



Thiết kế hướng đối tượng - Sử dụng UML (7)



Nội dung

- Khái niệm cơ bản hướng đối tượng
- Biểu đồ ca sử dụng
- Thiết kế cấu trúc tĩnh
- Thiết kế cấu trúc động
- Sinh mã



Hướng chức năng

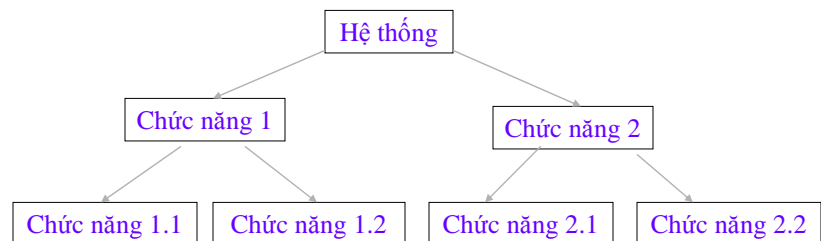
- Dựa vào các chức năng của hệ thống
 - Hệ thống là tập hợp các chức năng
- Chia nhỏ các chức năng và làm mịn dần
 - Hệ thống gồm các hệ thống con
 - Làm chủ độ phức tạp
- Các chức năng trao đổi với nhau bằng truyền tham số hoặc dữ liệu (chẳng hạn biến toàn cục) dùng chung

3



Hướng chức năng

- Phân cấp chức năng



4



Hướng chức năng

- Ưu điểm
 - Phân tích được các chức năng của hệ thống
 - Đưa lại kết quả mong đợi
- Nhược điểm
 - Chức năng → cấu trúc
 - Thay đổi về chức năng → khó khăn thay đổi cấu trúc
 - Tính mở của hệ thống thấp
 - Khó tái sử dụng
 - Chi phí sửa chữa lỗi lớn

5



Hướng đối tượng

- Lấy đối tượng làm trung tâm
- Hệ thống = tập hợp các đối tượng + quan hệ giữa các đối tượng
- Các đối tượng trao đổi bằng thông điệp (message)
 - Không sử dụng biến toàn cục
- Đóng gói
- Thừa kế

6



Hướng đối tượng

○ Phân biệt

- Lập trình cấu trúc
 - Thuật toán + cấu trúc dữ liệu = chương trình
- Lập trình HĐT
 - Σ đối tượng = chương trình
 - đối tượng = thuật toán + cấu trúc dữ liệu

7



Hướng đối tượng

○ Ưu điểm chính

- Gần gũi với thế giới thực
- Tái sử dụng dễ dàng
- Đóng gói, che dấu thông tin làm cho hệ thống tin cậy hơn
- Thừa kế làm giảm chi phí, hệ thống có tính mở cao hơn
- Xây dựng hệ thống lớn và phức tạp

8



Đối tượng

- Đối tượng (object) là khái niệm cho phép mô tả các sự vật/thực thể trong thế giới thực
- Các đối tượng duy trì các quan hệ giữa chúng
- Nguyễn Văn A là một đối tượng

9



Đối tượng

- Các tính chất của đối tượng
 - Đối tượng = trạng thái + hành vi + định danh
 - Trạng thái là các đặc tính của đối tượng tại một thời điểm
 - Hành vi thể hiện các chức năng của đối tượng
 - Định danh thể hiện sự tồn tại duy nhất của đối tượng

10



Đối tượng : trạng thái

- Trạng thái = tập hợp các thuộc tính
 - Mỗi thuộc tính mô tả một đặc tính
 - Tại một thời điểm cụ thể, các thuộc tính mang các giá trị trong miền xác định
 - Ví dụ
 - Một chiếc xe máy: màu xanh, 110 cm³, dream, 12000km, đứng yên, ...

11



Đối tượng : hành vi

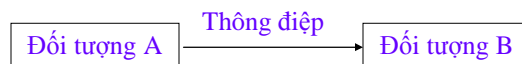
- Hành vi = tập hợp các phương thức
 - Phương thức: là một thao tác hoặc được thực hiện bởi chính nó, hoặc thực hiện khi có yêu cầu từ môi trường (thông điệp từ đối tượng khác)
 - Hành vi phụ thuộc vào trạng thái
 - Ví dụ:
 - một xe máy có các hành vi: khởi động, chạy, ...

12



Giao tiếp giữa các đối tượng

- Các đối tượng giao tiếp với nhau
 - Gửi thông điệp (message) cho nhau



- Các loại thông điệp
 - hàm dựng (constructor)
 - hàm hủy (destructor)
 - hàm chọn lựa (get)
 - hàm sửa đổi (set)
 - các hàm chức năng khác

13



Đối tượng

- Giữa các đối tượng có mối liên kết (link) với nhau
- Ví dụ



14



Lớp

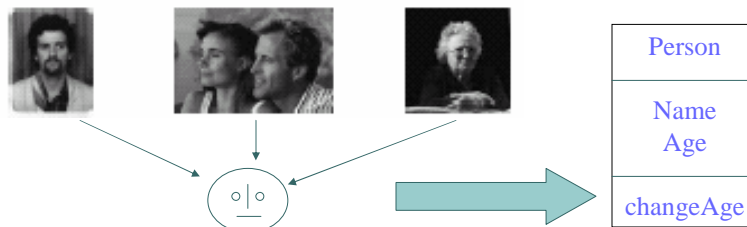
- Lớp là khái niệm dùng để mô tả một tập hợp các đối tượng có cùng một cấu trúc, cùng hành vi và có cùng những mối quan hệ với các đối tượng khác
- Lớp = các thuộc tính + các phương thức

15



Lớp

- Lớp là một bước trừu tượng hóa
 - Tìm kiếm các điểm giống nhau, bỏ qua các điểm khác nhau của đối tượng



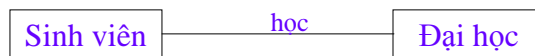
- Trừu tượng hóa làm giảm độ phức tạp

16



Lớp

- Quan hệ giữa các lớp: kết hợp
- Một kết hợp là một tập hợp các mối liên kết giữa các đối tượng



17



Lớp & Đối tượng

- Đối tượng là thể hiện (instance) của lớp
- Giá trị là thể hiện của thuộc tính
- Liên kết là thể hiện của kết hợp
- Lớp → Đối tượng
- Thuộc tính → Giá trị
- Kết hợp → Liên kết

18



Các tính chất của HĐT

- Tính đóng gói (encapsulation)
 - dữ liệu + xử lý dữ liệu = đối tượng
 - thuộc tính + phương thức = lớp
- Ưu điểm
 - Hạn chế ảnh hưởng khi có sự thay đổi cập nhật
 - Ngăn cản sự truy cập thông tin từ bên ngoài
 - Che dấu thông tin

19



Các tính chất của HĐT

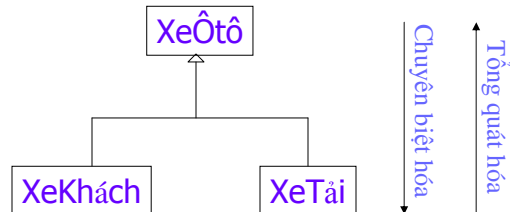
- Tính thừa kế (inheritance)
 - Một lớp được xây dựng từ một hoặc nhiều lớp khác bằng việc chia sẻ các thuộc tính và phương thức
 - *Lớp con* thừa kế các thuộc tính và phương thức từ *lớp cha*
 - Tổng quát hóa/chuyên biệt hóa
 - Tổng quát hóa (generalization): đặt các tính chất chung của các lớp khác nhau vào một lớp cha
 - Chuyên biệt hóa (specialization): tạo ra một lớp con có các tính chất riêng từ lớp cha

20



Các tính chất của HĐT

- Đơn thừa kế: một lớp con chỉ thừa kế từ một lớp cha duy nhất



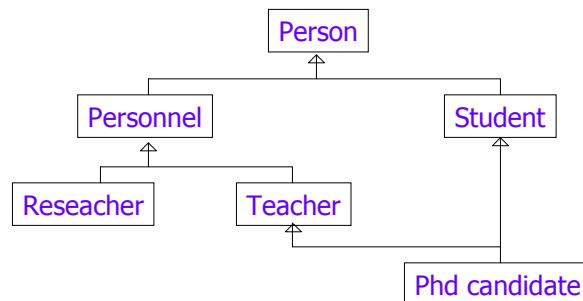
- Lớp trừu tượng hay lớp chung: XeÔtô
- Lớp cụ thể hay lớp chuyên biệt: XeKhách
- Lớp chuyên biệt có thể thay thế lớp chung trong tất cả các ứng dụng. Ví dụ: Ôtô tải **là một** ô tô.

21



Các tính chất của HĐT

- Đa thừa kế: một lớp con thừa kế từ nhiều lớp cha khác nhau



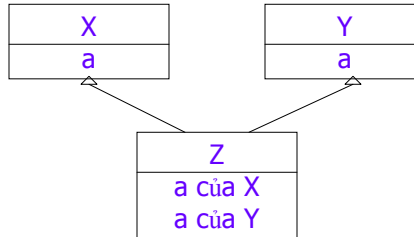
22



Các tính chất của HĐT

o Đa thừa kế

- Đụng độ tên các thuộc tính



- Đa thừa kế không được chấp nhận bởi một số ngôn ngữ: Java

23



Các tính chất của HĐT

o Ưu điểm của thừa kế

- Phân loại các lớp: các lớp được phân loại, sắp xếp theo một thứ bậc để dễ quản lí
- Xây dựng các lớp: các lớp con được xây dựng từ các lớp cha
- Tiết kiệm thời gian xây dựng, tránh lặp lại thông tin

24



Các tính chất của HĐT

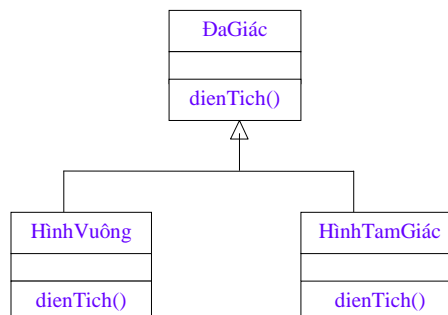
- Tính đa hình (polymorphism): của phương thức, tức là khả năng các phương thức khác nhau được thực hiện để trả lời cùng một yêu cầu
- Mỗi lớp con thừa kế đặc tả các phương thức từ lớp cha, và các phương thức này có thể được sửa đổi trong lớp con để thực hiện các chức năng riêng trong lớp đó
- Một phương thức (cùng một tên phương thức) có nhiều dạng (định nghĩa) khác nhau trong các lớp khác nhau

25



Các tính chất của HĐT

- Ví dụ tính đa hình



26



Nội dung

- Khái niệm cơ bản hướng đối tượng
- **Biểu đồ ca sử dụng**
- Thiết kế cấu trúc tĩnh
- Thiết kế cấu trúc động
- Sinh mã

27



Ca sử dụng (Use case)

- Bước đầu tiên của phân tích yêu cầu là xác định các ca sử dụng của hệ thống
- Một **ca sử dụng** là một tương tác giữa hệ thống và môi trường
- Tập hợp các ca sử dụng là mô tả toàn bộ hệ thống cần xây dựng

28



Ca sử dụng

- Ví dụ: *phát triển một phần mềm thảo văn bản*
- Các ca sử dụng có thể:
 - *Nhập văn bản mới*
 - *Sửa văn bản đã tồn tại*
 - *Tạo mục lục*
 - *Chép đoạn văn bản*
 - ...

