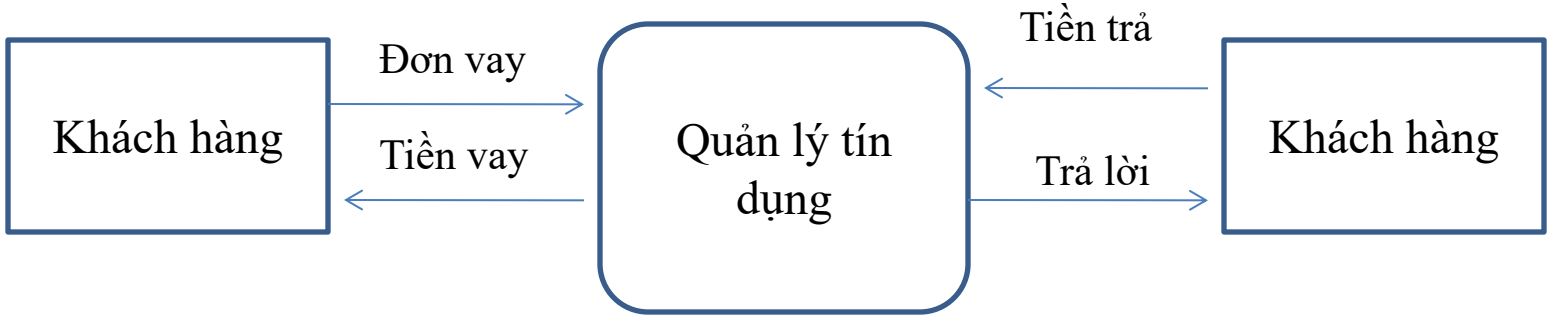
Giai đoạn Phân tích Thiết kế P2

Công nghệ phần mềm

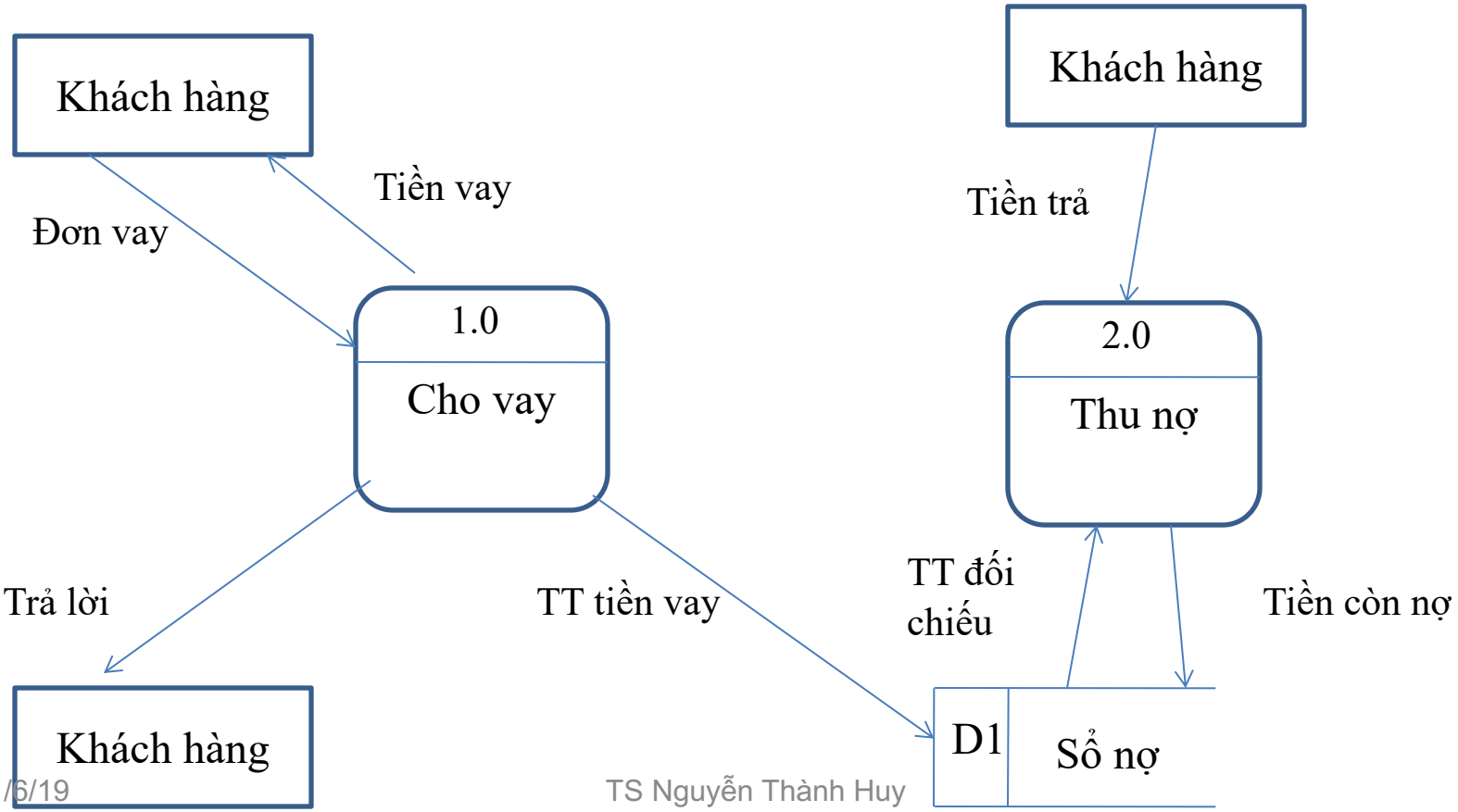
Ví dụ DFD: Quản lý tín dụng

- Sơ đồ ngữ cảnh



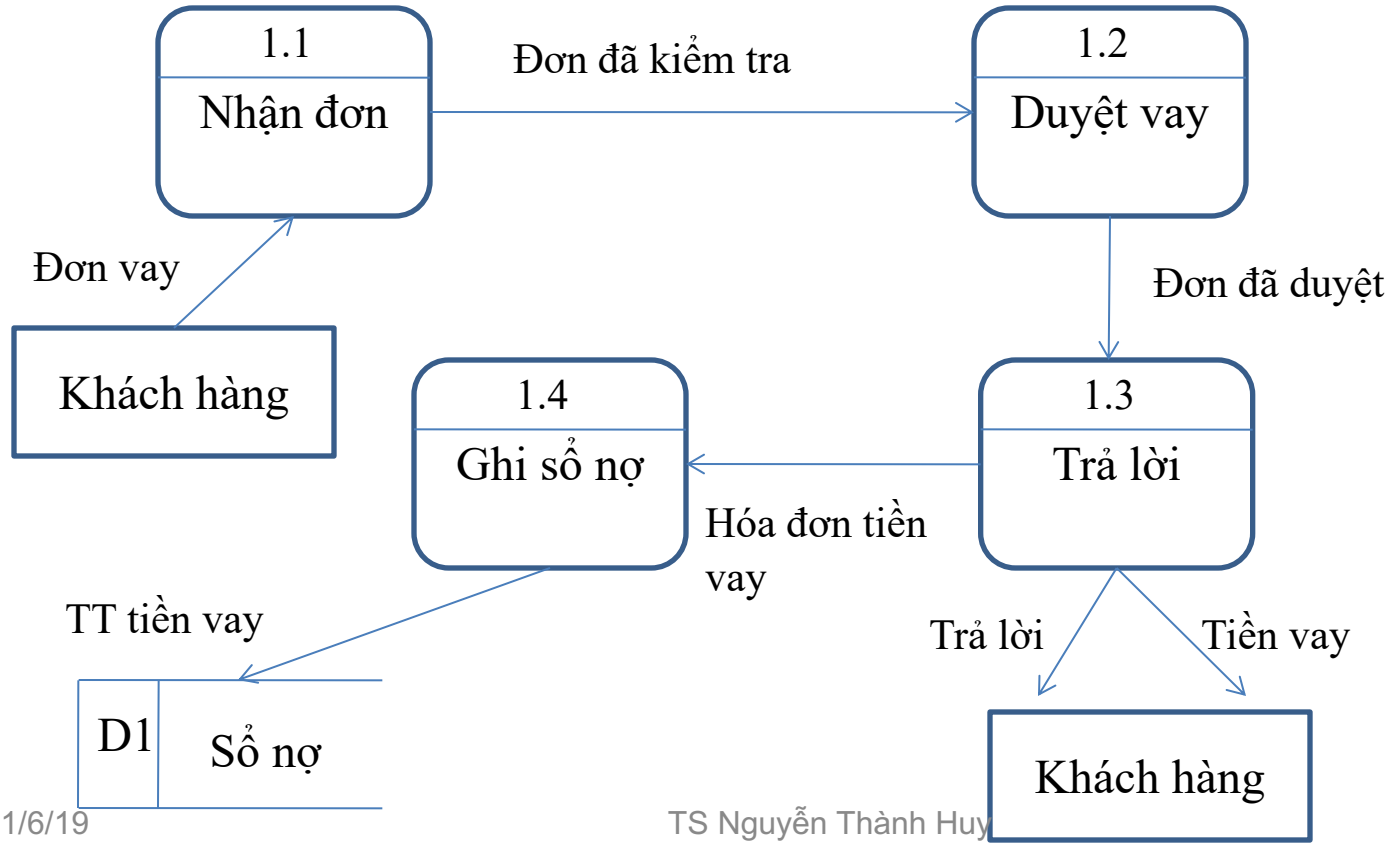
Ví dụ DFD: Quản lý tín dụng

- Sơ đồ mức 0



