



Phân tích và thiết kế hệ thống hướng đối tượng



Nội dung (1)

- Các khái niệm hướng đối tượng
- Tổng quan về ngôn ngữ mô hình hóa UML
- UML trong tiến trình phát triển phần mềm
- Mô hình hoá yêu cầu
 - Biểu đồ ca sử dụng
- Mô hình hoá khái niệm
- Biểu đồ hoạt động
- Biểu đồ trạng thái



Nội dung (2)

- Biểu đồ lớp
- Biểu đồ tương tác
 - Biểu đồ tuần tự
 - Biểu đồ cộng tác
- Mô hình hoá kiến trúc
 - Biểu đồ gói
 - Biểu đồ cài đặt
- Mã hoá
- Ứng dụng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

3



Tài liệu tham khảo

- G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.
- Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition, Addison-Wesley, 2004.
- Pierre-Alain Muller, Modélisation objet avec UML, Eyrolles, 1997.
- Bernard Boigelot, Cours de l'Ingénierie du logiciel orienté-objet.
- Olivier Gerbe, Cours UML, IFI, 2000.
- UML applied – Object Oriented Analysis and Design using the UML, Ariadne training, 2000.
- Đặng Văn Đức, Giáo trình nhập môn UML, NXB Lao động xã hội, 2003.
- Nguyễn Văn Ba, Phát triển hướng đối tượng với UML 2.0 và C++, NXB Đại học Quốc gia Hà nội, 2005.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

4



Các khái niệm cơ bản về hướng đối tượng (1)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

5



Nội dung trình bày

- Hướng chức năng
- Hướng đối tượng (HĐT)
- Đối tượng
- Lớp
- Các tính chất cơ bản của HĐT

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

6



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

7-Aug-07

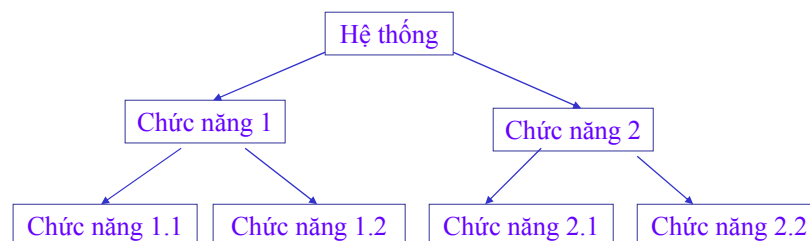
Phân tích và thiết kế hệ thống hướng đối tượng

7



Hướng chức năng

- Phân cấp chức năng



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

8



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

9



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

10



Hướng đối tượng

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

11



Hướng đối tượng

- Ưu điểm chính
 - PT & TK 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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

12



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



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



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, ...



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, ...



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

17



Đối tượng

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

18



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

19



Lớp

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



Person
Name Age
changeAge

- Trừu tượng hóa làm giảm độ phức tạp

7-Aug-07

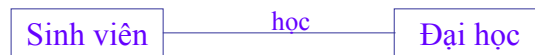
Phân tích và thiết kế hệ thống hướng đối tượng

20



Lớp

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



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



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

23



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

7-Aug-07

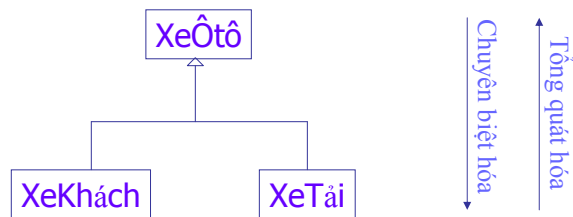
Phân tích và thiết kế hệ thống hướng đối tượng

24



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.

7-Aug-07

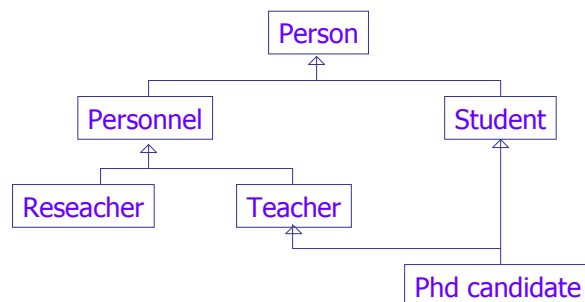
Phân tích và thiết kế hệ thống hướng đối tượng

25



Các tính chất của HĐT

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

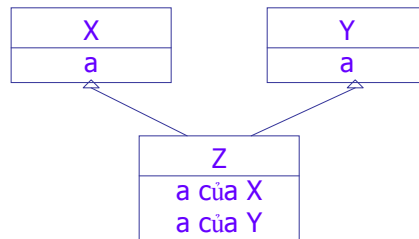
26



Các tính chất của HĐT

■ Đa thừa kế

■ Độ độ tên các thuộc tính



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

27



Các tính chất của HĐT

■ Ưu điểm của thừa kế

- Phân loại các lớp: các lớp được phân loại, sắp xếp theo một thứ bậc để dễ quản lý
- 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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

28



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

7-Aug-07

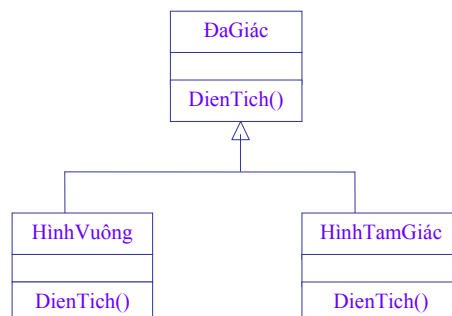
Phân tích và thiết kế hệ thống hướng đối tượng

29



Các tính chất của HĐT

- Ví dụ tính đa hình



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

30



Giới thiệu UML (2)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

31



Nội dung trình bày

- Khái niệm mô hình hóa
- Một số phương pháp mô hình hóa HDT
- Lịch sử phát triển UML
- Giới thiệu tổng quan UML
 - UML – một ngôn ngữ
 - Các khái niệm cơ bản của UML

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

32



Khái niệm mô hình hóa

- **Mô hình** (model) là một sự đơn giản hóa của hệ thống thực
 - Mô hình là sự biểu diễn bằng hình ảnh của hệ thống thực
- Mô hình diễn tả hệ thống
 - ở một mức trừu tượng,
 - theo một quan điểm,
 - bằng một hình thức biểu diễn (văn bản, hình ảnh, ...)
- **Mô hình hóa** (modeling) là quá trình dùng mô hình để diễn tả hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

33



Khái niệm mô hình hóa

- Tại sao mô hình hóa?
 - Mô hình hóa để dễ hiểu
 - Dễ nhận thức vấn đề
 - Mô hình hóa để trao đổi
 - Phương tiện trao đổi giữa những người phát triển
 - Mô hình hóa để hoàn chỉnh
 - Dễ dàng nhận sự phù hợp giữa mô hình và nhu cầu để cải tiến, hoàn thiện

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

34



Khái niệm mô hình hóa

- Các nguyên tắc mô hình hóa
 - Chọn mô hình thích hợp
 - Góc nhìn cơ sở dữ liệu: mô hình thực thể - kết hợp
 - Góc nhìn có cấu trúc: thuật toán
 - Góc nhìn hướng đối tượng: các lớp và các quan hệ
 - Các mô hình được thể hiện nhiều mức chính xác khác nhau
 - Hệ thống được diễn tả ở các mức chi tiết khác nhau, tùy theo nhu cầu người sử dụng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

35



Khái niệm mô hình hóa

- Các nguyên tắc mô hình hóa
 - Các mô hình phải liên hệ với thế giới thực
 - Các mô hình xây dựng gần gũi với mô hình hệ thống thực
 - Mô hình hướng đối tượng > Mô hình hóa có cấu trúc
 - Một hệ thống phải được mô hình hóa mỗi một tập hợp các mô hình
 - Chỉ một mô hình là không thể đủ
 - Phải thể hiện được nhiều góc nhìn khác nhau về hệ thống: sử dụng, thiết kế, cài đặt, ...

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

36



Một số phương pháp mô hình hóa HĐT

- OOD (Object Oriented Design)
- OOSE (Object Oriented Software Engineering)
- OMT (Object Modeling Technique)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

37



OOD (Object Oriented Design)

- Phương pháp được phát triển bởi Booch
- Mô hình tĩnh
 - Biểu đồ lớp
 - Biểu đồ đối tượng
- Mô hình động
 - Biểu đồ trạng thái
 - Biểu đồ thời gian

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

38



OOSE (Object Oriented Software Engineering)

- Phương pháp được phát triển bởi Jacobson
- Năm mô hình
 - Mô hình yêu cầu (kịch bản sử dụng)
 - Mô hình phân tích (mức khái niệm)
 - Mô hình thiết kế (mức logic)
 - Mô hình mã hóa (mức vật lý)
 - Mô hình kiểm thử

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

39



OMT (Object Modeling Technique)

- Phương pháp được phát triển bởi Rumbaugh
- Ba mô hình
 - Mô hình tĩnh
 - mô hình thực thể quan hệ
 - Mô hình động
 - biểu đồ trạng thái và chuyển tiếp
 - Mô hình chức năng
 - dựa trên biểu đồ luồng dữ liệu

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

40



Lịch sử phát triển UML

- 1975-1990
 - nhiều ngôn ngữ mô hình hóa HĐT được phát triển
- 1990-1994
 - hơn 50 phương pháp phát triển HĐT trong đó có các phương pháp OOD, OOSE và OMT
- 10/1994
 - Rumbaugh và Booch tiến hành dự án UML ở Rational

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

41



Lịch sử phát triển UML

- 10/1995
 - Phiên bản đầu tiên của phương pháp hợp nhất ra đời dưới cái tên « Unified Method » v0.8
- Cuối 1995
 - Jacobson tham gia nhóm của Booch và Rumbaugh
- 6/1996
 - Phiên bản UML v0.9 tích hợp OOSE ra đời

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

42



Lịch sử phát triển UML

- 1996
 - Hội thảo đầu tiên về UML được tổ chức
- 1997
 - Phiên bản UML v1.0 được đề xuất như là chuẩn với tổ chức OMG (Object Management Group)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

43



Lịch sử phát triển UML

- 7/1997
 - phiên bản UML v1.1 được đề nghị chuẩn hóa ở OMG
- 11/1997
 - UML v1.1 được chấp nhận là phương pháp chuẩn bởi OMG
- 6/1998
 - UML v1.2 được phát hành bởi OMG
- 8/1998
 - UML v1.3 được phát hành bởi OMG

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

44



UML – một ngôn ngữ

- Một *ngôn ngữ* cung cấp từ vựng, ngữ pháp (cú pháp) và ngữ nghĩa để mô tả sự giao tiếp
- Một *ngôn ngữ mô hình hóa* là ngôn ngữ mà trong đó từ vựng và luật được sử dụng để mô tả mức khái niệm và mức vật lý của một hệ thống
- UML cung cấp từ vựng và luật để mô tả các mô hình khác nhau của một hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

45



UML – ngôn ngữ để mô hình hóa trực quan

- UML cung cấp tập hợp các kí hiệu và luật để biểu diễn đồ họa những mô hình của hệ thống
- Biểu diễn đồ họa thường đem lại cái nhìn tốt hơn về hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

46



UML – ngôn ngữ để đặc tả

- Một đặc tả là một mô tả chính xác và không nhập nhằng về hệ thống
- UML cung cấp các công cụ để đặc tả hệ thống ở nhiều mức: phân tích, thiết kế và cài đặt

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

47



UML – ngôn ngữ để xây dựng

- Các mô hình UML có thể được chuyển sang một ngôn ngữ lập trình khá dễ dàng
- UML là khá chính xác và không nhập nhằng để cho phép thực hiện các mô phỏng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

48



UML – ngôn ngữ để lập tài liệu

- UML cho phép mô tả tất cả các bước xây dựng một hệ thống
- Các mô hình được xây dựng tạo nên một tài liệu đầy đủ và chính xác về hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

49



Các khái niệm cơ bản của UML

- Các phân tử
- Các quan hệ
- Các biểu đồ

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

50



Các phần tử

- Các phần tử cấu trúc
- Các phần tử động
- Các phần tử nhóm
- Các phần tử chú thích

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

51



Các phần tử cấu trúc

- Lớp (class)
- Giao diện (interface)
- Cộng tác (collaboration)
- Tình huống (case)
- Thành phần (component)
- Nút (node)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

52



Các phần tử cấu trúc

■ Lớp (class)

- Lớp là đặc tả một tập hợp các đối tượng có cùng thuộc tính, hành vi và các quan hệ
- Một lớp được biểu diễn bởi hình chữ nhật
- Ví dụ

Điểm
x, y : int
vẽĐiểm()
diChuyển(x, y)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

53



Các phần tử cấu trúc

■ Giao diện (interface)

- Một giao diện là một tập hợp đặc tả các hành vi/phương thức của một lớp hoặc một thành phần
- Một giao diện được biểu diễn đồ họa bởi một hình tròn
- Ví dụ



ImageObserver

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

54



Các phần tử cấu trúc

■ Cộng tác (collaboration)

- Một cộng tác định nghĩa một sự tương tác giữa các phần tử
- Một cộng tác được biểu diễn bởi một hình ê-líp có nét đứt
- Ví dụ

Kiểm tra một
đơn đặt hàng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

55



Các phần tử cấu trúc

■ Ca sử dụng (use case)

- Một ca sử dụng sử dụng là mô tả một tập hợp các hoạt động của hệ thống nhằm trả lời yêu cầu của một tác nhân (actor)
- Một ca sử dụng được thực hiện bởi một cộng tác
- Một ca sử dụng được biểu diễn bởi hình ê-líp
- Ví dụ

Thanh toán
hóa đơn

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

56



Các phần tử cấu trúc

- Thành phần (component)
 - Một thành phần là một phần vật lý và có thể thay thế được của một hệ thống
 - Một thành phần được biểu diễn như sau



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

57



Các phần tử cấu trúc

- Nút (node)
 - Một nút là một phần tử vật lý của hệ thống, nó tồn tại khi hệ thống thực thi
 - Nút biểu diễn một nguồn tài nguyên với bộ nhớ và khả năng xử lý
 - Nút được biểu diễn bởi hình hộp chữ nhật
 - Ví dụ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

58



Các phần tử động

- Tương tác
- Máy trạng thái

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

59



Các phần tử động

- Tương tác (interaction)
 - Một tương tác là một tập hợp các thông điệp (message) trao đổi giữa các đối tượng
 - Một thông điệp được biểu diễn bởi mũi tên
 - Ví dụ

print() →

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

60



Các phần tử động

- Máy trạng thái (state machine)
 - Máy trạng thái đặc tả chuỗi các trạng thái thay đổi của một đối tượng
 - Một trạng thái được biểu diễn như sau

Đợi



Các phần tử nhóm

- Gói (package)
 - Gói là kỹ thuật để nhóm các phần tử
 - Gói được biểu diễn như sau

Client



Các phần tử chú thích

■ Chú thích (note)

- Chú thích là một giải thích liên quan đến một phần tử UML mà cho phép mang lại thông tin bổ sung
- Một chú thích được biểu diễn bởi một hình chữ nhật có một góc bị gấp
- Ví dụ

Xem tệp tin
note.doc

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

63



Các quan hệ

- Quan hệ phụ thuộc (dependency)
- Quan hệ kết hợp (association)
- Quan hệ tổng quát hóa (generalization)
- Quan hệ cài đặt (realization)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

64



Quan hệ phụ thuộc

- Quan hệ phụ thuộc là quan hệ ngữ nghĩa giữa hai phần tử mà sự thay đổi của phần này sẽ ảnh hưởng đến phần tử kia
- Quan hệ phụ thuộc được biểu diễn một mũi tên đứt nét như sau:

.....→



Quan hệ kết hợp

- Quan hệ kết hợp là quan hệ cấu trúc mà định nghĩa một tập hợp các liên kết giữa các đối tượng
- Quan hệ được biểu diễn bởi một đoạn thẳng với các bội số/bản số (multiplicity) và vai trò

0..1 sở hữu 0..n



Quan hệ tổng quát hóa

- Quan hệ tổng quát hóa là quan hệ giữa một phần tử chung (lớp cha) và phần tử cụ thể (lớp con)
- Quan hệ tổng quát hóa thể hiện tính thừa kế
- Quan hệ tổng quát hóa được biểu diễn như sau



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

67



Quan hệ cài đặt

- Quan hệ cài đặt là một quan hệ ngữ nghĩa mà trong đó một phần tử sẽ cài đặt (thực hiện) phần tử khác
- Quan hệ cài đặt được biểu diễn bởi mũi tên với nét đứt như sau



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

68

Các biểu đồ (diagram)

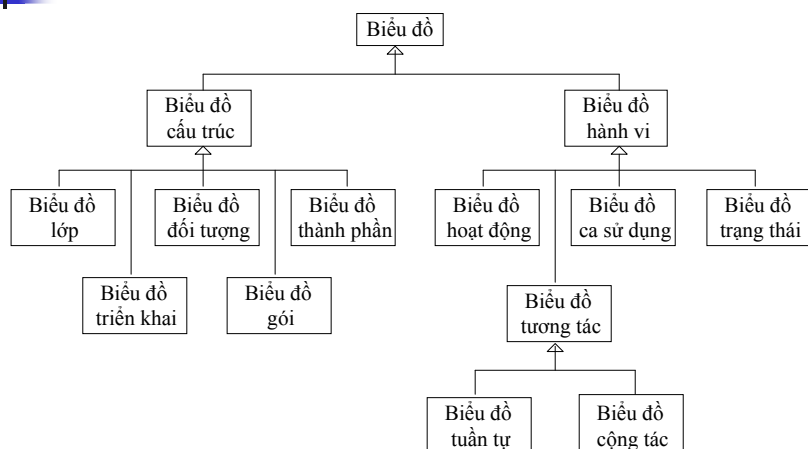
- Biểu đồ lớp (class diagram)
- Biểu đồ đối tượng (object diagram)
- Biểu đồ ca sử dụng (use case diagram)
- Biểu đồ trạng thái (state diagram)
- Biểu đồ hoạt động (activity diagram)
- Biểu đồ tương tác (interaction diagram)
 - Biểu đồ tuần tự (sequence diagram)
 - Biểu đồ cộng tác (collaboration diagram)
- Biểu đồ gói (package diagram)
- Biểu đồ thành phần (component diagram)
- Biểu đồ triển khai (deployment diagram)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

69

Các biểu đồ (diagram)



7-Aug-07

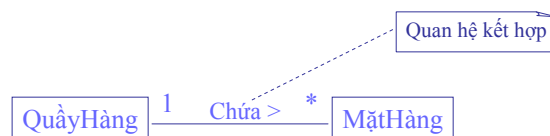
Phân tích và thiết kế hệ thống hướng đối tượng

70



Biểu đồ lớp

- Biểu đồ lớp (class diagram) mô tả tập hợp các lớp và các tương tác giữa chúng
- Các biểu đồ lớp mô tả cấu trúc tĩnh của hệ thống



7-Aug-07

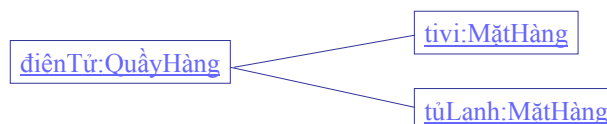
Phân tích và thiết kế hệ thống hướng đối tượng

71



Biểu đồ đối tượng

- Biểu đồ đối tượng (object diagram) mô tả tập hợp các đối tượng và các quan hệ giữa chúng
- Các biểu đồ đối tượng mô tả cùng các thông tin như biểu đồ lớp nhưng dưới cách nhìn các thể hiện của lớp



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

72

Biểu đồ ca sử dụng

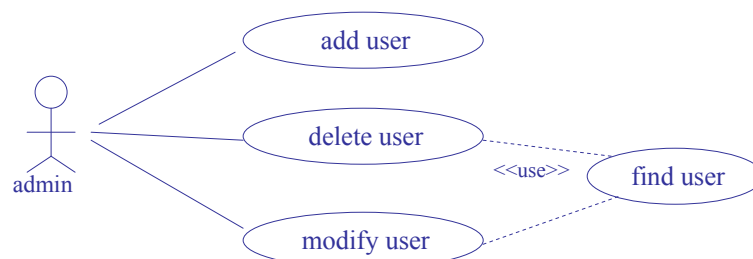
- Biểu đồ ca sử dụng (use case diagram) mô tả tập hợp các ca sử dụng, các tác nhân và những quan hệ giữa chúng
- Các biểu đồ ca sử dụng mô tả cái nhìn tĩnh về hệ thống dưới con mắt của người sử dụng
- Các biểu đồ ca sử dụng rất quan trọng để nắm bắt các chức năng của hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

