiều biểu đồ tương tác thể hiện cho từng luồng điều khiển đó.

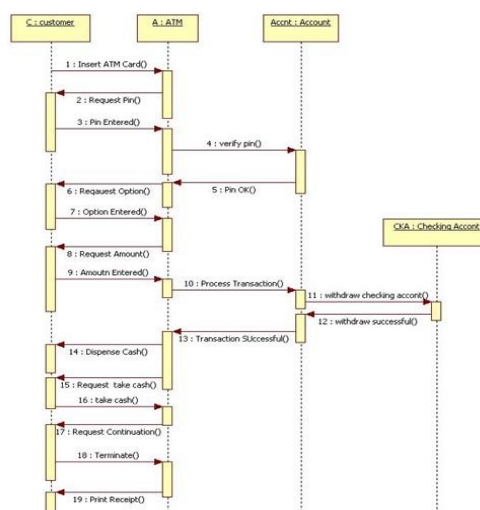
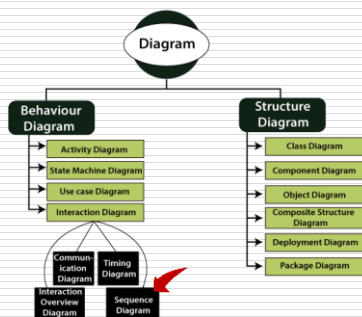


5

5

# 1. Tổng quan

## 1.2 Biểu đồ tương tác là gì?



6

# 1. Tổng quan

## 1.3 Mô hình hóa hệ thống bằng biểu đồ tương tác

❑ **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 cho mỗi ca sử dụng của hệ thống.

❑ Hai biểu đồ chính được sử dụng để diễn tả sự tương tác (một cách tương đương với nhau) là

- Biểu đồ trình tự (sequence diagram)
- Biểu đồ giao tiếp (collaboration diagram).



7

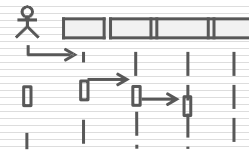
7

# 1. Tổng quan

## 1.3 Mô hình hóa hệ thống bằng biểu đồ tương tác

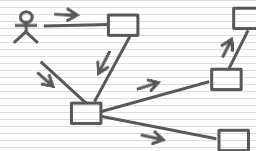
### ❑ Biểu đồ trình tự

- ❑ Một cách nhìn hướng về trình tự thời gian tương tác giữa các đối tượng.



### ❑ Biểu đồ giao tiếp

- ❑ Một cách nhìn thông điệp giữa các đối tượng hướng về cấu trúc của quá trình truyền thông điệp giữa các đối tượng.



8

8

# 1. Tổng quan

## 1.3 Mô hình hóa hệ thống bằng biểu đồ tương tác

- Dù hai biểu đồ này là khác nhau về hình thức thể hiện, song khi tạo 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.



9

9

## Nội dung trình bày

1. Tổng quan
2. MHH sự tương tác với biểu đồ trình tự
3. MHH sự tương tác với biểu đồ giao tiếp
4. Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
5. Bài tập tổng hợp



10

10

## 2. MHH sự tương tác với biểu đồ trình tự

### 2.1. Biểu đồ trình tự

### 2.2. Các thông điệp

### 2.3. MHH tương tác trong ca sử dụng với bđ trình tự

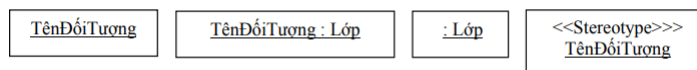


11

11

## 2.1. Biểu đồ trình tự

- ❑ **Biểu đồ trình tự** (Sequence Diagram) được sử dụng để xác định và chỉ rõ vai trò của các đối tượng tham gia vào luồng sự kiện của use case
- ❑ Là một loại biểu đồ tương tác, mô tả sự tương tác giữa các đối tượng, trong đó nhấn mạnh vào *trình tự thời gian của các thông điệp trao đổi giữa các đối tượng* đó.
- ❑ Các đối tượng ở đây thường là các cá thể có tên hay khuyết danh của các lớp, cũng còn có thể là các tác nhân, các thành phần hệ thống.



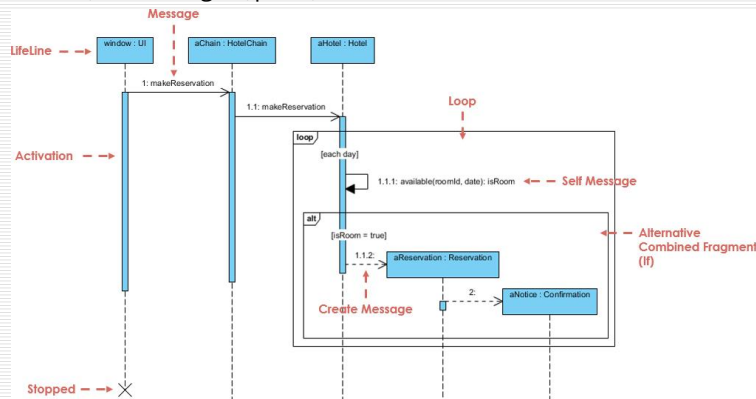
12

12

## 2.1. Biểu đồ trình tự

### □ Biểu đồ trình tự chỉ ra:

- Các đối tượng tham gia vào tương tác.
- Thời gian sống của các đối tượng
- Trình tự các thông điệp được trao đổi.



13

13

## 2.1. Biểu đồ trình tự

### □ 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 **đường đời (lifeline)**. Đường đời của đối tượng sẽ kết thúc bằng một dấu gạch chéo, khi đối tượng bị huỷ bỏ.

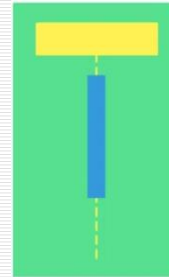


14

14

## 2.1. Biểu đồ trình tự

- 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 đường 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.

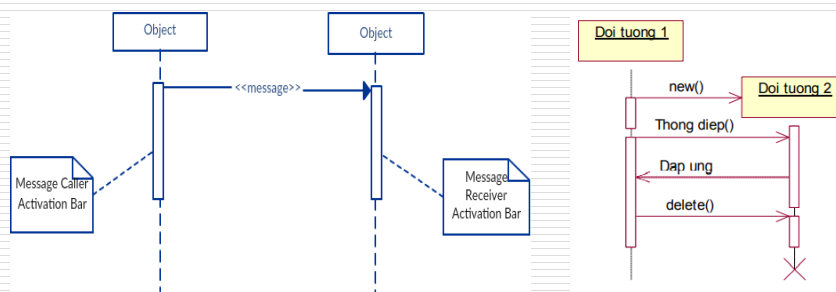


15

15

## 2.1. Biểu đồ trình tự

- Các thông điệp là những mũi tên nằm ngang nối đường đời của hai đối tượng và được vẽ lần lượt từ trên xuống theo thứ tự thời gian..



16

16



## 2.1 Biểu đồ trình tự

- ❑ Mỗi thông điệp gửi đi mang theo tên thông điệp dưới dạng **thông điệp (ds tham số)**.
- ❑ 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. Gọi đó là một **lộ trình điều khiển** (thread of control).
- ❑ Lộ trình điều khiển có thể **phẳng** (tuyến tính) hay **lồng** (do có sự trả lại).

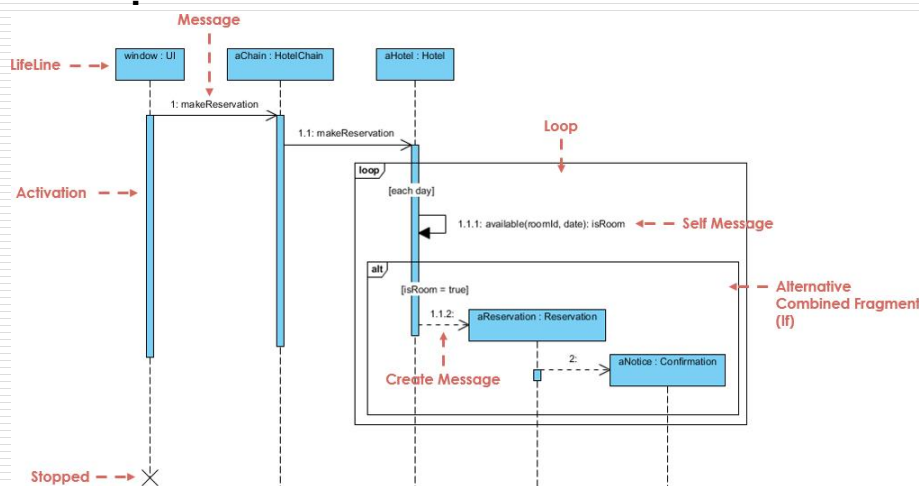


17

17

## 2.1 Biểu đồ trình tự

### ❑ Ví dụ

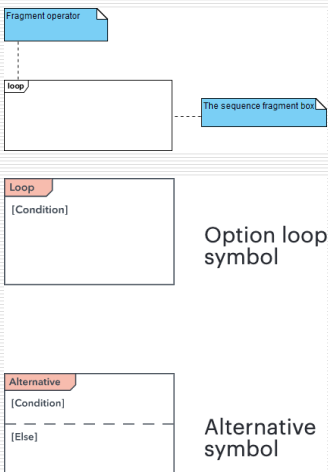


18

18

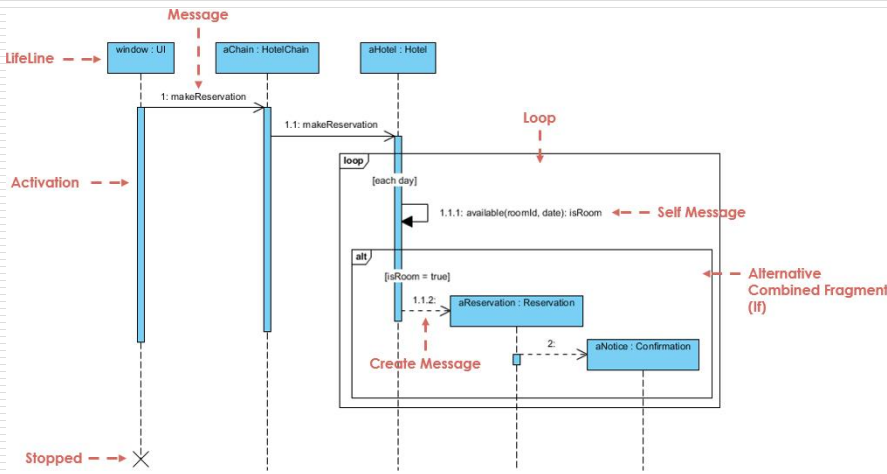
## 2.1. Biểu đồ trình tự

Khung tương tác (Combined fragment):  
mô tả những tương tác trong hệ thống  
được phân mảnh và biểu diễn dưới dạng  
phẳng. Một khung tương tác được định  
nghĩa bởi một toán tử tương tác  
(interaction operator)

