

Phân tích chức năng và cấu trúc

Week 4-5

Hai V. Pham

Tham khảo tài liệu giáo trình PGS. Nguyễn Văn Ba

1

Phân tích chức năng

2

Nội dung

1. Mục đích phân tích chức năng (PTCN)
2. Mô hình hóa (MHH) nghiệp vụ với biểu đồ hoạt động
3. MHH nhu cầu với biểu đồ ca sử dụng
4. Bài tập tổng hợp

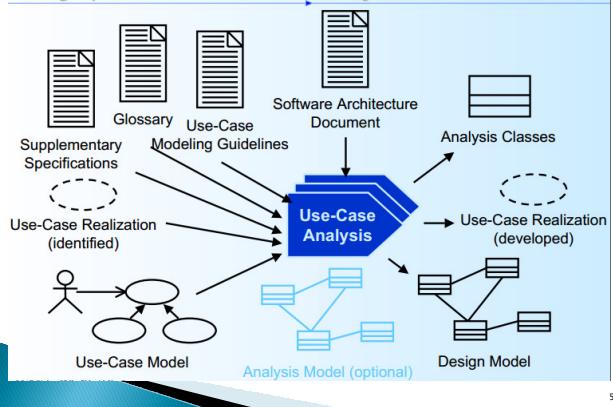
3

1. Mục đích PTCN

- ▶ Cho một cái nhìn sơ bộ về chức năng của hệ thống.
- ▶ Sự phân tích theo chức năng chỉ nhằm đưa ra các chức năng lớn (các nhu cầu đối với HT) mà sẽ không tiếp tục đi sâu vào các chức năng nhỏ, vì con đường PT&TK đã chọn là hướng đối tượng chứ không hướng chức năng.
- ▶ Trật tự trình bày:
 - Mô hình hoá nghiệp vụ với biểu đồ hoạt động.
 - Mô hình hoá nhu cầu với biểu đồ ca sỹ dụng.
 - Bài tập tổng hợp.

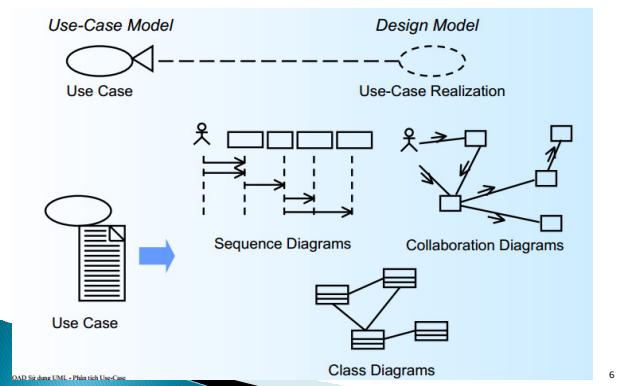
4

Tổng quan về Use-Case Analysis



5

Tổng quan ...



6

2. MHH nghiệp vụ với biểu đồ hoạt động

► Mục đích MHH nghiệp vụ

- **Nghiên cứu môi trường nghiệp vụ** của hệ thống. Trong môi trường bao gồm người, thiết bị, máy tính kết hợp với nhau hoạt động theo những quy trình nghiệp vụ nhất định. Quy trình nghiệp vụ thường được mô tả bằng các biểu đồ hoạt động.
- Biểu đồ hoạt động (Activity Diagram) là biểu đồ mô tả một nội dung hoạt động, theo các luồng đi từ việc này sang việc khác hay để diễn tả một quy trình nghiệp vụ một cách trình tự lôgic của một ca sử dụng, một kịch bản

7

2. MHH nghiệp vụ với biểu đồ hoạt động (2)

► Mục đích MHH nghiệp vụ (2)

- Có thể nói biểu đồ hoạt động là mô hình UML tương đương với sơ đồ khối hoặc với biểu đồ luồng dữ liệu trong các phương pháp phân tích và thiết kế cũ. Sau đây ta lần lượt đề cập các yếu tố tạo nên một biểu đồ hoạt động.

8

2. MHH nghiệp vụ với biểu đồ hoạt động (3)

► Các hoạt động và các dịch chuyển

- Biểu đồ hoạt động là một đồ thị có hướng, trong đó các nút (đỉnh) là các hoạt động, các cung là các dịch chuyển. Có một nút khởi đầu (một chấm đen) và có thể có hay không nhiều nút kết thúc (chấm đen có viền).
- Hoạt động (Activity) là một công việc, có thể là được xử lý bằng tay như là Đien mầu, hoặc xử lý bằng máy tính như là Hiển thị màn hình.
- Biểu diễn hoạt động: Tênhoạtđộng
- **Dịch chuyển** (Transition) là sự chuyển tiếp từ hoạt động này sang hoạt động khác.
- Biểu diễn dịch chuyển:



9

2. MHH nghiệp vụ với biểu đồ hoạt động (4)

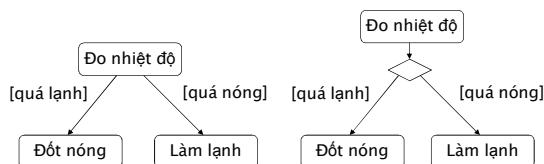
► Các cảnh giới

- Cảnh giới là một điều kiện Bus gắn với một dịch chuyển để diễn tả rằng dịch chuyển đó chỉ được thực hiện khi điều kiện này được thỏa mãn
- Để thực hiện sự rẽ nhánh, UML dùng một hình thoi nhỏ với ý nghĩa:
 - Hoặc là một *quyết định* (decision), khi nó có một luồng vào và nhiều luồng ra (các luồng ra phải mang các cảnh giới loại trừ lẫn nhau);
 - Hoặc là một *hoà nhập* (merge), khi nó có nhiều luồng vào và chỉ một luồng ra (diểm hòa nhập sẽ được vượt qua khi có một luồng vào xuất hiện).

10

2. MHH nghiệp vụ với biểu đồ hoạt động (5)

► Các cảnh giới (2)



11

2. MHH nghiệp vụ với biểu đồ hoạt động (6)

► Đòng bộ hóa

- Trong biểu đồ hoạt động ta có thể dùng các thanh đồng bộ hóa để mở hay đóng các nhánh thực hiện song song:
 - **mở** các nhánh song song bằng một thanh đồng bộ hóa khi nó có một dịch chuyển vào và nhiều dịch chuyển ra - ta gọi đó là một chia (fork);
 - **đóng** các nhánh song song bằng một thanh đồng bộ hóa khi nó có nhiều dịch chuyển vào và một dịch chuyển ra - ta gọi đó là một chum (join).
- Chum chỉ có thể vượt qua khi mọi nhánh vào nó đều đã hoàn tất.