73

Biểu đồ ca sử dụng



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

74



Biểu đồ trạng thái

- Biểu đồ trạng thái (state diagram) biểu diễn máy trạng thái (trạng thái, chuyển tiếp, sự kiện và hoạt động)
- Những biểu đồ trạng thái mô tả cái nhìn động về hệ thống
- Các biểu đồ trạng thái là rất quan trọng để hiểu rõ hành vi (behavior) của hệ thống



7-Aug-07

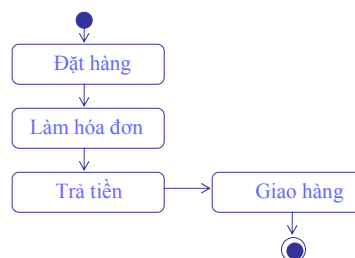
Phân tích và thiết kế hệ thống hướng đối tượng

75



Biểu đồ hoạt động

- Biểu đồ hoạt động (activity diagram) mô tả luồng thông tin từ một hoạt động này đến một hoạt động khác
- Những biểu đồ hoạt động mô tả cái nhìn động về hệ thống
- Các biểu đồ hoạt động là rất quan trọng để hiểu rõ hoạt động của hệ thống



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

76

Biểu đồ tương tác

- Biểu đồ tương tác (interaction diagram) gồm biểu đồ tuần tự và biểu đồ cộng tác
 - Biểu đồ tuần tự (sequence diagram) mô tả sự trao đổi các thông điệp giữa các đối tượng dựa theo thời gian

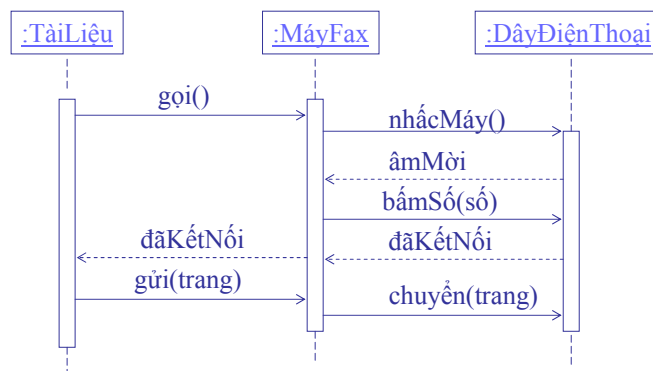
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

77

Biểu đồ tương tác

- Ví dụ



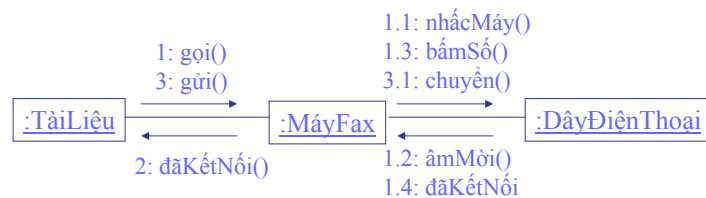
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

78

Biểu đồ tương tác

- Biểu đồ cộng tác (collaboration diagram) mô tả sự trao đổi các thông điệp giữa các đối tượng dựa theo cấu trúc quan hệ giữa các đối tượng



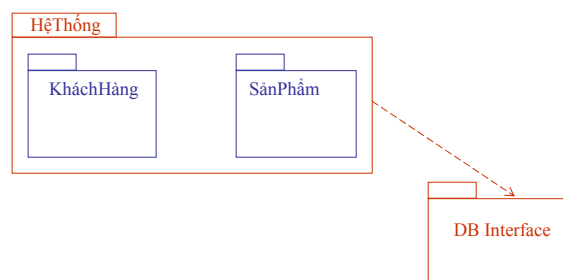
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

79

Biểu đồ gói

- Biểu đồ gói (package diagram) cho phép nhóm các lớp thành các gói quan hệ với nhau
- Tạo nên một **cấu trúc phụ thuộc giữa các gói** thay vì cấu trúc phụ thuộc giữa các lớp



7-Aug-07

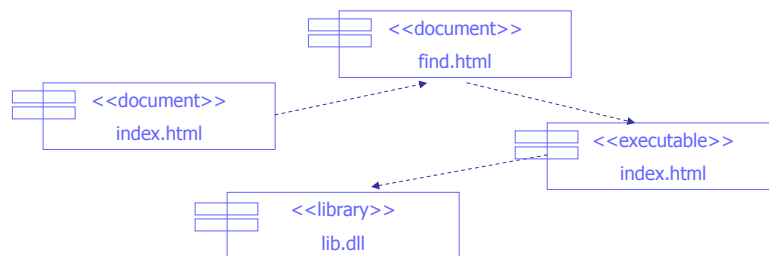
Phân tích và thiết kế hệ thống hướng đối tượng

80



Biểu đồ thành phần

- Biểu đồ thành phần (component diagram) chỉ ra tổ chức và sự phụ thuộc giữa các thành phần của hệ thống
- Các biểu đồ thành phần mô tả cái nhìn tĩnh về sự cài đặt của hệ thống



7-Aug-07

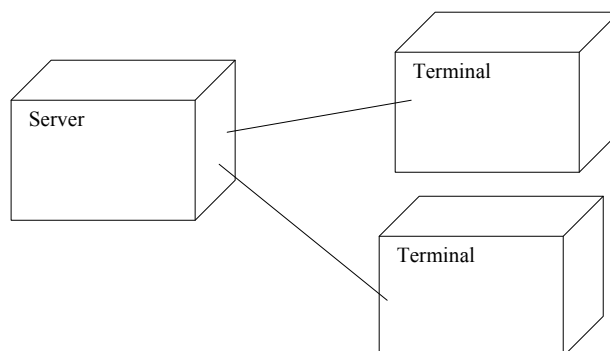
Phân tích và thiết kế hệ thống hướng đối tượng

81



Biểu đồ triển khai

- Biểu đồ triển khai (deployment diagram) mô tả các nút và các thành phần của hệ thống tại thời điểm thực thi



7-Aug-07

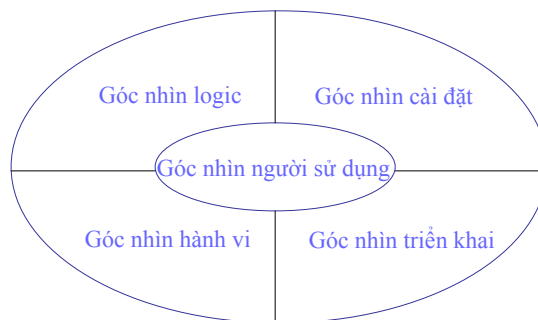
Phân tích và thiết kế hệ thống hướng đối tượng

82



Góc nhìn hệ thống

- Hệ thống được thể hiện dưới năm góc nhìn khác nhau



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

83



Góc nhìn hệ thống

- Góc nhìn người sử dụng (use case view) mô tả hệ thống dưới góc nhìn của người sở hữu và người sử dụng cuối cùng
 - Mô tả các mục tiêu và mục đích của hệ thống
 - Mô tả các chức năng của hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

84



Góc nhìn hệ thống

- Góc nhìn logic (logical view) mô tả mặt tĩnh (static) và cấu trúc của hệ thống
- Góc nhìn hành vi (behavior view) mô tả mặt động của hệ thống
 - Mô tả các tương tác và cộng tác giữa các phần tử của hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

85



Góc nhìn hệ thống

- Góc nhìn cài đặt (component view) mô tả thực hiện giải pháp
 - Mô tả tổ chức mã chương trình của giải pháp
- Góc nhìn triển khai (deployment view) mô tả sự thực thi của giải pháp
 - Mô tả các phần tử vật lý của giải pháp (bộ xử lý, thiết bị, ...)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

86



Góc nhìn và biểu đồ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

87



Ví dụ

- Xây dựng phần mềm tính tiền tại một cửa hàng
- Ca sử dụng (use case)
 - Khách hàng mua hàng và trả bằng tiền mặt
 - Khách hàng mua hàng và trả bằng thẻ
 - ...

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

88



Ví dụ

■ Lớp (class)

■ *BánHàng*

- *Dữ liệu: ngày, giờ, mặtHàng, đangThựcHiện...*
- *Thao tác: tínhTổng(), trả(sốTiền), ...*

■ *CửaHàng*

- *Dữ liệu: tên, địaChỉ, ...*
- *...*



Ví dụ

■ Hoạt động (activity)

- *Các mặt hàng mua bởi khách hàng được ghi/nhập lần lượt từng mặt hàng, sau đó máy hiển thị tổng số tiền, ...*
- *...*

■ Trạng thái (state)

- *Một BánHàng đang được thực hiện hay đã kết thúc,*
- *...*



Ví dụ

- Tương tác (interaction)
 - *Tổng số tiền thu được tính bằng cách cộng tất cả cả tổng số tiền mà mỗi khách hàng trả, ...*
 - ...



UML trong tiến trình phát triển phần mềm (3)



Nội dung trình bày

- Khái niệm công nghệ phần mềm
- Các tiến trình phát triển phần mềm
- Tiến trình hợp nhất

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

93



Công nghệ phần mềm

- Công nghệ phần mềm: lý thuyết, phương pháp và công cụ phát triển các hệ thống phần mềm
- Động cơ
 - Phần mềm lớn và phức tạp
 - Đạt tiêu chuẩn chất lượng: hiệu quả, bảo trì, kiểm thử, tái sử dụng, ...
 - Chi phí thấp, hạn chế các nguy cơ

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

94



Công nghệ phần mềm

- Các hoạt động phát triển phần mềm
 - Xác định yêu cầu (đặc tả yêu cầu)
 - Phân tích (phần mềm làm cái gì)
 - Thiết kế (phần mềm được làm như thế nào)
 - Cài đặt (mã hóa)
 - Kiểm thử (phần mềm đúng yêu cầu)
- Sự tổ chức các hoạt động này tạo nên một *tiến trình phát triển (development process)*

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

95



UML là ngôn ngữ kí hiệu

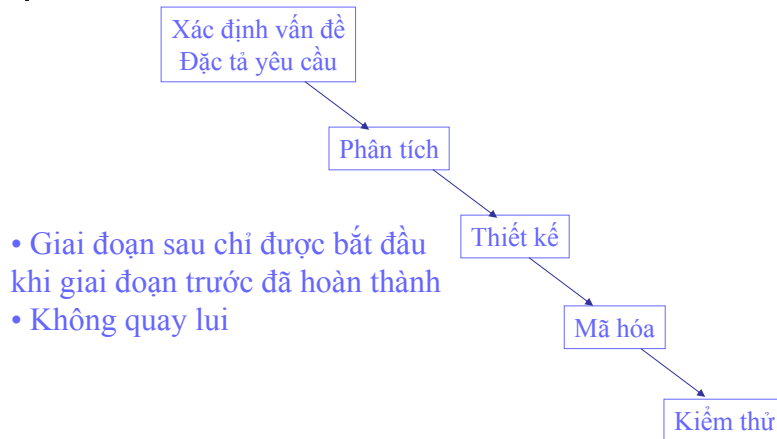
- UML chỉ là ngôn ngữ mô hình hóa
- UML không cho chúng ta biết *làm thế nào* để phát triển phần mềm
- Cần sử dụng UML trong một tiến trình phát triển

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

96

Tiến trình thác đổ (waterfall)



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

97

Tiến trình thác đổ

■ Nhược điểm

- Cứng nhắc: giai đoạn sau chỉ bắt đầu khi giai đoạn trước đã kết thúc, không quay lui
 - Khó dò tìm các vấn đề, lỗi
 - Tìm thấy vấn đề quá trễ
 - Dễ tạo nên sản phẩm có cấu trúc tồi
- Đối với hệ thống phức tạp
 - Cần hiểu toàn bộ hệ thống → tăng độ phức tạp
 - Một giai đoạn có thể rất dài (vài năm)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

98



Tiến trình thác đổ

- Cải tiến: tiến trình V
 - Cho phép quay lui
 - Bề gãy đường phát triển
 - Nhấn mạnh vấn đề kiểm thử

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

99

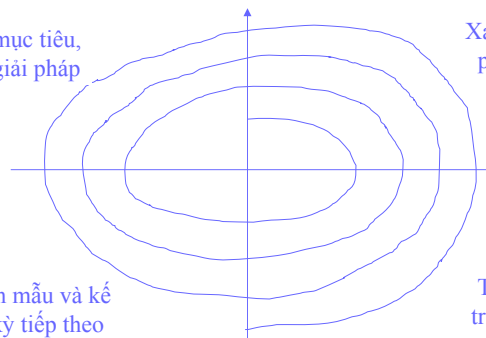


Tiến trình xoắn ốc (spiral model)

- Phát triển theo chu kỳ, mỗi chu kỳ cho một sản phẩm (nguyên mẫu) thực thi được, mỗi chu kỳ gồm 4 giai đoạn

Xác định các mục tiêu, ràng buộc và giải pháp

Xác định và tìm giải pháp cho nguy cơ



Kiểm thử nguyên mẫu và kế hoạch cho chu kỳ tiếp theo

Thiết kế và phát triển nguyên mẫu

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

100



Tiến trình xoắn ốc (spiral model)

- Giảm các nguy cơ tiềm ẩn
- Nhận được phản hồi từ người sử dụng sớm
- Giảm độ phức tạp
- Sự thay đổi công nghệ có thể được thích ứng dễ dàng
- Tuy nhiên, tiến trình khó để quản lý: tiến trình thác nước phù hợp với kỹ thuật quản lý mạng Gantt, còn tiến trình xoắn ốc thì yêu cầu kỹ thuật khác

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

101



Tiến trình hợp nhất

- Sự mở rộng của tiến trình xoắn ốc
- Hình thức hơn
- Chặt chẽ hơn

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

102



Tiến trình hợp nhất

- Tiến trình hợp nhất có thể được nhìn dưới hai góc nhìn khác nhau
 - **Góc nhìn quản lý**: quan tâm đến lĩnh vực kinh tế, chiến thuật, con người
 - Tiến trình gồm **bốn giai đoạn**
 - **Góc nhìn kỹ thuật**: quan tâm đến công nghệ, kiểm tra chất lượng, phương pháp
 - Tiến trình gồm **nhiều bước lặp**

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

103



Tiến trình hợp nhất: góc nhìn quản lý

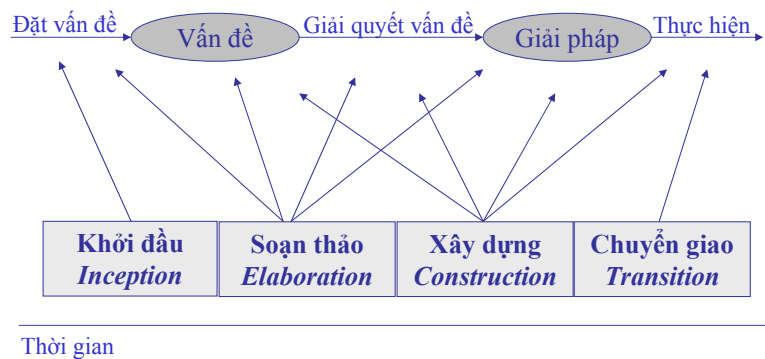
- Tiến trình phát triển phần mềm được chia làm 4 giai đoạn
 - Khởi đầu (inception): đánh giá tính khả thi
 - Soạn thảo (elaboration): phân tích, thiết kế kiến trúc
 - Xây dựng (construction): thực hiện, cụ thể hóa sản phẩm
 - Chuyển giao (transition): chuyển giao cho người sử dụng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

104

Tiến trình hợp nhất: góc nhìn quản lý



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

105

Tiến trình hợp nhất: góc nhìn quản lý

■ Khởi đầu (inception)

- Định nghĩa mục đích, yêu cầu
- Tìm hiểu vấn đề và phác thảo phương án thực hiện
- Đánh giá lợi ích
- Xác định các nguy cơ
- Định nghĩa các tiêu chuẩn đánh giá
- Đánh giá thời gian thực hiện và chi phí
- Xây dựng phương án thực hiện
- Đánh giá tính khả thi về kỹ thuật

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

106



Tiến trình hợp nhất: góc nhìn quản lý

- Khởi đầu

- Được thực hiện bởi các chuyên gia về thị trường và nhà tin học
- Quyết định thực hiện hay không

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

107



Tiến trình hợp nhất: góc nhìn quản lý

- Soạn thảo (elaboration)

- Phân tích yêu cầu
- Xác định các chức năng của hệ thống
- Lựa chọn kiến trúc hệ thống
- Xây dựng kế hoạch thực hiện

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

108



Tiến trình hợp nhất: góc nhìn quản lý

- Soạn thảo
 - Thực hiện bởi các nhà tin học
 - Chủ yếu dựa trên các ca sử dụng (use case)
 - Giúp người sử dụng hiểu rõ cái họ cần
 - Chi tiết hóa dần các ca sử dụng
 - Xây dựng mô hình các lớp (class)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

109



Tiến trình hợp nhất: góc nhìn quản lý

- Soạn thảo
 - Kết quả
 - Mô tả chức năng của hệ thống (biểu đồ ca sử dụng, kịch bản sử dụng và biểu đồ lớp)
 - Kiến trúc thực thi được của hệ thống
 - Kế hoạch hoàn chỉnh để phát triển hệ thống
 - Kế hoạch chi tiết các bước lập

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

110



Tiến trình hợp nhất: góc nhìn quản lý

- Xây dựng (construction)
 - Phát triển phần mềm cho người sử dụng
 - Thực hiện bởi nhiều bước lặp
 - Xây dựng kiến trúc chi tiết
 - Phát triển một phần hệ thống
 - Kiểm thử một phần hệ thống
 - Cài đặt một phần hệ thống

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

111



Tiến trình hợp nhất: góc nhìn quản lý

- Xây dựng
 - Mỗi bước lặp cho một nguyên mẫu thực thi được (executable prototype)
 - Thêm dần dần các chức năng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

112



Tiến trình hợp nhất: góc nhìn quản lý

- Chuyển giao (transition)
 - Giai đoạn phát triển gần như kết thúc
 - Đánh giá dự án
 - Chuyển giao phần mềm cho người sử dụng
 - Huấn luyện người sử dụng
 - Tiếp thị, phân phối, bán

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

113



Tiến trình hợp nhất: góc nhìn quản lý

- Chuyển giao
 - Kết quả
 - Hệ thống thực thi được
 - Phiên bản β
 - Phiên bản chính thức
 - Hướng dẫn cài đặt
 - Hướng dẫn sử dụng

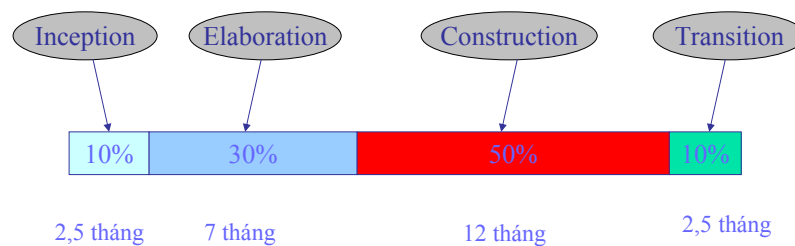
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

114

Tiến trình hợp nhất: góc nhìn quản lý

- Ví dụ điển hình: dự án trong hai năm



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

115

Tiến trình hợp nhất: góc nhìn kỹ thuật

- Góc nhìn kỹ thuật tập trung vào việc triển khai, tổ chức các hoạt động kỹ thuật để tạo ra phần mềm
- Tiến trình phát triển là một dãy các bước lặp (iteration)
- Mỗi bước lặp tạo ra một nguyên mẫu thực thi được
- Hệ thống lớn dần theo số bước lặp cho đến khi hoàn thành
- Mỗi bước lặp trải qua tất cả các giai đoạn (hoạt động) phát triển cổ điển

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

116



Tiến trình hợp nhất: góc nhìn kỹ thuật

- Mỗi bước lặp gồm các hoạt động:
 - Đặc tả
 - Phân tích
 - Thiết kế
 - Mã hóa
 - Kiểm thử
 - Cài đặt
- **Mỗi bước lặp là một tiến trình thác đổ**

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

117



Tiến trình hợp nhất: góc nhìn kỹ thuật

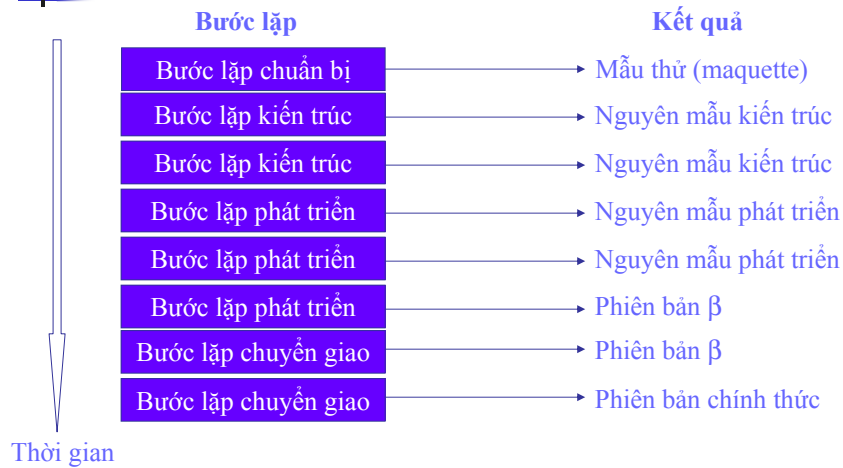
- Các bước lặp cho phép kiểm tra sự lựa chọn công nghệ sử dụng
- Kiểm tra tính khả thi về kỹ thuật
- Bước lặp đầu tiên được phát triển rất nhanh, cho ra mẫu thử (thường bị loại bỏ)
- Các bước lặp cuối cho ra phiên bản β