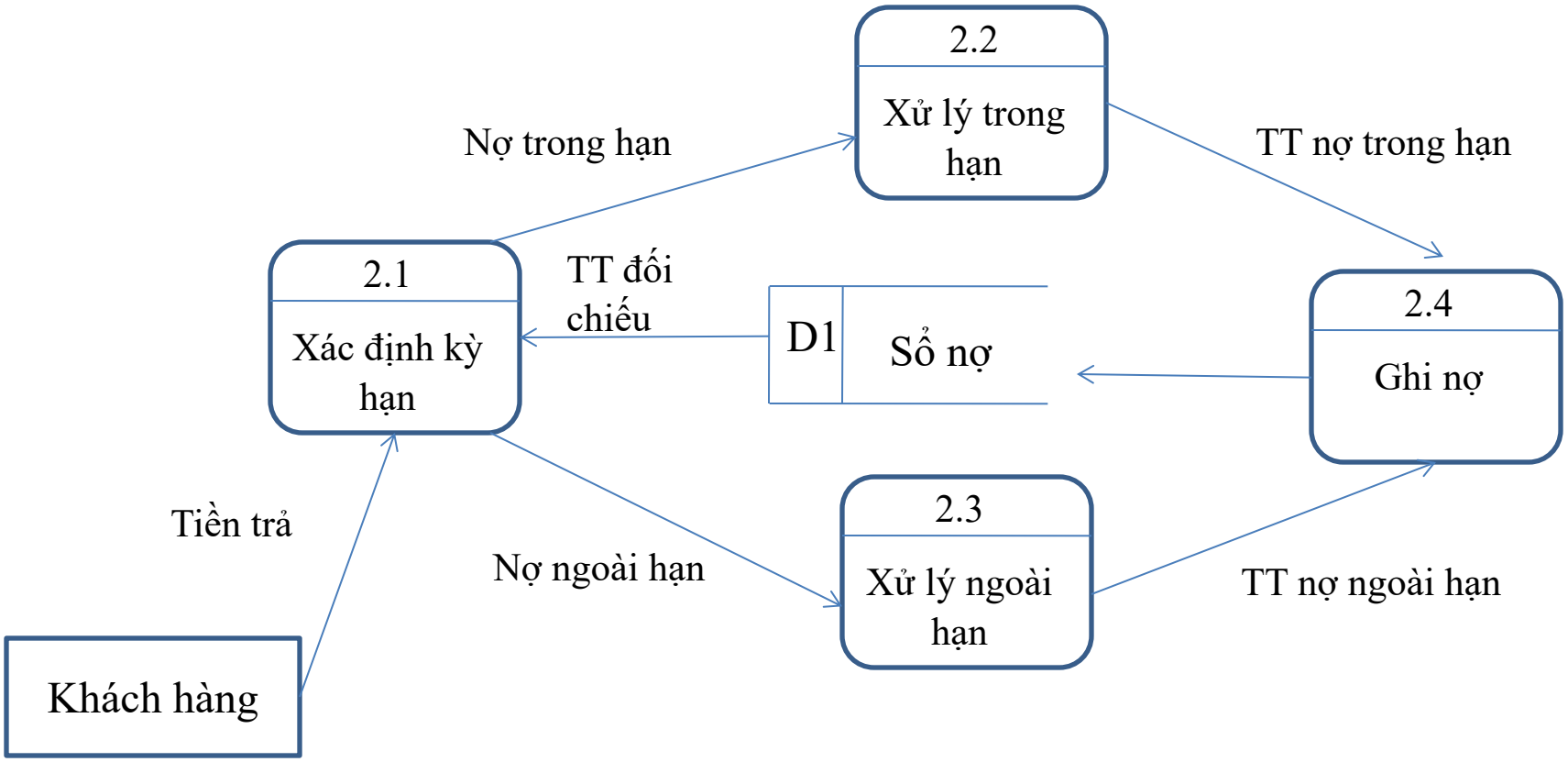
Ví dụ DFD: Quản lý tín dụng

- Sơ đồ mức 1 của 1.0



Ví dụ DFD: Quản lý tín dụng

- Sơ đồ mức 1 của 2.0



+

Ví dụ 2: Quản lý bán hàng

Khi nhận được đơn đặt hàng. Bộ phận kiểm tra tính hợp lệ của đơn hàng Nếu không hợp lệ sẽ trả lại cho khách hàng, ngược lại, sẽ tiến hành kiểm tra mặt hàng trong kho. Nếu đã hết, báo khách hàng chờ hàng và gởi phiếu yêu cầu nhập hàng cho bộ phận nhập hàng. Các đơn hàng được đáp ứng sẽ chuyển sang bộ phận làm phiếu xuất, phiếu xuất được giao cho khách hàng đem đến bộ phận thanh toán. Tại đây sau khi thu tiền, hóa đơn sẽ được lưu trữ tại chổ và giao cho kế toán1 liên, bộ phận giao hàng 1 liên. Căn cứ trên hóa đơn và phiếu xuất, bộ phận giao hàng sẽ giao hàng và hóa đơn cho khách, phiếu xuất sẽ được lưu trữ tại đây.