12

2. MHH nghiệp vụ với biểu đồ hoạt động (7)

► Phân tuyến và phân vùng

- Biểu đồ hoạt động có thể **phân tuyến**, như trong một bể bơi. Mỗi hoạt động phải đặt gọn trong một tuyến, và mỗi tuyến dành cho một hay một số đối tượng thực hiện. Các dịch chuyển có thể đổi tuyến tự do.
- Biểu đồ hoạt động cũng có thể được **phân vùng**, mỗi vùng gồm các hoạt động cùng hướng vào một mục đích chung nào đó. Có thể có một vùng thực hiện kịch bản chính, còn các vùng khác thực hiện các kịch bản phụ.

13

2. MHH nghiệp vụ với biểu đồ hoạt động (8)

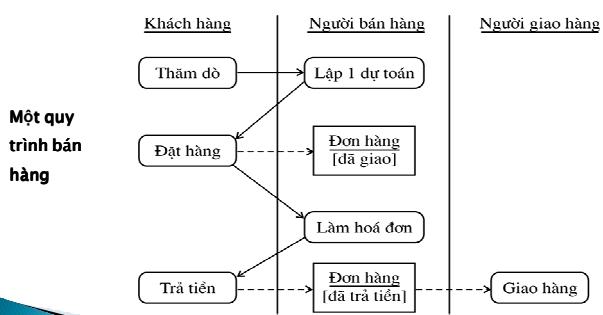
► Tạo lập đối tượng và xuất nhập sự kiện

- Một luồng đối tượng từ một hoạt động đến một đối tượng, rồi lại tiếp tục đi vào một hoạt động khác có thể xem là một luồng điều khiển (một dịch chuyển) giữa hai hoạt động đó.
- *Thí dụ:* Hình sau đây diễn tả một quy trình bán hàng, trong đó có chỉ rõ một đối tượng. Đơn hàng đã được tạo lập và thay đổi trạng thái qua các bước hoạt động:

14

2. MHH nghiệp vụ với biểu đồ hoạt động (9)

► Tạo lập đối tượng và xuất nhập sự kiện (2)



15

3. MHH nhu cầu với biểu đồ ca sử dụng

► Mục đích MHH nhu cầu

- Làm rõ các chức năng lớn của HT mà người dùng mong đợi (gọi là các nhu cầu).
- Diễn tả kết quả PT bằng một BD ca sử dụng của UML.
- Trình tự công việc như sau:
 - MHH môi trường của HT bằng các đối tác
 - MHH nhu cầu bằng các ca sử dụng
 - Thiết lập các mối liên quan
 - Lập biểu đồ ca sử dụng

16

3. MHH nhu cầu với biểu đồ ca sử dụng(2)

► MHH môi trường của HT bằng các đối tác

- Đối tác là **vai trò** của một hay nhiều người hoặc vật thể ở ngoài HT và có tương tác với HT.
- Có 4 loại đối tác:
 - Đối tác chính
 - Đối tác phụ
 - Thiết bị ngoài
 - Hệ thống khác

► Biểu diễn đối tác:



17

3. MHH nhu cầu với biểu đồ ca sử dụng (3)

► MHH nhu cầu bằng các ca sử dụng

- Nhận định các ca sử dụng:
 - **Ca sử dụng** (use case) là một biểu diễn của một tập hợp các chuỗi hành động của HT nhằm cung cấp một kết quả cho một đối tác.
- Các đặc điểm:
 - Ca SD phải liên kết với một hay một số đối tác.
 - Ca SD phải dẫn tới một kết quả cụ thể.
 - Ca SD phải là tập hợp của nhiều chuỗi hành động (các kịch bản).

18

3. MHH nhu cầu với biểu đồ ca sử dụng (4)

► MHH nhu cầu bằng các ca sử dụng (2)

◦ Biểu diễn ca SD:

 Rút tiền

◦ Đặc tả một ca SD:

- Bằng ngôn ngữ tự nhiên
 - Tên và mục đích ca SD
 - Đối tác chính, phụ
 - Điều kiện đầu vào
 - Kịch bản chính (thông lệ)
 - Các kịch bản khác (biệt lệ, sai hỏng)
- Bằng BD hoạt động hay bằng BD trình tự hệ thống (giới thiệu sau)

19

3. MHH nhu cầu với biểu đồ ca sử dụng (5)

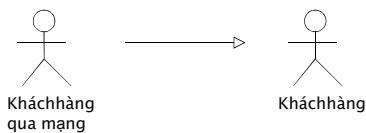
► Thiết lập các mối liên quan

Có nhiều loại liên quan:

a) **Liên quan khái quát hóa giữa các đối tác**

► ĐT A là **khái quát hóa** của ĐT B nếu B thừa kế mọi đặc điểm của A.

► Biểu diễn:



20

3. MHH nhu cầu với biểu đồ ca sử dụng (6)

► Thiết lập các mối liên quan (2)

b) **Liên kết giao tiếp giữa một đối tác với một ca sử dụng**

► Đó là trường hợp đối tác và ca SD có trao đổi thông tin với nhau.

► Biểu diễn:



21

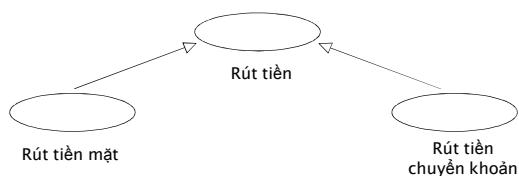
Nếu thông tin trao đổi là một chiều thì 

3. MHH nhu cầu với biểu đồ ca sử dụng (7)

► Thiết lập các mối liên quan (3)

c) **Liên quan Khái quát hoá giữa hai ca SD**

- Ca SD X là **khái quát hoá** của ca SD Y nếu Y thừa kế mọi đặc điểm của X (có thể điều chỉnh và thêm mới).
- Biểu diễn:



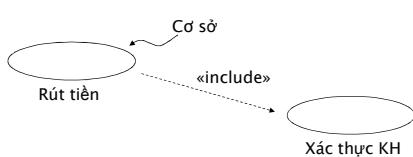
22

3. MHH nhu cầu với biểu đồ ca sử dụng (8)

► Thiết lập các mối liên quan (4)

d) **Liên quan Bao hàm giữa hai ca SD**

- Đó là trường hợp mà một ca SD X ghép nội dung của một ca SD Y vào nội dung của nó (tại một điểm được chỉ rõ trong đặc tả). X gọi là ca SD cơ sở.
- Biểu diễn:



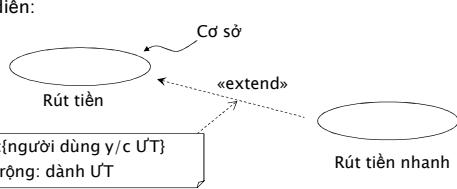
23

3. MHH nhu cầu với biểu đồ ca sử dụng (9)

► Thiết lập các mối liên quan (5)

e) **Liên quan Mở rộng giữa hai ca SD**

- Đó là trường hợp mà một ca SD X ghép (có điều kiện) nội dung của một ca SD Y vào nội dung của nó (tại một điểm được chỉ rõ). X gọi là ca SD cơ sở.
- Biểu diễn:

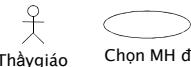


24

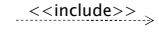
3. MHH nhu cầu với biểu đồ ca sử dụng (10)

► Lập biểu đồ ca sử dụng

- Các nút là các đối tác và ca sử dụng:
- Các cung gồm 4 dạng sau:

• **Khái quát hoá** đối tác hay ca SD: 

• **Liên kết giao tiếp** giữa ĐT và ca SD: 

• **Bao hàm** giữa hai ca SD: 

• **Mở rộng** giữa hai ca SD: 

25

4. Bài tập tổng hợp

(A) MHH nghiệp vụ

Đầu bài: Đề nâng cấp hệ thống thông tin của mình, một công ty muốn, trong bước đầu, vạch ra một quy trình đào tạo (QTĐT) cho nhân viên của mình, để sau đó tin học hoá một phần quy trình này.

1) QTĐT bắt đầu khi có một nhân viên (NV) gửi đến người phụ trách đào tạo (PTĐT) một đề nghị được đi đào tạo. Người PTĐT xem xét đề nghị này và đưa ra trả lời đồng ý hay không đồng ý.

2) Nếu đồng ý, người PTĐT tìm trong một catalô một nơi có các lớp đào tạo thích hợp, thông báo nội dung đào tạo lại cho NV đã xin đi đào tạo, cùng với một danh sách các kỳ học sẽ mở tới đây. Khi người NV đã chọn kỳ học, người PTĐT gửi một đăng ký cho NV đó tới cơ sở đào tạo.

26

4. Bài tập tổng hợp (2)

(A) MHH nghiệp vụ (2)

Đầu bài (tt)

3) Nếu có gì cần trả, người NV phải báo sớm cho người PTĐT để huỷ đăng ký hay huỷ đề nghị đào tạo.

4) Sau khi đào tạo xong, người NV nộp lại cho người PTĐT một giấy xác nhận sự có mặt và giấy nhận xét kết quả học tập.

5) Người PTĐT kiểm tra lại hoá đơn mà cơ sở đào tạo gửi tới, trước khi chuyển cho kế toán trả tiền.

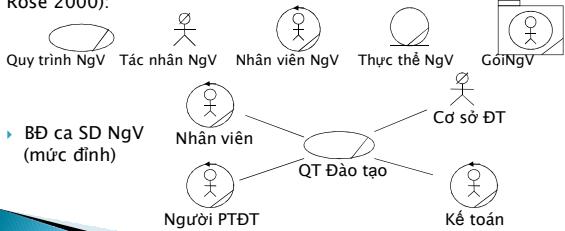
27

4. Bài tập tổng hợp (3)

(A) MHH nghiệp vụ (3)

Câu hỏi 1: MHH quy trình đào tạo trên bằng một BD ca sử dụng (mức định).

- Đề MHH nghiệp vụ (NgV) ta dùng các biểu tượng sau, do Jacobson đề xuất (vẫn được dùng trong RUP và trong Rational Rose 2000):

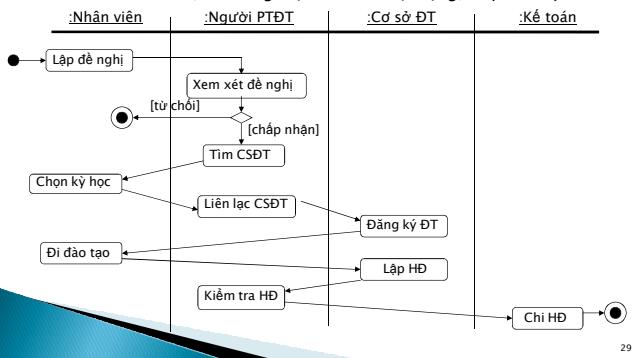


28

4. Bài tập tổng hợp (4)

(A) MHH nghiệp vụ (4)

Câu hỏi 2: Mô tả QTĐT bằng một biểu đồ hoạt động, có phân tuyển.

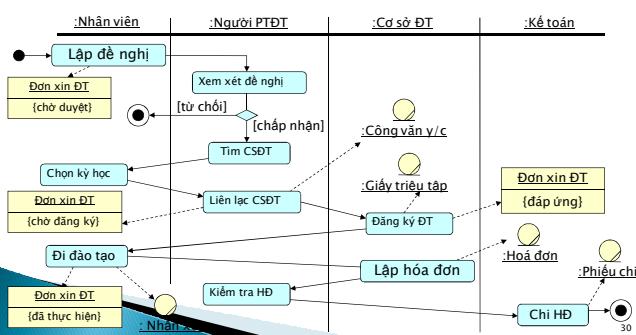


29

4. Bài tập tổng hợp (5)

(A) MHH nghiệp vụ (5)

Câu hỏi 2 (tt): Thêm vào BĐ hoạt động các đối tượng NgV đã được sản sinh cùng với trạng thái của chúng nếu có.