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

118

Tiến trình hợp nhất: góc nhìn kỹ thuật

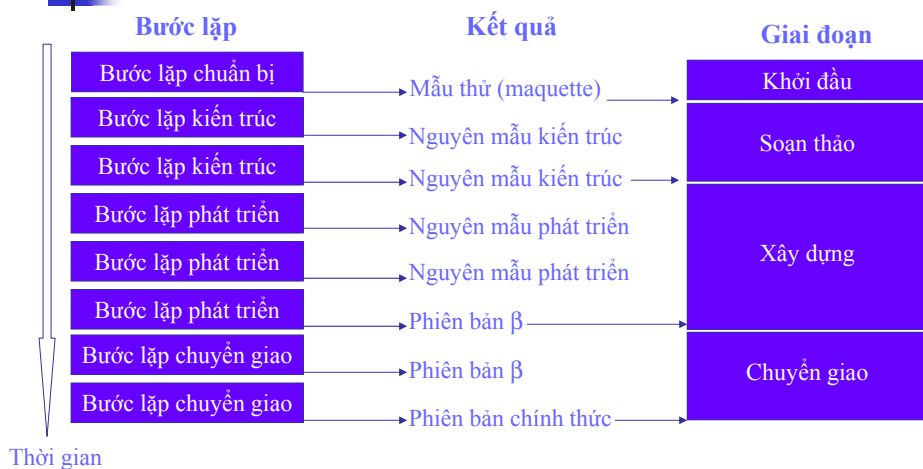


7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

119

Tiến trình hợp nhất: kết hợp hai góc nhìn



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

120



Rational Unified Process

- Tiến trình hợp nhất được phát triển bởi công ty Rational
 - Rational Unified Process (RUP)
 - Hỗ trợ phát triển hướng đối tượng
 - Phần mềm RUP

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

121



UML và Unified Process



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

122

UML và Unified Process

■ Hai giai đoạn chính sử dụng UML

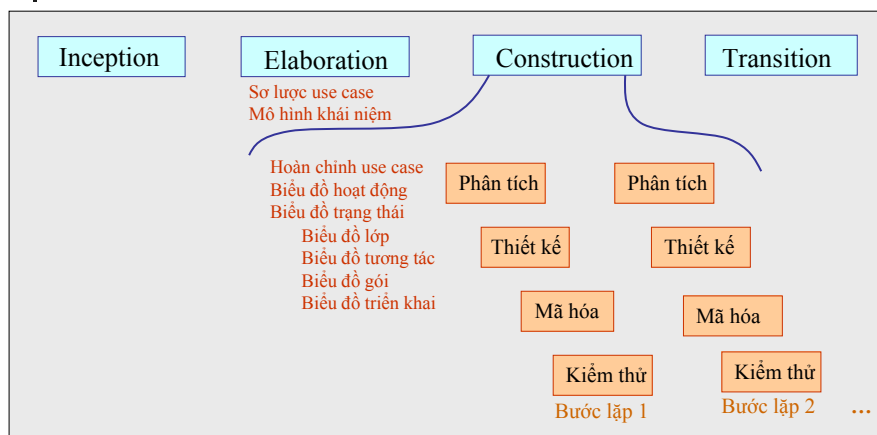
- Elaboration
 - Short Use Case diagrams
 - Analysis Class diagrams
- Construction
 - Complete Use Case diagrams
 - Design Class diagrams
 - Interactions diagrams
 - State diagrams
 - Activity diagrams
 - Package diagrams
 - Deployment diagrams

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

123

UML và Unified Process



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

124



Mô hình hoá yêu cầu – **Biểu đồ ca sử dụng (4)**

Nguyễn Thanh Bình

Khoa Công Nghệ Thông Tin

Trường Đại học Bách khoa, Đại học Đà Nẵng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

125



Nội dung trình bày

- Ca sử dụng
- Tác nhân
- Biểu đồ ca sử dụng
- Cách xác định các ca sử dụng
- Sắp xếp các ca sử dụng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

126



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

127



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*
 - ...

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

128



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

129



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

130



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
 - Xác định số tiền rút
 - Lấy thẻ ra
 - Lấy tiền
 - Lấy phiếu rút tiền
- Các tương tác trên có là các ca sử dụng không ?

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

131



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

132



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*

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

133



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

134



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

135



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

136



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

137



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>

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

138



Đặc tả ca sử dụng

■ Các sự kiện chính (tiếp)

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

139



Đặc tả ca sử dụng

■ Các sự kiện chính (tiếp)

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

140



Đặc tả ca sử dụng

■ Các sự kiện phụ

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

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

141



Ca sử dụng

- 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

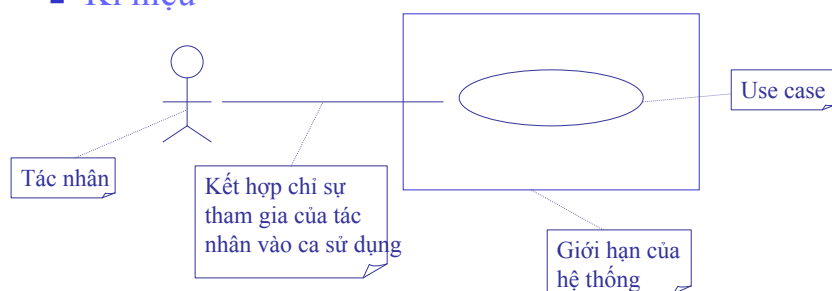
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

142

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



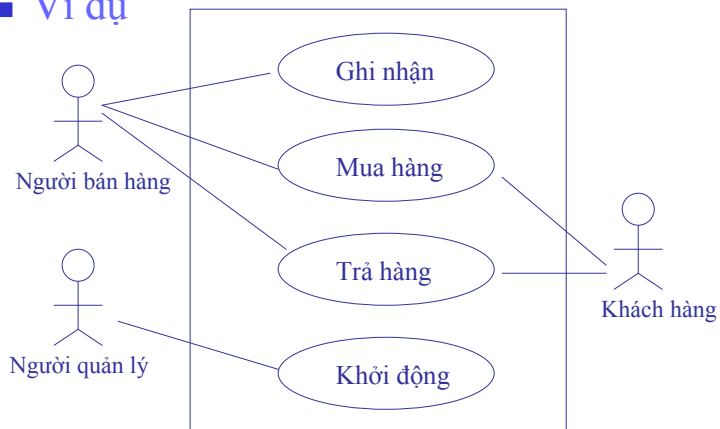
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

143

Biểu đồ ca sử dụng

- Ví dụ



7-Aug-07

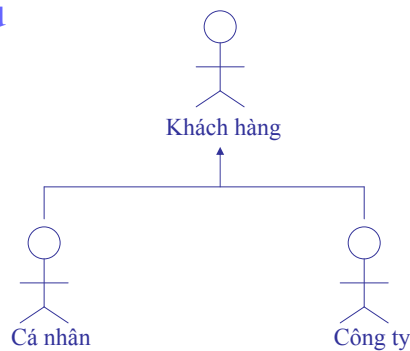
Phân tích và thiết kế hệ thống hướng đối tượng

144



Biểu đồ ca sử dụng

- Các tác nhân có thể có quan hệ thừa kế
- Ví dụ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

145



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

146



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*”

7-Aug-07

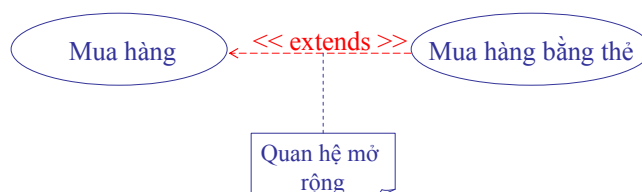
Phân tích và thiết kế hệ thống hướng đối tượng

147



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

148



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*”

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

149



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.*

7-Aug-07

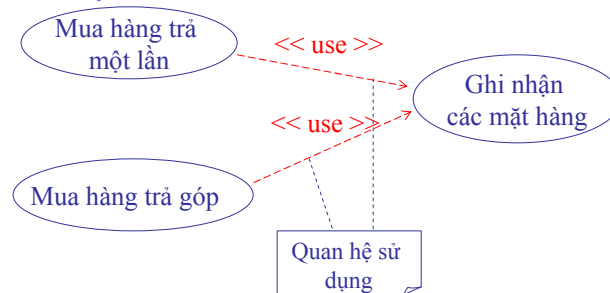
Phân tích và thiết kế hệ thống hướng đối tượng

150



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

151



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

152



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

153



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

154



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?

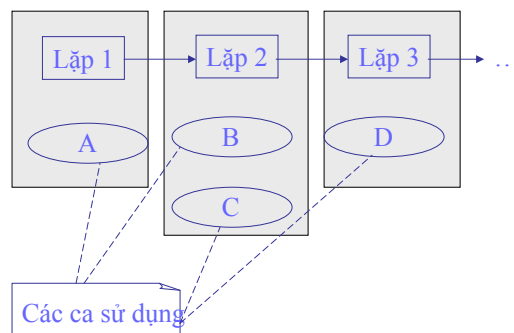
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

155



Sắp xếp các ca sử dụng



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

156



Sắp xếp các ca sử dụng

- Các ca sử dụng nên được thực hiện trước
 - Các ca sử dụng chứa các 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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

157



Hoàn chỉnh các ca sử dụng

- Các ca sử dụng đã được trình bày chi tiết
- Chỉ mô tả ngắn gọn các ca sử dụng : tên ca sử dụng, các tác nhân, mô tả ca sử dụng
- Hoàn chỉnh ca sử dụng :
 - Thêm vào kịch bản các luồng sự kiện
 - Thêm vào tham chiếu đến đặc tả yêu cầu
 - Thêm các điều kiện trước và điều kiện sau khi ca sử dụng xảy ra

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

158



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng

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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

159



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng hoàn chỉnh hơn

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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

160



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng hoàn chỉnh hơn (2) 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>

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

161



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng hoàn chỉnh hơn (3) Các sự kiện chính

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>

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

162



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng hoàn chỉnh hơn (4)

Các sự kiện chính

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

163



Hoàn chỉnh các ca sử dụng

■ Ví dụ ca sử dụng hoàn chỉnh hơn (5)

Các sự kiện phụ

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

164



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

165



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)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

166

Bài tập 1

■ Các ca sử dụng

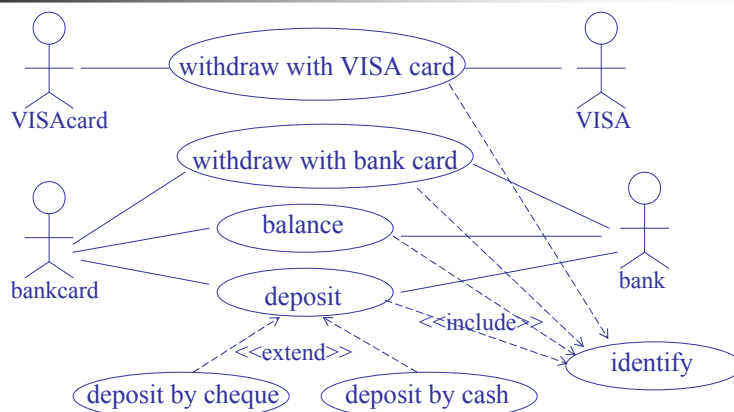
- Rút tiền với thẻ ngân hàng (withdraw by bankcard)
- Rút tiền với thẻ VISA (withdraw by VISAcad)
- Kiểm tra mã PIN (identify)
- Xem số tiền còn trong tài khoản (balance)
- Bỏ tiền vào tài khoản bằng ngân phiếu hoặc tiền mặt (deposit)
- Nạp tiền vào máy (put money)
- Lấy thẻ bị nuốt trong máy (get cards)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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

167

Bài tập 1



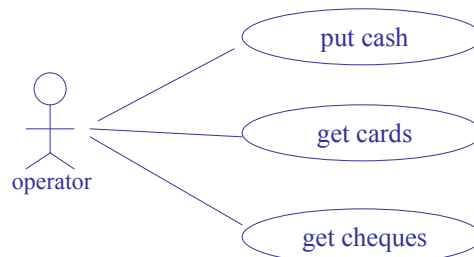
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

168



Bài tập 1



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

169



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

170



Bài tập 3

- **Quản lý bán cơm hộp:** Công ty XXX có một hệ thống nhà hàng XXX chuyên cung cấp cơm hộp tại nhà. Hiện tại các đơn đặt hàng được thực hiện bằng điện thoại đến các cửa hàng XXX. Công ty chỉ có thể xử lý một số có giới hạn các đơn đặt hàng và mỗi khách hàng phải biết rõ thực đơn cung cấp bởi cửa hàng mà họ liên lạc (có thể thay đổi tùy theo cửa hàng). Công ty XXX muốn tin học hóa tiến trình đặt hàng / chế biến / giao hàng.
- Công ty muốn xây dựng phần mềm quản lý từ xa các đơn đặt hàng, các nhà hàng, các nhân viên. Phần mềm cũng cho phép truy cập xem các loại cơm hộp qua internet, mỗi hộp cơm có tên, giá và ảnh. Đối với các hộp cơm có món ăn nóng, một khoảng thời gian từ lúc chuẩn bị đến lúc giao hàng sẽ được gán cho hộp cơm, nếu khách hàng nhận hộp cơm trễ hơn so với thời gian quy định thì có thể yêu cầu trả tiền lui. Tuy nhiên, để hạn chế khả năng này, thì việc đề nghị trả lui không được thực hiện qua internet mà khách hàng phải viết đơn đề nghị và gửi đến người quản lý công ty.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

171



Bài tập 3

- Ngược lại, đơn đặt hàng có thể thực hiện bất kỳ lúc nào qua internet. Khách hàng cần phải có thể tin dụng. Khi đặt hàng, khách hàng cần phải cung cấp thông tin: họ tên và địa chỉ giao hàng. Một đơn đặt hàng có thể yêu cầu nhiều hộp cơm, loại hộp cơm khác nhau và số lượng từng loại hộp cơm. Một khi đã đặt hàng, khách hàng có thể theo dõi đơn đặt hàng của mình. Khi mà đơn đặt hàng chưa được xử lý thì khách hàng có thể hủy đơn đặt hàng.
- Các nhà hàng XXX mở cửa 24/24h. Để bảo đảm phục vụ trên nhiều địa bàn và 24/24h, công ty hợp đồng nhiều nhân viên làm việc bán thời gian, chủ yếu là sinh viên. Mỗi nhân viên được cấp một điện thoại di động để liên lạc. Khi cần báo rằng nhân viên đó rảnh rồi, chỉ cần ấn nút trên điện thoại di động hoặc ngược lại ấn một nút khác để thông báo nhân viên đó bận. Người quản lý có thể xem trạng thái của toàn hệ thống thông qua Internet. Người quản lý có thể giao nhiệm vụ cho nhân viên làm việc ở cửa hàng hoặc đi giao hàng. Một nhân viên có thể thay đổi địa điểm làm việc hoặc vai trò (làm việc tại cửa hàng hay đi giao hàng) nhiều lần trong một ngày phụ thuộc vào các đơn đặt hàng.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

172



Bài tập 3

- Khi đặt hàng, khách hàng không chỉ rõ đặt ở cửa hàng nào, mà chính người quản lý sẽ yêu cầu cửa hàng nào cung cấp và nhân viên nào giao hàng. Mục đích là để người quản lý đề ưu thời gian, khoảng cách, hoạt động giữa các cửa hàng và các nhân viên. Mỗi nhân viên giao hàng có phương tiện riêng để đi giao hàng. Hơn nữa, khi đi giao hàng nhân viên sẽ được cung cấp thiết bị chuyên dụng có tích hợp GPS, cho phép định vị chính xác thông qua kết nối vệ tinh. Thiết bị có một màn hình cho phép nhân viên xem thông tin các đơn đặt hàng mà nhân viên đó cần thực hiện giao hàng, tra bản đồ và định vị các cửa hàng và địa chỉ khách hàng. Nhân viên giao hàng cũng sử dụng thiết bị này để thông báo khi nào anh ta thực hiện đơn đặt hàng nào và khi nào giao hàng cho khách hàng.
- Trong mỗi cửa hàng có một nhân viên đóng vai trò điều phối viên. Tại cửa hàng, chỉ có điều phối viên là người duy nhất sử dụng phần mềm còn các nhân viên khác chuẩn bị các món ăn. Điều phối viên xem các đơn đặt hàng cần thực hiện, chỉ rõ khi nào bắt đầu chuẩn bị món ăn cho đơn đặt hàng, khi nào kết thúc và khi nào thì giao cho nhân viên giao hàng.
- Xây dựng biểu đồ ca sử dụng, chỉ xét các tác nhân là người sử dụng.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

173



Mô hình khái niệm (5)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

174



Nội dung trình bày

- Mô hình khái niệm
- Xác định các khái niệm
- Xác định các thuộc tính
- Xác định các kết hợp

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

175



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)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

176



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

177



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

178



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

179



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*, ...