29



Ca sử dụng

- Một ca sử dụng tương ứng với một chức năng của hệ thống dưới **góc nhìn của người sử dụng**
- Một ca sử dụng có thể lớn hoặc nhỏ
- Một ca sử dụng chỉ ra làm thế nào một **mục tiêu của người sử dụng** được thỏa mãn bởi hệ thống

30



Ca sử dụng

- Cần phân biệt các **mục tiêu** của người sử dụng và các **tương tác** của họ với hệ thống
 - Mục tiêu: cái mà người sử dụng mong đợi
 - Tương tác: kỹ thuật cho phép đáp ứng mục tiêu
- Ví dụ
 - Mục tiêu: *có được một văn bản trình bày đẹp*
 - Tương tác: *chọn định dạng trang, chọn font chữ, định nghĩa các kiểu tiêu đề (heading), ...*
- Thực tế, chúng ta xác định các mục tiêu trước, sau đó chọn tập hợp các tương tác đáp ứng các mục tiêu đó

31



Ca sử dụng

- Ví dụ: cần xây dựng một hệ thống ATM cho phép rút tiền
- Có thể có vài tương tác chung trong một kịch bản sau:
 - Đưa thẻ vào
 - Nhập mã PIN
 - Chọn số tiền rút
 - Khẳng định số tiền rút
 - Lấy thẻ ra
 - Lấy tiền
 - Lấy phiếu rút tiền
- Các tương tác trên có là các ca sử dụng không ?

32



Ca sử dụng

- Câu trả lời: không.
- Tại sao ?
- Vì chẳng hạn “Nhập mã PIN” không đáp ứng một mục tiêu nào của người sử dụng.
- Mục tiêu của người sử dụng là “Rút tiền”, vậy đó nên là một ca sử dụng.

33



Tác nhân (Actor)

- **Tác nhân** đóng vai trò một người sử dụng hoặc một thực thể bên ngoài tương tác với hệ thống
- Ví dụ: *Cần phát triển hệ thống tính tiền ở siêu thị*
 - Các tác nhân có thể là: *Khách hàng, Người bán hàng, Người quản lý, Kho hàng*
- Cần phân biệt: tác nhân (actor) và người sử dụng (user)
 - Nhiều người sử dụng có thể tương ứng một tác nhân: *nhiều người bán hàng khác nhau đóng cùng vai trò đối với hệ thống*
 - Một người sử dụng có thể tương ứng với nhiều tác nhân khác nhau: *cùng một người có thể đồng thời đóng hai vai trò là người bán hàng và người quản lý*

34



Tác nhân

- Tác nhân không nhất thiết luôn luôn là con người
- Tác nhân có thể là môi trường, hệ thống khác, thực thể bên ngoài tương tác với hệ thống
- Ví dụ
 - *Kho hàng* là có thể một cơ sở dữ liệu

35



Đặc tả ca sử dụng

- Đặc tả điển hình của một ca sử dụng:
 - Ca sử dụng: tên ca sử dụng thường bắt đầu bởi một động từ
 - Các tác nhân: danh sách các tác nhân liên quan
 - Mô tả: tóm tắt các xử lý cần thực hiện
- Ví dụ
 - **Ca sử dụng:** *Mua hàng*
 - **Các tác nhân:** *Khách hàng, Người bán hàng*
 - **Mô tả:** Một *khách hàng* sau khi đã chọn các mặt hàng, mang giỏ hàng đến quầy thu tiền. *Người bán hàng* ghi nhận các mặt hàng, thông báo tổng số tiền, thu tiền và trả tiền còn lại cho khách hàng. Khách hàng mang hàng đi.

36



Đặc tả ca sử dụng

- Đặc tả ca sử dụng có thể thêm:
 - Tham chiếu (reference) đến mục liên quan trong đặc tả yêu cầu
 - Điều kiện trước và điều kiện sau khi thực hiện ca sử dụng
- Ví dụ
 - **Ca sử dụng:** *Mua hàng*
 - **Các tác nhân:** *Khách hàng, Người bán hàng*
 - **Tham chiếu:** R1.2, R2.3
 - **Điều kiện trước:** Người bán hàng đã đăng nhập thành công.
 - **Điều kiện sau:** Các mặt hàng bán đã được ghi nhận và đã ghi nhận thanh toán tiền.
 - **Mô tả:** Một *khách hàng* sau khi đã chọn các mặt hàng, mang giỏ hàng đến quầy thu tiền. *Người bán hàng* ghi nhận các mặt hàng, thông báo tổng số tiền, thu tiền và trả tiền còn lại cho khách hàng. Khách hàng mang hàng đi.

37



Đặc tả ca sử dụng

- Ngoài ra, đối với mỗi ca sử dụng ta có thể xây dựng một kịch bản (scenario) hành động mô tả các sự kiện xảy ra
- Kịch bản: gồm các sự kiện chính và các sự kiện ngoại lệ
- Các sự kiện chia làm hai luồng
 - Luồng tương ứng với các tác nhân
 - Luồng tương ứng với hệ thống

38



Đặc tả ca sử dụng

○ Các sự kiện chính

Hành động của tác nhân	Hành động của hệ thống
<p>1. Một <i>khách hàng</i> đưa hàng đã chọn mua đến quầy tính tiền.</p> <p>2. <i>Người bán hàng</i> ghi nhận từng mặt hàng.</p> <p>Nếu một mặt hàng có số lượng nhiều hơn một thì <i>người bán hàng</i> có thể nhập vào một số.</p>	<p>3. Xác định mặt hàng, hiển thị các thông tin và giá mặt hàng.</p> <p>Số này được hiển thị.</p>

39



Đặc tả ca sử dụng

○ Các sự kiện chính (tiếp)

Hành động của tác nhân	Hành động của hệ thống
<p>4. Sau khi đã ghi nhận tất cả các mặt hàng, <i>người bán hàng</i> báo hiệu kết thúc việc ghi nhận hàng.</p> <p>6. <i>Người bán hàng</i> thông báo tổng số tiền phải trả cho <i>khách hàng</i>.</p> <p>7. <i>Khách hàng</i> trả tiền cho <i>người bán hàng</i>.</p>	<p>5. Tính và hiển thị tổng số tiền.</p>

40



Đặc tả ca sử dụng

o Các sự kiện chính (tiếp)

Hành động của tác nhân	Hành động của hệ thống
8. <i>Người bán hàng nhập số tiền khách hàng trả.</i>	9. <i>Hiển thị tiền dư và in phiếu bán hàng</i>
10. <i>Người bán hàng xác nhận sự trả tiền, lấy tiền dư trả cho khách hàng và đưa cho khách hàng phiếu bán hàng.</i>	11. <i>Ghi nhận phiên bán hàng.</i>
12. <i>Khách hàng rời quầy thu tiền với túi hàng</i>	

41



Đặc tả ca sử dụng

o Các sự kiện phụ

Hành động của tác nhân	Hành động của hệ thống
7. <i>Khách hàng không có đủ tiền. Người bán hàng hủy bỏ việc bán.</i>	3. <i>Sự xác nhận mặt hàng không đúng. Hiển thị lỗi.</i>

Lưu ý: định dạng đặc tả các ca sử dụng không cần thiết phải chặt chẽ.

42



Ca sử dụng ở giai đoạn Elaboration

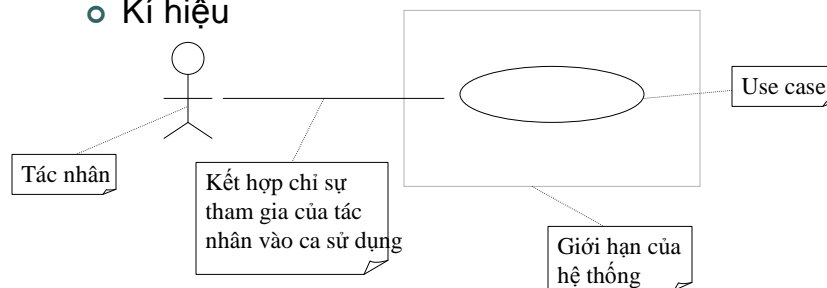
- Xác định càng nhiều ca sử dụng một cách có thể
- Không đi vào quá chi tiết, nhằm giảm độ phức tạp
- Một **mô tả ngắn gọn về mỗi ca sử dụng** là đủ, có thể bỏ qua phần kịch bản, tham chiếu đến đặc tả yêu cầu, điều kiện trước và điều kiện sau.
- Bảo đảm rằng các ca sử dụng bao quát hết các yêu cầu của hệ thống

43



Biểu đồ ca sử dụng

- Biểu đồ ca sử dụng mô tả quan hệ giữa các tác nhân và các ca sử dụng của một hệ thống.
- Kí hiệu

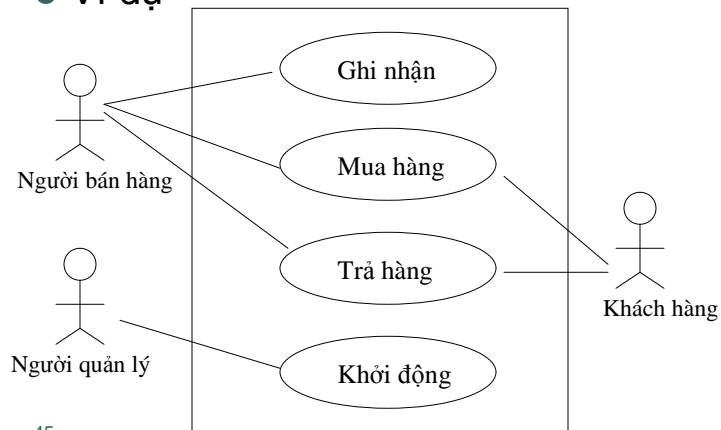


44



Biểu đồ ca sử dụng

◦ Ví dụ



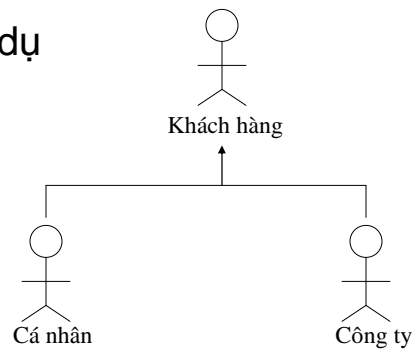
45



Biểu đồ ca sử dụng

◦ Các tác nhân có thể có quan hệ thừa kế

◦ Ví dụ



46



Quan hệ mở rộng

- Có thể xảy ra trường hợp: một ca sử dụng tương tự với một ca sử dụng khác, tuy nhiên nó gồm thêm một số hành động
- Ví dụ
 - **Ca sử dụng:** *Mua hàng bằng thẻ tín dụng*
 - **Các tác nhân:** *Khách hàng, Người bán hàng*
 - **Mô tả:** Một *khách hàng* sau khi đã chọn các mặt hàng, mang giỏ hàng đến quầy thu tiền. *Người bán hàng* ghi nhận các mặt hàng, thông báo tổng số tiền. *Khách hàng* đưa thẻ vào máy và nhập mã PIN. *Khách hàng* nhận phiếu bán hàng và mang hàng đi.