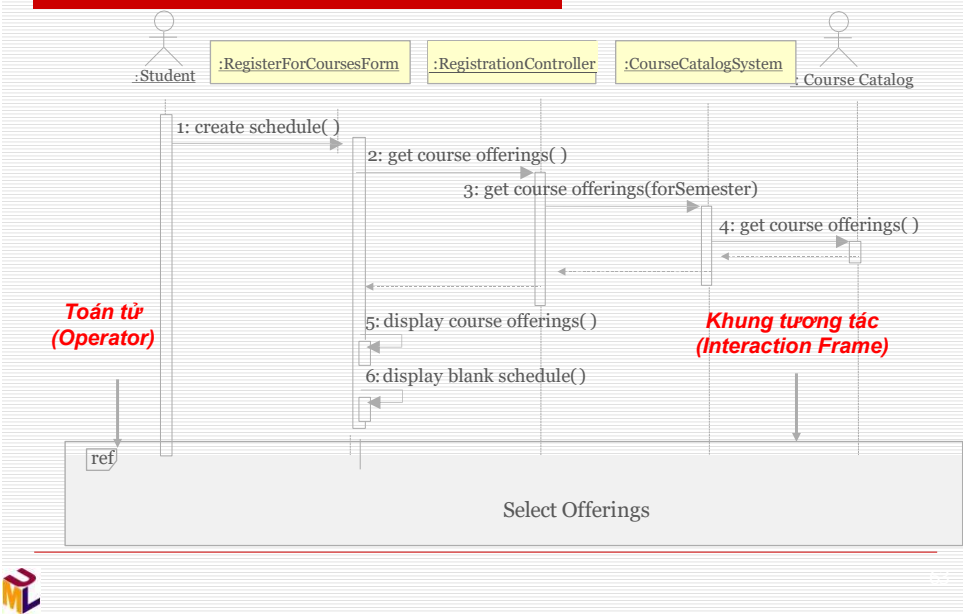


## 2.1. Biểu đồ trình tự

Ví dụ: biểu đồ trình tự UC: Xác nhận đặt phòng khách sạn



Biểu đồ trình tự: Khung tương tác



21

2.1 Biểu đồ trình tự

□ Ý nghĩa toán tử tương tác

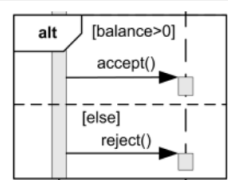
Toán tử	Ý nghĩa
alt	Khung lựa chọn nhiều, chỉ có lựa chọn có điều kiện đúng sẽ được thực hiện
opt	Tùy chọn, chỉ thực hiện khi điều kiện thỏa mãn
par	Song song, mỗi khung chạy song song
loop	Lặp lại, khung có thể được thực hiện nhiều lần
region	Vùng then chốt, tại một thời điểm chỉ có một luồng chạy nó
ref	Tham chiếu đến một tương tác khác trong biểu đồ khác, vẽ trùm trên các lifetime liên quan, có thể có tham số và giá trị trả về
seq	Song song nhưng có ưu tiên tuần tự ở mức yếu



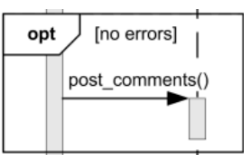
22

22

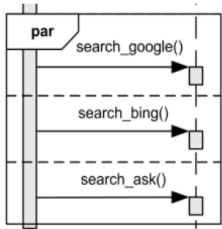
## 2.1 Biểu đồ trình tự



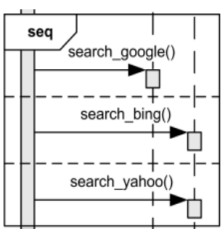
Call accept() if balance > 0, call reject() otherwise.



Post comments if there were no errors.

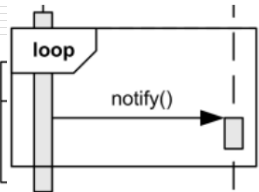


Search Google, Bing and Ask in any order, possibly parallel.

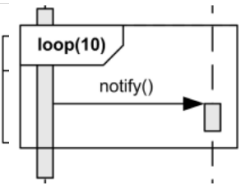


Search Google possibly parallel with Bing and Yahoo, but search Bing before Yahoo.

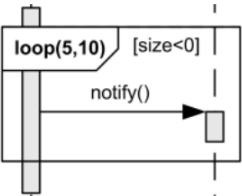
## 2.1 Biểu đồ trình tự



Potentially infinite loop.



Loop to execute exactly 10 times.



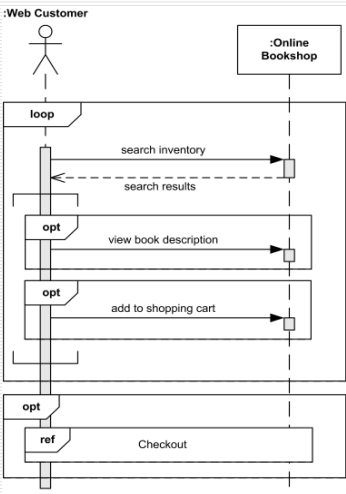
We may guess that as per UML 2.3, the loop is expected to execute minimum 5 times and no more than 10 times.  
If guard condition [size < 0] becomes false loop terminates regardless of the minimum number of iterations specified.

## 2.1 Biểu đồ trình tự

**Khung tương tác.**

**Ví dụ 1: Online bookshop**

- ✓ Tìm kiếm các cuốn sách ( tìm lặp đi lặp lại cho tới lúc tìm thấy cuốn sách thì thôi
- ✓ Cho phép nhìn thấy danh sách kết quả tìm kiếm
- ✓ Xem chi tiết thông tin sách ( có thể lựa chọn thực hiện)
- ✓ Thêm sách vào giỏ hàng ( có thể lựa chọn thực hiện)
- ✓ Xác nhận giỏ hàng và thanh toán (có thể lựa chọn thực hiện

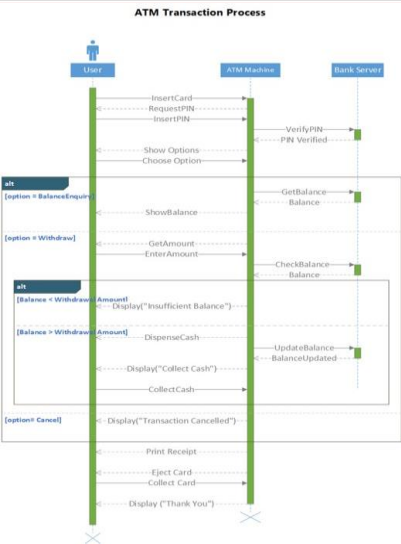


## 2.1 Biểu đồ trình tự

**Khung tương tác**

**Ví dụ 2: Mô tả quy trình giao dịch ATM**

- ✓ Người dùng đưa thẻ ATM vào máy → Hệ thống yêu cầu nhập mã PIN
- ✓ Người dùng nhập mã PIN → Hệ thống xác thực mã PIN
- ✓ Người dùng nhìn thấy màn hình giao dịch chính và có thể đưa ra các lựa chọn
- ✓ Người dùng có thể xem số dư tài khoản
- ✓ Người dùng có thể rút tiền
- ✓ Người dùng có thể hủy bỏ giao dịch
- ✓ Người dùng nhận lại thẻ ATM và hóa đơn giao dịch
- ✓ ATM hiển thị thông điệp “Thanks You” trên màn hình



## 2. MHH sự tương tác với biểu đồ trình tự

### 2.1. Biểu đồ trình tự

### 2.2. Các thông điệp

### 2.3. MHH tương tác trong ca sử dụng với bề trình tự



28

28

## 2.2 Các thông điệp

- **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 thông điệp.
  - **Tự gọi** (self-call) 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.
  - **Hủy bỏ** (destroy): Hủy một đối tượng. Một đối tượng cũng có thể hủy bỏ chính nó (terminate).









29

29

## 2.2 Các thông điệp

- các loại thông điệp sau:
  - Thông điệp đồng bộ (Synchronous call)
  - Thông điệp không đồng bộ (Asynchronous call)
  - Thông điệp trả về (Reply message)
  - Thông điệp mất hút (Lost message)
  - Thông điệp tìm thấy (Found message)
  - Thông điệp tạo lập (Create message)
  - Thông điệp hủy bỏ (Delete message)

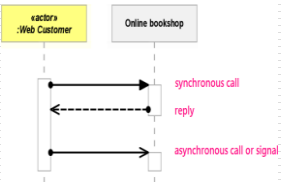
Symbol	Name	Description
	Synchronous message symbol	Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.
	Asynchronous message symbol	Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.
	Asynchronous return message symbol	Represented by a dashed line with a lined arrowhead.
	Asynchronous create message symbol	Represented by a dashed line with a lined arrowhead. This message creates a new object.
	Reply message symbol	Represented by a dashed line with a lined arrowhead, these messages are replies to calls.
	Delete message symbol	Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

## 2.2 Các thông điệp

- Thông điệp đồng bộ**, biểu diễn bằng mũi tên đầu tam giác đặc:
  - A *synchronous message* được sử dụng khi bên gửi chờ bên nhận xử lý thông điệp và trả lại kết quả quyền điều khiển rồi mới chuyển sang xử lý một thông điệp khác tiếp theo.