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

180



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

181



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

182



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>

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

183



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 <u>mặt hàng</u>, <u>người bán hàng</u> báo hiệu kết thúc <u>việc ghi nhận hàng</u>.</p> <p>6. <u>Người bán hàng</u> thông báo <u>tổng số tiền</u> phải trả cho <u>khách hàng</u>.</p> <p>7. <u>Khách hàng</u> trả tiền cho <u>người bán hàng</u>.</p>	<p>5. Tính và hiển thị <u>tổng số tiền</u>.</p>

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

184



Xác định các khái niệm

■ Ví dụ

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

185



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

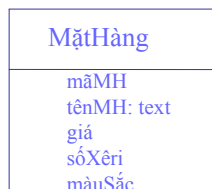
186



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



Phương án 1 (chưa tốt)

7-Aug-07

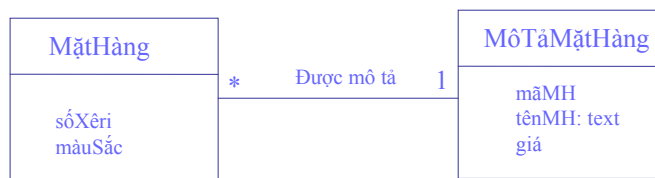
Phân tích và thiết kế hệ thống hướng đối tượng

187



Xác định các khái niệm

■ Lớp “MôTả”



Phương án 2 (tốt hơn)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

188



Xác định các khái niệm

■ 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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

189



Xác định các khái niệm

■ 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

7-Aug-07

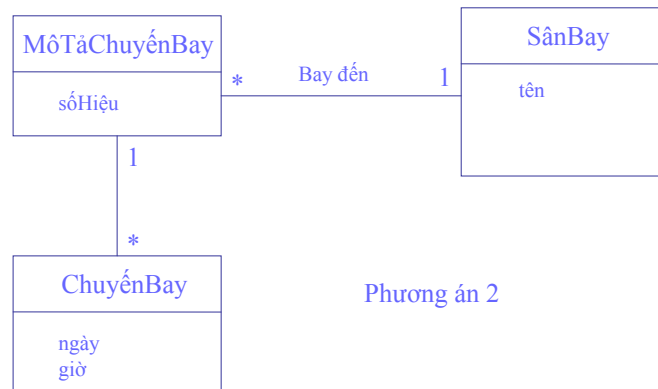
Phân tích và thiết kế hệ thống hướng đối tượng

190



Xác định các khái niệm

■ Lớp “MôTả”



Phương án 2

7-Aug-07

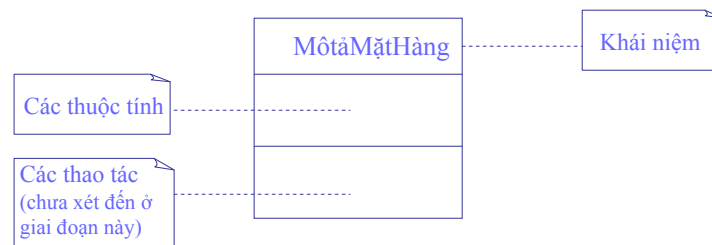
Phân tích và thiết kế hệ thống hướng đối tượng

191



Biểu diễn khái niệm

■ Sử dụng kí hiệu của biểu đồ lớp



7-Aug-07

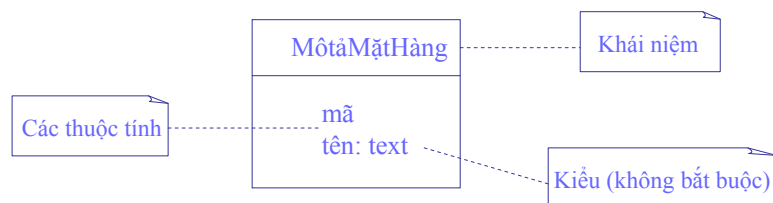
Phân tích và thiết kế hệ thống hướng đối tượng

192



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



7-Aug-07

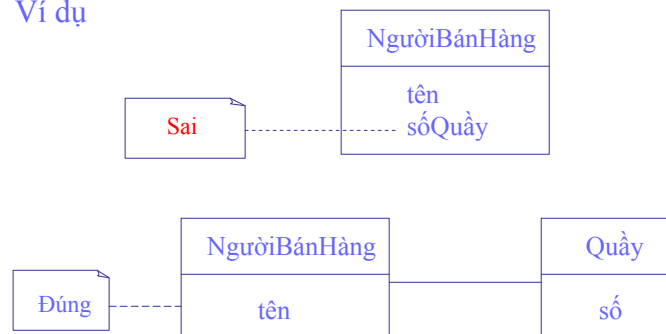
Phân tích và thiết kế hệ thống hướng đối tượng

193



Thuộc tính

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

194



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

7-Aug-07

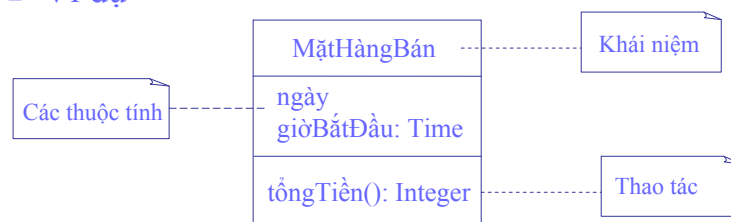
Phân tích và thiết kế hệ thống hướng đối tượng

195



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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

196



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

7-Aug-07

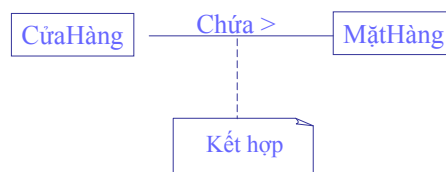
Phân tích và thiết kế hệ thống hướng đối tượng

197



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



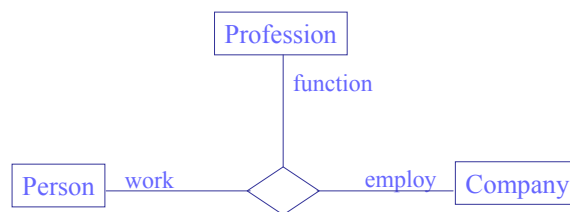
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

198

Kết hợp

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

199

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



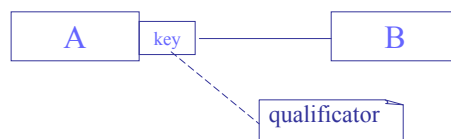
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

200

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

201

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

202



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)

7-Aug-07

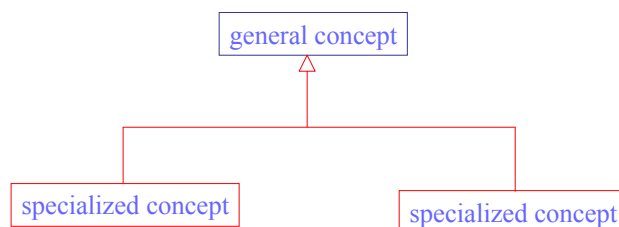
Phân tích và thiết kế hệ thống hướng đối tượng

203



Chuyên biệt hóa

- Kí hiệu



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

204

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

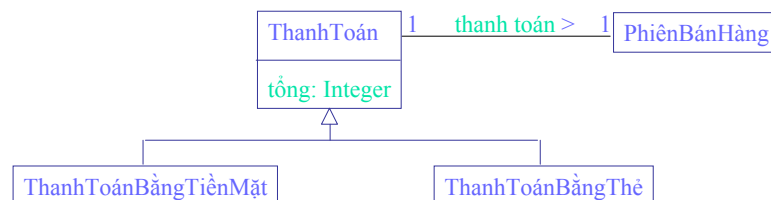
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

205

Thừa kế

- Ví dụ



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

7-Aug-07

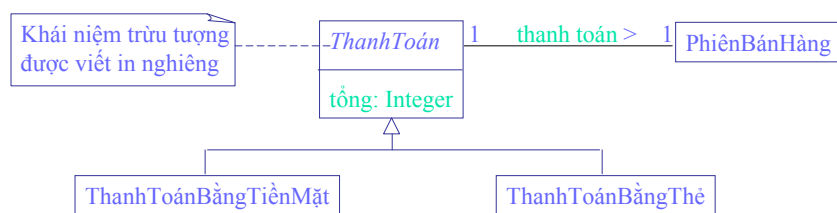
Phân tích và thiết kế hệ thống hướng đối tượng

206



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

207



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



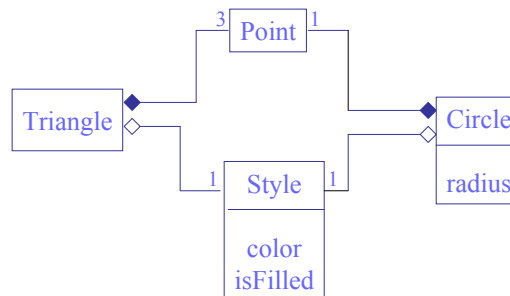
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

208

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”

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

209

Quan hệ hợp thành và quan hệ kết tập

- Quan hệ hợp thành **nhân 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



7-Aug-07

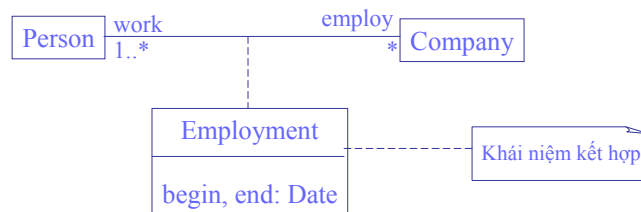
Phân tích và thiết kế hệ thống hướng đối tượng

210



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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

211



Bài tập 1

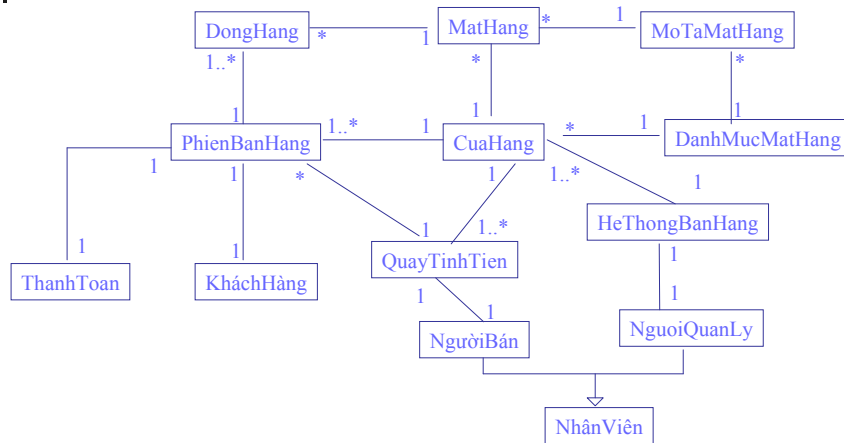
- Xây dựng mô hình khái niệm của hệ thống/phần mềm bán hàng tại siêu thị
 - Phần mềm bán hàng sử dụng tại siêu thị nhằm giúp ghi nhận hoạt động bán hàng, xử lý các thanh toán với khách hàng. Phần mềm được sử dụng bởi người bán hàng và được quản lý bởi người quản lý siêu thị. Phần mềm nhằm tự động hóa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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

212

Bài tập 1



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

213

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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

214



Bài tập 3

- **Quản lý bán cơm hộp:** Công ty XXX có một hệ thống nhà hàng XXX chuyên cung cấp cơm hộp tại nhà. Hiện tại các đơn đặt hàng được thực hiện bằng điện thoại đến các cửa hàng XXX. Công ty chỉ có thể xử lý một số có giới hạn các đơn đặt hàng và mỗi khách hàng phải biết rõ thực đơn cung cấp bởi cửa hàng mà họ liên lạc (có thể thay đổi tùy theo cửa hàng). Công ty XXX muốn tin học hóa tiến trình đặt hàng / chế biến / giao hàng.
- Nhờ vào phần mềm đó, công ty muốn quản lý từ xa các đơn đặt hàng, các nhà hàng, các nhân viên. Phần mềm cũng cho phép truy cập xem các loại cơm hộp qua internet, mỗi hộp cơm có tên, giá và ảnh. Đối với các hộp cơm có món ăn nóng, một khoảng thời gian từ lúc chuẩn bị đến lúc giao hàng sẽ được gán cho hộp cơm, nếu khách hàng nhận hộp cơm trễ hơn so với thời gian quy định thì có thể yêu cầu trả tiền lui. Tuy nhiên, để hạn chế khả năng này, thì việc đề nghị trả lui không được thực hiện qua internet mà khách hàng phải viết đơn đề nghị và gửi đến người quản lý công ty.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

215



Bài tập 3

- Ngược lại, đơn đặt hàng có thể thực hiện bất kỳ lúc nào qua internet. Khách hàng cần phải có thể tin dụng. Khi đặt hàng, khách hàng cần phải cung cấp thông tin: họ tên và địa chỉ giao hàng. Một đơn đặt hàng có thể yêu cầu nhiều hộp cơm loại hộp cơm khác nhau và số lượng từng loại hộp cơm. Một khi đã đặt hàng, khách hàng có thể theo dõi đơn đặt hàng của mình. Khi mà đơn đặt hàng chưa được xử lý thì khách hàng có thể hủy đơn đặt hàng.
- Các nhà hàng XXX mở cửa 24/24h. Để bảo đảm phục vụ trên nhiều địa bàn và 24/24h, công ty hợp đồng nhiều nhân viên làm việc bán thời gian, chủ yếu là sinh viên. Mỗi nhân viên được cấp một điện thoại di động để tiện liên lạc. Khi cần báo rằng nhân viên đó rảnh rồi, chỉ cần ấn nút trên điện thoại di động hoặc ngược lại ấn một nút khác để thông báo nhân viên đó bận. Người quản lý có thể xem trạng thái của toàn hệ thống thông qua Internet. Người quản lý có thể giao nhiệm vụ cho nhân viên làm việc ở cửa hàng hoặc đi giao hàng. Một nhân viên có thể thay đổi địa điểm làm việc hoặc vai trò (làm việc tại cửa hàng hay đi giao hàng) nhiều lần trong một ngày phụ thuộc vào các đơn đặt hàng.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

216



Bài tập 3

- Khi đặt hàng, khách hàng không chỉ rõ đặt ở cửa hàng nào, mà chính người quản lý sẽ yêu cầu cửa hàng nào cung cấp và nhân viên nào giao hàng. Mục đích là để người quản lý đề tối ưu thời gian, khoảng cách, hoạt động giữa các cửa hàng và các nhân viên. Mỗi nhân viên giao hàng có phương tiện riêng để đi giao hàng. Hơn nữa, khi đi giao hàng nhân viên sẽ được cung cấp thiết bị chuyên dụng có tích hợp GPS, cho phép định vị chính xác thông qua kết nối vệ tinh. Thiết bị có một màn hình cho phép nhân viên xem thông tin các đơn đặt hàng mà nhân viên đó cần thực hiện giao hàng, tra bản đồ và định vị các cửa hàng và địa chỉ khách hàng. Nhân viên giao hàng cũng sử dụng thiết bị này để thông báo khi nào anh ta thực hiện đơn đặt hàng nào và khi nào giao hàng cho khách hàng.
- Trong mỗi cửa hàng có một nhân viên đóng vai trò điều phối viên. Tại cửa hàng, chỉ có điều phối viên là người duy nhất sử dụng phần mềm còn các nhân viên khác chuẩn bị các món ăn. Điều phối viên xem các đơn đặt hàng cần thực hiện, chỉ rõ khi nào bắt đầu chuẩn bị món ăn cho đơn đặt hàng, khi nào kết thúc và khi nào thì giao cho nhân viên giao hàng.
- Xây dựng mô hình khái niệm.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

217



Biểu đồ hoạt động (6)

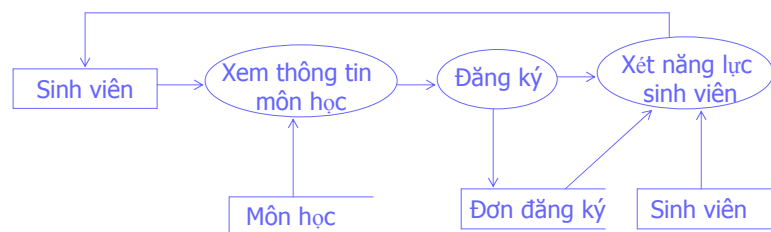
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

218

Biểu đồ hoạt động

- Phương pháp phân tích và thiết kế truyền thống
 - Biểu đồ luồng dữ liệu (DFD – Data-Flow Diagram)
 - Ví dụ



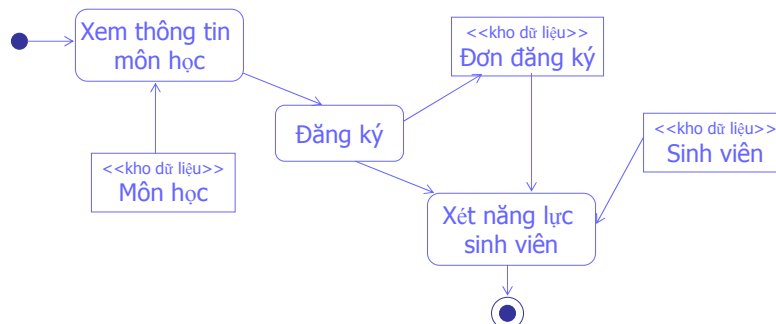
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

219

Biểu đồ hoạt động

- Phương pháp phân tích và thiết kế HĐT
 - Biểu đồ hoạt động



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

220



Biểu đồ hoạt động

- **Biểu đồ hoạt động** (activity diagram) cho phép mô tả hoạt động của hệ thống so với một hoặc nhiều ca sử dụng
- Một biểu đồ hoạt động định nghĩa:
 - **Các hoạt động** (activity) của hệ thống và của các tác nhân
 - **Thứ tự** mà các hoạt động này được thực hiện
 - **Phụ thuộc** có thể giữa các hoạt động này
- Một hoạt động tương ứng với một công việc ở mức trừu tượng cao có mục tiêu xác định

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

221



Biểu đồ hoạt động

- Các **hoạt động** không tương ứng với các **thao tác** trong mô hình khái niệm, các thao tác đó liên quan đến khái niệm, chứ không liên quan đến hệ thống hay các tác nhân
- Thông thường, các biểu đồ hoạt động được định nghĩa trước hoặc trong khi xây dựng mô hình khái niệm
 - Dựa vào biểu đồ hoạt động các thao tác sẽ được thêm vào mô hình khái niệm

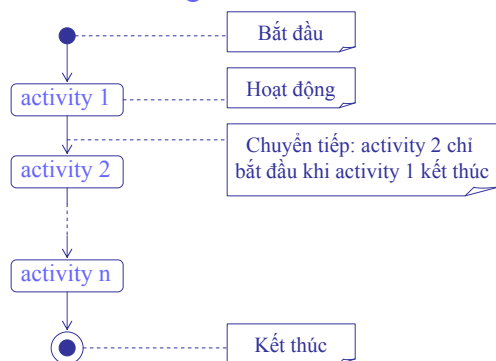
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

222

Biểu đồ hoạt động

■ Kí hiệu của các hoạt động



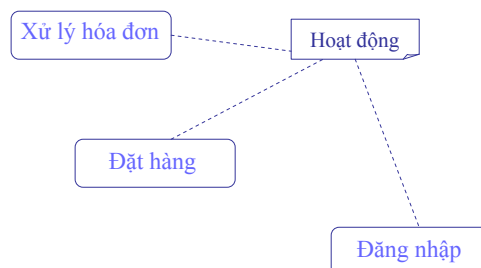
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

223

Biểu đồ hoạt động

■ Ví dụ các hoạt động



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

224

Biểu đồ hoạt động

■ Đồng bộ hóa các hoạt động



- Các activity 1', ..., n' (cũng như 1, ..., n) có thể thực hiện trong **bất cứ thứ tự** nào. Hoặc các activity này có thể thực hiện **đồng thời**.

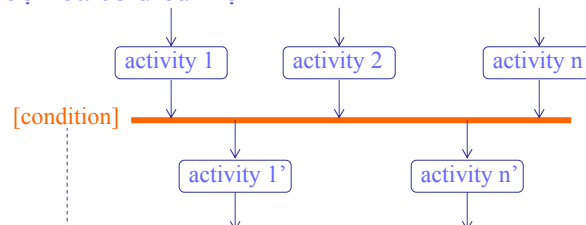
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

225

Biểu đồ hoạt động

■ Đồng bộ hóa có điều kiện



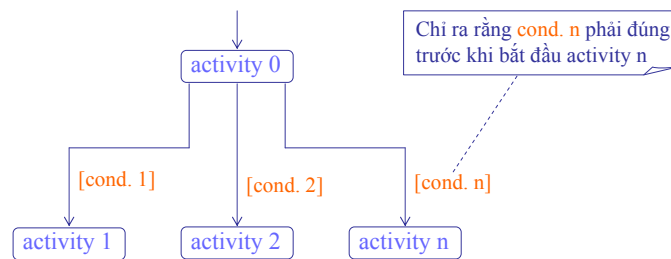
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

226

Biểu đồ hoạt động

- Quyết định



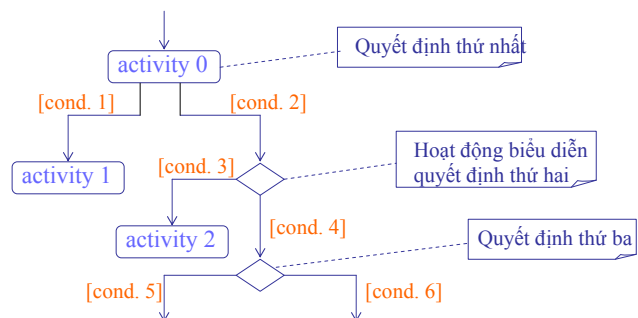
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

227

Biểu đồ hoạt động

- Quyết định kết hợp: trong trường hợp nếu có nhiều quyết định đi liền nhau, thì cần phải biểu diễn bằng hoạt động riêng



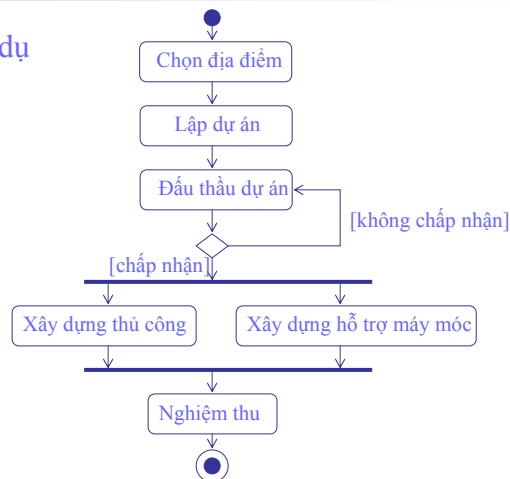
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

228

Biểu đồ hoạt động

■ Ví dụ



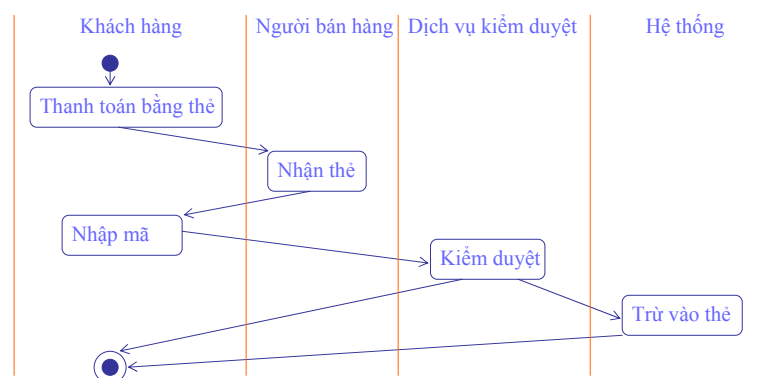
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

229

Biểu đồ hoạt động

■ Hành lang (swimlane): dùng để phân luồng các hoạt động liên quan đến các tác nhân hoặc hệ thống



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

230



Biểu đồ hoạt động

■ Bài tập: Máy rút tiền ATM

- Mô tả một máy ATM. 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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

231



Biểu đồ hoạt động

■ Bài tập

- Mô tả việc rút tiền với một thẻ VISA bằng biểu đồ hoạt động. Thẻ có thể là không hợp lệ. Nếu thẻ hợp lệ, khách hàng có thể nhập mã PIN. Thẻ sẽ bị nuốt sau ba lần nhập không thành công. Hệ thống ngân hàng cho phép rút một số lượng tiền tối đa hoặc từ chối việc rút tiền. Một thẻ không được lấy lại bởi khách hàng sau khi rút sẽ bị nuốt. Tiền mặt không lấy bởi khách hàng sẽ bị máy thu hồi lại. Phiếu xác nhận việc rút tiền luôn được in trong khi khách hàng chờ lấy tiền.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

232



Biểu đồ trạng thái (7)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

233



Biểu đồ trạng thái

- Biểu đồ trạng thái (state diagram) mô tả hành vi của một thể hiện của một khái niệm (hay hành vi của một đối tượng)
- Biểu đồ trạng thái bao gồm:
 - Các **trạng thái** của một thể hiện (đối tượng)
 - Các **chuyển tiếp** cho phép chuyển từ trạng thái này sang trạng thái khác
 - Các **sự kiện** cho phép vượt qua các chuyển tiếp

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

234



Trạng thái

- Một **trạng thái** là một tình huống mà thể hiện/đối tượng của một khái niệm/lớp tồn tại ở một thời điểm nào đó
- Một thể hiện ở trong một trạng thái trong một khoảng thời gian nhất định
- Trong khoảng thời gian đó, thể hiện có thể thực hiện các hoạt động và chờ đợi các sự kiện
- Kí hiệu



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

235



Chuyển tiếp

- Các **chuyển tiếp** (transition) gắn liền với các hành động kích hoạt bởi các sự kiện
- Có hai loại chuyển tiếp
 - **Chuyển tiếp trong** (interne transition) một trạng thái, cho phép trả lời một sự kiện mà không rời trạng thái đó.
 - **Chuyển tiếp giữa các trạng thái** hay **chuyển tiếp ngoài** (externe transition): mô tả một sự chuyển trạng thái.