47



Quan hệ mở rộng

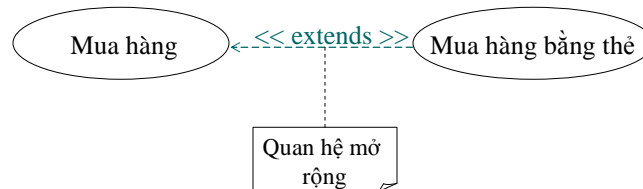
- Ca sử dụng này là một biến thể của ca sử dụng “*mua hàng*”, tuy nhiên thêm vào các hành động liên quan đến trả tiền bằng thẻ
- Ca sử dụng “*mua hàng bằng thẻ tín dụng*” là một **sự mở rộng** của ca sử dụng “*mua hàng*”

48



Quan hệ mở rộng

- Kí hiệu



- Nếu một ca sử dụng kết hợp với một tác nhân, thì tất cả các ca sử dụng mở rộng đều kết hợp với tác nhân đó

49



Quan hệ sử dụng

- Trường hợp nhiều ca sử dụng **chia sẻ cùng một dãy các hành động**. Nếu phần chung là quan trọng và hướng tới một mục tiêu rõ ràng, như thế ta có thể xây dựng một ca sử dụng riêng
- Ví dụ: chúng ta muốn chấp nhận *mua hàng trả tiền một lần* và *mua hàng trả góp*
- Hai ca sử dụng “*mua hàng trả tiền một lần*” và “*mua hàng trả góp*” thực hiện một dãy các hành động mà có thể được mô tả bởi ca sử dụng “*ghi nhận các mặt hàng*”

50



Quan hệ sử dụng

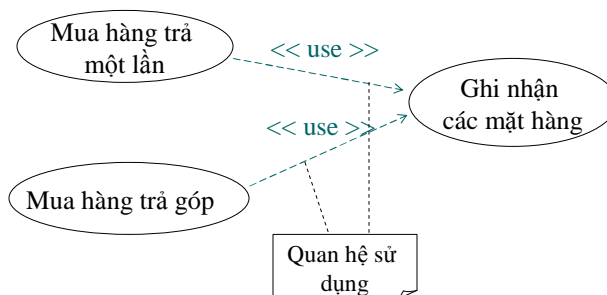
- Đặc tả của ca sử dụng “*ghi nhận các mặt hàng*”
 - **Ca sử dụng:** *ghi nhận các mặt hàng*
 - **Các tác nhân:** người bán hàng, khách hàng
 - **Mô tả:** *Khách hàng mang các mặt hàng đến quầy tính tiền. Người bán hàng ghi nhận các mặt hàng và thông báo tổng số tiền phải trả.*

51



Quan hệ sử dụng

- Kí hiệu



Ngược với quan hệ mở rộng, các ca sử dụng trong quan hệ sử dụng không nhất thiết kết hợp với cùng tác nhân.

52



Cách xác định các ca sử dụng

- Phương pháp phỏng vấn
 - Khó khăn, vì hai người khác nhau được phỏng vấn có thể đưa ra ý kiến khác nhau
- Phương pháp hội thảo (workshop)
 - Tập hợp tất cả những ai liên quan đến hệ thống để thảo luận: các nhà tin học và khách hàng (người sử dụng)
 - Mỗi người đều đưa ra ý kiến

53



Cách xác định các ca sử dụng

- Cách tiến hành hội thảo
 - Liệt kê tất cả các tác nhân có thể
 - Liệt kê tất cả các ca sử dụng có thể
 - Phân tích, biện chứng mỗi ca sử dụng bằng cách viết ra một mô tả đơn giản
 - Mô hình hóa các ca sử dụng và tác nhân

54



Cách xác định các ca sử dụng

- Khuyến khích
 - Không nên cố gắng tìm mọi ca sử dụng,
 - Trong quá trình phát triển các ca sử dụng sẽ lộ diện dần
 - Nếu không thể biện chứng cho một ca sử dụng
 - Có thể đó không phải là ca sử dụng

55



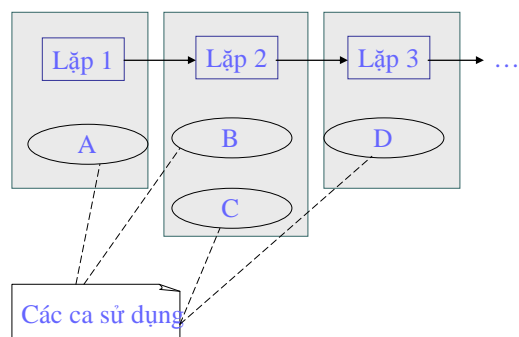
Sắp xếp các ca sử dụng

- Khi tất cả các ca sử dụng đã được xác định
- Tiến trình phát triển gồm nhiều bước lặp
- Mỗi bước lặp thực hiện thiết kế, mã hóa và kiểm thử chỉ **một vài ca sử dụng**
- Làm sao chia các ca sử dụng vào các bước lặp?

56



Sắp xếp các ca sử dụng



57



Sắp xếp các ca sử dụng

- Các ca sử dụng nên được thực hiện trước
 - Các ca sử dụng chứa các rủi ro/nguy cơ
 - Các ca sử dụng kiến trúc chính
 - Các ca sử dụng đòi hỏi nghiên cứu mới, công nghệ mới
 - Các ca sử dụng mà khách hàng quan tâm hơn

58



Bài tập 1

- **Máy rút tiền ATM** có các chức năng chính như sau:
 - Cấp phát tiền cho những ai có thẻ ngân hàng (cho phép rút một số lượng tiền bởi hệ thống thông tin của ngân hàng) và những ai có thẻ VISA (cho phép từ xa bởi hệ thống VISA)
 - Cho xem kiểm tra số tiền tài khoản và bỏ tiền vào tài khoản bằng tiền mặt hoặc ngân phiếu đối với những ai có thẻ ngân hàng
- Tất cả các giao tác đều được kiểm tra an toàn
 - Kiểm tra mã PIN
 - Mã PIN nhập sai 3 lần thì thẻ sẽ bị “nuốt”
- Cần phải thường xuyên nạp tiền vào máy, lấy ngân phiếu và các thẻ bị nuốt ra
- Xác định các tác nhân, các ca sử dụng và vẽ biểu đồ ca sử dụng

59



Bài tập 1

- Các tác nhân
 - Người có thẻ ngân hàng (bankcard)
 - Người có thẻ VISA (VISAcards)
 - Người vận hành máy (operator)
 - Hệ thống VISA (VISA)
 - Hệ thống thông tin ngân hàng (bank)

60



Bài tập 1

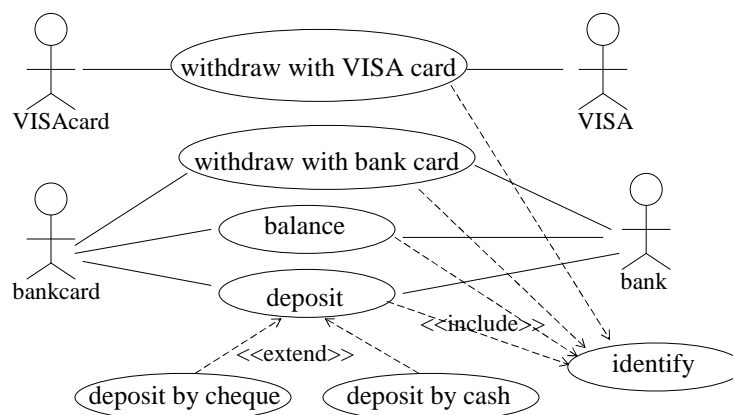
o Các ca sử dụng

- Rút tiền với thẻ ngân hàng (withdraw by bankcard)
- Rút tiền với thẻ VISA (withdraw by VISAcards)
- Kiểm tra mã PIN (identify)
- Xem số tiền còn trong tài khoản (balance)
- Bỏ tiền vào tài khoản bằng ngân phiếu hoặc tiền mặt (deposit)
- Nạp tiền vào máy (put money)
- Lấy thẻ bị nuốt trong máy (get cards)
- Lấy ngân phiếu trong máy (get cheques)

61



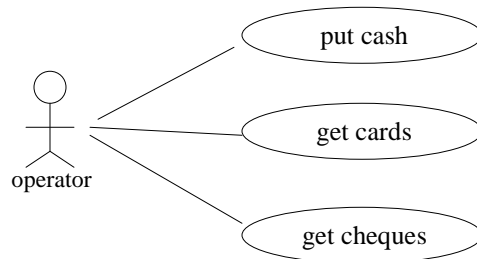
Bài tập 1



62



Bài tập 1



63



Bài tập 2

- **Quản lý đào tạo nhân viên:** Một công ty muốn mô tả bằng UML việc đào tạo nhân viên để tin học hóa một số công việc. Việc đào tạo được bắt đầu khi người quản lý đào tạo nhận được yêu cầu đào tạo của một nhân viên. Nhân viên này có thể xem danh mục các chuyên đề đào tạo của các đơn vị đào tạo ký kết với công ty. Yêu cầu của nhân viên được xem xét bởi người quản lý đào tạo và người quản lý sẽ trả lời là chấp nhận hay từ chối đề nghị đó. Trong trường hợp chấp nhận, người quản lý sẽ xác định chuyên đề phù hợp trong danh mục các chuyên đề, sau đó gửi cho nhân viên nội dung của chuyên đề và danh sách các khóa đào tạo. Nhân viên sẽ chọn khóa đào tạo và người quản lý sẽ đăng ký khóa học với đơn vị đào tạo cho nhân viên. Trong trường hợp muốn hủy bỏ đăng ký khóa đào tạo, nhân viên phải thông báo sớm cho người quản lý biết để người quản lý thực hiện hủy bỏ. Cuối khóa đào tạo, nhân viên chuyển phiếu đánh giá kết quả học về cho công ty. Người quản lý sẽ kiểm tra hóa đơn thanh toán tiền của đơn vị đào tạo.
- Xây dựng biểu đồ ca sử dụng.