- Bên gửi chờ cho đến khi bên nhận hoàn thành xong việc xử lý thông điệp.
- Bên gửi chỉ có thể tiếp tục làm việc tiếp khi nó biết rằng bên nhận đã xử lý xong thông điệp trước đó mà nó đã gửi đi. (ví dụ: nhận được thông điệp phản hồi - reply message)

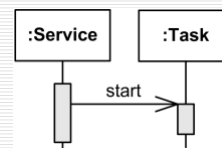


## 2.2 Các thông điệp

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



- 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.

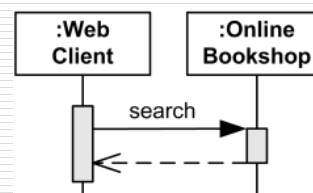
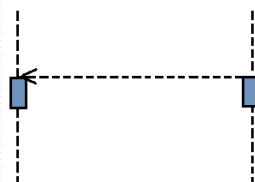


32

32

## 2.2 Các thông điệp

- **Thông điệp trả về (reply message)** 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, báo hiệu đã hoàn thành xong thông điệp và trả lại quyền điều khiển cho bên nhận.



33

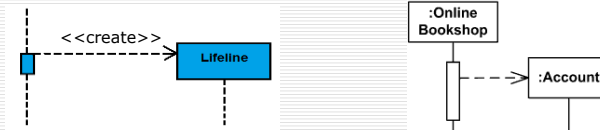
33



## 2.2 Các thông điệp

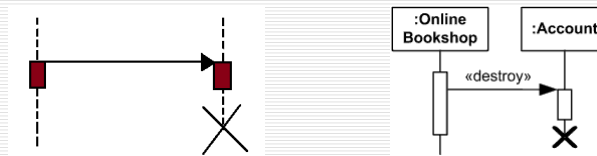
- **Thông điệp tạo lập** (create message): là thông điệp được gửi để yêu cầu tạo lập một đối tượng.

Biểu diễn:



- **Thông điệp hủy bỏ** (found message): là thông điệp được gửi để yêu cầu hủy bỏ (kết thúc hoạt động) của một đối tượng nào đó.

Biểu diễn



34

34

## 2.2 Các thông điệp

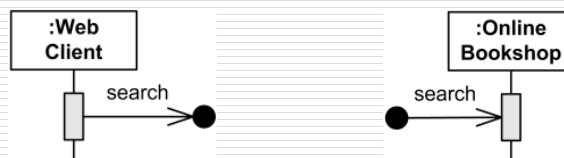
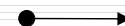
- **Thông điệp mất hút** (lost message): là thông điệp mà nơi phát thì biết rõ, song nơi nhận thì không được biết (vì ở ngoài phạm vi mô tả, hoặc vì đó là một sự phát tán).

Biểu diễn bằng một mũi tên có hình tròn đen ở cuối:



- **Thông điệp tìm thấy** (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).

Biểu diễn bằng một mũi tên có hình tròn đen ở gốc:

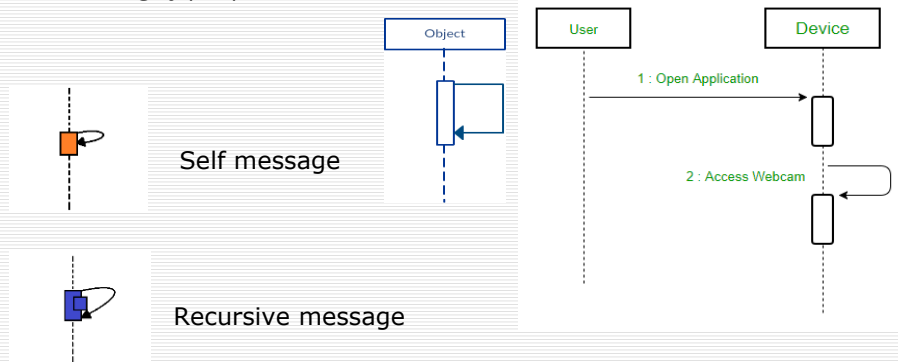


35

35

## 2.2 Các thông điệp

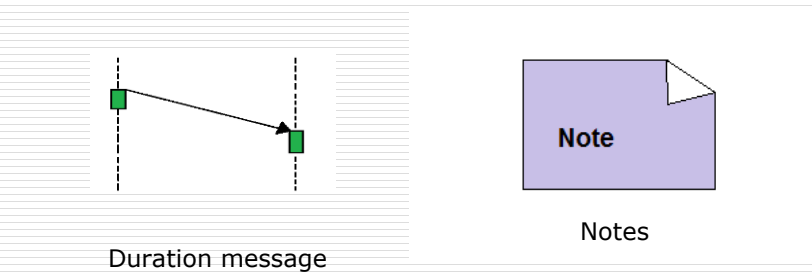
- **Thông điệp phản thân** (reflexive message): là thông điệp mà đối tượng tự gọi chính nó. Điểm xuất phát của mũi tên thông điệp và điểm kết thúc của nó nằm trên cùng một đường đời của một đối tượng.
- Sử dụng ký pháp mũi tên chữ U



36

36

## 2.2 Các thông điệp



37

37

## 2. MHH sự tương tác với biểu đồ trình tự

### 2.1. Các thông điệp

### 2.2. Biểu đồ trình tự

### 2.3. MHH tương tác trong ca sử dụng với biểu đồ trình tự



38

38

### 2.3. MHH tương tác trong ca sử dụng với biểu đồ trình tự

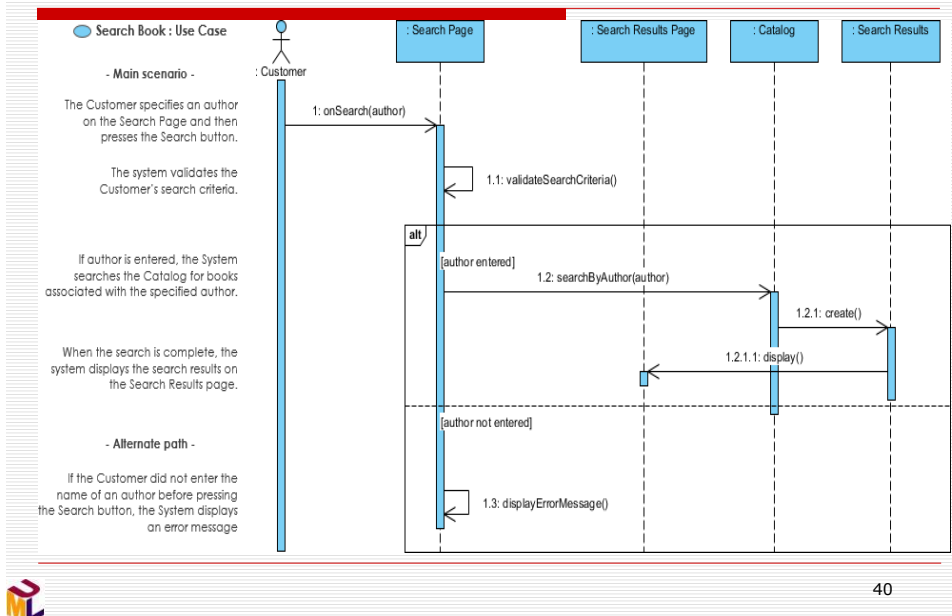
- Một ca sử dụng có thể gồm nhiều kịch bản tương ứng với nhiều luồng điều khiển khác nhau. Một biểu đồ trình tự chỉ có thể diễn tả một luồng điều khiển, mặc dầu trong biểu đồ cũng có thể diễn tả sự rẽ nhánh hay lặp đơn giản.
- Vậy *thông thường ta lập một số biểu đồ trình tự cho một ca sử dụng*. Một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



39

39

### 2.3. MHH tương tác trong ca sử dụng với bể trình tự



40

### 2.3. MHH tương tác trong ca sử dụng với bể trình tự

□ Các bước tiến hành để lập một biểu đồ trình tự cho một kịch bản của ca sử dụng

- **Bước 1:** 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, dõi theo từng bước trong kịch bản để xem các đối tượng đóng vai trò gì trong bước đó (biên (view- form- boundary), điều khiển (controller), thực thể(model -entity).
- **Bước 2:** Dàn các đối tượng thành hàng ngang trên đỉnh biểu đồ trình tự. Bố trí các đối tượng quan trọng ở bên trái, các đối tượng phụ trợ đặt ở bên phải. Tác nhân, nếu có, thì đặt ở lề trái.



41

41

### 2.3. MHH tương tác trong ca sử dụng với bề trình tự

- **Bước 3:** Vẽ đường 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.
- **Bước 4:** 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 đường đời. Chỉ rõ đặc điểm của mỗi thông điệp (như là các tham số của nó). Nếu cần thì cho thêm giải thích về ngữ nghĩa của tương tác.



42

42

### 2.3. MHH tương tác trong ca sử dụng với bề trình tự

- **Bước 5:** 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 đường 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 thời gian thích hợp.