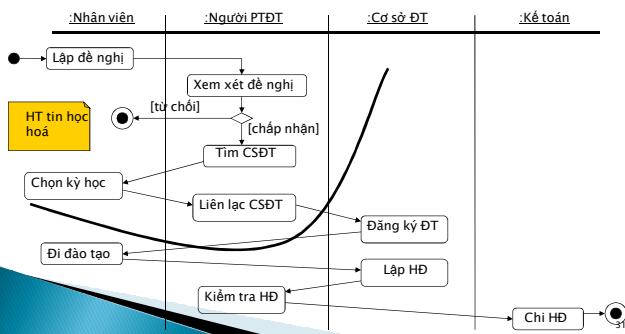


30

4. Bài tập tổng hợp (6)

(B) XĐ phạm vi, nhu cầu

Câu hỏi 3: Từ mô hình nghiệp vụ, hãy xác định phạm vi của HT tin học hóa.



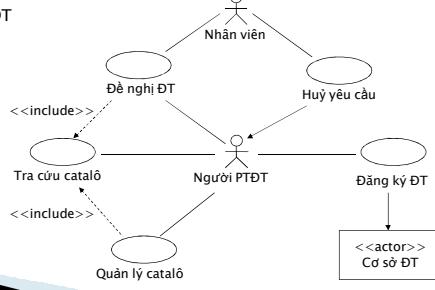
4. Bài tập tổng hợp (7)

(B) XĐ phạm vi, nhu cầu (3)

Câu hỏi 5: Hãy lập BD ca SD cho hệ YCDT

Từ các hoạt động nghiệp vụ, phát hiện các ca SD sau:

- ▶ Đề nghị được ĐT
- ▶ Đăng ký ĐT
- ▶ Huỷ yêu cầu
- ▶ Quản lý catalô



Phân tích cấu trúc

Nội dung

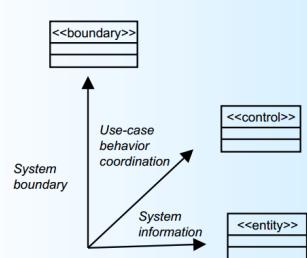
1. Mục đích
2. Đối tượng và lớp
3. Phát hiện các lớp lĩnh vực
4. Phát hiện các lớp tham gia các ca sử dụng
5. Bài tập tổng hợp

34

1. Mục đích

- ▶ Sơ bộ phát hiện các lớp chính tạo nên hệ thống.

Thế nào là một Analysis Class?



35

2. Đối tượng và lớp

- 2.1. Định nghĩa và biểu diễn đối tượng và lớp
- 2.2. Các thuộc tính
- 2.3. Các thao tác
- 2.4. Mối liên quan
 - 2.4.1. Mối liên quan phụ thuộc
 - 2.4.2. Mối liên quan khái quát hóa
 - 2.4.3. Liên kết
- 2.5. Biểu đồ lớp và biểu đồ đối tượng

36

2.1. Định nghĩa và biểu diễn 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**, 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.

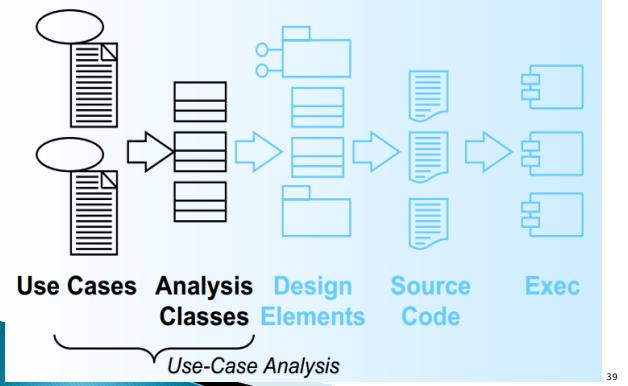
37

2.1. Định nghĩa và biểu diễn đối tượng và lớp

- ▶ **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).

38

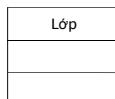
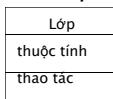
Các Analysis Class: Bước đầu tiên đến cài đặt



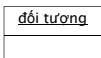
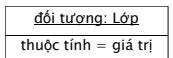
59

2.1. Định nghĩa và biểu diễn đối tượng và lớp

► Biểu diễn lớp



► Biểu diễn đối tượng



40

2.2. 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}]

41

2.2. Các thuộc tính

► **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.
- **Gói** (package), ký hiệu bởi dấu '~', nếu thuộc tính có thể truy cập từ các phần tử thuộc cùng một gói (hộp nhất) với lớp.

42

2.2. Các thuộc tính

- ▶ Cú pháp của thuộc tính:

[tên nhỉn] [/] tên [: Kiểu] [cơ số] [= giá trị đầu] [[xâu tính chất]]

- **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.
- **Cơ số** (multiplicity), chỉ số các giá trị có thể nhận, chẳng hạn [0..1] để chỉ thuộc tính này là tuỳ chọn (không nhận giá trị nào, hay có nhận một giá trị).

43

2.2. Các thuộc tính

- ▶ **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 chỉ các giá trị có thể gán cho thuộc tính, thường dùng đối với một kiểu liệt kê, chẳng hạn:
tình trạng: Tình trạng = chưatrà {chưatrà, đãträ}

44

2.2. Các 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).

45

2.3. 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:

[tâm nhìn] tên [(danh sách tham số)] [: Kiểu trả lại] [(xâu tính chất)]

- Tâm nhìn hoàn toàn giống tâm 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]

46

2.3. Các thao tác

- Hướng có thể lấy các giá trị in, out, inout và return tùy thuộc tham số là in – không thể điều chỉnh, out – có thể điều chỉnh để đưa thông tin cho bên gọi, hoặc là inout có thể điều chỉnh được, hay là return 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 đe, các tác động lên trạng thái đối tượng...

47

2.4. Mối liên quan

► 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

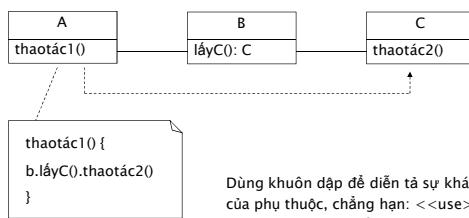
48

2.4.1. 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).

49

2.4.1. Mối liên quan phụ thuộc



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

50

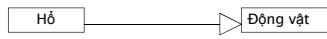
2.4.2. Mối liên quan khái quát hóa

- Khái quát hóa** (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 hóa** (specialization): từ một lớp đã cho ta tăng cường thêm một số đặc điểm mới, tạo thành một lớp chuyên biệt, 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).