Ví dụ 2: Quản lý bán hàng

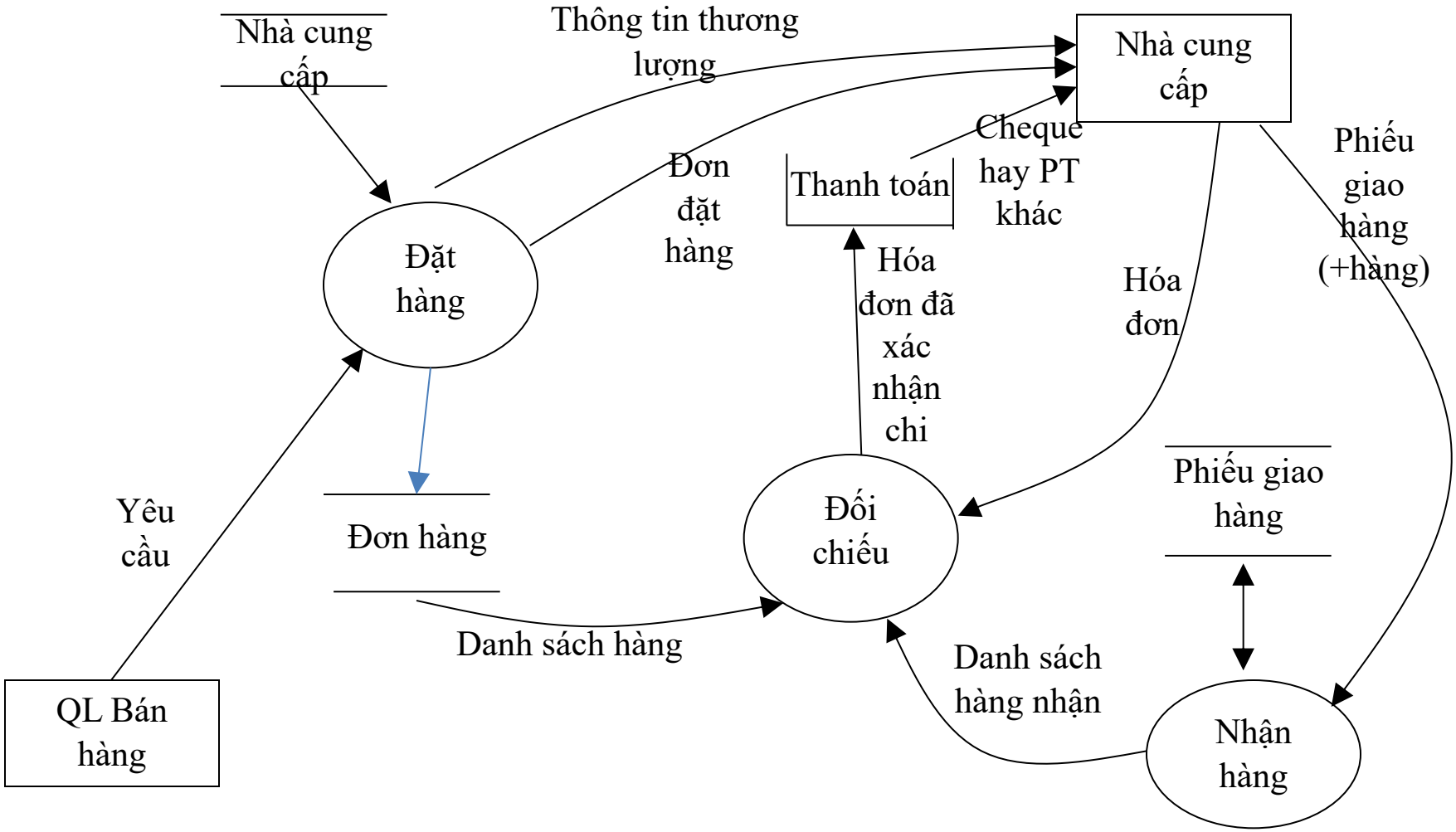
+

Ví dụ 3: Quản lý nhập hàng

Khi nhận được yêu cầu nhập hàng, hệ thống sẽ tham khảo hồ sơ các nhà cung cấp để xác định nhà cung cấp thích hợp. Các thông tin thương lượng và đơn đặt hàng được chuyển đến nhà cung cấp, đồng thời đơn hàng sẽ được lưu trữ (để đối chiếu sau nầy).