43

43

## 2.3. MHH tương tác trong ca sử dụng với bđ trình tự

### Các bước thực hành để phát triển biểu đồ Trình tự:

- ✓ **Step 1:** Pick a use case to develop sequence diagram
- ✓ **Step 2:** Identify all objects or actors that would be involved in the sequence
- ✓ **Step 3:** Write a detailed description on what the use case does. From this description, figure out the interactions (that should go in the sequence diagram) that would occur between the objects above
- ✓ **Step 4:** From each of above interaction (in step 3), specify what messages should be exchanged between the objects in the sequence diagram. Once it's clear, you can go ahead and start drawing the sequence diagram.
- ✓ **Step 5:** Repeat Step 4 for all of interactions

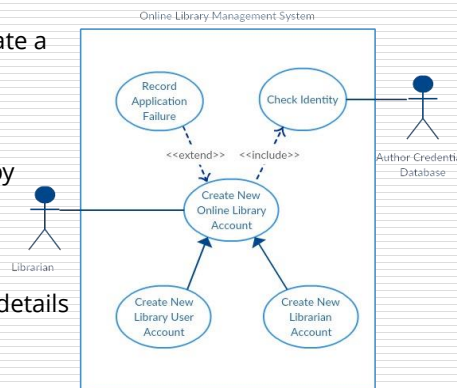


44

## 2.3. MHH tương tác trong ca sử dụng với bđ trình tự

### UC: Create new online library account

- The librarian requests the system to create a new online library account
- The librarian enters the user's details
- The system will verify user information by checking User Credentials Database
- The new library user account is created
- A summary of the of the new account's details are then emailed to the user



Use case diagram

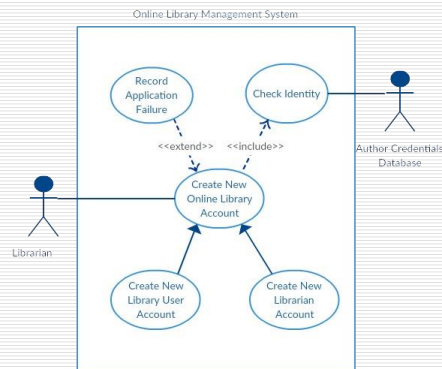


45

## 2.3. MHH tương tác trong ca sử dụng với bđ trình tự

### □ Step 1: Pick a use case to develop sequence diagram

✓ We will focus on the use case named 'Create New User Account' to draw sequence diagram



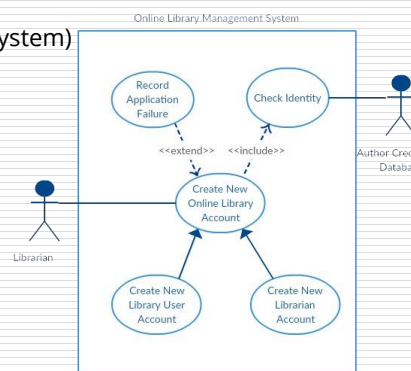
46

## 2.3. MHH tương tác trong ca sử dụng với bđ trình tự

### Step 2:

✓ Identify the objects or participants in the use case "Create New User Account":

- Librarian
- Online Library Management System (System)
- User credentials database (Database)
- Email System / Service



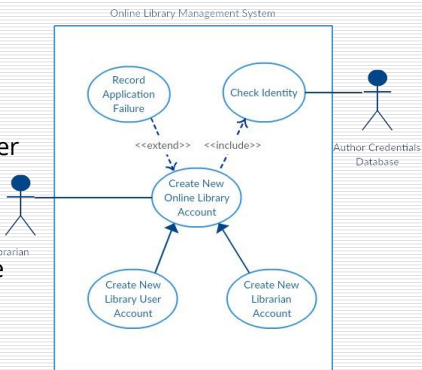
47

2.3. MHH tương tác trong ca sử dụng với bđ trình tự

Step 3:

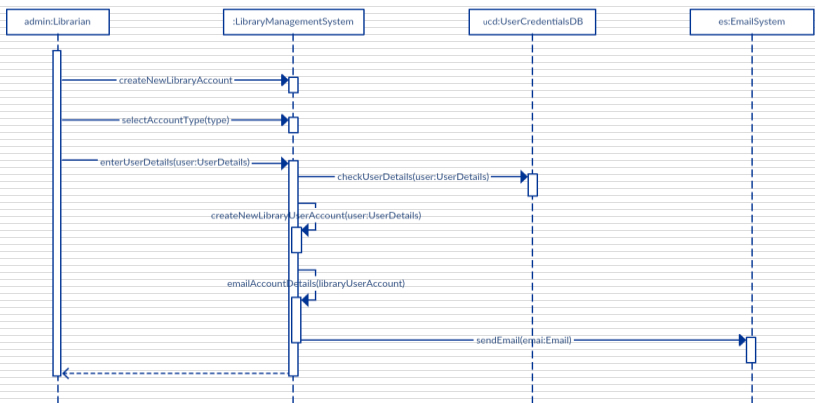
✓List down the steps involved in the execution of the use case:

- The librarian requests the system to create a new online library account
- The librarian then selects the library user account type
- The librarian enters the user's details
- The user's details are checked using the user Credentials Database
- The new library user account is created
- A summary of the of the new account's details are then emailed to the user



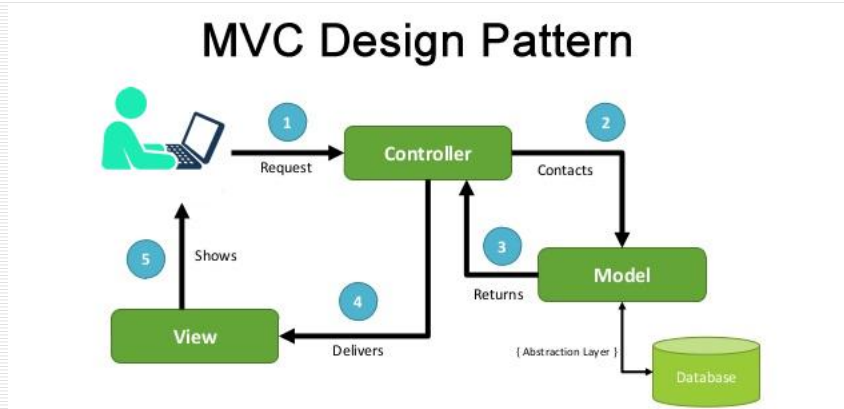
2.3. MHH tương tác trong ca sử dụng với bđ trình tự

Step 4 & 5: Sequence diagram





## Bài tập ví dụ



50

50

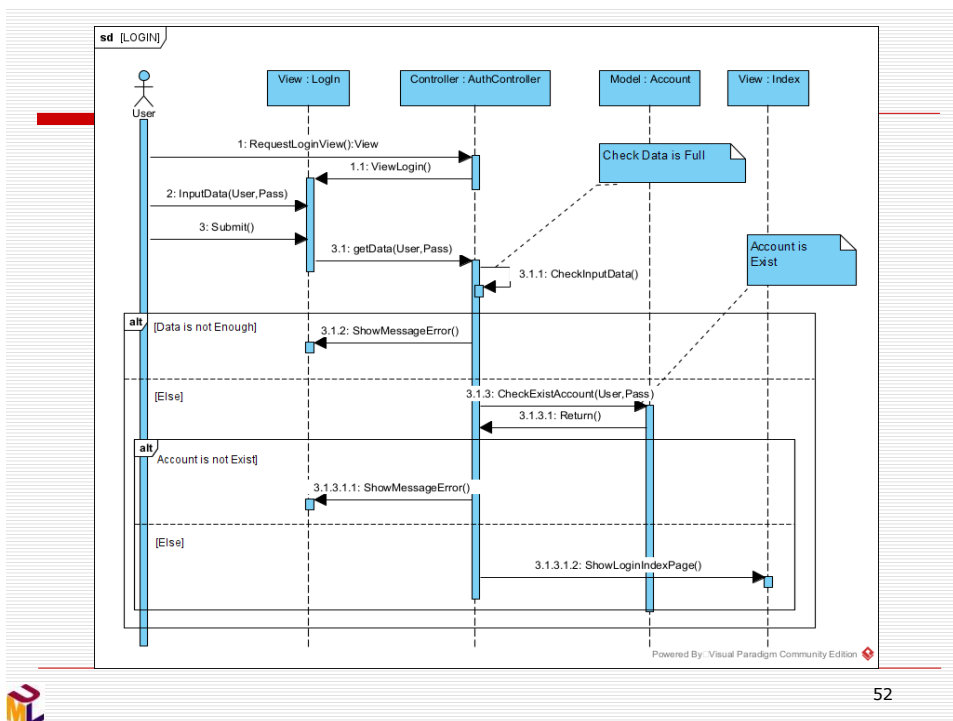
## Bài tập ví dụ

Vẽ ví dụ Demo vẽ Sequence Diagram cho: Usecase Login qua tài khoản MyApp trong hệ thống e\_Commere MyApp



51

51



52

52

## Nội dung

1. Tổng quan
2. MHH sự tương tác với biểu đồ trình tự
3. MHH sự tương tác với biểu đồ giao tiếp
4. Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
5. Bài tập tổng hợp



53

53

### 3. MHH sự tương tác với bđ giao tiếp

#### 3.1. Bđ giao tiếp

#### 3.2. MHH tương tác trong ca sử dụng với bđ giao tiếp



54

54

### 3.1. Biểu đồ giao tiếp

- ❑ **Biểu đồ giao tiếp** (Communication Diagram) là một trong hai biểu đồ tương tác chính, với chủ ý làm nổi bật khung cảnh tổ chức của sự tương tác. Tên cũ trong UML 1.x là biểu đồ hợp tác (Collaboration diagram).
- ❑ Nó 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 nói đây thường là các cá thể có tên hay khuyết danh của các lớp, song thể vào chỗ các đối tượng, cũng còn có thể là các đối tác, các hợp tác, 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.