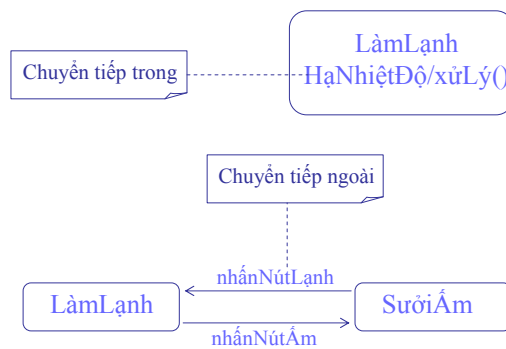
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

236

Chuyển tiếp

- Ví dụ: các trạng thái của một máy điều hòa



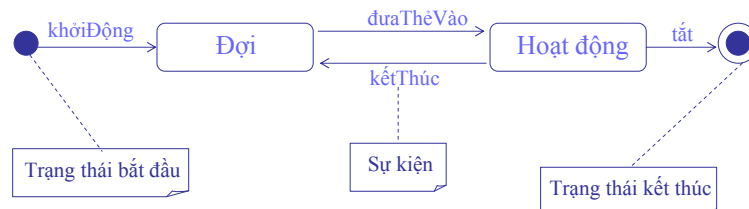
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

237

Ví dụ

- Mô tả các trạng thái của khái niệm “máy ATM”



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

238



Sự kiện

- Một chuyển tiếp có dạng tổng quát như sau:

SựKiện [ĐiềuKiện] / HànhĐộng

- “SựKiện” là tên sự kiện dẫn đến sự vượt qua một chuyển tiếp
- “ĐiềuKiện” là một biểu thức logic phải đúng để có thể vượt qua một chuyển tiếp
- “HànhĐộng” định nghĩa một thao tác được thực hiện khi vượt qua chuyển tiếp.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

239



Sự kiện

- Ví dụ: các trạng thái của “bóng đèn”



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

240



Sự kiện đặc biệt

- Có ba **sự kiện đặc biệt** dành cho chuyển tiếp trong một trạng thái
 - **entry** và **exit** cho phép mô tả các hoạt động cần phải thực hiện lúc **vào** hoặc lúc **ra** trạng thái
 - **do** đặc tả một hoạt động cần phải được thực hiện khi **ở trong** trạng thái
- Lưu ý: khi các chuyển tiếp trong có một trong ba sự kiện trên thì chúng không thể có các biểu thức điều kiện

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

241



Sự kiện đặc biệt

- Ví dụ

NhậpMậtKhẩu

entry / không hiển thị các kí tự nhập vào
exit / hiển thị các kí tự nhập vào
do/ quản lý nhập vào của người sử dụng

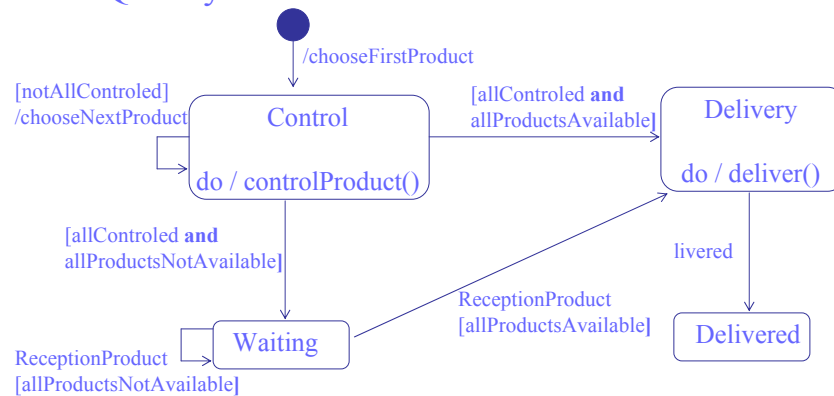
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

242

Ví dụ biểu đồ trạng thái

■ Quản lý “order”



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

243

Trạng thái phức hợp

- Nhiều trạng thái và chuyển tiếp nối các trạng thái này có thể nhóm lại tạo nên một **trạng thái phức hợp** (composite state)
- Một trạng thái phức hợp chứa một trạng thái đầu
- Ví dụ

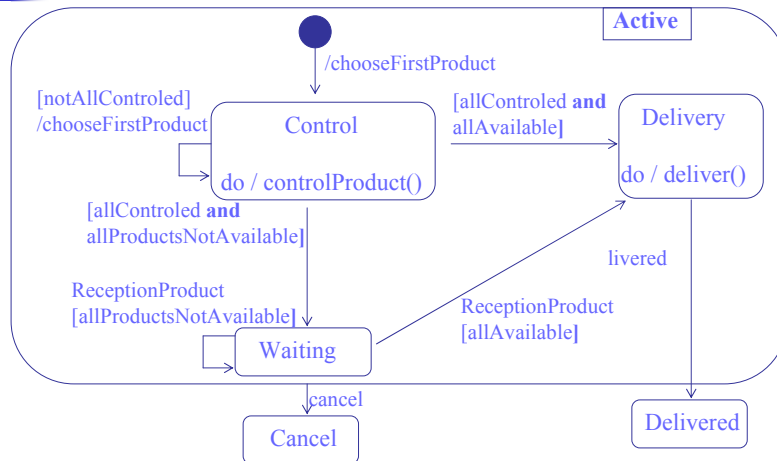
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

244



Trạng thái phức hợp



7-Aug-07

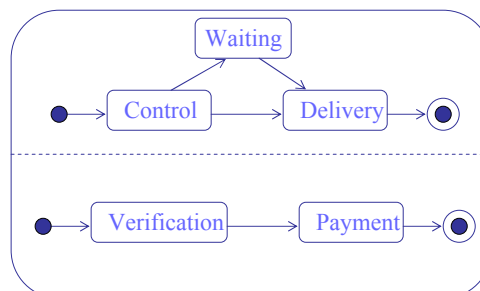
Phân tích và thiết kế hệ thống hướng đối tượng

245



Trạng thái cạnh tranh

- Các **trạng thái cạnh tranh** (concurrent state) tồn tại đồng thời bên trong một trạng thái phức hợp
- Ví dụ: xử lý đồng thời “order” và “payment”



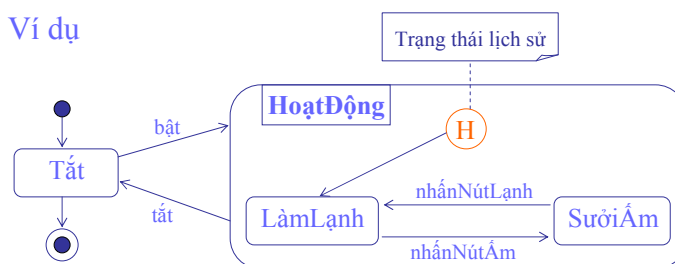
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

246

Trạng thái lịch sử

- Đôi khi cần ghi nhớ lại trạng thái cuối cùng khi đi ra khỏi một trạng thái phức hợp, ta sử dụng **trạng thái lịch sử** (history state)
- Ví dụ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

247

Ví dụ

- Hãy mô tả biểu đồ trạng thái của các khái niệm sau:
 - Hóa đơn
 - Hệ thống bán hàng (phần mềm bán hàng ở siêu thị)
 - Thang máy

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

248



Ví dụ

■ Hóa đơn



7-Aug-07

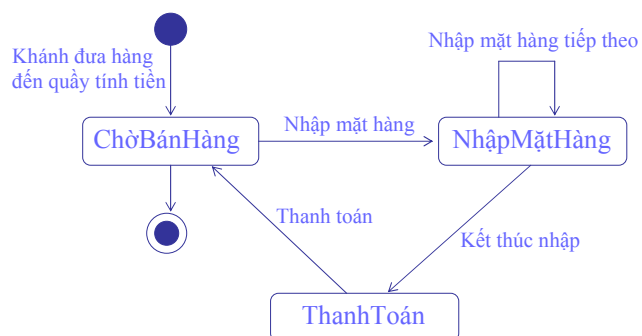
Phân tích và thiết kế hệ thống hướng đối tượng

249



Ví dụ

■ Hệ thống bán hàng



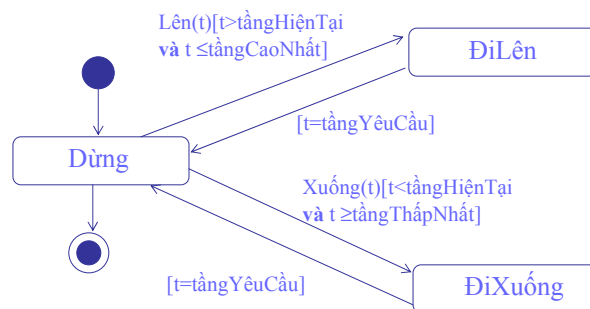
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

250

Ví dụ

■ Thang máy



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

251

Bài tập 1

- **Tiến trình sửa xe hơi ở một garage:** Có nhiều xe hơi của khách hàng yêu cầu được sửa chữa, chúng được đặt trong garage đợi để sửa. Việc sửa chữa có thể được tiến hành bất kỳ lúc nào. Khi sửa chữa có thể cần phải thay phụ tùng mới, trong trường hợp đó cần phải đợi phụ tùng mới được mua về để thay thế. Khi sửa xong xe, xe sẽ được lái thử. Theo kết quả thử, sẽ xác định cần sửa chữa khác hoặc xe đã sẵn sàng để giao cho khách hàng. Khi đó, xe được đưa vào garage đợi khách hàng đến lấy.
- Hãy xây dựng biểu đồ trạng thái mô tả các trạng thái của xe hơi cần sửa chữa.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

252



Bài tập 2

- Mô tả các trạng thái của một **cuốn sách trong thư viện**
 - Khi cuốn sách được nhập vào thư viện, nếu cuốn sách đó là bản duy nhất thì nó chỉ được đọc tại chỗ, nếu có nhiều bản của cuốn sách đó thì nó có thể dùng để cho mượn. Thậm chí có nhiều bản của cuốn sách, người thủ thư có quyền thay đổi các bản đó thành loại tài liệu chỉ được đọc tại chỗ. Nếu một cuốn sách được cho mượn một thời gian xác định mà nó không được trả lại thì coi như là bị mất và sau đó 1 năm thì bị xóa ra khỏi CSDL. Tuy nhiên, nếu sau đó tìm thấy thì nó lại sẵn sàng để cho mượn. Trong trường hợp cuốn sách quá cũ và hư hỏng thì nó sẽ bị loại bỏ. Cuốn sách có thể được thay thế cuốn mới hoặc bị xóa khỏi CSDL. Không thể đặt trước sách mà hiện tại không có trong thư viện.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

253



Biểu đồ lớp

- Nhắc lại: giai đoạn elaboration đã xây dựng mô hình khái niệm hay còn gọi là biểu đồ lớp phân tích
- Giai đoạn construction xây dựng **biểu đồ lớp** (class diagram) chi tiết hay còn gọi là biểu đồ lớp thiết kế (design class diagram)
- Biểu đồ lớp như thế là một bước chi tiết hóa mô hình khái niệm

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

254



Biểu đồ lớp (8)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

255



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

256



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

257



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

7-Aug-07

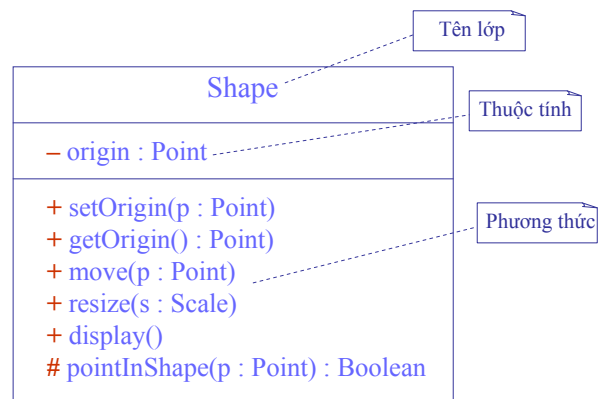
Phân tích và thiết kế hệ thống hướng đối tượng

258



Biểu đồ lớp

■ Ví dụ



7-Aug-07

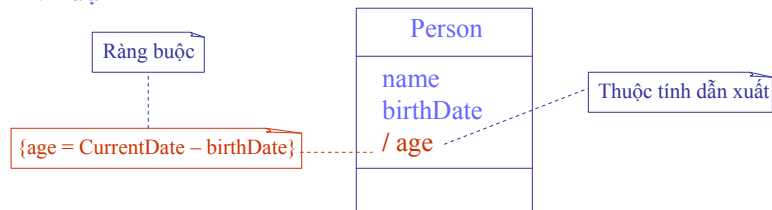
Phân tích và thiết kế hệ thống hướng đối tượng

259



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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

260



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)

7-Aug-07

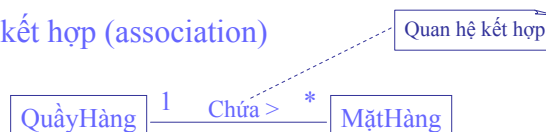
Phân tích và thiết kế hệ thống hướng đối tượng

261

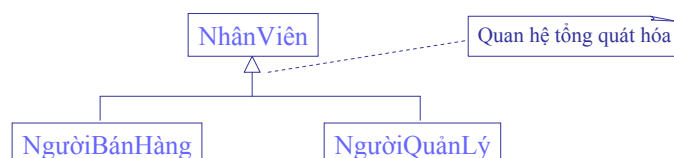


Biểu đồ lớp

- Quan hệ kết hợp (association)



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



7-Aug-07

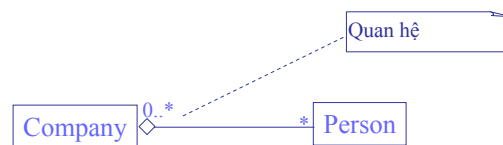
Phân tích và thiết kế hệ thống hướng đối tượng

262

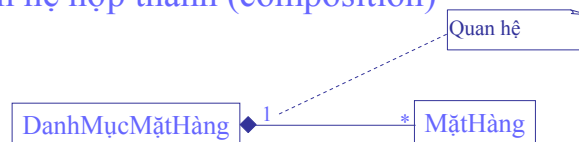


Biểu đồ lớp

- Quan hệ kết tập (agregation)



- Quan hệ hợp thành (composition)



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

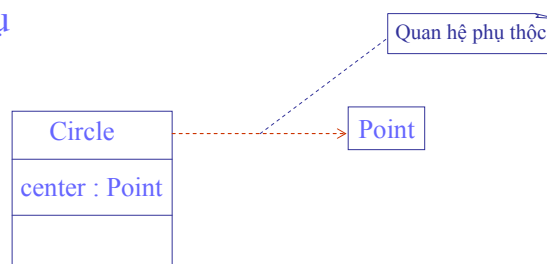
263



Biểu đồ lớp

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

- Ví dụ



7-Aug-07

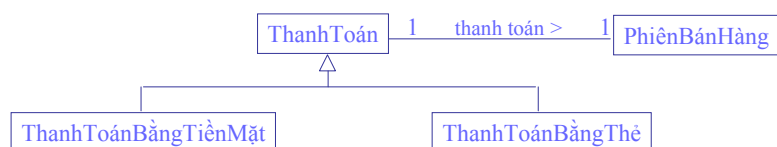
Phân tích và thiết kế hệ thống hướng đối tượng

264



Biểu đồ lớp

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



7-Aug-07

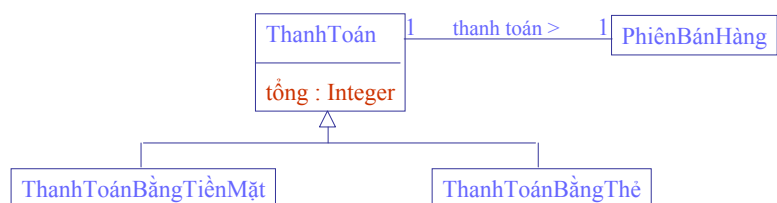
Phân tích và thiết kế hệ thống hướng đối tượng

265



Biểu đồ lớp

- Chi tiết các thuộc tính



7-Aug-07

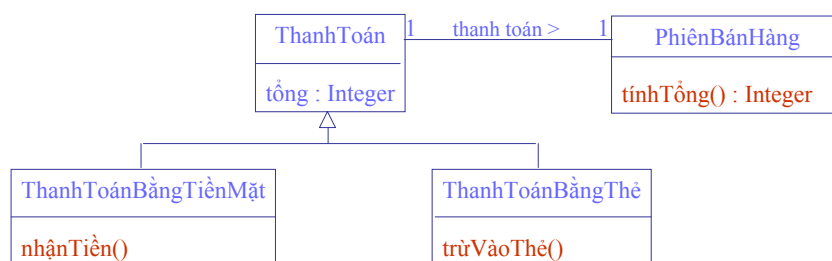
Phân tích và thiết kế hệ thống hướng đối tượng

266



Biểu đồ lớp

■ Chi tiết các phương thức



7-Aug-07

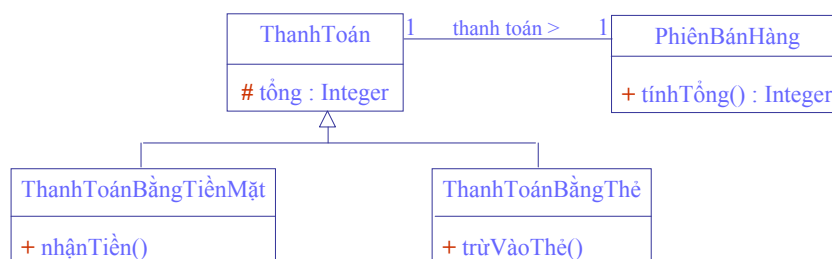
Phân tích và thiết kế hệ thống hướng đối tượng