51

2.4.2. Mối liên quan khái quát hóa

- Thuật ngữ **thừa kế** vốn được dùng nhiều trong các NNLT, nhằm diễn tả lớp con có mọi thuộc tính, thao tác và liên kết 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 hình/đa xạ** (polymorphism).
- Biểu diễn khái quát hóa:



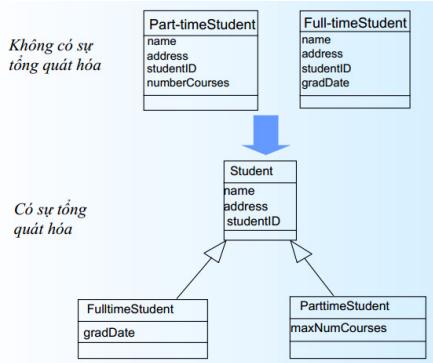
52

2.4.2. Mối liên quan khái quát hóa

- Thông qua sự khái quát hóa 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 xen kẽ và có thể kèm thêm tính chất {abstract}.
- Thí dụ: Động vật là lớp trừu tượng được khái quát hóa từ các lớp Ngựa, Hổ, Dơi. Nó có một thao tác trừu tượng là ngủ(). Thao tác trừu tượng này sẽ được cụ thể hóa 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).

53

2.4.2. Mối liên quan khái quát hóa



54

2.4.3. Liên kết

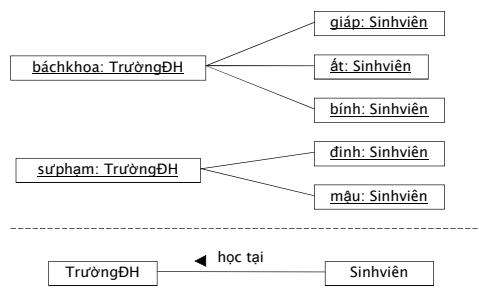
▶ Kết nối và liên kết

- 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).

55

2.4.3. Liên kết

▶ Biểu diễn kết nối và liên kết



56

2.4.3. Liên kết

▶ 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. Bây 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.

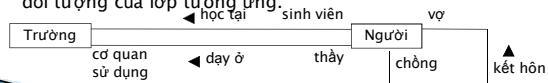


57

2.4.3. Liên kết

► 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**.



58

2.4.3. Liên kết

► Cơ số

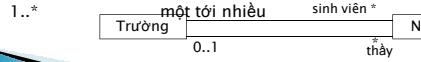
- Mỗi đầu của liên kết còn có thể chứa thêm một **cơ số** (multiplicity), cho biết số cá thể (tối thiểu và tối đa) của đầu đó tham gia liên kết với **một** cá thể ở đầu kia. Các giá trị của cơ số thường dùng là:

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



59

2.4.3. Liên kết

► 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.

60

2.4.3. Liên kết

▶ Hạn định (tt)

- 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.

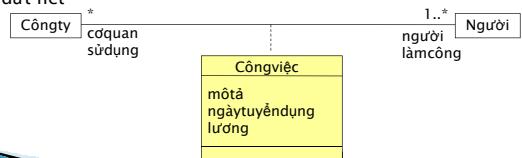


61

2.4.3. Liên kết

▶ Lớp liên kết

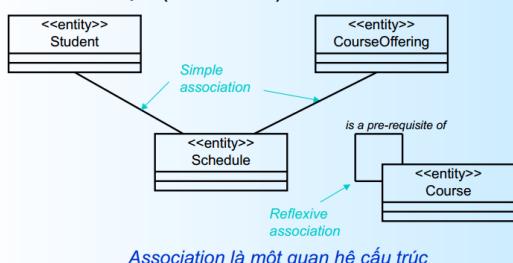
- Bản thân các liên kết cũng có thể cần có các tính chất đặc trưng cho nó. UML thể hiện điều này bằng các **lớp liên kết** (association class). Lớp liên kết là một lớp như bình thường, nghĩa là có thể có các thuộc tính, các thao tác và tham gia liên kết với những lớp khác, song ngắn tên có thể có tên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



62

Ví dụ: Lớp liên kết (association class).

- Mô hình hóa một liên hệ ngữ nghĩa giữa các thể hiện (instances) của các class

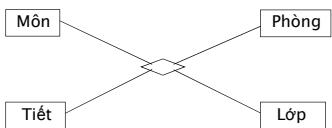


63

2.4.3. Liên kết

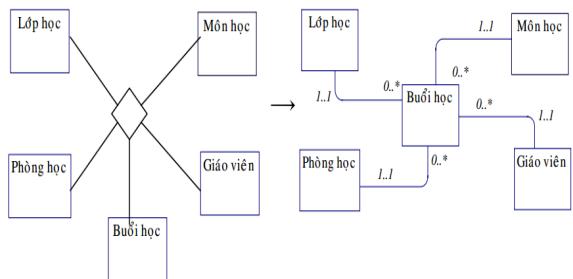
► Liên kết nhiều bên

- 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连线 với các lớp tham gia và cũng có thể có một lớp liên kết cho nó.



64

Liên kết nhiều bên

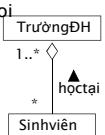


65

2.4.3. Liên kết

► Kết nhập

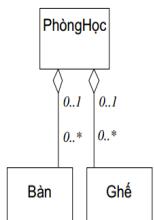
- Trong một liên kết, hai bên tham gia được xem là bình đẳng, không bên nào được nhấn mạnh hơn bên nào. Tuy nhiên cũng có lúc ta muốn mô hình hóa mối quan hệ "toàn thể/bộ phận" giữa một lớp các vật thể lớn (cái "toàn thể") với một lớp các vật thể bé (các "bộ phận") bao gồm trong chúng. Đó là một loại 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



66

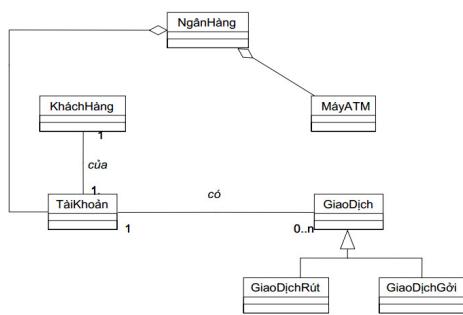
Kết nhập (aggregation)

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



67

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

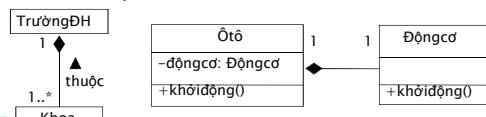


68

2.4.3. Liên kết

Hợp thành

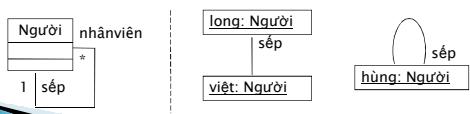
- Một **hợp thành** (composition) là một loại kết nhập đặc biệt với quan hệ sở hữu mạnh hơn, trong đó một bộ phận chỉ thuộc vào một cái toàn thể duy nhất và cái toàn thể có trách nhiệm tạo lập và hủy bỏ cái bộ phận. Như vậy khi cái toàn thể bị hủy bỏ thì cái bộ phận cũng buộc phải hủy bỏ theo. Hợp thành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



69

2.5. 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 hóa, 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 hóa cấu trúc tĩnh và nó bao gồm mọi phần tử khai báo, cho nên nó là cái nền (cái nòng đõ) cho các hoạt động chức năng của hệ thống.