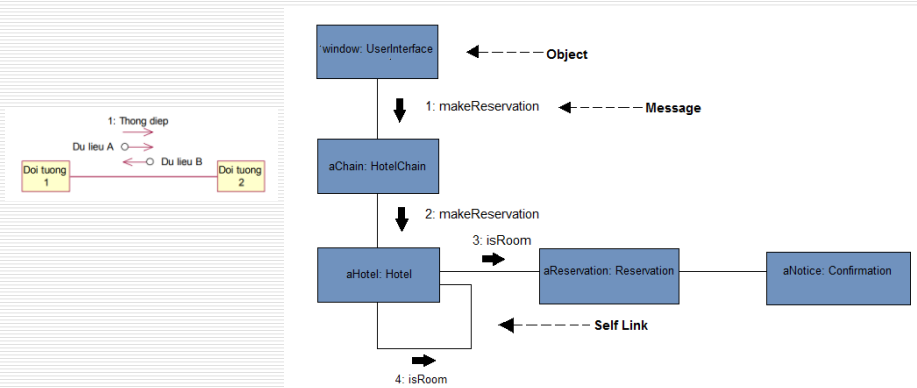


55

55

### 3.1 Biểu đồ giao tiếp

#### □ Hình ảnh minh họa:



56

56

### 3.1. Biểu đồ giao tiếp

- Các đối tượng cũng như 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  $\longrightarrow$ ) phải kèm theo tiêu đề của thông điệp mà cú pháp đầy đủ là như sau:

[Bước trước] [Biểu thức trình tự] [Trả lời :=] Tên Thông điệp (DsThamsố)  
 trong đó trừ Tên Thông điệp là bắt buộc, còn các phần khác đều là tùy ý.



57

57

### 3.1. Biểu đồ giao tiếp

- **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.
- Vậy đây là một yêu cầu đồng bộ hoá của lộ trình điều khiển.  
Thí dụ: 1.1, 2.3/



58

58

### 3.1. Biểu đồ giao tiếp

- **Biểu thứ 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.



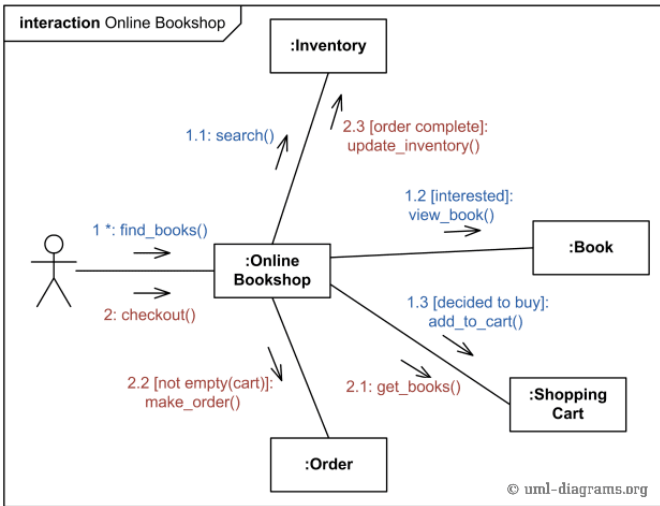
59

59

### 3.1. Biểu đồ giao tiếp

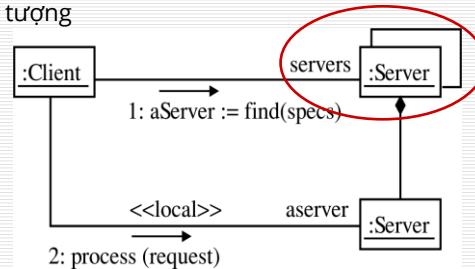
- Các số lại có thể thay bởi các dãy ký tự, nhằm mục đích:
  - Diễn tả **lộ trình điều khiển**, như là A3, B4/ C3.1.2:
  - Diễn tả **điều kiện chọn**, như là 4.2. [x<0]:
  - Diễn tả **thông điệp lặp**, như là 1.1.\*: hay 2.1.\* [i:=1..n]:
- **Trả lời:** là tên của kết quả trả về bên gửi. Tên này sau 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.
- **TênThôngđiệp(DsThamsố):** là tên của thông điệp, kèm theo danh sách tham số (có thể rỗng). Tên này thường lấy trùng với thao tác cần được huy động ở bên nhận.

### 3.1. Biểu đồ giao tiếp



### 3.1. Biểu đồ giao tiếp

- 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. Lúc này ta nên diễn tả bằng **đối tượng bộ**.
  - **Đối tượng bộ (multiobject)** là một ký pháp UML dùng để biểu diễn trong chỉ một ký hiệu có nhiều đối tượng của cùng một lớp. Nó được vẽ thành hai đối tượng chồng lên nhau.
- Dùng đối tượng bộ, 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 ....



63

63

## 3. MHH sự tương tác với bđ giao tiếp

### 3.1. Bđ giao tiếp

### 3.2. MHH tương tác trong ca sử dụng với bđ giao tiếp



64

64

### 3.2. MHH tương tác trong ca sử dụng với bđ giao tiếp

□ Để lập biểu đồ giao tiếp cho một kịch bản của ca sử dụng ta tiến hành các bước sau:

- **Bước 1:** 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, dõi theo từng bước trong kịch bản để xem đối tượng đóng vai trò gì trong bước đó.
- **Bước 2:** 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.



65

65

### 3.2. MHH tương tác trong ca sử dụng với bđ giao tiếp

- **Bước 3:** 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 gán 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 đập như là <<become>> hay <<copy>>.
- **Bước 4:** Xác định các kết nối giữa các đối tượng, cùng với các thông điệp có thể có trên đó.
  - Vẽ các kết nối liên kết trước; đó là những kết nối quan trọng nhất vì chúng diễn tả cấu trúc tĩnh.
  - Tiếp theo vẽ các kết nối khác và ghi chú với các khuôn đập thích hợp (như là <<global>>, <<local>>) để làm rõ các đối tượng liên hệ với nhau theo kiểu gì.



66

66



### 3.2. MHH tương tác trong ca sử dụng với bề giao tiếp

- **Bước 5:** 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.



67

67

### Nội dung trình bày

1. Tổng quan
2. MHH sự tương tác với biểu đồ trình tự
3. MHH sự tương tác với biểu đồ giao tiếp
4. Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
5. Bài tập tổng hợp



68

68

## 4. Đối chiếu và chỉnh sửa các mô hình

- Tới 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 là như sau:

- 4.1. Thêm bớt các lớp
- 4.2. Thêm và chỉnh lý các thao tác trong các lớp
- 4.3. Thêm các kiểu cho các thuộc tính, các tham số và các trả lời
- 4.4. Thêm và chỉnh lý các mối liên quan giữa các lớp



69

69

### 4.1. Thêm bớt các lớp

- Một số lớp đã được phát hiện ra 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.



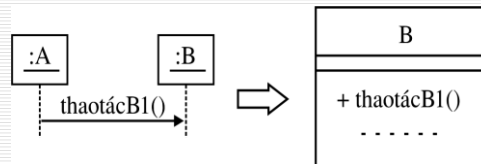
70

70

## 4.2. Thêm và chỉnh lý các thao tác trong các lớp

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 <<public>> 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.



71

71

## 4.2. Thêm và chỉnh lý các thao tác trong các lớp

Tuy nhiên quy luật trên cũng có những ngoại lệ:

- ❑ Thông điệp đến một tác nhân:
  - 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 giao thức thích hợp. Bấy giờ thông điệp sẽ kích hoạt một thao tác trong lớp này.



72

72

## 4.2. Thêm và chỉnh lý các thao tác trong các lớp

- 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.



73

73

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

- Các kiểu có thể là các kiểu nguyên thuỷ hay các kiểu do người dùng đưa vào, hoặc 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.



74

74

#### 4.4. Thêm và chỉnh lý 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 (hay ở trong tầm nhìn) của bên gửi:
  - Nếu sự hiểu biết đó là 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.



75

75

## Nội dung

1. Tổng quan
2. MHH sự tương tác với biểu đồ trình tự
3. MHH sự tương tác với biểu đồ giao tiếp
4. Đối chiếu, chỉnh sửa các MH cấu trúc và tương tác
5. Bài tập tổng hợp



76

76

## 5. Bài tập tổng hợp

□ 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 SD, song có khi là một gói ca SD.
- 2) Xác định các **thao tác hệ thống**: Đó là các yêu cầu mà một đối tác đưa ra đối với HT (tức là một kịch bản mà đối tác khởi động).
- 3) Lập một BD tương tác cho mỗi thao tác HT.



77

77

## 5. Bài tập tổng hợp

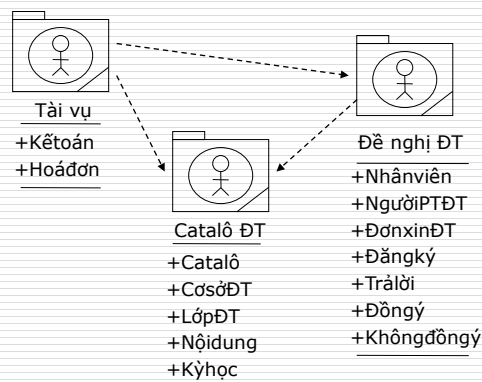
**Câu hỏi 17:** Xác định các vòng lặp cho việc MHH tương tác.

□ Nhìn các phụ thuộc trong

BD gói, ta thấy một cách tự nhiên là nên bắt đầu với gói Catalô ĐT, rồi đến gói Đề nghị ĐT.

□ Vậy trên BD ca SD

ta chia cắt các vòng lặp như ở trang sau.