64



Nội dung

- Khái niệm cơ bản hướng đối tượng
- Biểu đồ ca sử dụng
- **Thiết kế cấu trúc tĩnh**
- Thiết kế cấu trúc động
- Sinh mã

65



Cấu trúc tĩnh

- Mô hình khái niệm
- Biểu đồ lớp
- Biểu đồ đối tượng

66



Mô hình khái niệm

- Xác định các “khái niệm” quan trọng trong hệ thống
- Mô hình khái niệm (conceptual model) mô tả các khái niệm trong các quan hệ của chúng
- UML không cung cấp mô hình khái niệm, tuy nhiên cung cấp kí hiệu và cú pháp để biểu diễn mô hình đó chính là **biểu đồ lớp**
- Ở giai đoạn này, mô hình khái niệm còn được gọi biểu đồ lớp phân tích (analysis class diagram) – lưu ý, khác với biểu đồ lớp thiết kế (design class diagram)
- Ngoài ra, mô hình khái niệm cũng còn được gọi là mô hình lĩnh vực (domain model)

67



Mô hình khái niệm

- Mô hình khái niệm gồm:
 - **Các khái niệm** của lĩnh vực nghiên cứu
 - **Các thuộc tính và các thao tác** của các khái niệm này
 - **Các quan hệ** của các khái niệm này
- Một khái niệm là biểu diễn ở mức cao (trừu tượng) về một sự vật
- Một khái niệm là một phần tử của lĩnh vực nghiên cứu, chứ không phải một phần tử của phần mềm hay hệ thống

68



Mô hình khái niệm

- Trong mô hình khái niệm, chúng ta sẽ nắm bắt các khái niệm nhận biết bởi khách hàng
- Ví dụ các khái niệm đúng: **khái niệm gắn liền với vấn đề**
 - **Thang máy** trong hệ thống điều khiển thang máy
 - **Vé máy bay** trong hệ thống đặt vé máy bay
 - **Đặt hàng** trong hệ thống mua bán hàng qua mạng
- Ví dụ tồi về khái niệm: **khái niệm gắn liền với giải pháp**
 - **DanhSachKhachHang** – bảng các khách hàng
 - **EventTrigger** – tiến trình thực hiện duyệt hệ thống 10 phút một lần

69



Mô hình khái niệm

- Làm sao biết được một khái niệm là đúng hay không?
- Nguyên tắc: **“Nếu khách hàng không hiểu khái niệm, rất có thể đó không phải là khái niệm”**
- Mô hình khái niệm sẽ được chuyển dần sang biểu đồ lớp thiết kế trong giai đoạn xây dựng

70



Xác định các khái niệm

- Để xác định các khái niệm, dựa vào đặc tả yêu cầu, mà cụ thể hơn là dựa vào các ca sử dụng
- Ví dụ: ca sử dụng “*mua hàng*”
 - Các khái niệm có thể: *KháchHàng*, *NgườiBánHàng*, *TínhTiền*, *MuaHàng*, *MặtHàng*, ...

71



Xác định các khái niệm

- Một số ứng cử viên của khái niệm từ đặc tả hoặc ca sử dụng:
 - Các đối tượng vật lý (xe ô tô)
 - Các vị trí, địa điểm (nhà ga)
 - Các giao tác (thanh toán)
 - Các vai trò của con người (người bán)
 - Các hệ thống khác ở bên ngoài (cơ sở dữ liệu từ xa)
 - Danh từ trừu tượng (sự khát, ăn uống)
 - Các tổ chức (đại học)
 - Các sự kiện (cấp cứu)
 - Nguyên tắc/chính sách

72



Xác định các khái niệm

- Cách khác để xác định các khái niệm
 - Các danh từ và cụm danh từ trong đặc tả yêu cầu hoặc đặc tả ca sử dụng có thể là các khái niệm
 - Dựa vào hiểu biết và kinh nghiệm loại bỏ các danh từ và cụm danh từ không là các khái niệm
- Ví dụ: dựa vào kịch bản ca sử dụng “*mua hàng*”
 - Gạch chân các danh từ và cụm danh từ

73



Xác định các khái niệm

- Ví dụ

Hành động của tác nhân	Hành động của hệ thống
<p>1. Một <u>khách hàng</u> đưa <u>hàng đã chọn</u> mua đến <u>quầy tính tiền</u>.</p> <p>2. <u>Người bán hàng</u> ghi nhận từng <u>mặt hàng</u>.</p> <p>Nếu một <u>mặt hàng</u> có số lượng nhiều hơn một thì <u>người bán hàng</u> có thể nhập vào <u>một số</u>.</p>	<p>3. Xác định <u>mặt hàng</u>, hiển thị các <u>thông tin</u> và <u>giá mặt hàng</u>.</p> <p><u>Số</u> này được hiển thị.</p>

74



Xác định các khái niệm

○ Ví dụ

Hành động của tác nhân	Hành động của hệ thống
<p>4. Sau khi đã ghi nhận tất cả các mặt hàng, <i>người bán hàng</i> báo hiệu kết thúc <i>việc ghi nhận hàng</i>.</p> <p>6. <i>Người bán hàng</i> thông báo <i>tổng số tiền</i> phải trả cho <i>khách hàng</i>.</p> <p>7. <i>Khách hàng</i> trả tiền cho <i>người bán hàng</i>.</p>	<p>5. Tính và hiển thị <i>tổng số tiền</i>.</p>

75



Xác định các khái niệm

○ Ví dụ

Hành động của tác nhân	Hành động của hệ thống
<p>8. <i>Người bán hàng</i> nhập <i>số tiền khách hàng trả</i>.</p> <p>10. <i>Người bán hàng</i> xác nhận sự trả tiền, lấy <i>tiền dư</i> trả cho <i>khách hàng</i> và đưa cho khách hàng <i>phiếu bán hàng</i>.</p> <p>12. <i>Khách hàng</i> rời <i>quầy thu tiền</i> với <i>túi hàng</i></p>	<p>9. Hiển thị <i>tiền dư</i> và in <i>phiếu bán hàng</i></p> <p>11. Ghi nhận <i>phiên bán hàng</i>.</p>

76



Xác định các khái niệm

- Phân biệt giữa khái niệm (concept) và thuộc tính (attribut)
 - Nếu một phần tử của lĩnh vực nghiên cứu không là một con số hoặc một chuỗi kí tự thì có thể đó là một khái niệm
 - Ví dụ: Cần xây dựng phần mềm quản lý các chuyến bay. *Đích* của một chuyến bay là thuộc tính của một chuyến bay hay là một khái niệm khác ?
 - Trả lời: *Đích* một chuyến bay là một sân bay, không phải là một con số hay văn bản, đó là một khái niệm

77



Xác định các khái niệm

- Lớp “MôTả”
 - Lớp MôTả là lớp chứa thông tin mô tả các đối tượng khác
 - Ví dụ: Lớp MặtHàng chứa các thông tin về Mặt Hàng

MặtHàng
mãMH tênMH: text giá sốXêri màuSắc

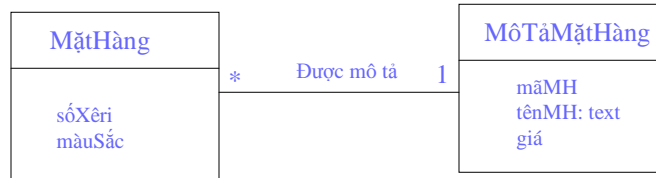
Phương án 1 (chưa tốt)

78



Xác định các khái niệm

o Lớp “MôTả”



Phương án 2 (tốt hơn)

79



Xác định các khái niệm

o Lớp “MôTả”

- Khi nào sử dụng lớp “MôTả”
 - Khi cần giảm bớt sự dư thừa, trùng lặp thông tin
 - Khi cần mô tả về đối tượng độc lập với các đối tượng cụ thể
 - Khi cần duy trì thông tin về đối tượng cho dù các đối tượng cụ thể bị xóa

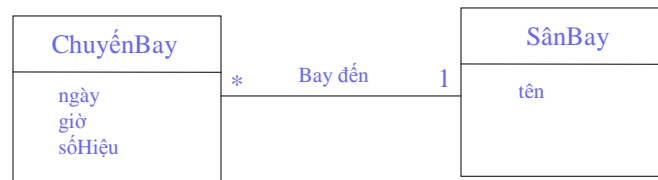
80



Xác định các khái niệm

o Lớp “MôTả”

- Ví dụ: trong lĩnh vực hàng không, cần mô tả quan hệ giữa các chuyến bay và các sân bay



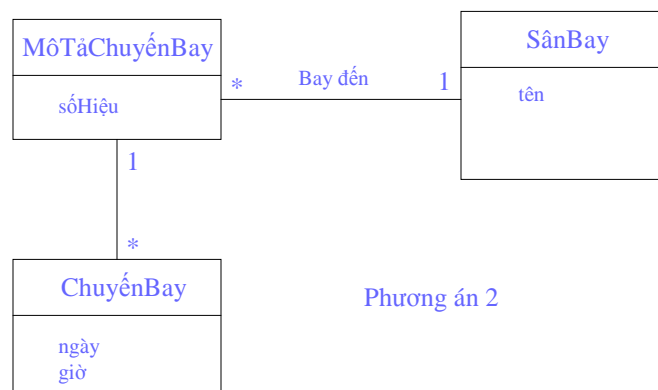
Phương án 1

81



Xác định các khái niệm

o Lớp “MôTả”



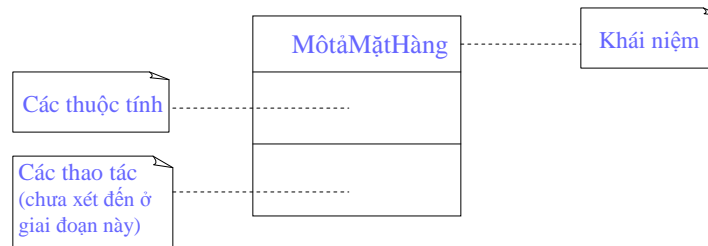
Phương án 2

82



Biểu diễn khái niệm

- Sử dụng kí hiệu của biểu đồ lớp

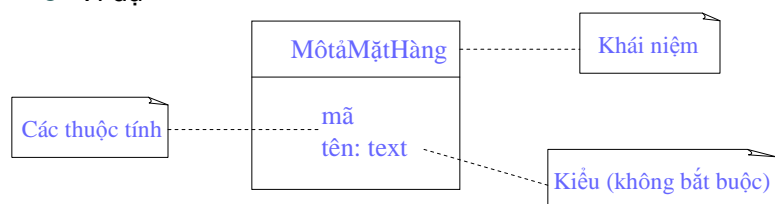


83



Thuộc tính

- Các thuộc tính (attribut) của một khái niệm biểu diễn dữ liệu cần thiết cho các thể hiện (instance) của khái niệm
- Ví dụ

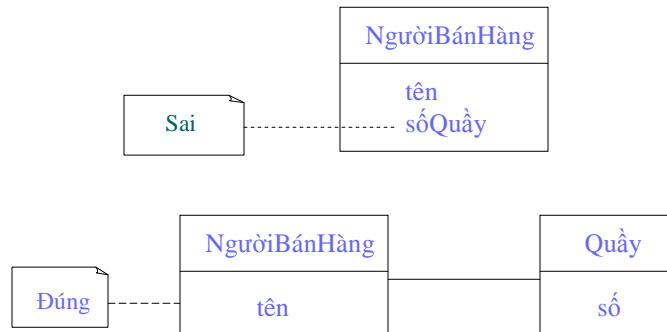


84



Thuộc tính

- Một thuộc tính chỉ đại diện cho các dữ liệu liên quan đến khái niệm **sở hữu** thuộc tính đó
- Ví dụ