267



Biểu đồ lớp

■ Xác định các mức khả kiến



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

268



Biểu đồ lớp

- Bài tập 1: Trung tâm đào tạo tin học
 - Cần xây dựng phần mềm quản lý đào tạo ở một trung tâm tin học tư nhân. Một chuyên đề/môn học được mô tả bởi một số thông tin như tên, thời lượng, ... Mỗi chuyên đề có thể được mở nhiều khóa trong một năm. Mỗi khóa mang một số thông tin như ngày bắt đầu, giá, ... Một khóa học thường được dạy bởi nhiều giáo viên, trong đó có một giáo viên chịu trách nhiệm chính. Mỗi giáo viên dạy nhiều khóa trong một năm. Phần mềm cần ghi nhận số giờ dạy mỗi giáo viên. Mỗi giáo viên có một vài thông tin cá nhân, như tên, địa chỉ, ... Mỗi khóa học có một số học viên. Học viên gồm hai loại: các học viên tự do và các học viên là nhân viên các công ty đối tác. Mỗi học viên có một số thông tin cá nhân, như tên, địa chỉ, ... Nếu học viên là nhân viên các công ty đối tác cần ghi nhận thêm tên và địa chỉ công ty. Phần mềm cần quản lý riêng rẽ (đặc biệt là việc thu tiền) của hai loại học viên trên.
 - Xây dựng biểu đồ lớp.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

269



Biểu đồ lớp

- Bài tập 2: Công ty XXX có một hệ thống nhà hàng XXX chuyên cung cấp cơm hộp tại nhà.
 - Chi tiết mô hình khái niệm đã xây dựng bằng cách thêm các thuộc tính

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

270



Biểu đồ tương tác (9)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

271



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

272



Biểu đồ tương tác

- Có hai loại biểu đồ tương tác:
 - Biểu đồ tuần tự
 - Biểu đồ cộng tác

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

273



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

7-Aug-07

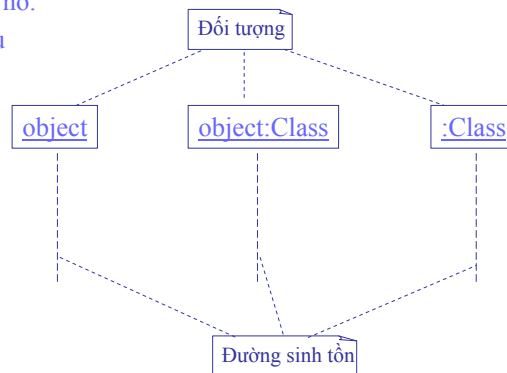
Phân tích và thiết kế hệ thống hướng đối tượng

274



Biểu đồ tuần tự

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



7-Aug-07

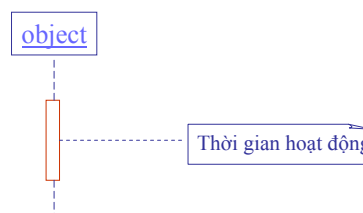
Phân tích và thiết kế hệ thống hướng đối tượng

275



Biểu đồ tuần tự

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

276



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)

7-Aug-07

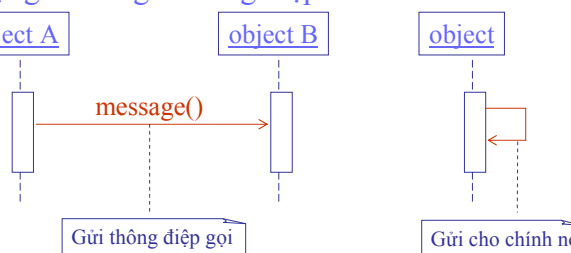
Phân tích và thiết kế hệ thống hướng đối tượng

277



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 object A object B object



7-Aug-07

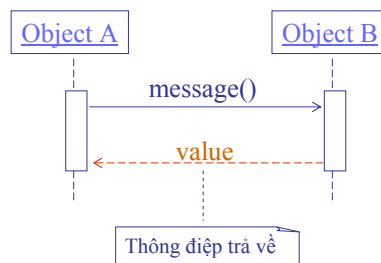
Phân tích và thiết kế hệ thống hướng đối tượng

278



Biểu đồ tuần tự

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



7-Aug-07

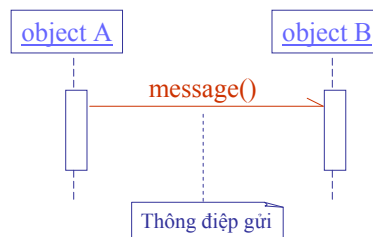
Phân tích và thiết kế hệ thống hướng đối tượng

279



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



7-Aug-07

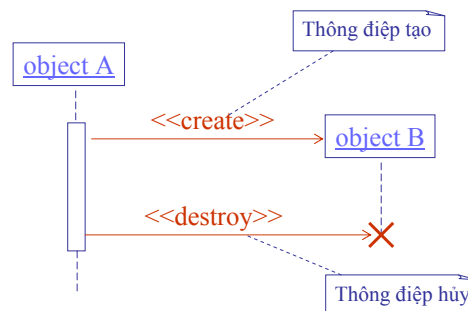
Phân tích và thiết kế hệ thống hướng đối tượng

280



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



7-Aug-07

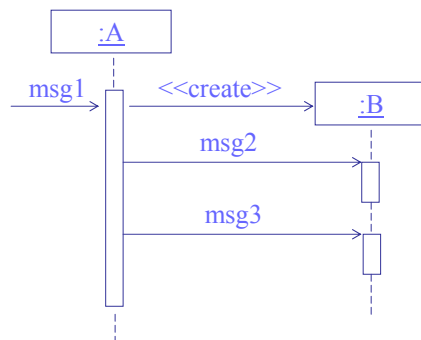
Phân tích và thiết kế hệ thống hướng đối tượng

281



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() { ... }
}
```

7-Aug-07

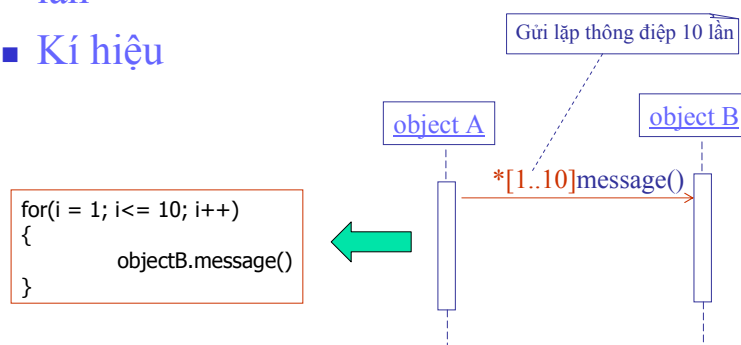
Phân tích và thiết kế hệ thống hướng đối tượng

282



Biểu đồ tuần tự

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



7-Aug-07

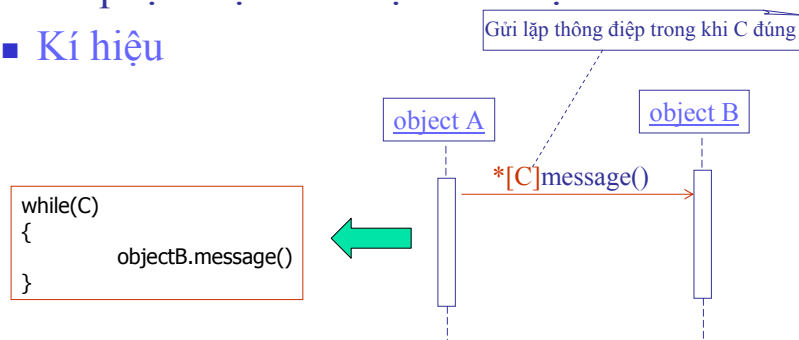
Phân tích và thiết kế hệ thống hướng đối tượng

283



Biểu đồ tuần tự

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



7-Aug-07

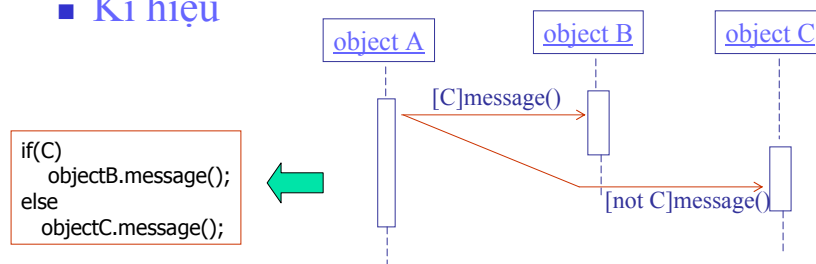
Phân tích và thiết kế hệ thống hướng đối tượng

284



Biểu đồ tuần tự

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



7-Aug-07

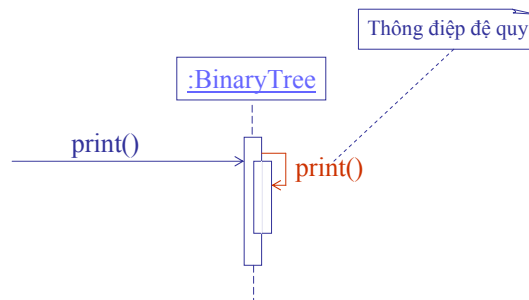
Phân tích và thiết kế hệ thống hướng đối tượng

285



Biểu đồ tuần tự

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



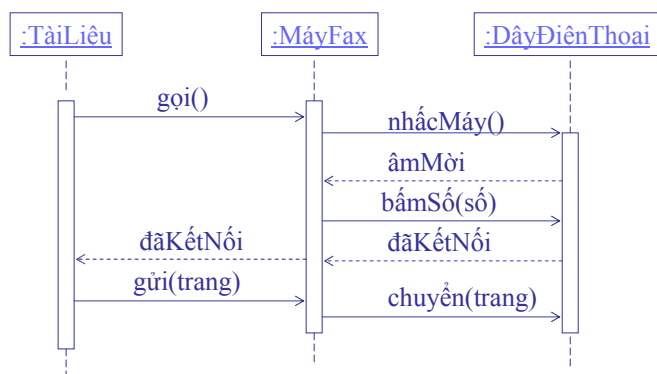
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

286

Biểu đồ tuần tự

■ Ví dụ



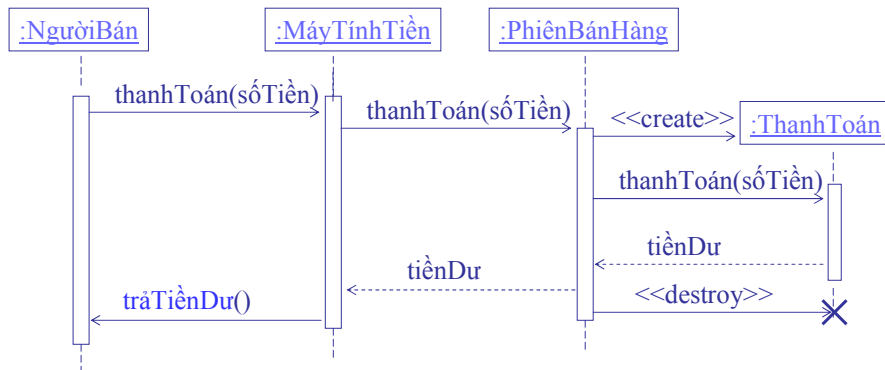
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

287

Biểu đồ tuần tự

■ Ví dụ



7-Aug-07

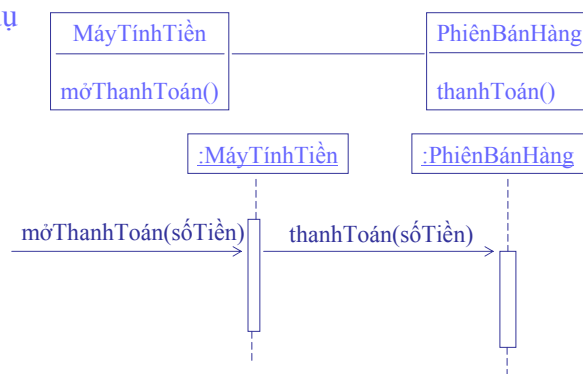
Phân tích và thiết kế hệ thống hướng đối tượng

288



Biểu đồ tuần tự

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



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

289



Biểu đồ tương tác

- Có hai loại biểu đồ tương tác:
 - Biểu đồ tuần tự
 - Biểu đồ cộng tác

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

290



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

291



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.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

292

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 $t < 10$.

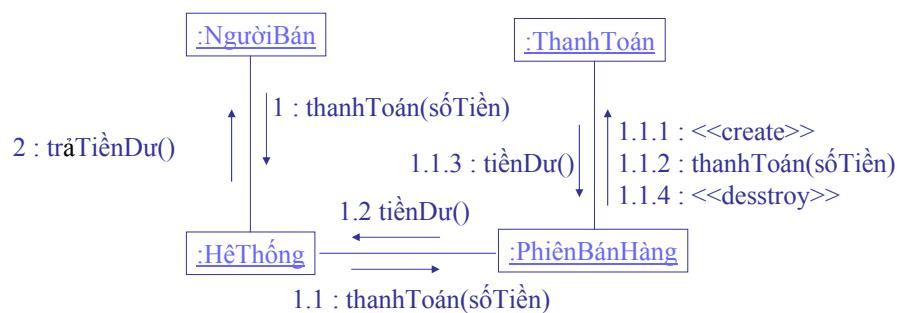
7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

293

Biểu đồ cộng tác

■ Ví dụ biểu đồ cộng tác



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

294



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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

295



Biểu đồ gói (10)

7-Aug-07

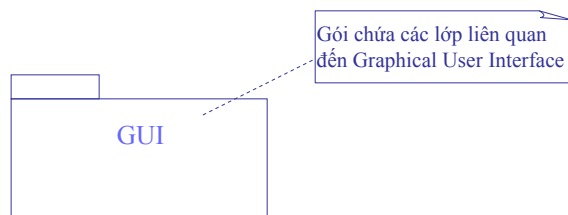
Phân tích và thiết kế hệ thống hướng đối tượng

296



Biểu đồ gói

- Một gói (package) nhóm các lớp liên quan lại với nhau (tương tự khái niệm thư mục trong hệ điều hành)
- Kí hiệu



7-Aug-07

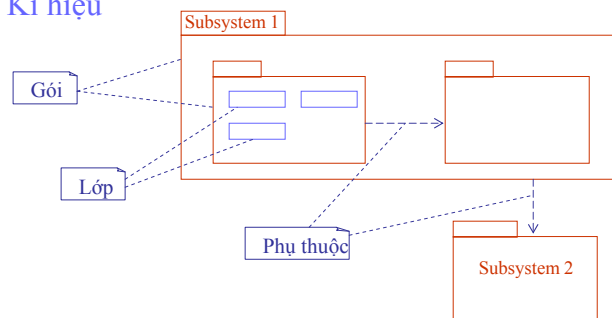
Phân tích và thiết kế hệ thống hướng đối tượng

297



Biểu đồ gói

- Biểu đồ gói (package diagram) mô tả các các gói và quan hệ giữa chúng
- Kí hiệu



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

298



Biểu đồ gói

- Tạo sao phải “gói” ?
 - Chia các hệ thống lớn thành hệ thống con để dễ quản lý
 - Cho phép phát triển lặp đồng thời
- Nguyên tắc đóng gói
 - Sự cố kết (cohesion) cao: tính cố kết thể hiện các lớp trong cùng một gói phải hợp tác cùng nhau, hướng tới cùng mục đích
 - Sự móc nối (coupling) kém: là sự ràng buộc ảnh hưởng nhau giữa các gói. Sự móc nối giữa các gói càng lỏng lẻo càng tốt.

7-Aug-07

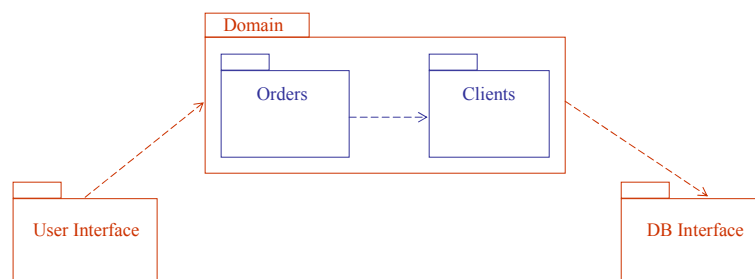
Phân tích và thiết kế hệ thống hướng đối tượng

299



Biểu đồ gói

- Ví dụ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

300



Biểu đồ triển khai (11)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

301



Biểu đồ triển khai

- **Biểu đồ triển khai** (deployment diagram) chỉ ra sự tương ứng giữa các phần tử phần mềm và các phần tử phần cứng của hệ thống
- Biểu đồ triển khai gồm:
 - Các **nút** (node), biểu diễn các đơn vị xử lý
 - Các **thành phần** (component) biểu diễn các phần tử phần mềm
 - Các **kết nối** (connection) chỉ ra kết nối vật lý giữa các nút

7-Aug-07

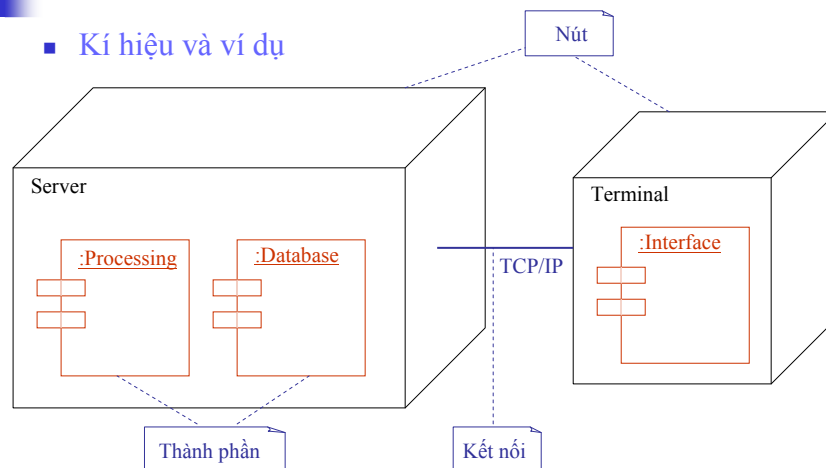
Phân tích và thiết kế hệ thống hướng đối tượng

302



Biểu đồ triển khai

■ Kí hiệu và ví dụ



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

303



Mã hóa (12)

Nguyễn Thanh Bình

Khoa Công Nghệ Thông Tin

Trường Đại học Bách khoa, Đại học Đà Nẵng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

304



Mã hóa

- Lập trình hướng đối tượng
- Sinh mã
 - Chuyển các mô hình thiết kế sang mã chương trình hướng đối tượng

7-Aug-07

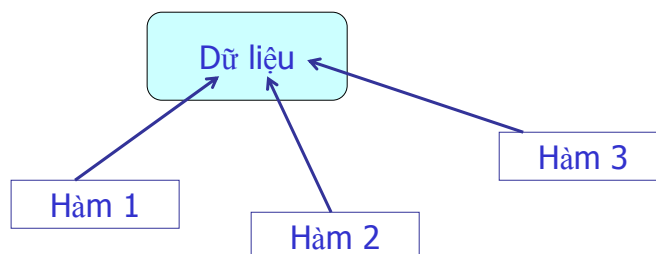
Phân tích và thiết kế hệ thống hướng đối tượng

305



Lập trình hướng đối tượng

- Lập trình thủ tục: C, Pascal, ...



7-Aug-07

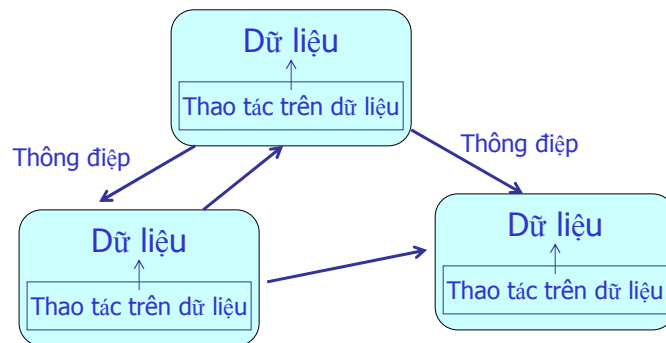
Phân tích và thiết kế hệ thống hướng đối tượng

306



Lập trình hướng đối tượng

- Lập trình hướng đối tượng: C++, Java, ...



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

307



Lập trình hướng đối tượng

- Lập trình hướng đối tượng: C++, Java
 - Định nghĩa lớp
 - Thuộc tính
 - Phương thức
 - Constructor và Destructor
 - Thừa kế
 - Lớp trừu tượng và giao diện

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

308



Lập trình hướng đối tượng

■ Định nghĩa lớp trong C++

```
class User {  
[private:]  
    string name;  
    int age;  
  
public:  
    User(string n, int a):name(n), age(a) {}  
    string getName() {return name;}  
    int  getAge() {return age;}  
    void setName(string n) {name = n;}  
    void setName(int a) {age = a;}  
    void print();  
    ...  
};  
  
void      User::print(){  
    cout << "name: " << name << " age: " << age << endl;  
}  
  
...  
User u(" Nguyen Van A ", 35);  
User* p = new User( " Nguyen Van A ", 35 ); /*phải sử dụng delete p;*/
```

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

309



Lập trình hướng đối tượng

■ Định nghĩa lớp trong Java