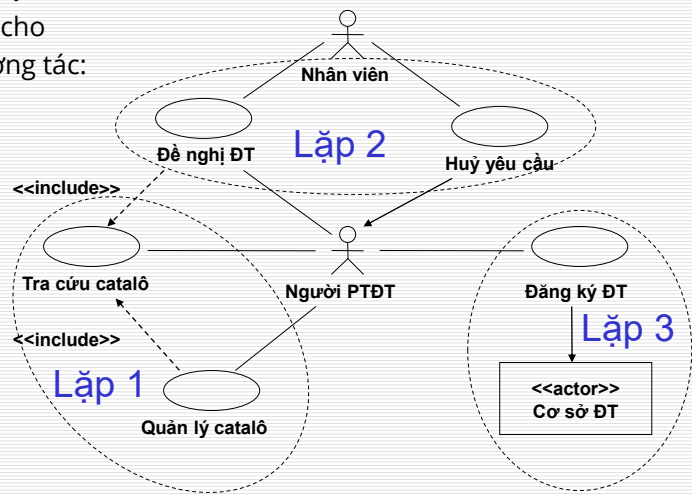


78

78

5. Bài tập tổng hợp

**Câu hỏi 17 (tt):**  
Các vòng lặp cho  
việc MHH tương tác:



5. Bài tập tổng hợp

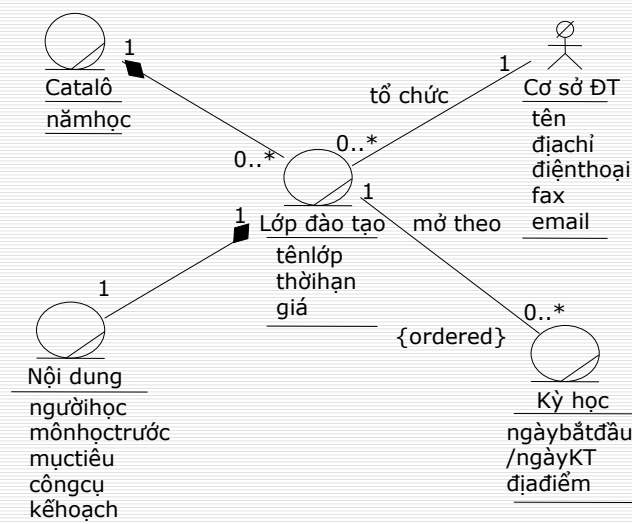
**Câu hỏi 18:** Định nghĩa các thao tác HT cho vòng lặp 1.

- Với mỗi thao tác HT, Larman đề nghị chỉ ra một “hợp đồng 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)
- Nghiên cứu ca sử dụng ‘Quản lýcatalô’,  
ta đưa ra được các thao tác HTcho vòng lặp 1.

Hệ thống
tạoLớpDT() điềuchỉnhLớpDT() tạoLớpCơsởDT() điềuchỉnhCơsởDT() tạoLớpChủđề() điềuchỉnhChủđề() tạoLớpKỹhọc() điềuchỉnhKỹhọc()

## 5. Bài tập tổng hợp

□ BD lớp của gói Catalô DT

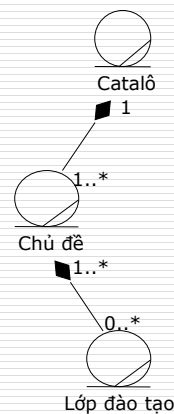


## 5. Bài tập tổng hợp

Câu hỏi 18 (tt):

Lập hợp đồng của thao tác HT:  
"tạo lập Lớp DT".

□ Trước hết ta cần lưu ý các lớp thuộc gói Catalô DT, vì thao tác HT "tạo lập Lớp DT" sẽ đề cập các lớp này. Song ta thấy thiếu lớp Chủ đề, mà ta sẽ phải bổ sung vào gói đó như ở hình bên. Bây giờ có thể lập hợp đồng như sau:

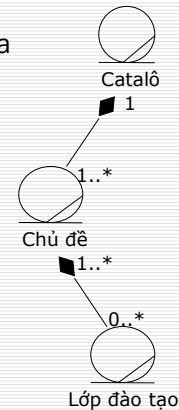




## 5. Bài tập tổng hợp

### Câu hỏi 18 (tt):

- 1) Tên: tạoLớpĐT
- 2) Trách nhiệm: Tạo lập một lớp ĐT mới theo mô tả của CSĐT và đặt lớp đó vào ít nhất một Chủ đề.
- 3) Tham chiếu: Ca SD 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 intranet.



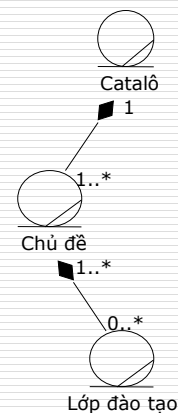
83

83

## 5. Bài tập tổng hợp

### Câu hỏi 18 (tt):

- 5) Các điều kiện ra:
  - Một lớp ĐT 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



84

84

## 5. Bài tập tổng hợp

**Câu hỏi 19:** *Lập một BD tương tác (BD trình tự hay BD giao tiếp) thực hiện thao tác HT “tạo lập LớpĐT”.*

- Các điều kiện ra đã kể ở trên cho ta thấy cái đích cần đạt được:
  - Một **Lớp ĐT** **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.



85

85

## 5. Bài tập tổng hợp

- 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.

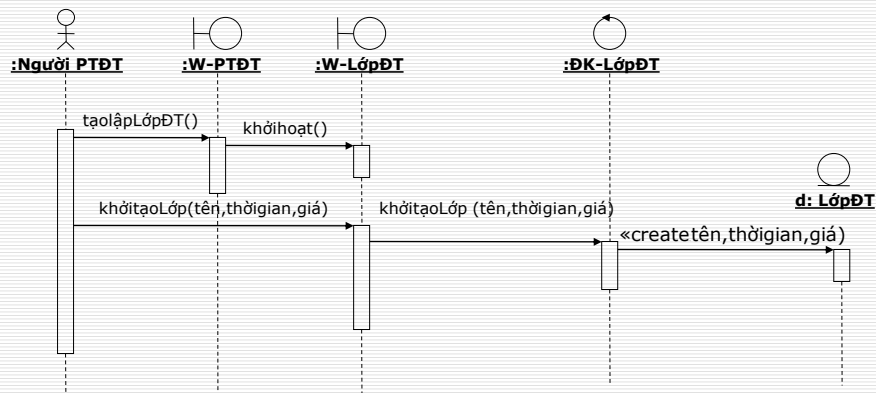


86

86

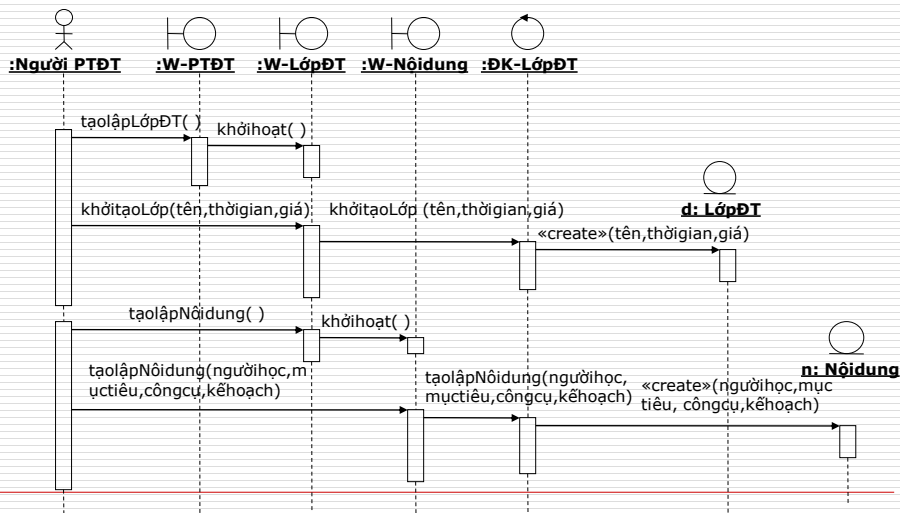
### 5. Bài tập tổng hợp

Thực hiện câu 1 bằng một BD trình tự:



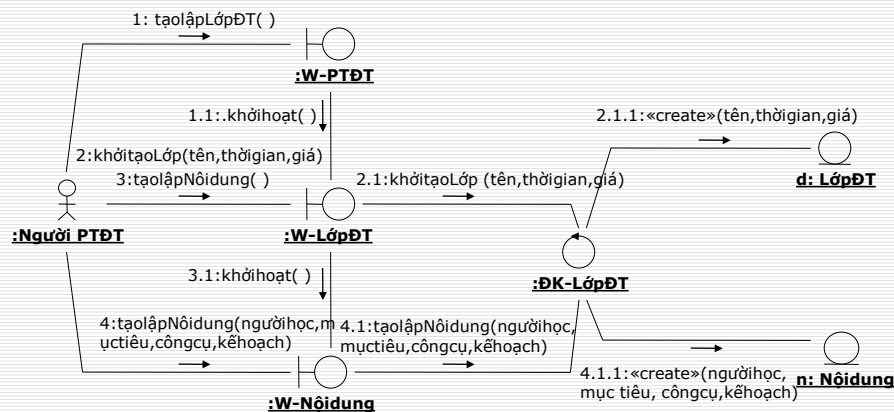
### 5. Bài tập tổng hợp

Thực hiện câu 1 và 2 bằng một BD trình tự:



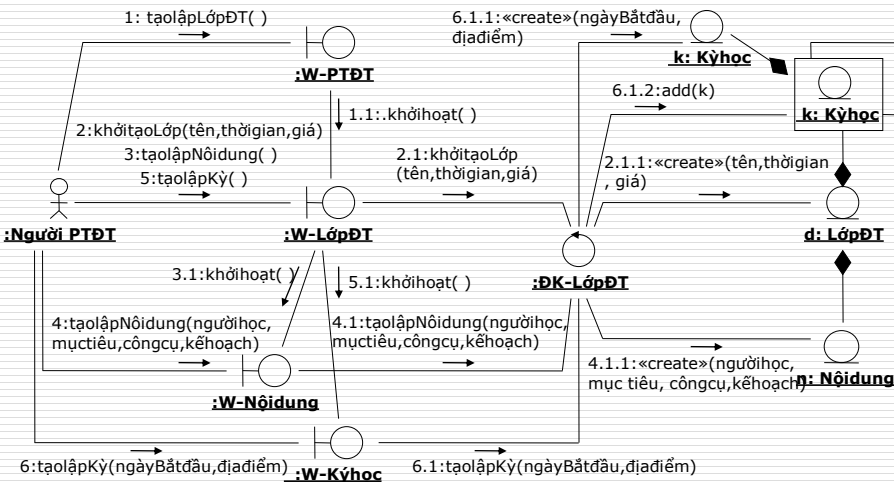
5. Bài tập tổng hợp

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



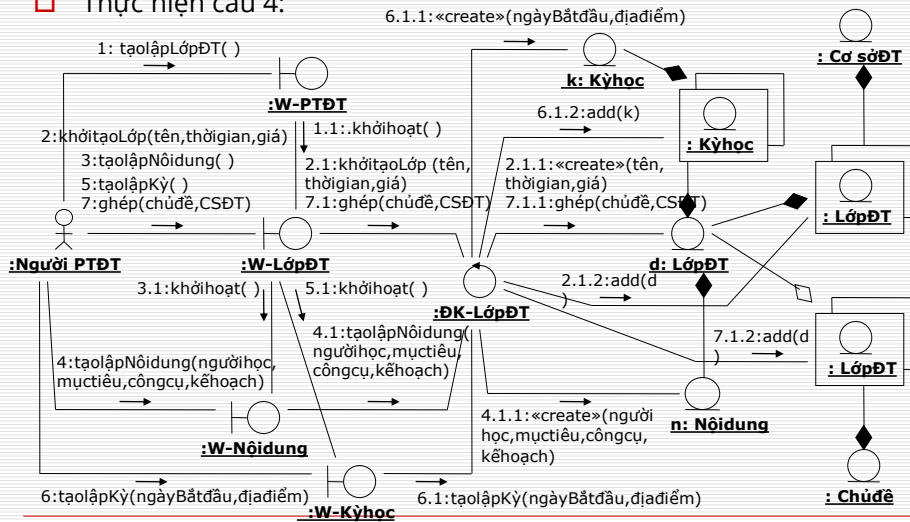
5. Bài tập tổng hợp

❑ Thực hiện câu 3 bằng cách dùng đối tượng bội:



## 5. Bài tập tổng hợp

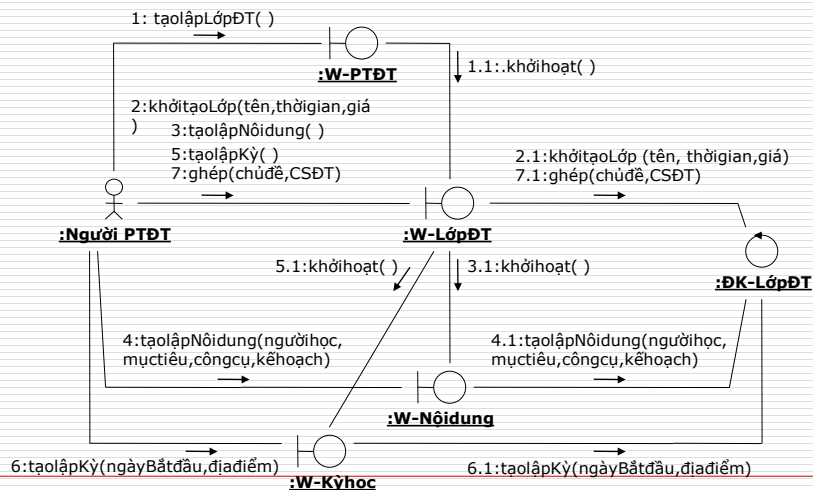
### Thực hiện câu 4:



91

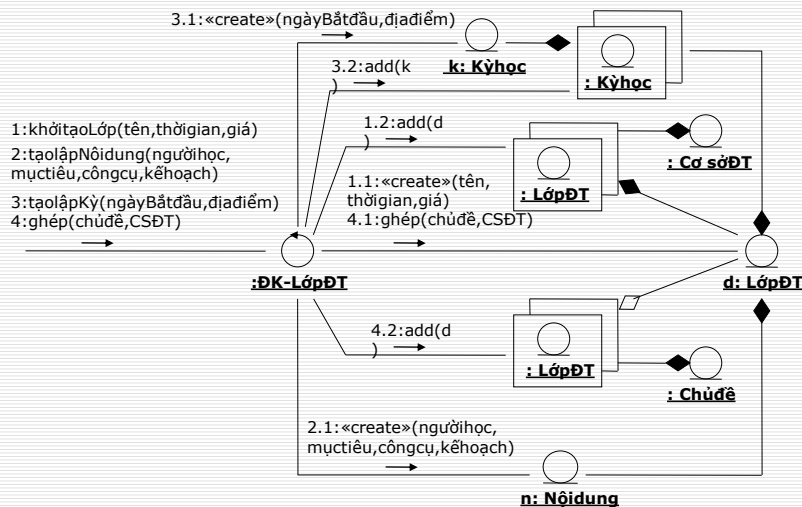
## 5. Bài tập tổng hợp

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



92

## 5. Bài tập tổng hợp



93

93

## 5. Bài tập tổng hợp

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

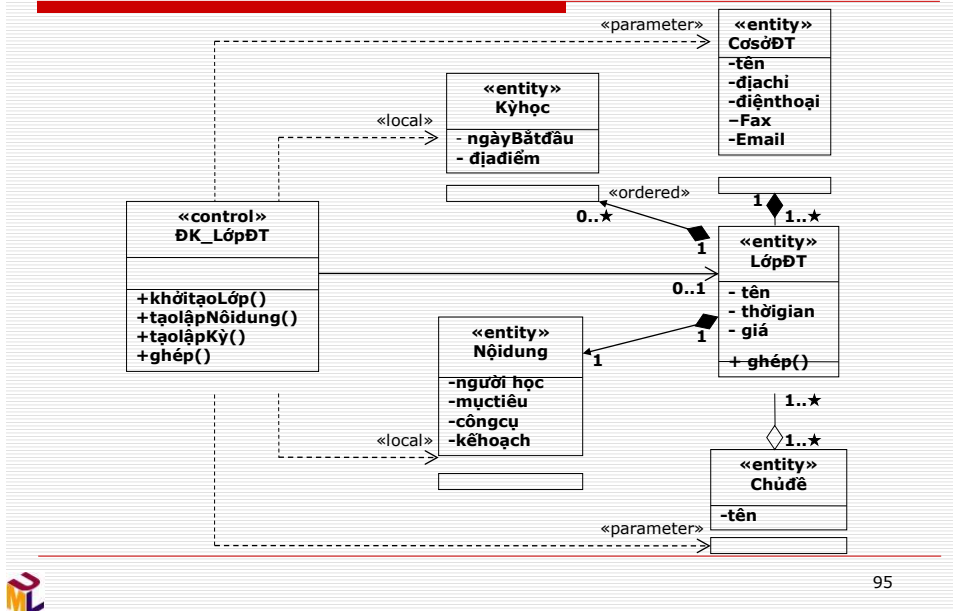
- Nội dung bổ sung gồm:
  - Các thao tác: Một thông điệp gửi tới một đối tượng phải tương ứng với một thao tác có sẵn trong lớp của đối tượng đó.
  - Lưu hành: Tùy theo các kết nối giữa các đối tượng trong biểu đồ giao tiếp (giá đỡ cho các thông điệp) là «association», «self», hay «parameter», «global», «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 BD giao tiếp đang xét là BD 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(...)):



94

94

## 5. Bài tập tổng hợp



95

95

## 5. Bài tập tổng hợp

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 BD tương tác khác mà ta chưa xét tới, 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 BD 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 BD lớp vừa lập ta thấy lớp ĐK\_LớpĐT tương liên với mọi lớp còn lại. Đó một là hiện tượng nên tránh. Bởi vậy cần san sẻ bớt trách nhiệm (chẳng hạn các việc tạo lập các đối tượng Nội dung và Kỳ học) cho lớp khác (chẳng hạn cho lớp LớpĐT) như sau:

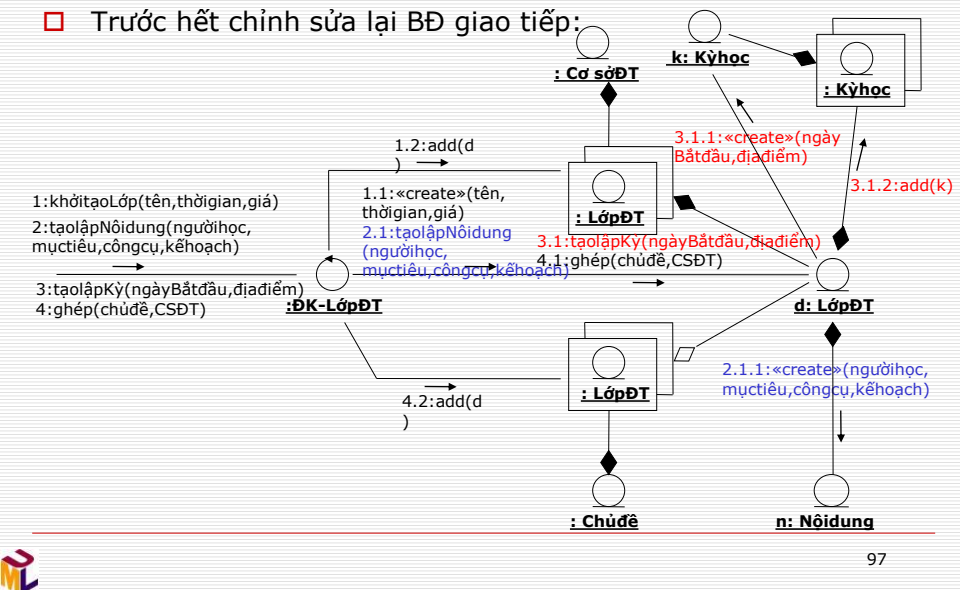


96

96

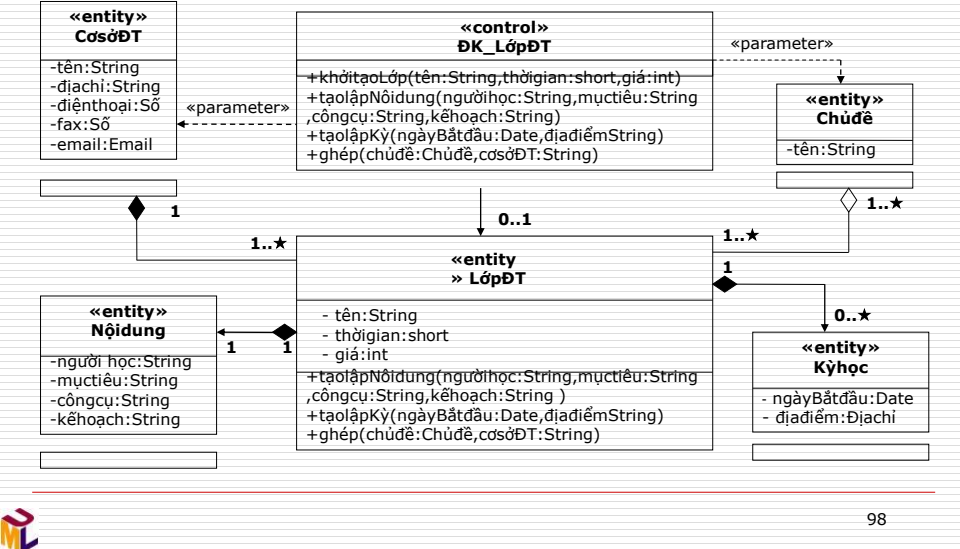
5. Bài tập tổng hợp

Trước hết chỉnh sửa lại BD giao tiếp:



5. Bài tập tổng hợp

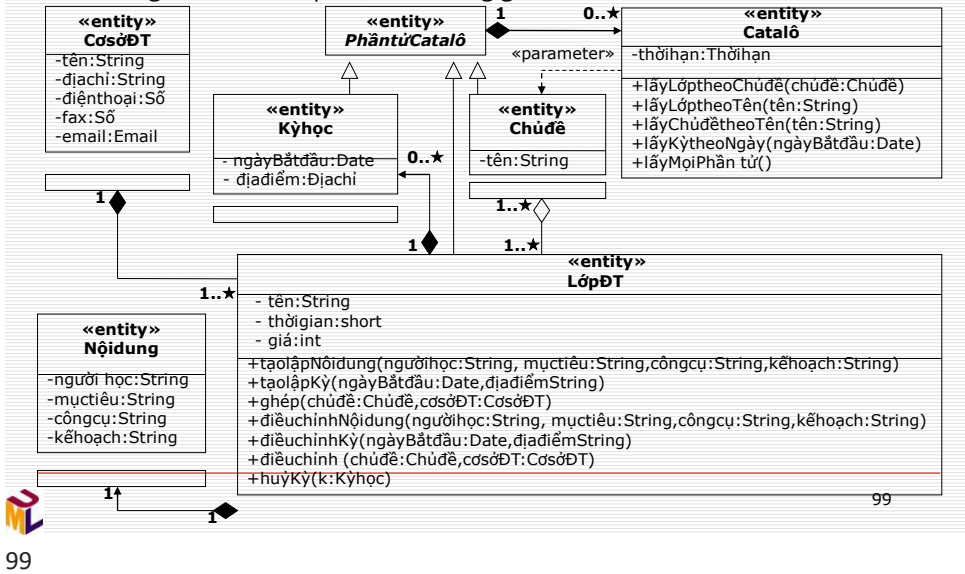
Tiếp đến từ BD giao tiếp đã chỉnh sửa, ta lập lại BD các lớp tham gia:





5. Bài tập tổng hợp

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:



5. Bài tập tổng hợp

IV. Bài tập tổng hợp: MHH tương tác (tt)

Câu hỏi 21: Định nghĩa các thao tác HT cho vòng lặp 2.

- Đọc lại các bản mô tả của ca SD ‘Đề nghị đào tạo’ và ca SD ‘Huỷ đào tạo’ (hoặc xem BD trình tự HT tương ứng), và lưu ý rằng các ca SD này có 2 đối tác là Nhân viên và NgườiPTĐT, ta tìm ra các thao tác HT cho vòng lặp 2 như sau:

Hệ thống
lậpđơnxinĐT() chấpnhậnĐơn() từchốiĐơn() hủyĐơn() đềnghicácký() chọnKý()

## 5. Bài tập tổng hợp

**Câu hỏi 22:** *Lập hợp đồng của các thao tác HT: "lậpĐơnxinĐT" và "từ chốiĐơn".*

□ Lập hợp đồng cho thao tác HT 'lậpĐơnxinĐT'

- 1) Tên: LậpĐơnxinĐT.
- 2) Trách nhiệm: Lập một đơn xin ĐT, dựa trên kết quả tra cứu từ catalô và gửi đơn đó cho người PTĐT để được duyệt.
- 3) Tham chiếu: Ca SD 'Đề nghị ĐT'.
- 4) Các điều kiện vào:
  - Catalô ĐT đã có trước.
  - Nhân viên đang kết nối với intranet.
  - Một đối tượng nv biểu diễn cho nhân viên đó đã có trong HT.



101

101

## 5. Bài tập tổng hợp

**Câu hỏi 22 (tt):**

- 5) Các điều kiện ra:
  - Một đối tượng đơn xin ĐT 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 ca talô (điều có vẻ không phù hợp lắm với BD MHH nghiệp vụ).
- 6) Ngoại lệ: Nhân viên có thể xin huỷ đơn xin của mình trước khi đơn được duyệt.



102

102

## 5. Bài tập tổng hợp

### Câu hỏi 22 (tt):

#### □ Lập hợp đồng của thao tác HT: “từ chối Đơn”.

- 1) Tên: từ chối Đơn.
- 2) Trách nhiệm: Từ chối một đơn xin DT của một nhân viên, có đưa ra lý do từ chối.
- 3) Tham chiếu: Ca SD ‘Đề nghị DT’.
- 4) Các điều kiện vào:
  - Một đối tượng đơn xin DT dx đã có trước.
  - Người PTĐT đang kết nối với intranet.
  - Một đối tượng nv biểu diễn cho nhân viên đã có trong HT và đã kết nối với dx.



103

103

## 5. Bài tập tổng hợp

### Câu hỏi 22 (tt):

- 5) Các điều kiện ra:
  - Đối tượng đơn xin DT 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.

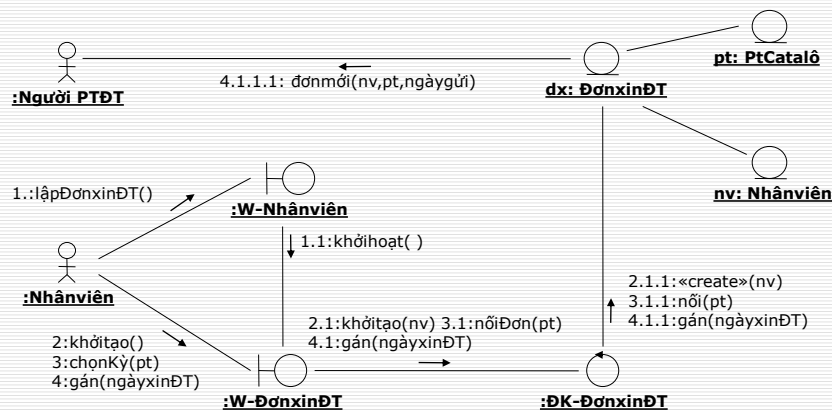


104

104

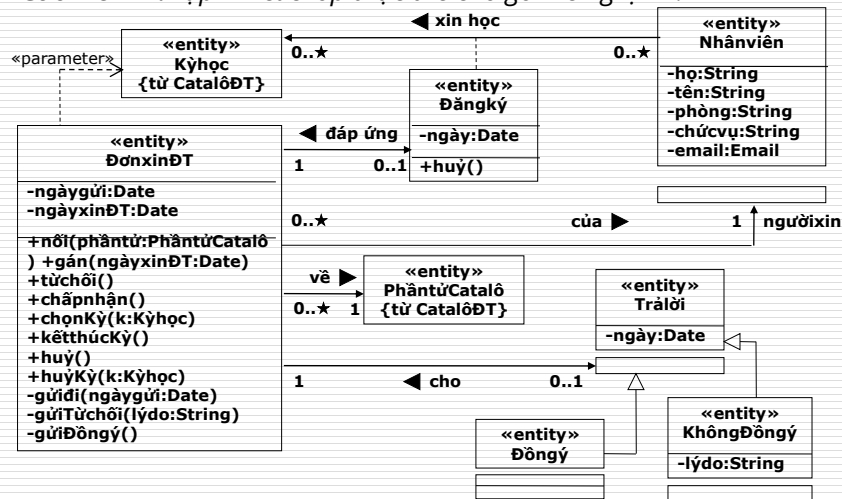
5. Bài tập tổng hợp

❑ Câu hỏi 23: Lập BD giao tiếp thực hiện thao tác HT 'lậpĐơnxinĐT'



5. Bài tập tổng hợp

❑ Câu hỏi 24: Lập BD các lớp thực thể cho gói Đề nghị ĐT.



## Hỏi - đáp

---



107

107

## Lời hay ý đẹp

---

**"Sự đơn giản là điều khó khăn nhất ở đời: đó là hạn định cuối cùng của kinh nghiệm và nỗ lực cuối cùng của tài năng"**

**DUDEVANT GEROG SAND**



108

108