85



Thuộc tính

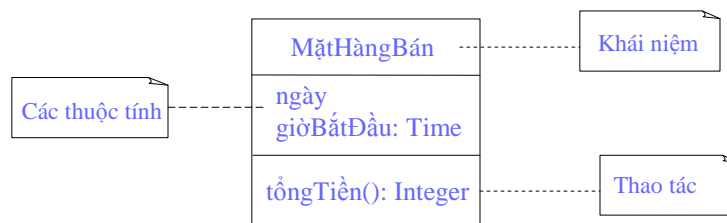
- Cách xác định các thuộc tính
 - Các con số và chuỗi kí tự là các thuộc tính
 - Nếu một tính chất của một khái niệm không thể làm được điều gì thì rất có thể đó là thuộc tính
 - Nếu nghi ngờ một thuộc tính là khái niệm, thì đơn giản hãy coi đó là khái niệm
 - Ví dụ: *lương* là thuộc tính hay khái niệm so với khái niệm *công nhân* ?
 - Nếu nghi ngờ đó là khái niệm thì coi như *lương* và *công nhân* là hai khái niệm tách rời

86



Thao tác

- Khái niệm có thể có các thao tác (operation)
- Thao tác của khái niệm chính là khả năng thực hiện của một thể hiện của khái niệm
- Ví dụ



87



Thao tác

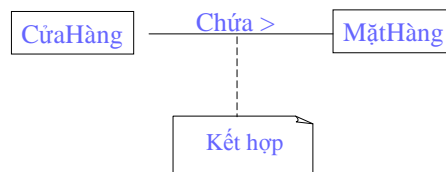
- Ở giai đoạn elaboration, mô hình khái niệm có thể **không nhất thiết phải mô tả các thao tác** của khái niệm
- Giai đoạn construction sẽ thực hiện công việc này một cách chi tiết và đầy đủ

88



Kết hợp

- **Kết hợp** (association) biểu diễn quan hệ giữa các thể hiện của các khái niệm
- Ví dụ: kết hợp *chứa* giữa khái niệm *cửa hàng* và khái niệm *mặt hàng*
- Kí hiệu

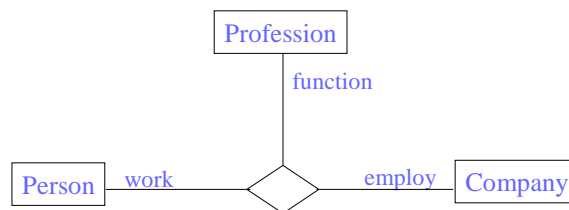


89



Kết hợp

- Có thể tồn tại kết hợp của nhiều hơn hai khái niệm
- Ví dụ



90



Kết hợp

- **Bội số** (multiplicity) của vai trò chỉ ra số thể hiện có thể của quan hệ tham gia vào quan hệ
- Các bội số có thể
 - 1: chỉ đúng một
 - 1..*: từ một đến nhiều
 - *: từ 0 đến nhiều
 - m..n: từ m đến n
- Ví dụ

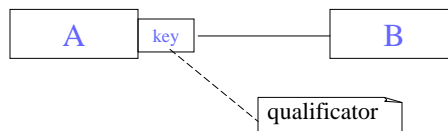


91



Hạn chế kết hợp (qualificator)

- Giảm số thể hiện tham gia vào một kết hợp
- Kí hiệu



- Mỗi thể hiện A với giá trị *key* xác định một tập con các thể hiện B tham gia vào kết hợp

92

Hạn chế kết hợp (qualificator)

- Ví dụ

- Phân biệt các sinh viên học tại một đại học dựa vào mã số sinh viên



- Phân biệt các mặt hàng thuộc vào một danh mục mặt hàng dựa vào mã số mặt hàng



93

Chuyên biệt hóa

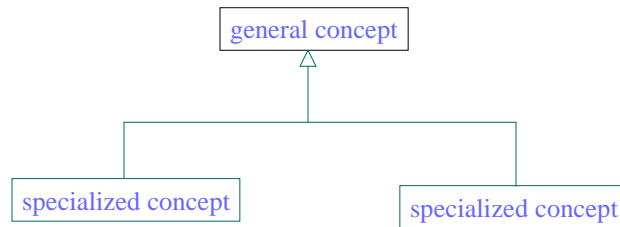
- Một khái niệm có thể về cơ bản giống với một khái niệm khác, chỉ có một vài sự khác nhau trên một số tính chất (thuộc tính, thao tác, các kết hợp)
- Khái niệm thứ nhất được gọi là **chuyên biệt hóa** (specialization) của khái niệm thứ hai
- Khái niệm thứ nhất được gọi là khái niệm chuyên biệt hóa (specialized concept), khái niệm thứ hai được gọi là khái niệm chung (general concept)

94



Chuyên biệt hóa

- Kí hiệu



95



Thừa kế

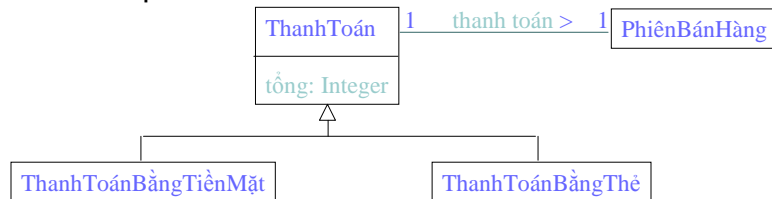
- Khái niệm chuyên biệt hóa **thừa kế** (inheritance) tất cả các tính chất của của khái niệm chung. Các tính chất này bao gồm:
 - Các thuộc tính
 - Các thao tác
 - Các kết hợp với khái niệm khác

96



Thừa kế

- Ví dụ



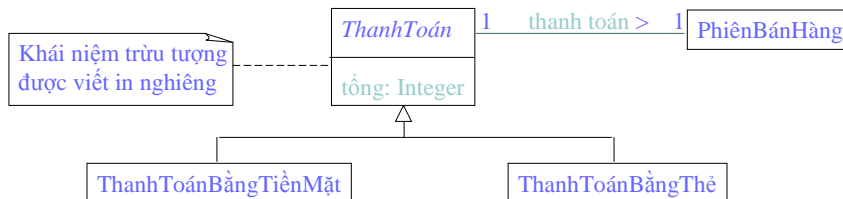
- Các khái niệm “ThanhToánBằngTiềnMặt” và “ThanhToánBằngThẻ” đều có thuộc tính “tổng” và kết hợp “thanh toán” với khái niệm “PhiênBánHàng”

97



Khái niệm trừu tượng

- Tương tự như lớp, khái niệm trừu tượng không có các thể hiện



- Lưu ý, trong tài liệu viết tay có thể viết {abstract} dưới tên khái niệm trừu tượng

98



Quan hệ hợp thành và quan hệ kết tập

- **Quan hệ hợp thành** (composition) và **quan hệ kết tập** (agregation) là hai **kết hợp đặc biệt** chỉ **sự sở hữu**

- Quan hệ hợp thành: một khái niệm thành phần chỉ thuộc vào một khái niệm toàn phần



- Quan hệ kết tập: một khái niệm thành phần có thể thuộc vào nhiều khái niệm toàn phần

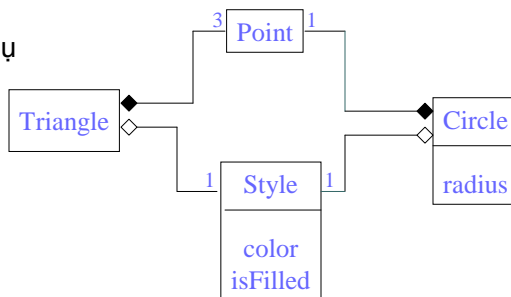


99



Quan hệ hợp thành và quan hệ kết tập

- Ví dụ



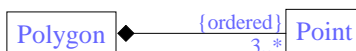
- Một thể hiện của “Point” không thể đồng thời thuộc vào một thể hiện của “Triangle” và một thể hiện của “Circle”

100



Quan hệ hợp thành và quan hệ kết tập

- Quan hệ hợp thành **nhấn mạnh sự sở hữu**: nếu khái niệm toàn phần bị hủy bỏ thì khái niệm thành phần cũng bị hủy bỏ theo
- Trong trường hợp, **thứ tự** của các khái niệm thành phần là quan trọng, thì thêm vào kết hợp điều kiện {ordered}
- Ví dụ

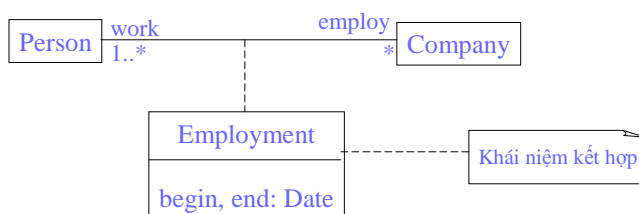


101



Khái niệm kết hợp

- Có thể mô tả các **tính chất** của một kết hợp giữa hai khái niệm bởi một **khái niệm kết hợp** (association concept)
- Ví dụ



102



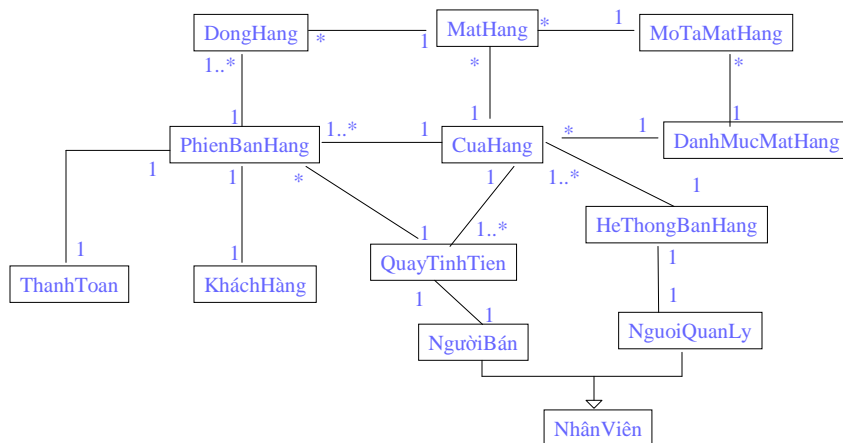
Bài tập 1

- Xây dựng mô hình khái niệm của hệ thống/phần mềm bán hàng tại siêu thị
 - Phần mềm bán hàng sử dụng tại siêu thị nhằm giúp ghi nhận hoạt động bán hàng, xử lý các thanh toán với khách hàng. Phần mềm được sử dụng bởi người bán hàng và được quản lý bởi người quản lý siêu thị. Phần mềm nhằm tự động hóa công việc của người bán hàng tại quầy thu tiền.

103



Bài tập 1



104



Bài tập 2

- **Quản lý đào tạo ở trung tâm tin học:** Một công ty muốn mô tả bằng UML việc đào tạo nhân viên để tin học hóa một số công việc. Việc đào tạo được bắt đầu khi người quản lý đào tạo nhận được yêu cầu đào tạo của một nhân viên. Nhân viên này có thể xem danh mục các chuyên đề đào tạo của các đơn vị đào tạo ký kết với công ty. Yêu cầu của nhân viên được xem xét bởi người quản lý đào tạo và người quản lý sẽ trả lời là chấp nhận hay từ chối đề nghị đó. Trong trường hợp chấp nhận, người quản lý sẽ xác định chuyên đề phù hợp trong danh mục các chuyên đề, sau đó gửi cho nhân viên nội dung của chuyên đề và danh sách các khóa đào tạo. Nhân viên sẽ chọn khóa đào tạo và người quản lý sẽ đăng ký khóa học với đơn vị đào tạo cho nhân viên. Trong trường hợp muốn hủy bỏ đăng ký khóa đào tạo, nhân viên phải thông báo sớm cho người quản lý biết để người quản lý thực hiện hủy bỏ. Cuối khóa đào tạo, nhân viên chuyển phiếu đánh giá kết quả học về cho công ty. Người quản lý sẽ kiểm tra hóa đơn thanh toán tiền của đơn vị đào tạo.
- Xây dựng biểu mô hình khái niệm.