```
class User {  
private    String name;  
private    int age;  
  
public     User(String n, int a) {name = n; age = a;}  
public     String getName() {return name;}  
public     int  getAge() {return age;}  
public     void setName(String n) {name = n;}  
public     void setName(int a) {age = a;}  
public     void print(){  
            System.out.println( "name: " + name + " age: " + age );  
        }  
        ...  
}  
  
...  
User u = new User ("Nguyen Van A", 35 );
```

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

310



Lập trình hướng đối tượng

■ Constructor và Destructor trong C++

- Constructor là phương thức khởi gán giá trị cho các thuộc tính
- Destructor là phương thức giải phóng bộ nhớ
- Destructor là bắt buộc khi có thuộc tính là con trỏ và có sự cấp phát bộ nhớ

```
class X {};  
class Y {  
    X* p;  
    public:  
        Y( X* q ) : p( new X(*q) ) {}  
        ~Y(){ delete p; }  
};
```

Constructor

Destructor

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

311



Lập trình hướng đối tượng

■ Constructor và Destructor trong Java

- Có constructor nhưng không có destructor
- Không có con trỏ
- Không có cấp phát bộ nhớ

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

312



Lập trình hướng đối tượng

■ Thừa kế trong C++

```
class StudentUser : public User {  
    string schoolEnrolled;  
    public: StudentUser(string n, int a, string school) : User(n, a){  
        schoolEnrolled = school;  
    }  
    void print() {  
        User::print();  
        cout << "School Enrolled: " << schoolEnrolled << endl;  
    }  
};
```

■ C++ cho phép đa thừa kế

```
class StudentUser : public User, public Student { ... };
```

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

313



Lập trình hướng đối tượng

■ Thừa kế trong Java

```
class StudentUser extends User {  
    private String schoolEnrolled;  
    public StudentUser( String n, int a, String school ) {  
        super(n, a);  
        schoolEnrolled = school;  
    }  
  
    public void print() {  
        super.print();  
        System.out.print( " School: " + schoolEnrolled );  
    }  
}
```

■ Java không cho phép đa thừa kế

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

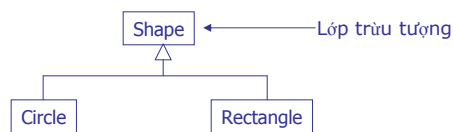
314



Lập trình hướng đối tượng

■ Lớp trừu tượng và giao diện

- C++ và Java đều cho phép định nghĩa lớp trừu tượng



- Java có khái niệm giao diện, C++ thì không

- Bởi vì C++ cho phép đa thừa kế
- Java không cho phép đa thừa kế, nhưng cho phép thừa kế từ một lớp và cài đặt nhiều giao diện
- Giao diện gần giống lớp trừu tượng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

315



Lập trình hướng đối tượng

■ Lớp trừu tượng trong C++

```
class Shape {
public:
    virtual double area( ) = 0;
    virtual double circumference() = 0;
    ....
}
```

■ Lớp trừu tượng trong Java

```
abstract class Shape {
    abstract public double area( );
    abstract public double circumference();
    ....
}
```

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

316



Lập trình hướng đối tượng

■ Giao diện trong Java

```
interface MyInterface {  
    public double area( );  
    public double circumference();  
    ...  
}  
  
class MyClass implements MyInterface {  
    // cài đặt các phương thức khai báo trong MyInterface  
}
```

■ Thay thế đa thừa kế

```
class MyClass extends SuperClass implements MyInterface1, MyInterface2 {  
    // cài đặt các phương thức khai báo trong MyInterface1 , MyInterface2  
}
```

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

317



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

7-Aug-07

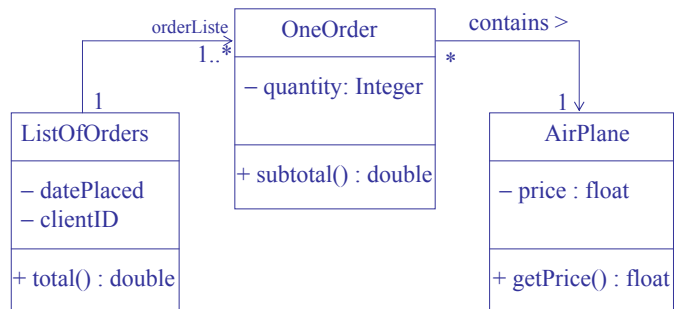
Phân tích và thiết kế hệ thống hướng đối tượng

318



Sinh mã

■ Ví dụ: biểu đồ lớp



7-Aug-07

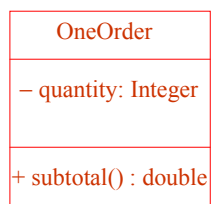
Phân tích và thiết kế hệ thống hướng đối tượng

319



Sinh mã

■ Mã lớp **OneOrder**