70

3. Phát hiện các lớp lĩnh vực

- Mục đích và trình tự tiến hành
- Nhận định các khái niệm lĩnh vực
- Thêm các liên kết và các thuộc tính
- Khái quát hóa các lớp

71

3.1. Mục đích và trình tự tiến hành

- Xuất phát từ các khái niệm về các sự vật trong lĩnh vực ứng dụng, ta trừu tượng hóa 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 giữ và cung cấp các thông tin về các sự vật đó (cho nên cũng gọi là lớp thực thể).
- Trình tự tiến hành:
 - Nhận định các khái niệm của lĩnh vực.
 - Thêm các liên kết và các thuộc tính.
 - Khái quát hóa các khái niệm.

72

3.2. 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.

73

3.2. Nhận định các khái niệm của lĩnh vực

Cách nắm bắt các khái niệm

- ▶ 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 đẹp, máy bay, cảng biển...
 - 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...

74

3.2. 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.



Thông tin cần thiết để đăng ký
học và tính học phí cho SV.
Sinh viên là người được đăng
ký theo học các lớp giảng của
trường ĐH.

75

3.2. Nhận định các khái niệm của lĩnh vực

Đặt tên và gán trách nhiệm (tt)

- ▶ Việc gán tên và trách nhiệm cho một lớp đè cùi 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 đè cùi 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 đồng 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.

76

3.3. 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.

77

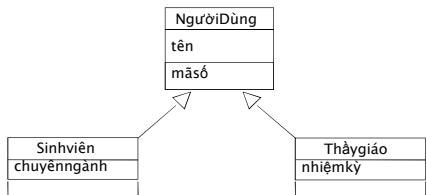
3.3. Thêm các liên kết và các thuộc tính

- ▶ 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, địa chỉ ... cho lớp Sinh viên.
 - 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 Sinh viên và Lớp giảng, với tên liên kết có thể là "đăng ký".

78

3.4. Khái quát hóa 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 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.



79

4. Phát hiện các lớp tham gia ca sử dụng

- Mục đích
- Phát hiện các đối tượng/lớp tham gia ca sử dụng
- Lập biểu đồ lớp cho ca SD

80

4.1. Mục đích

- Trong khi mô hình hoá lĩnh vực (công việc vừa làm), ta đã tiến hành nghiên cứu lĩnh vực, mà không xem xét tới ứng dụng. Các lớp phát hiện được chỉ là các lớp lĩnh vực.
- Đặc thù của mỗi ứng dụng nằm ở các ca sử dụng. Vậy để nghiên cứu ứng dụng, ta phải đi sâu phân tích 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 phụ trợ khác nữa.
- Mục đích của bước này chính là phân tích cấu trúc tĩnh của các ca SD. Ta sẽ phải lần lượt thực hiện các việc sau:
 - Phát hiện các đối tượng tham gia ca sử dụng.
 - Thêm các mối liên quan giữa các lớp để lập một biểu đồ lớp cho mỗi ca SD.

81

4.2. 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**.

82

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

- ▶ Các **lớp biên** (boundary) hay lớp đối thoại (dialog):
 - Đó là các lớp nhằm chuyên đổi thông tin giao tiếp giữa các đối tác và hệ thống:
 - Điện hình là các màn hình giao lưu với các người dùng, cho phép thu thập thông tin hay xuất các kết quả.
 - Đó 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.

83

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

- ▶ Các **lớp điều khiển** (control):
 - Đó 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 chuyên 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.

84

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

- ▶ Các **lớp lĩnh vực** (domain) hay lớp thực thể (entity):
 - Đây 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 đây 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 chử dụng trong nó là được đề cập đến trong ca sử dụng.
- ▶ Biểu diễn: <<boundary>> <<control>> <<entity>>



85

4.3. Lập biểu đồ lớp cho ca SD

- ▶ Mục đích và yêu cầu:
 - 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 sau.

86

4.3. Lập biểu đồ lớp cho ca SD

- Biểu đồ các lớp tham gia một ca sử dụng phải bao gồm đủ các lớp đã được phát hiện cùng các yếu tố cấu trúc (thuộc tính, thao tác, liên kết) cần thiết cho lớp.
 - Các thuộc tính phải lưu giữ đủ 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.

87

4.3. Lập biểu đồ lớp cho ca SD

▶ Bổ sung các 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 thường cứu 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.

88

4.3. Lập biểu đồ lớp cho ca SD

▶ Bổ sung các liên kết:

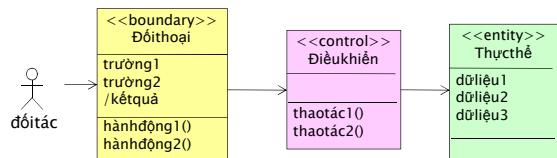
- Các lớp biên chỉ được nối với các lớp điều khiển hay với các lớp biên khác. Nói chung thì các liên kết là một chiều từ lớp biên đến lớp điều khiển, trừ khi lớp điều khiển lại tạo lập một đối thoại mới (chẳng hạn một trang các kết quả).
- Các lớp thực thể chỉ được nối với các lớp điều khiển hay lớp 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.