105



Biểu đồ lớp

- **Biểu đồ lớp định nghĩa:**
 - **Các lớp (class)**
 - Các **thuộc tính** (attribut) của lớp: các biến và kiểu của chúng
 - Các **thao tác** (operation) của lớp: các phương thức (method), các tham đối và có thể giá trị trả về
 - **Các quan hệ giữa các lớp**

106



Biểu đồ lớp

- Biểu đồ lớp có cùng quy tắc cú pháp với mô hình khái niệm
 - Thực ra, mô hình khái niệm sử dụng các cú pháp của biểu đồ lớp trong UML
 - **Tất cả các kí hiệu và quy tắc** (đã trình bày) **đối với mô hình khái niệm đều được sử dụng để xây dựng biểu đồ lớp**
- Biểu đồ lớp được xây dựng dựa trên mô hình khái niệm
- Các **lớp** có thể chủ yếu là các **khái niệm** hoặc các **thành phần khác**
- Biểu đồ lớp sẽ là nền tảng cho bước mã hóa

107



Biểu đồ lớp

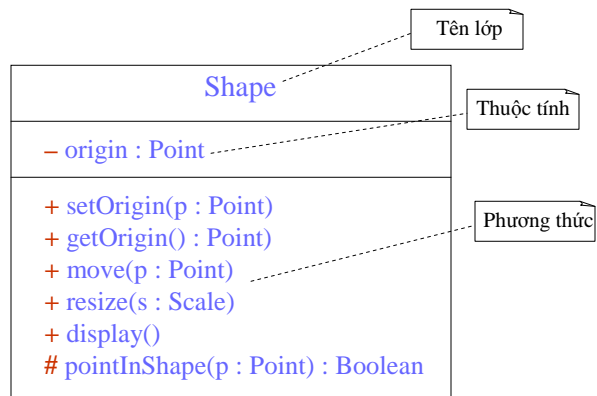
- Đối với biểu đồ lớp, mỗi thuộc tính hay mỗi phương thức có thể có thêm mức khả kiến – khả năng nhìn thấy (visibility)
- Kí hiệu
 - “-” **mức riêng** (private), thuộc tính hay phương thức chỉ được nhìn thấy bởi đối tượng của lớp đó
 - “#” **mức bảo vệ** (protected), thuộc tính hay phương thức chỉ được nhìn thấy bởi đối tượng của lớp đó và đối tượng của các lớp thừa kế lớp đó
 - “+” **mức chung** (public), thuộc tính hay phương thức chỉ được nhìn thấy bởi đối tượng của tất cả các lớp

108



Biểu đồ lớp

○ Ví dụ

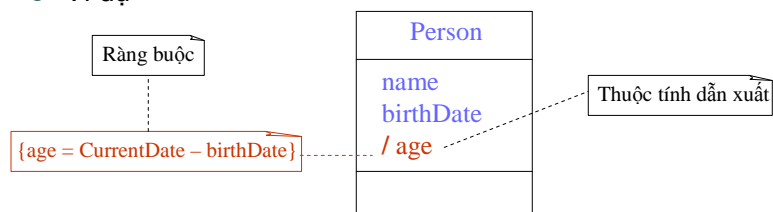


109



Biểu đồ lớp

- Thuộc tính dẫn xuất (derived attribut) là các thuộc tính của biểu đồ lớp mà có thể suy ra từ các thuộc tính khác.
- Kí hiệu: thuộc tính dẫn xuất bắt đầu bởi **"/"**, một **ràng buộc** có thể đi kèm để giải thích sự dẫn xuất
- Ví dụ



110



Biểu đồ lớp

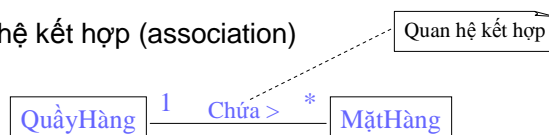
- Các quan hệ giữa các lớp
 - Quan hệ kết hợp (association)
 - Quan hệ chuyên biệt hóa/tổng quát hóa (specialization/generalization)
 - Quan hệ hợp thành (composition)
 - Quan hệ kết tập (agregation)
 - Quan hệ phụ thuộc (dependence)

111

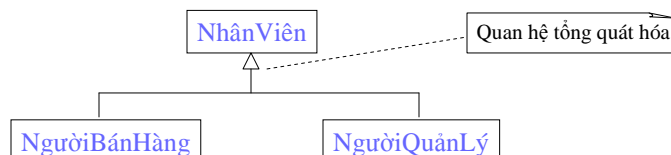


Biểu đồ lớp

- Quan hệ kết hợp (association)



- Quan hệ chuyên biệt hóa/tổng quát hóa (specialization/generalization)

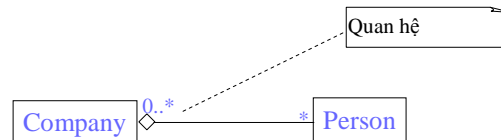


112

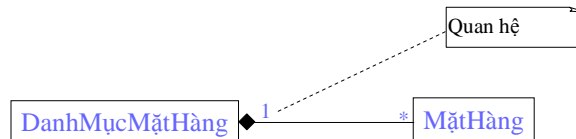


Biểu đồ lớp

- Quan hệ kết tập (agregation)



- Quan hệ hợp thành (composition)

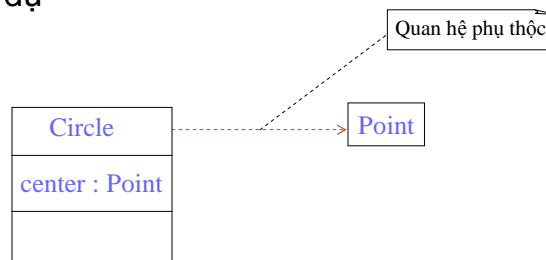


113



Biểu đồ lớp

- Quan hệ phụ thuộc (dependence): mô tả một lớp **phụ thuộc** vào lớp khác
- Ví dụ

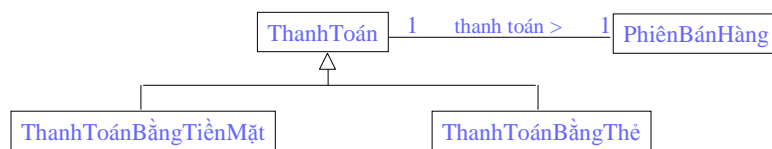


114



Biểu đồ lớp

- Ví dụ: **chuyển đổi mô hình khái niệm thành biểu đồ lớp**
- Giả sử mô hình khái niệm

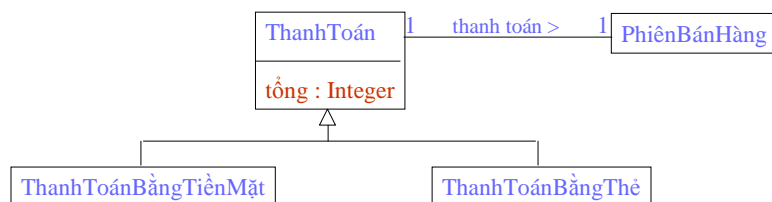


115



Biểu đồ lớp

- Chi tiết các thuộc tính

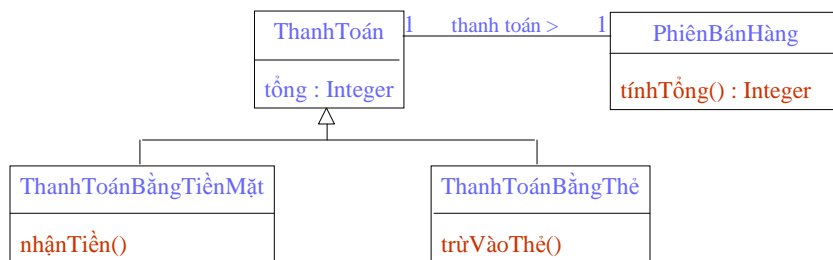


116



Biểu đồ lớp

- Chi tiết các phương thức

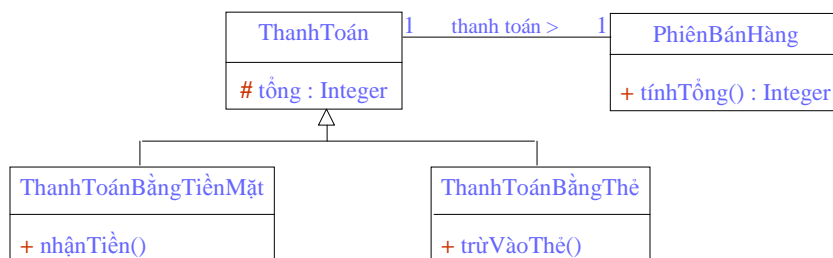


117



Biểu đồ lớp

- Xác định các mức khả kiến



118



Nội dung

- Khái niệm cơ bản hướng đối tượng
- Biểu đồ ca sử dụng
- Thiết kế cấu trúc tĩnh
- **Thiết kế cấu trúc động**
- Sinh mã

119



Cấu trúc động

- Biểu đồ tương tác
 - Biểu đồ tuần tự
 - Biểu đồ cộng tác

120



Biểu đồ tương tác

- Biểu đồ tương tác mô tả **hành vi** của hệ thống
- Mỗi biểu đồ tương tác tương ứng một **tác vụ được thực hiện bởi một số các đối tượng**
- Biểu đồ tương tác xây dựng dựa trên nền tảng của biểu đồ hoạt động và biểu đồ trạng thái
- Biểu đồ tương tác mô tả các hành động của các đối tượng để thực hiện một tác vụ. Các hành động của đối tượng bao gồm:
 - **gửi các thông điệp** (message) giữa các đối tượng
 - **tạo** (create) và **hủy** (destroy) các đối tượng

121



Biểu đồ tuần tự

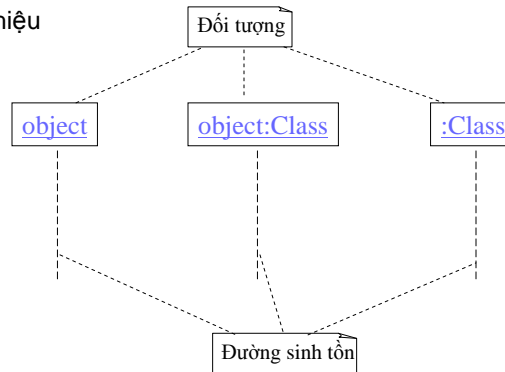
- Biểu đồ tuần tự (sequence diagram) biểu diễn sự tương tác giữa các đối tượng bằng việc nhấn mạnh thứ tự trao đổi thông điệp giữa các đối tượng
- Biểu đồ tuần tự gồm:
 - các đối tượng
 - các thông điệp trao đổi giữa các đối tượng