```
public class OneOrder
{
    public double subtotal()
    {
    }
    private int quantity;
}
```

7-Aug-07

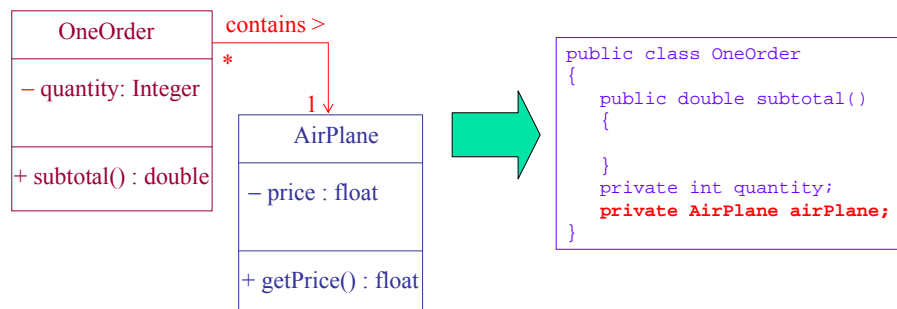
Phân tích và thiết kế hệ thống hướng đối tượng

320



Sinh mã

■ Mã lớp OneOrder



7-Aug-07

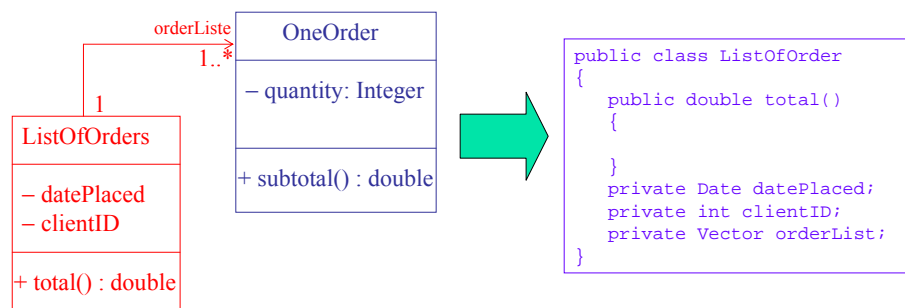
Phân tích và thiết kế hệ thống hướng đối tượng

321



Sinh mã

■ Mã lớp ListOfOrders



7-Aug-07

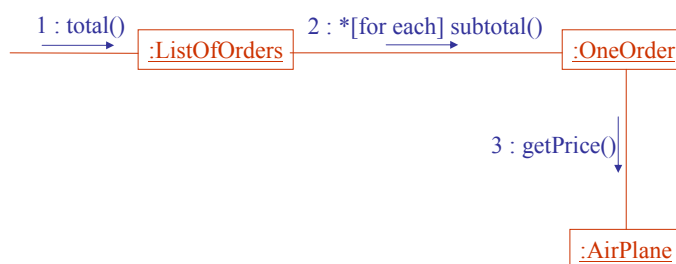
Phân tích và thiết kế hệ thống hướng đối tượng

322



Sinh mã

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



7-Aug-07

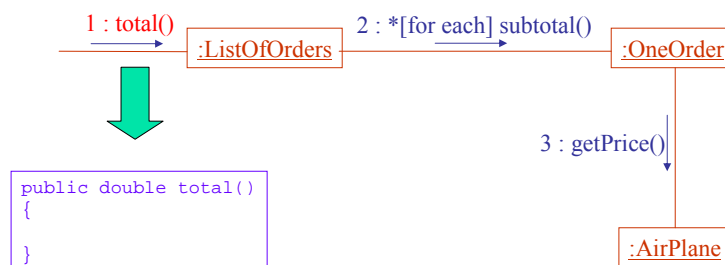
Phân tích và thiết kế hệ thống hướng đối tượng

323



Sinh mã

- Mã phương thức **total()**



7-Aug-07

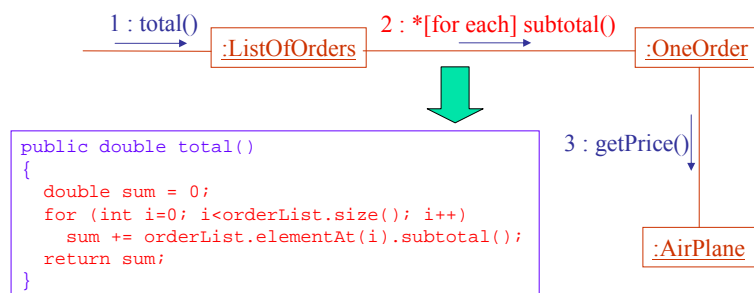
Phân tích và thiết kế hệ thống hướng đối tượng

324



Sinh mã

■ Mã phương thức **total()**



7-Aug-07

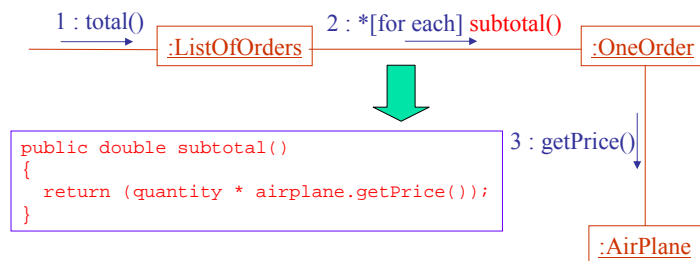
Phân tích và thiết kế hệ thống hướng đối tượng

325



Sinh mã

■ Mã phương thức **subTotal()**



7-Aug-07

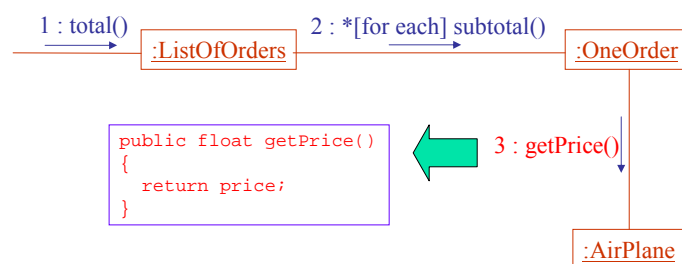
Phân tích và thiết kế hệ thống hướng đối tượng

326



Sinh mã

■ Mã phương thức `getPrice()`



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

327



Công cụ

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

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

328



Ứng dụng (13)

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

329



Vấn đề

- Vấn đề rất đơn giản, nhằm mô tả các bước phân tích và thiết kế một cách chi tiết
- Trích dẫn từ tài liệu “Applying UML and Patterns” của Claig Larman
- Xây dựng chương trình trò chơi súc sắc: người chơi tung hai con súc sắc 10 lần, nếu mỗi lần tung tổng số hai con súc sắc là 7, người chơi ghi được 10 điểm. Kết thúc cuộc chơi, số điểm của người chơi được ghi vào bảng điểm thành tích của những người chơi.

7-Aug-07

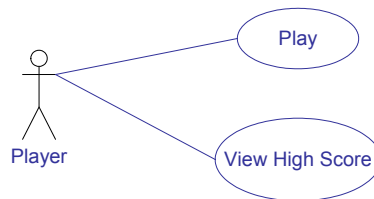
Phân tích và thiết kế hệ thống hướng đối tượng

330



Phân tích và thiết kế

- Xác định các chức năng cần thiết
 - Biểu đồ ca sử dụng



Play: người chơi tung hai con súc sắc 10 lần, nếu mỗi lần tung tổng số hai con súc sắc là 7, người chơi ghi được 10 điểm.

View High Score: người chơi xem số điểm của mình.

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

331



Phân tích và thiết kế

- Biểu đồ hoạt động
 - Mô tả các xử lý chung và các giao tiếp giữa người sử dụng và hệ thống
 - Một số các hoạt động được liên kết với các mẫu giao diện sử dụng

7-Aug-07

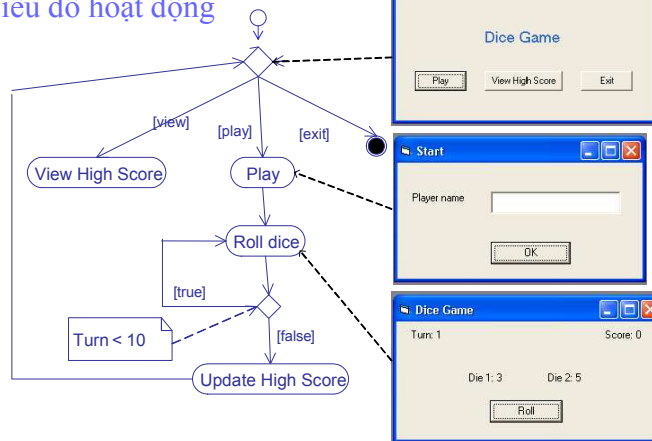
Phân tích và thiết kế hệ thống hướng đối tượng

332



Phân tích và thiết kế

■ Biểu đồ hoạt động



7-Aug-07

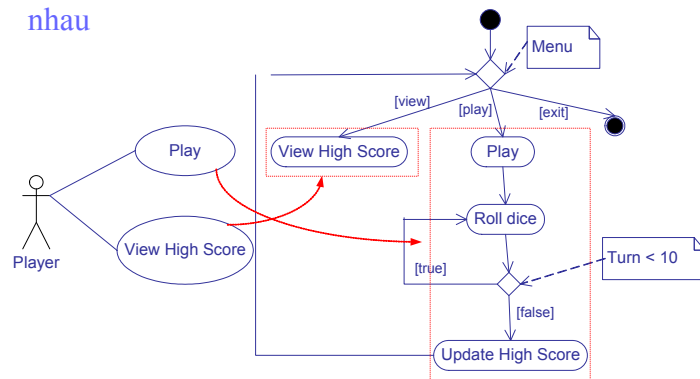
Phân tích và thiết kế hệ thống hướng đối tượng

333



Phân tích và thiết kế

■ Giữa các ca sử dụng và các hoạt động có mối quan hệ với nhau



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

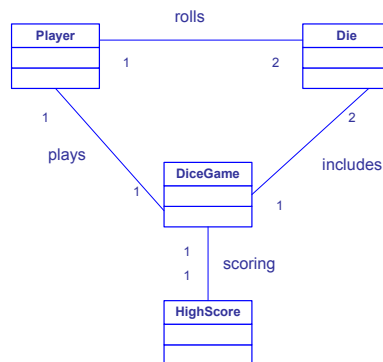
334



Phân tích và thiết kế

■ Mô hình khái niệm

- Các khái niệm: Player, Die, DiceGame, HighScore



7-Aug-07

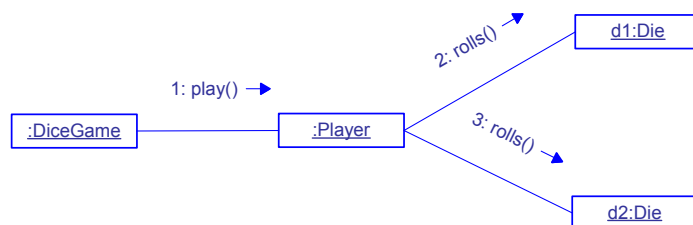
Phân tích và thiết kế hệ thống hướng đối tượng

335



Phân tích và thiết kế

■ Biểu đồ cộng tác



7-Aug-07

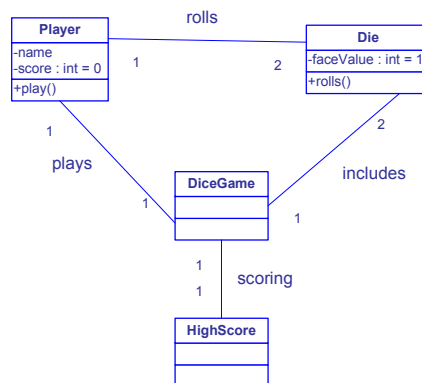
Phân tích và thiết kế hệ thống hướng đối tượng

336



Phân tích và thiết kế

- Mô hình khái niệm chi tiết thành biểu đồ lớp



7-Aug-07

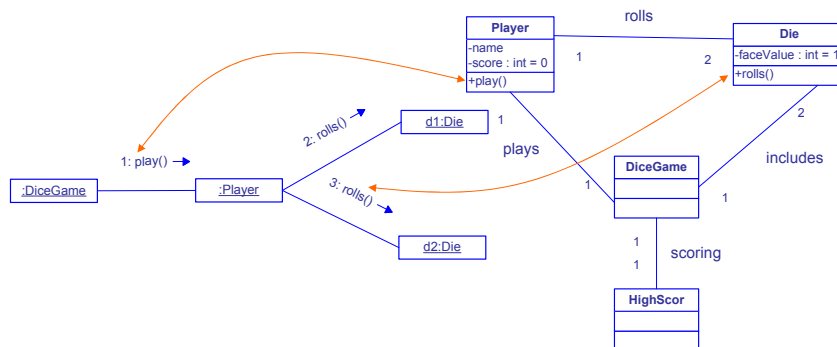
Phân tích và thiết kế hệ thống hướng đối tượng

337



Phân tích và thiết kế

- Quan hệ giữa biểu đồ cộng tác và biểu đồ lớp



7-Aug-07

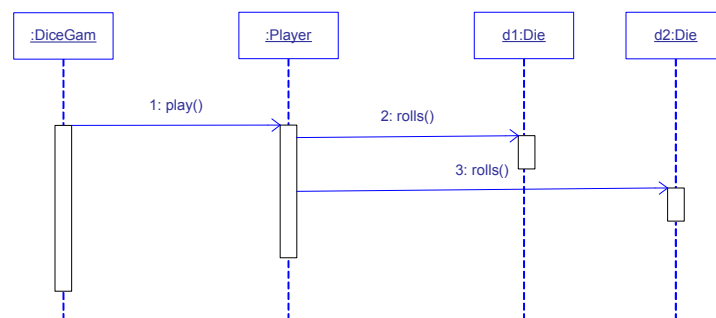
Phân tích và thiết kế hệ thống hướng đối tượng

338



Phân tích và thiết kế

■ Biểu đồ tuần tự



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

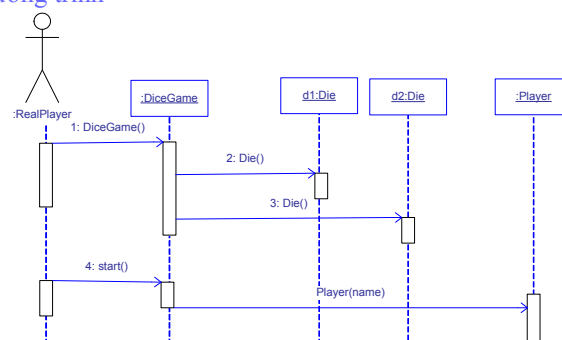
339



Phân tích và thiết kế

■ Biểu đồ tuần tự

- Chúng ta mô tả chi tiết hơn việc tạo ra các đối tượng khi bắt đầu chương trình



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

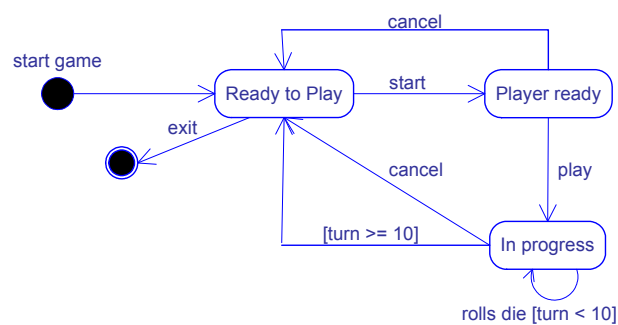
340



Phân tích và thiết kế

■ Biểu đồ trạng thái

- Mô tả các trạng thái của một ván chơi



7-Aug-07

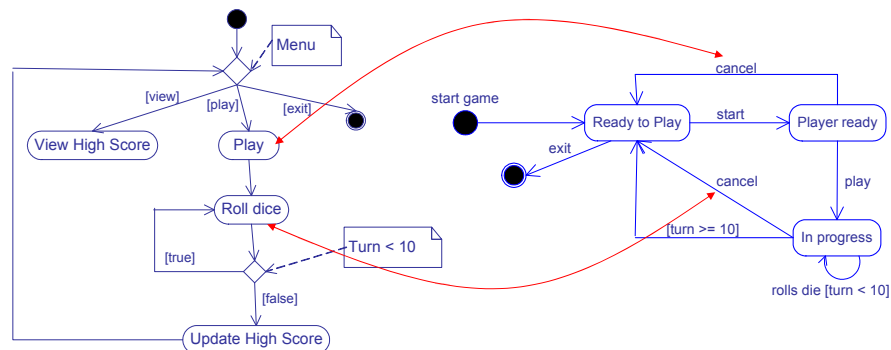
Phân tích và thiết kế hệ thống hướng đối tượng

341



Phân tích và thiết kế

- Sự không tương thích giữa biểu đồ trạng thái và biểu đồ hoạt động



7-Aug-07

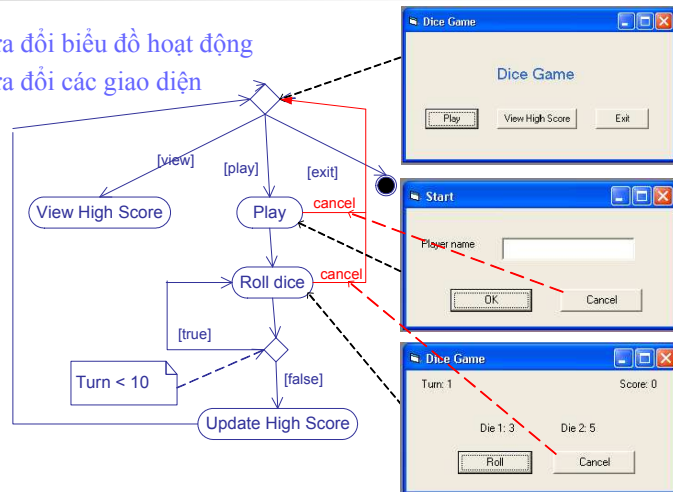
Phân tích và thiết kế hệ thống hướng đối tượng

342



Phân tích và thiết kế

- Sửa đổi biểu đồ hoạt động
- Sửa đổi các giao diện



7-Aug-07

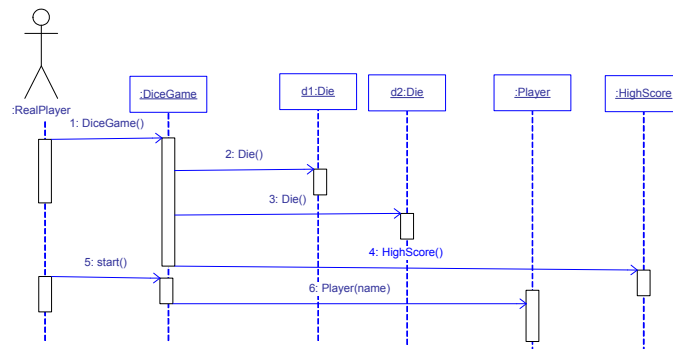
Phân tích và thiết kế hệ thống hướng đối tượng

343



Phân tích và thiết kế

- Trong các bước thiết kế, bảng điểm (High Score) chưa được xử lý: tạo ra và cập nhật High Score, cần bổ sung



7-Aug-07

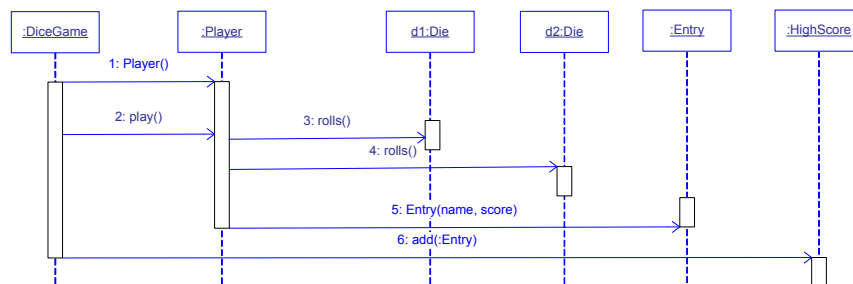
Phân tích và thiết kế hệ thống hướng đối tượng

344



Phân tích và thiết kế

- Trong các bước thiết kế, bảng điểm (High Score) chưa được xử lý: tạo ra và cập nhật High Score, cần bổ sung



7-Aug-07

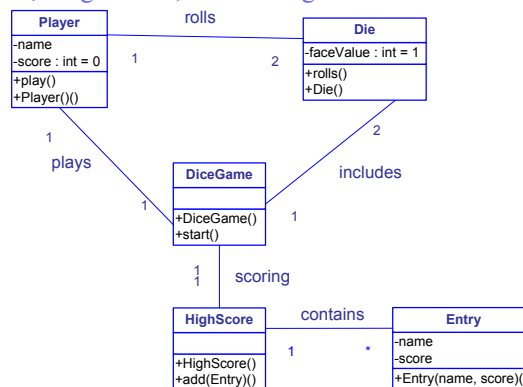
Phân tích và thiết kế hệ thống hướng đối tượng

345



Phân tích và thiết kế

- Trong các bước thiết kế, bảng điểm (High Score) chưa được xử lý: tạo ra và cập nhật High Score, cần bổ sung



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

346



Phân tích và thiết kế

- Kiến trúc tổng quan
 - Thiết kế kiến trúc 3 tầng/lớp

Trình diễn (Presentation)



Ứng dụng (Application)



CSDL (Data Base)



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

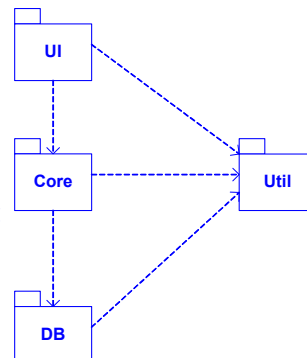
347



Phân tích và thiết kế

- Biểu đồ các gói tương ứng với kiến trúc

- UI: ứng với tầng trình diễn
- Core: ứng với tầng ứng dụng
- DB: ứng với tầng CSDL
- Util: chứa các dịch vụ dùng chung



7-Aug-07

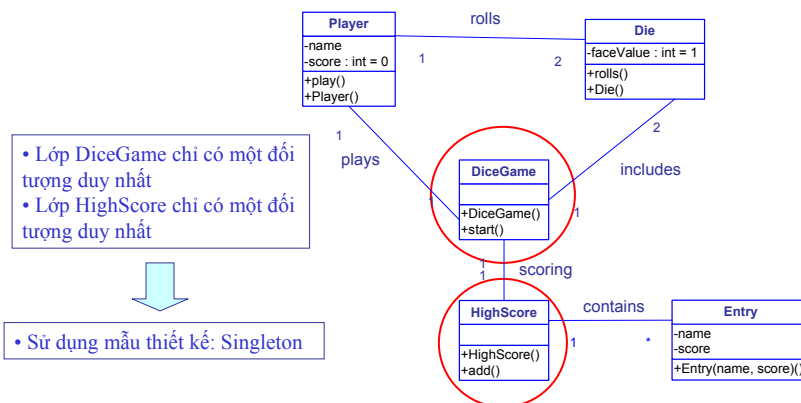
Phân tích và thiết kế hệ thống hướng đối tượng

348



Phân tích và thiết kế

- Sử dụng các Design Pattern để cải tiến biểu đồ lớp trong gói “Core”



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

349



Phân tích và thiết kế

- Nhắc lại mẫu thiết kế Singleton



- Áp dụng cho các lớp DiceGame và HighScore
- Thêm vào một số các phương thức cho các lớp

7-Aug-07

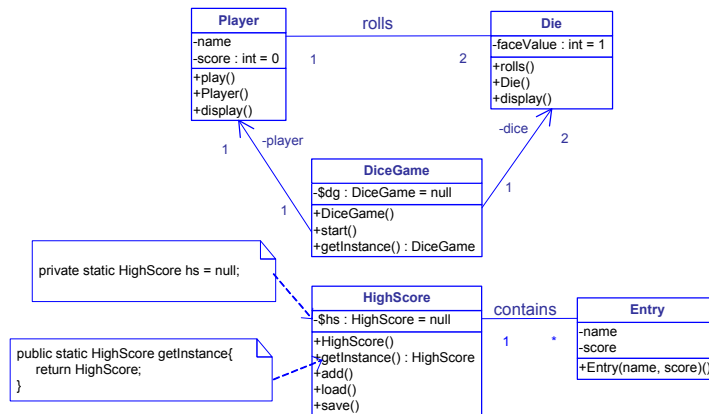
Phân tích và thiết kế hệ thống hướng đối tượng

350



Phân tích và thiết kế

- Sử dụng mẫu thiết kế **Singleton** để cải tiến biểu đồ lớp gói “Core”



7-Aug-07

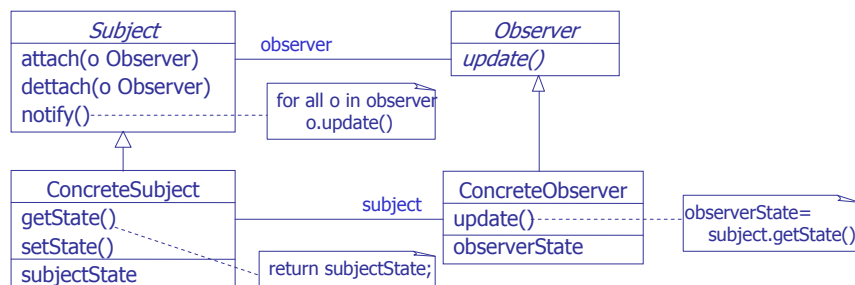
Phân tích và thiết kế hệ thống hướng đối tượng

351



Phân tích và thiết kế

- Sử dụng mẫu thiết kế **Observer** để cải tiến biểu đồ lớp gói “Core”
- Nhắc lại mẫu **Observer**



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

352



Phân tích và thiết kế

- Sử dụng mẫu thiết kế **Observer** để cải tiến biểu đồ lớp gói “Core”
 - Các lớp **Die** và **Player** là các *ConcreteSubject*
 - Sự hiển thị các đối tượng của **Die** và **Player** là các *ConcreteObserver*
 - Để hiển thị các đối tượng của **Die** và **Player**, sử dụng các đối tượng *Jpanel* của Java, tức là một vùng của cửa sổ

7-Aug-07

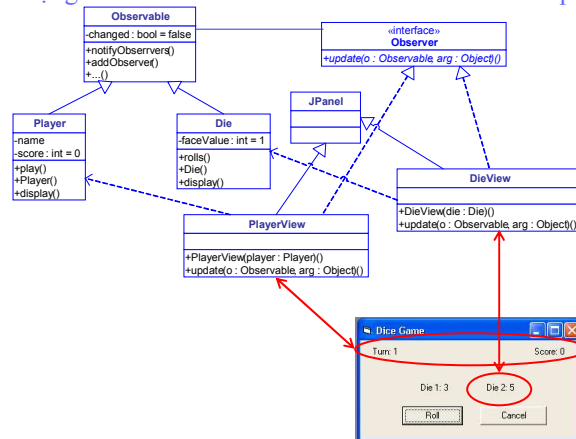
Phân tích và thiết kế hệ thống hướng đối tượng

353



Phân tích và thiết kế

- Sử dụng mẫu thiết kế **Observer** để cải tiến biểu đồ lớp gói “Core”



7-Aug-07

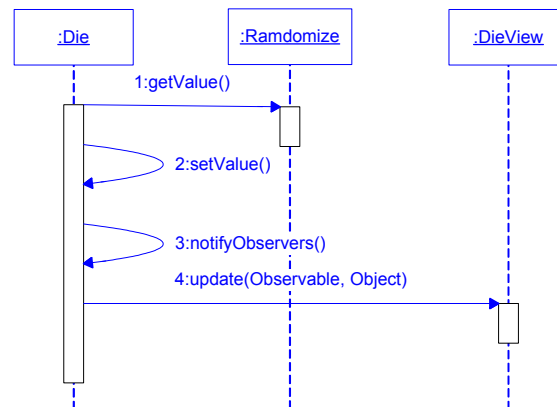
Phân tích và thiết kế hệ thống hướng đối tượng

354



Phân tích và thiết kế

- Biểu đồ tuần tự mô tả sự tương tác



7-Aug-07

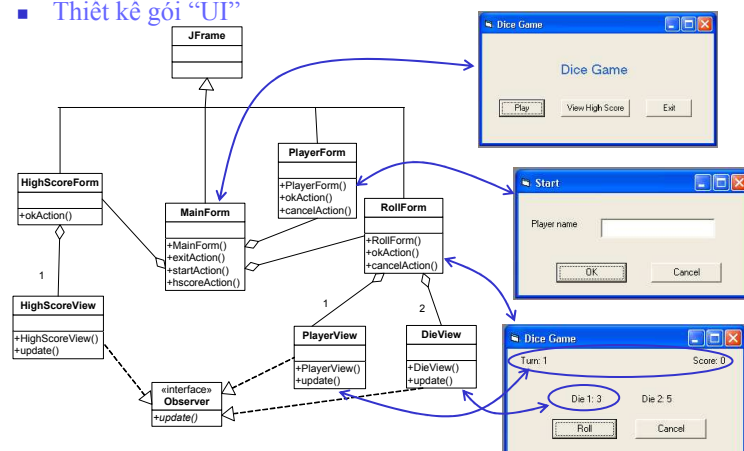
Phân tích và thiết kế hệ thống hướng đối tượng

355



Phân tích và thiết kế

- Thiết kế gọi “UI”



7-Aug-07

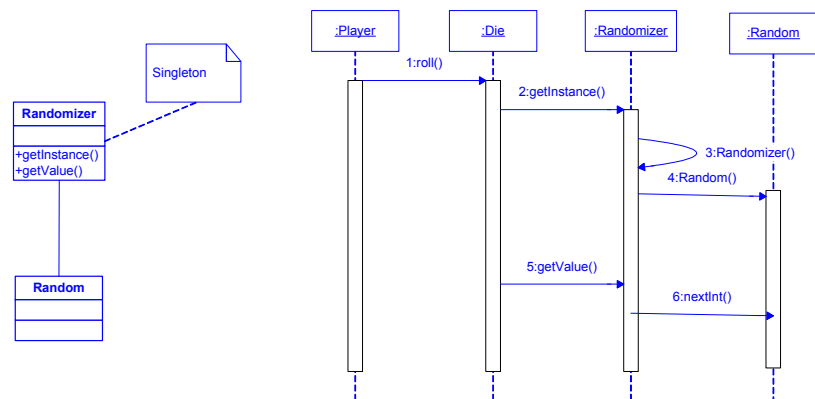
Phân tích và thiết kế hệ thống hướng đối tượng

356



Phân tích và thiết kế

- Thiết kế gói “Util”: chứa lớp Randomizer sử dụng lớp có sẵn Random



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

357



Phân tích và thiết kế

- Thiết kế gói “DB”
 - Làm sao để tạo ra sự độc lập giữa gói “Core” và gói “DB”, nghĩa là có thể sử dụng nhiều loại lưu trữ dữ liệu khác nhau
 - Tập tin truy cập tuần tự
 - CSDL thông qua JDBC, ...
 - Sử dụng mẫu thiết kế Factory Method

7-Aug-07

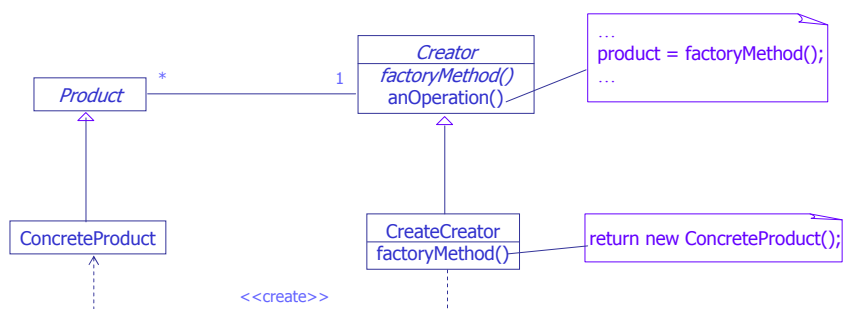
Phân tích và thiết kế hệ thống hướng đối tượng

358



Phân tích và thiết kế

- Thiết kế gói “DB”
 - Nhắc lại mẫu thiết kế Factory Method



7-Aug-07

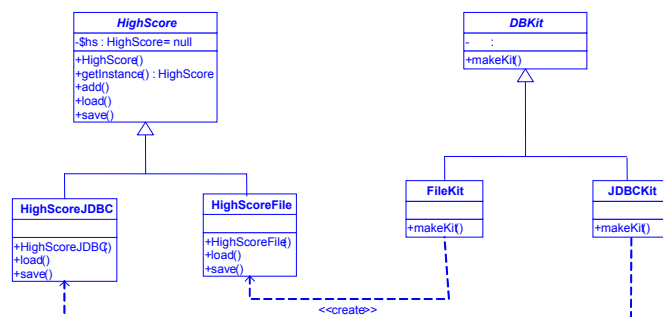
Phân tích và thiết kế hệ thống hướng đối tượng

359



Phân tích và thiết kế

- Thiết kế gói “DB”: biểu đồ lớp



• Lưu ý: Không có quan hệ kết hợp giữa HighScore và DBKit vì HighScore là lớp được thiết kế theo mẫu Singleton

7-Aug-07

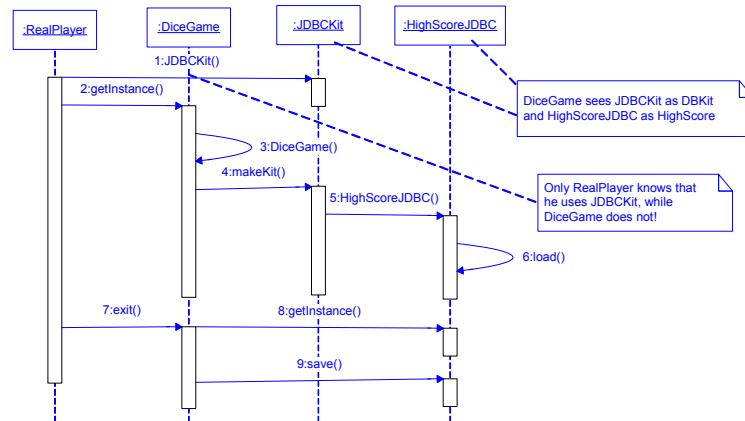
Phân tích và thiết kế hệ thống hướng đối tượng

360



Phân tích và thiết kế

■ Thiết kế gói “DB”: biểu đồ tuần tự



7-Aug-07

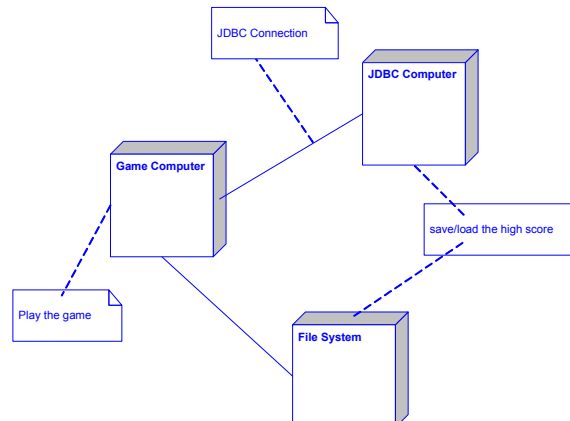
Phân tích và thiết kế hệ thống hướng đối tượng

361



Phân tích và thiết kế

■ Biểu đồ triển khai



7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

362



Phân tích và thiết kế

- Cần tiếp tục hoàn thiện các biểu đồ lớp chi tiết phù hợp với ngôn ngữ sử dụng cài đặt
- Hoàn thiện tiếp các biểu đồ tương tác
- Mã hóa bằng ngôn ngữ lập trình
- ...

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

363



Kết luận (14)



Kết luận

- Các khái niệm cơ bản về hướng đối tượng
- Lịch sử phát triển của các phương pháp phân tích và thiết kế hướng đối tượng
- Các tiến trình phát triển phần mềm

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

365



Kết luận

- Các bước phát triển hệ thống hướng đối tượng
 - Mô hình hoá yêu cầu
 - Mô hình hoá khái niệm
 - Mô hình hoá cấu trúc động
 - Mô hình hoá cấu trúc tĩnh
 - Mã hoá
- Ứng dụng

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

366



Bài tập lớn

- Chia nhóm từ 3 đến 4 sinh viên
- Tự chọn đề tài
 - Ví dụ: hệ thống quản lý thư viện, hệ thống đào tạo, ...
- Phân tích và thiết kế hướng đối tượng đề tài đã chọn
- Kết quả
 - Báo cáo (20-40 trang)
 - Nộp vào ngày thi học kỳ

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

367



Đánh giá

- Bài tập lớn
 - 40 %
- Thi học kỳ
 - 60 %

7-Aug-07

Phân tích và thiết kế hệ thống hướng đối tượng

368