89

4.3. Lập biểu đồ lớp cho ca SD

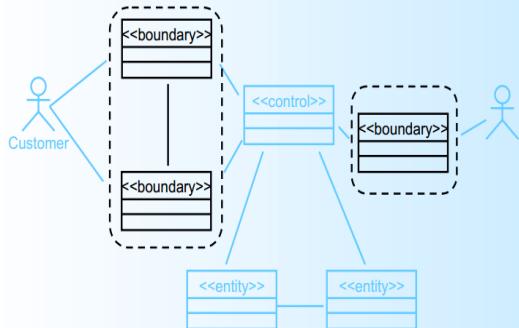
▶ Thêm các đối tác:

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



90

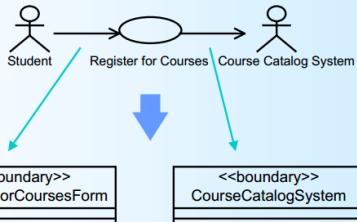
Vai trò của một Boundary Class



91

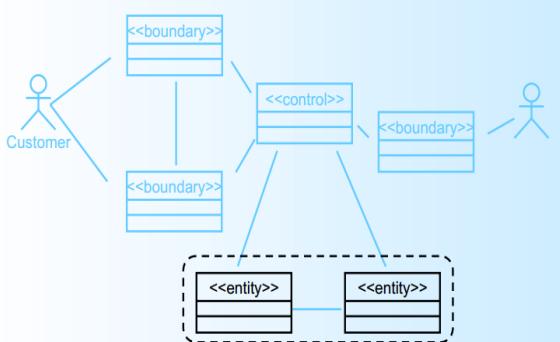
Ví dụ:

- Một boundary class cho 1 cặp actor/use case

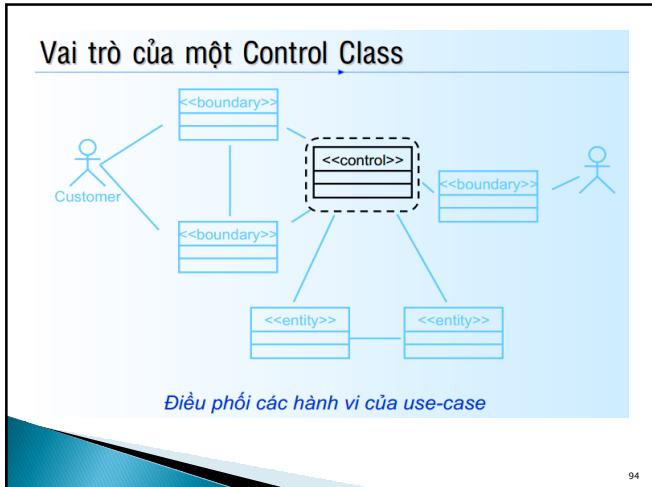


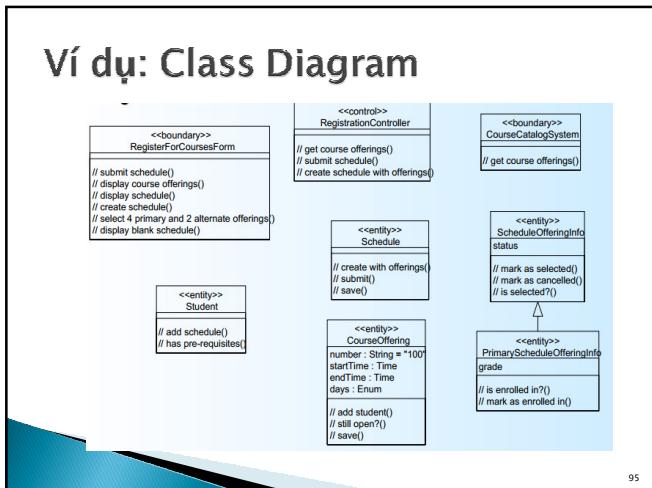
92

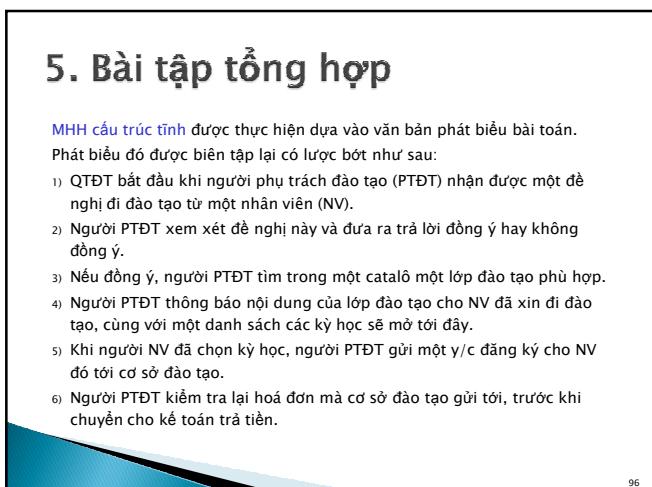
Vai trò của một Entity Class



93

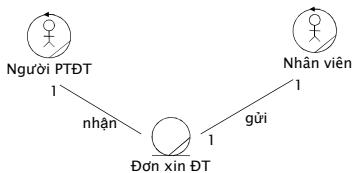






5. Bài tập tổng hợp

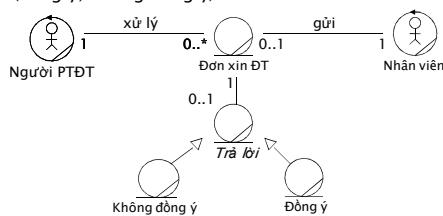
- Câu hỏi 8:** MHH câu 1, dùng các biểu tượng của Jacobson.
- 'QTĐT bắt đầu khi người phụ trách đào tạo (PTĐT) nhận được một đề nghị để đào tạo từ một nhân viên (NV)'.
- Chú ý các danh từ. QTĐT đã xác định từ câu hỏi 1 là một quy trình nghiệp vụ, vậy không phải là lớp. Còn lại các danh từ: người PTĐT, đề nghị ĐT, nhân viên sẽ được MHH thành các lớp.



97

5. Bài tập tổng hợp

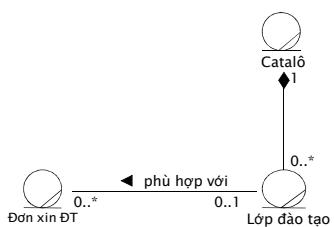
- Câu hỏi 9:** MHH câu 2.
- 'Người PTĐT xem xét đề nghị này và đưa ra trả lời đồng ý hay không đồng ý'.
- Mở rộng mô hình trên, có chỉnh sửa vài chi tiết cho thích hợp hơn và thêm một thực thể trùu tượng (trả lời) cùng với 2 thực thể chuyên biệt (đồng ý, không đồng ý).