122



Biểu đồ tuần tự

- Mỗi **đối tượng** có một đường sinh tồn (lifeline) biểu diễn thời gian tồn tại của nó.
- Kí hiệu

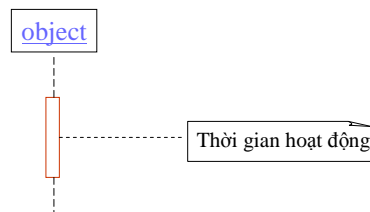


123



Biểu đồ tuần tự

- Thời gian hoạt động (activation) là thời gian mà đối tượng đang thực hiện một thao tác
- Kí hiệu



124



Biểu đồ tuần tự

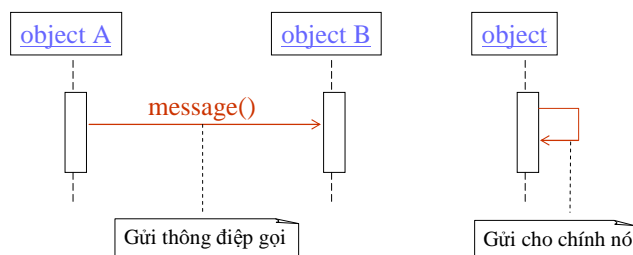
- Một thông điệp đặc tả trao đổi giữa các đối tượng
- Các loại thông điệp
 - Gọi (call)
 - Trả về (return)
 - Gửi (send)
 - Tạo (create)
 - Hủy (destroy)

125



Biểu đồ tuần tự

- Thông điệp gọi gọi một phương thức/thao tác trên đối tượng
 - Đối tượng gọi phải đợi thông điệp được thực hiện kết thúc mới có thể thực hiện công việc khác (thông điệp đồng bộ)
- Một đối tượng có thể gửi thông điệp cho chính nó
- Kí hiệu

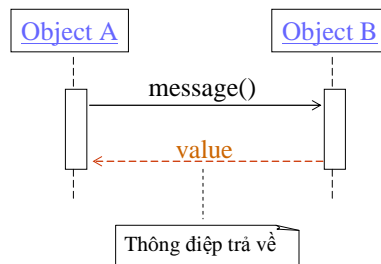


126



Biểu đồ tuần tự

- Thông điệp trả về trả về một giá trị cho đối tượng gọi
- Kí hiệu

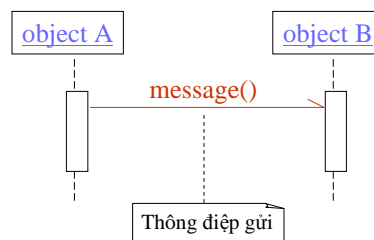


127



Biểu đồ tuần tự

- Thông điệp gửi gửi một tín hiệu đến một đối tượng
 - Khác với thông điệp gọi, khi đối tượng gửi thông điệp gửi nó không chờ đợi, mà tiếp tục thực hiện công việc khác (thông điệp không đồng bộ)
- Kí hiệu

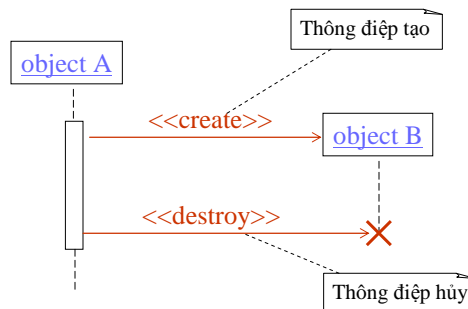


128



Biểu đồ tuần tự

- Thông điệp tạo gọi phương thức tạo một đối tượng
- Thông điệp hủy gọi phương thức hủy một đối tượng
- Kí hiệu

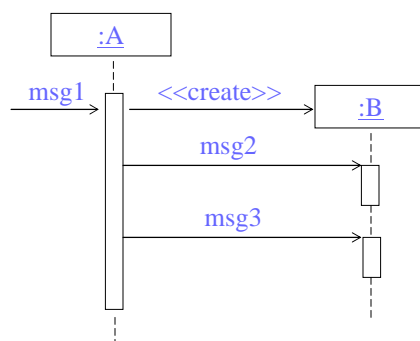


129



Biểu đồ tuần tự

- Ví dụ



```
public class A
{
    private B objB;
    public void msg1()
    {
        objB = new B();
        objB.msg2();
        objB.msg3();
    }
}

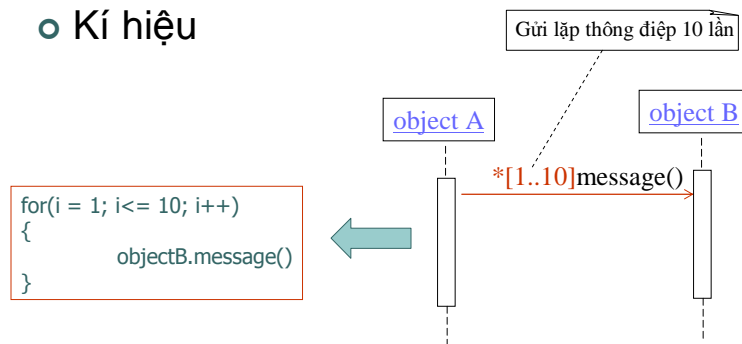
public class B
{
    ...
    public void msg2() { ... }
    public void msg3() { ... }
}
```

130



Biểu đồ tuần tự

- Một thông điệp có thể được gửi lặp nhiều lần
- Kí hiệu

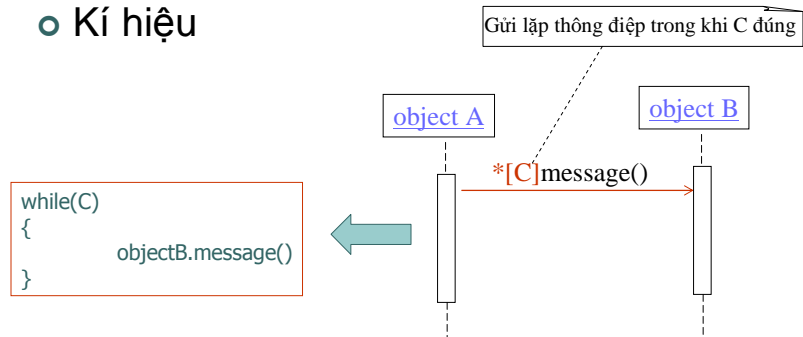


131



Biểu đồ tuần tự

- Một thông điệp có thể được gửi lặp nhiều lần phụ thuộc vào một điều kiện
- Kí hiệu

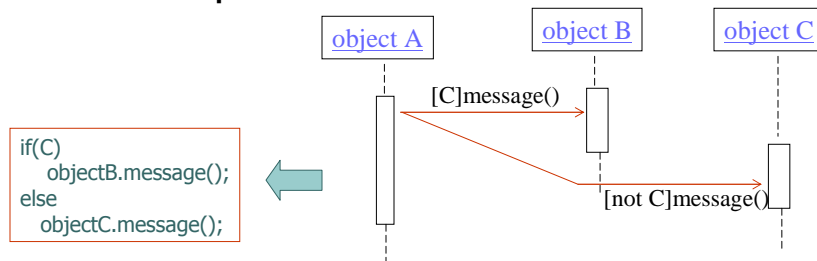


132



Biểu đồ tuần tự

- Một thông điệp có thể được gửi phụ thuộc vào điều kiện rẽ nhánh
- Kí hiệu

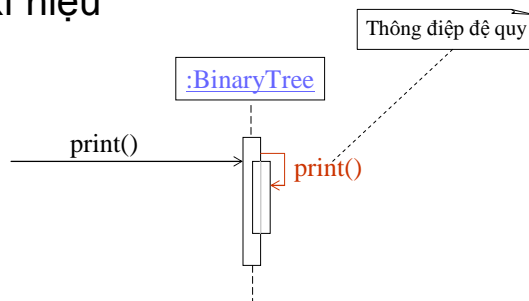


133



Biểu đồ tuần tự

- Một thông điệp có thể được gọi đệ quy
- Kí hiệu

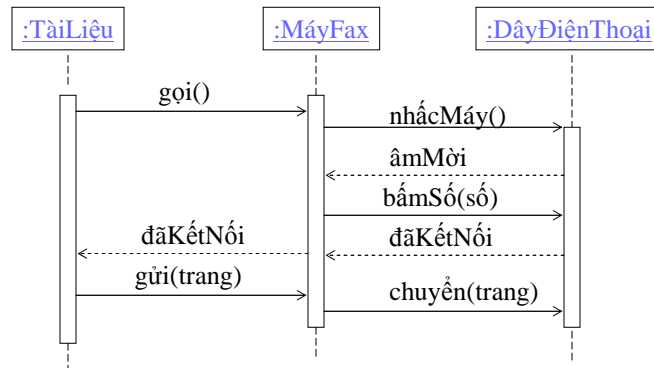


134



Biểu đồ tuần tự

◉ Ví dụ

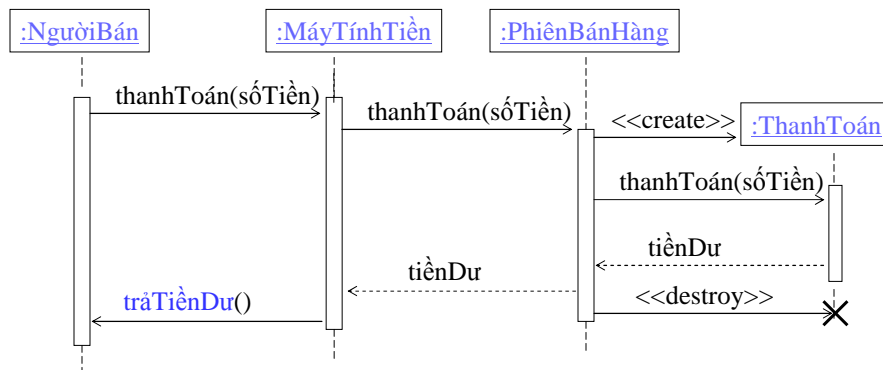


135



Biểu đồ tuần tự

◉ Ví dụ

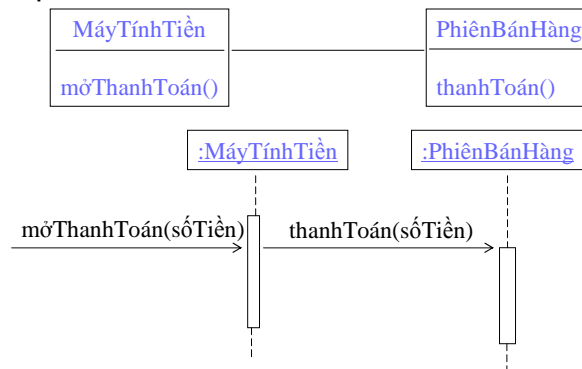


136



Biểu đồ tuần tự

- Giữa biểu đồ tương tác và biểu đồ lớp và có mối quan hệ chặt chẽ với nhau
- Ví dụ



137



Biểu đồ cộng tác

- Biểu đồ cộng tác (collaboration diagram) mô tả sự tương tác giữa các đối tượng bằng việc nhấn mạnh cấu trúc kết hợp giữa các đối tượng và những thông điệp trao đổi giữa chúng
- Biểu đồ cộng tác là sự mở rộng của biểu đồ đối tượng
- Biểu đồ cộng tác chỉ ra
 - thứ tự gửi các thông điệp: mỗi thông điệp được gán một số tuần tự
 - điều kiện gửi các thông điệp

138



Biểu đồ cộng tác

- Cấu trúc thông điệp được mô tả dạng tổng quát như sau:

`precondition / condition sequence * *|| iteration : result := message(parameters)`

- “**precondition** /” : danh sách số tuần tự của các thông điệp trước thông điệp cần gửi. Thông điệp chỉ được gửi đi khi tất cả các thông điệp trước nó đã được gửi đi.
- “**condition**” : thông điệp chỉ được gửi đi khi điều kiện được thỏa mãn.
- “**sequence**” : số tuần tự của thông điệp cần gửi. Ví dụ, việc gửi thông điệp 1.3.5 theo sau việc gửi thông điệp 1.3.4, cả hai thông điệp này nằm trong luồng 1.3.
- “*****” : chỉ ra thông điệp được gửi đi nhiều lần một cách tuần tự.
- “***||**” : chỉ ra thông điệp được gửi đi nhiều lần một cách đồng thời.
- “**iteration**” : chỉ ra số lần gửi thông điệp một cách tuần tự hoặc đồng thời
- “**result**” : chỉ ra giá trị trả về của thông điệp.
- “**message**” : tên thông điệp
- “**parameters**” : danh sách các tham số của thông điệp.

139



Biểu đồ cộng tác

- Ví dụ

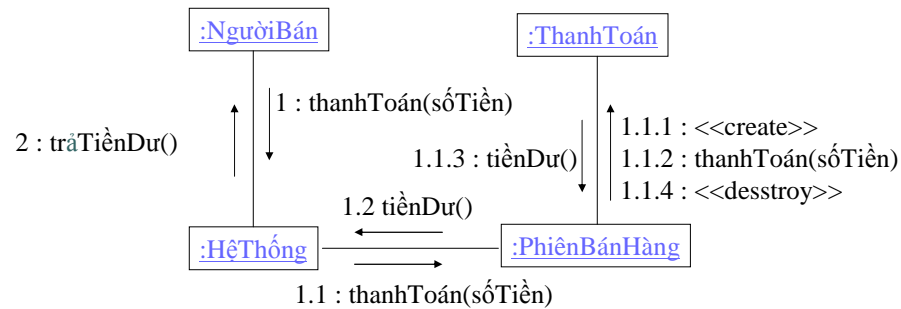
- **4 : hello()** : thông điệp có số tuần tự là 4.
- **[time = 12h] 1 : lunch()** : thông điệp này chỉ được gửi đi nếu là lúc 12h.
- **1.3.5 * call()** : thông điệp này được gửi đi nhiều lần.
- **3 / *|| [i:= 1..5] 1.2 : close()** : thông điệp này được gửi đi năm lần một cách đồng thời và sau thông điệp số 3.
- **1.2, 2.3 / [t < 10] 3.1 name = getName()** : thông điệp này được gửi đi sau các thông điệp 1.2, 2.3 và với điều kiện $t < 10$.

140



Biểu đồ cộng tác

◦ Ví dụ biểu đồ cộng tác



141



Biểu đồ tương tác

◦ Bài tập 1: Máy rút tiền ATM

- Xây dựng biểu đồ tuần tự cho ca sử dụng rút tiền trong trường hợp thành công
- Xây dựng biểu đồ tuần tự cho ca sử dụng xem số tiền dư trong tài khoản

142



Nội dung

- Khái niệm cơ bản hướng đối tượng
- Biểu đồ ca sử dụng
- Thiết kế cấu trúc tĩnh
- Thiết kế cấu trúc động
- **Sinh mã**

143



Sinh mã

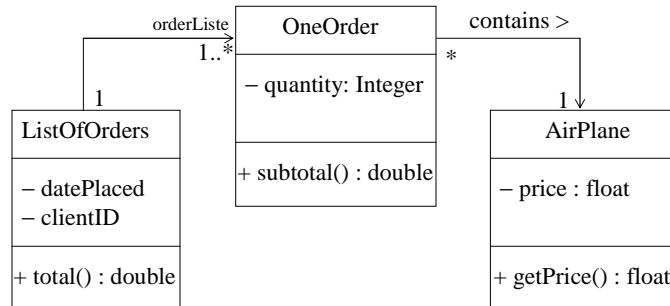
- Chuyển các mô hình thiết kế sang mã chương trình (C++, Java, ...)
- Mã chương trình hướng đối tượng
 - Định nghĩa các lớp và giao diện
 - Định nghĩa các phương thức
- Các **biểu đồ lớp** sẽ được chuyển sang **mã chương trình định nghĩa các lớp** tương ứng
- Các **biểu đồ tương tác** sẽ được chuyển thành **mã chương trình định nghĩa các phương thức**
- Các biểu đồ khác sẽ hỗ trợ cho quá trình mã hóa

144



Sinh mã

- o Ví dụ: biểu đồ lớp

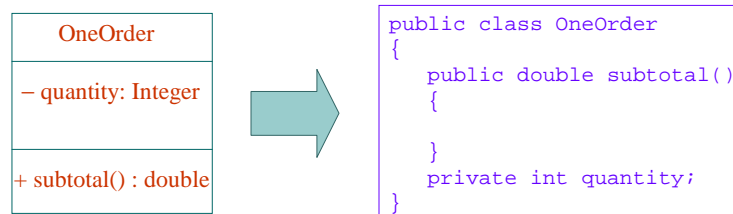


145



Sinh mã

- o Mã lớp **OneOrder**

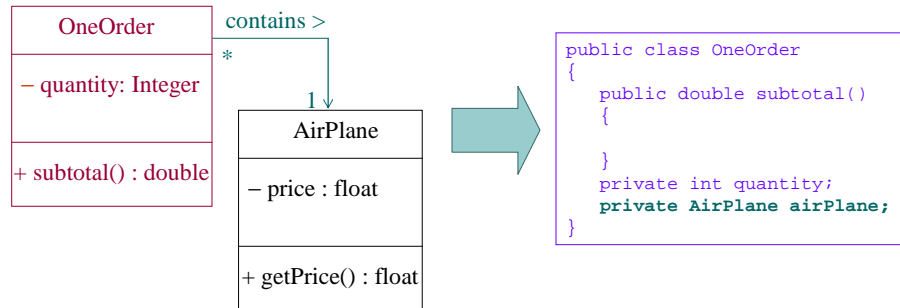


146



Sinh mã

o Mã lớp **OneOrder**

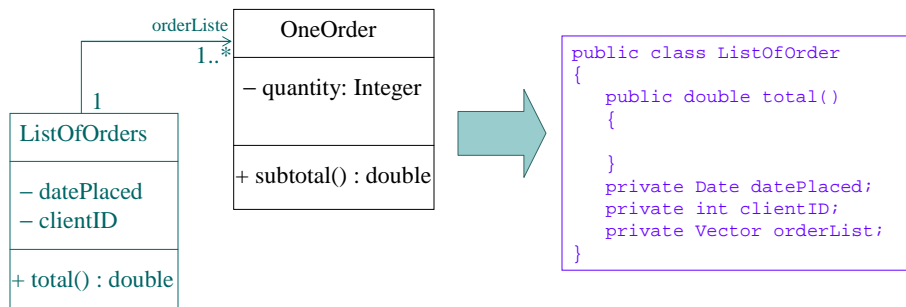


147



Sinh mã

o Mã lớp **ListOfOrders**

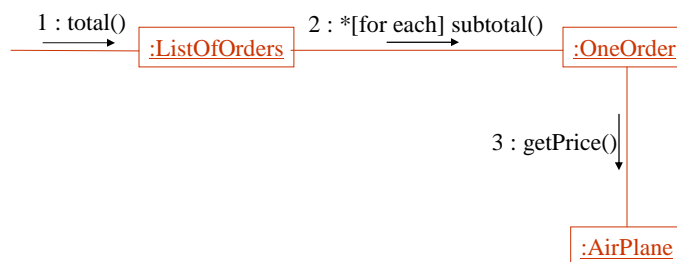


148



Sinh mã

- Biểu đồ cộng tác thực hiện phương thức **total()**

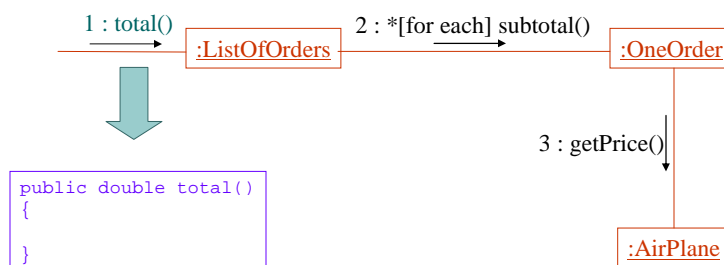


149



Sinh mã

- Mã phương thức **total()**

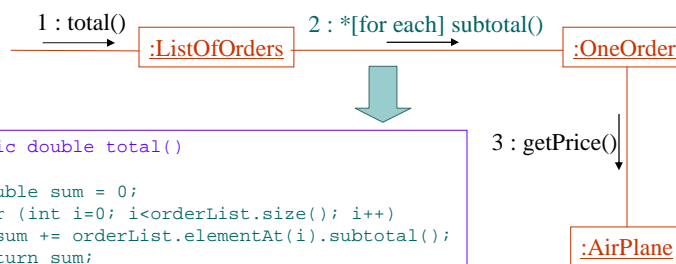


150



Sinh mã

o Mã phương thức **total()**

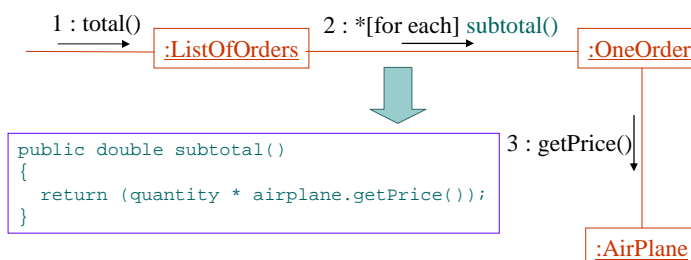


151



Sinh mã

o Mã phương thức **subTotal()**

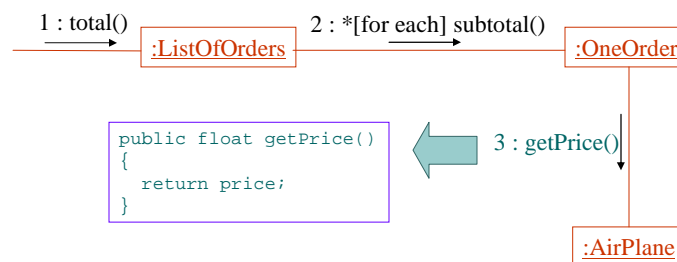


152



Sinh mã

- Mã phương thức **getPrice()**



153



Công cụ

- Phần mềm Rational Rose, Poisedon for UML, Umbrello
 - Thiết kế các biểu đồ UML
 - Sinh mã chương trình
 - C++
 - Java
 - VB
 - Ada

154