98

5. Bài tập tổng hợp

- Câu hỏi 10:** MHH câu 3.
- 'Nếu đồng ý, người PTĐT tìm trong một catalô một lớp đào tạo phù hợp'.

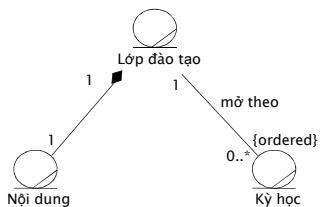


99

5. Bài tập tổng hợp

Câu hỏi 11: MHH câu 4.

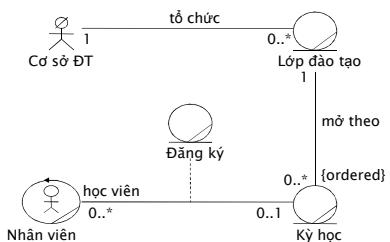
'Người PTĐT thông báo nội dung của lớp đào tạo cho NV đã xin đi đào tạo, cùng với một danh sách các kỳ học sẽ mở tới đây'.

10
0

5. Bài tập tổng hợp

Câu hỏi 12: MHH câu 5.

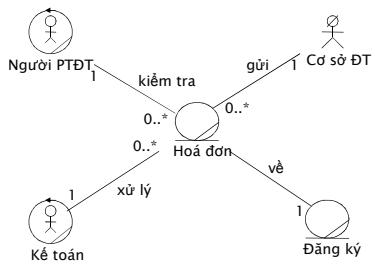
'Khi người NV đã chọn kỳ học, người PTĐT gửi một y/c đăng ký cho NV đó tới cơ sở đào tạo'.

10
1

5. Bài tập tổng hợp

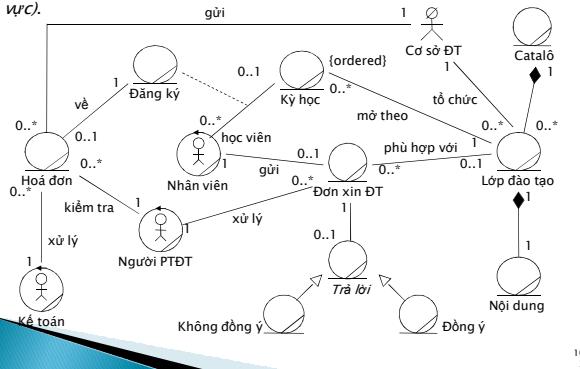
Câu hỏi 13: MHH câu 6.

'Người PTĐT kiểm tra lại hoá đơn mà cơ sở đào tạo gửi tới, trước khi chuyển cho kế toán trả tiền'.

10
2

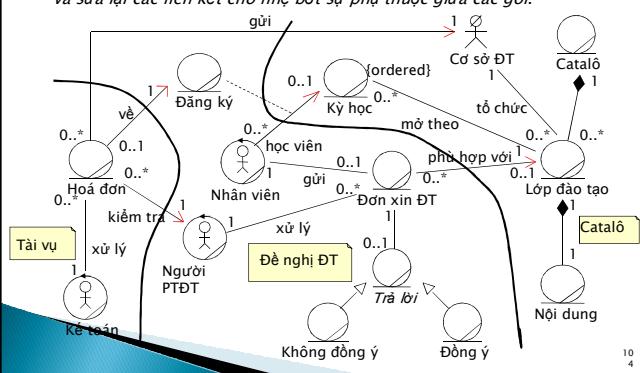
5. Bài tập tổng hợp

- Câu hỏi 14: Tổng hợp các kết quả trên vào một BD lớp nghiệp vụ (lĩnh vực).

10
3

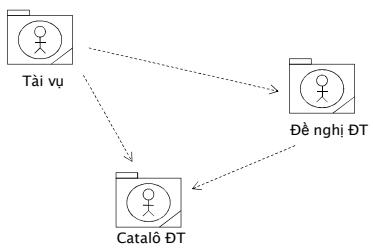
5. Bài tập tổng hợp

- Câu hỏi 15: Chia cắt mô hình thành các gói theo các đơn vị nghiệp vụ và sửa lại các liên kết cho nhẹ bớt sự phụ thuộc giữa các gói.

10
4

5. Bài tập tổng hợp

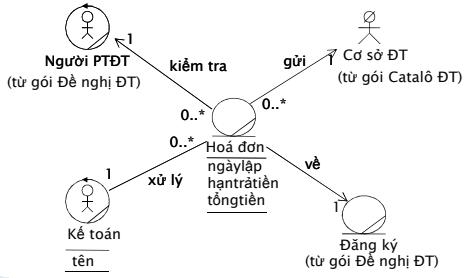
- Biểu đồ gói thu được là như sau (các phụ thuộc là một chiều):

10
5

5. Bài tập tổng hợp

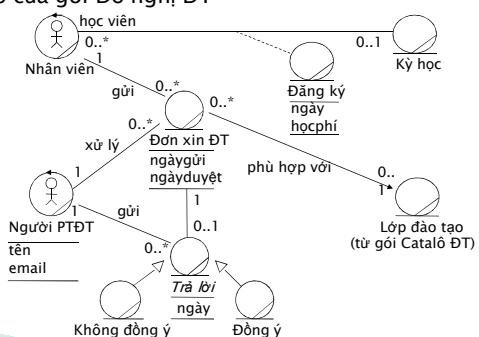
Câu hỏi 16: Thêm các thuộc tính trường cữu vào các lớp (vẽ riêng cho từng gói)

- ▶ BĐ lớp của gói Kế toán

10
6

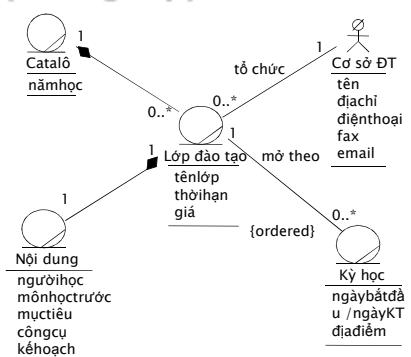
5. Bài tập tổng hợp

- ▶ BĐ lớp của gói Đề nghị ĐT

10
7

5. Bài tập tổng hợp

- ▶ BĐ lớp của gói Catalô ĐT

10
8