Khi nhà cung cấp giao hàng, một hóa đơn, phiếu giao hàng sẽ được gởi đến, hệ thống sẽ đối chiếu danh sách hàng nhận (thực nhận) với đơn hàng và danh sách hàng trên hóa đơn, dựa trên kết quả đối chiếu hóa đơn sẽ được thanh toán (bằng cheque hay các phương thức khác)

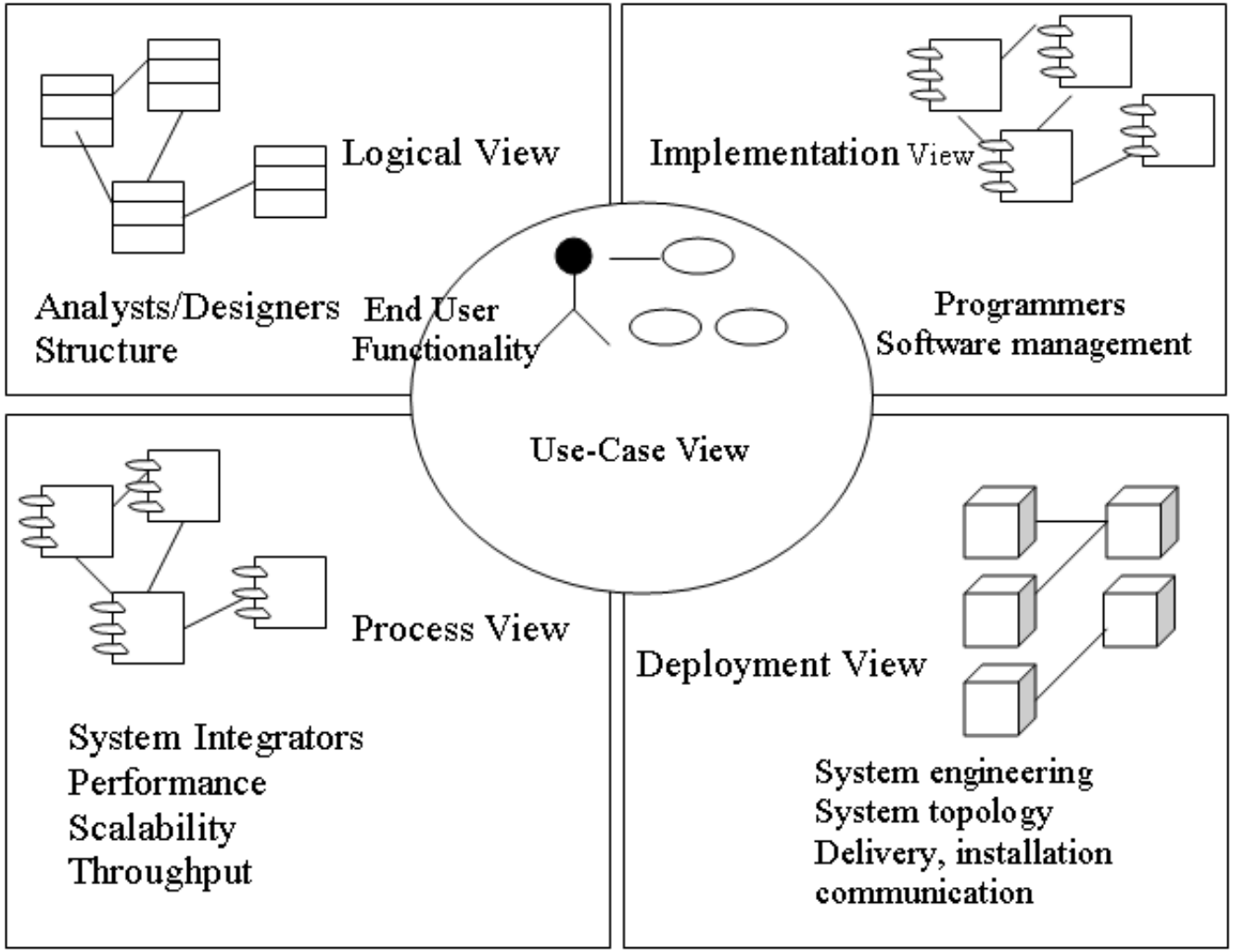
Ví dụ 3: Quản lý nhập hàng



I.2. Phân tích các xử lý của hệ thống theo hướng đối tượng

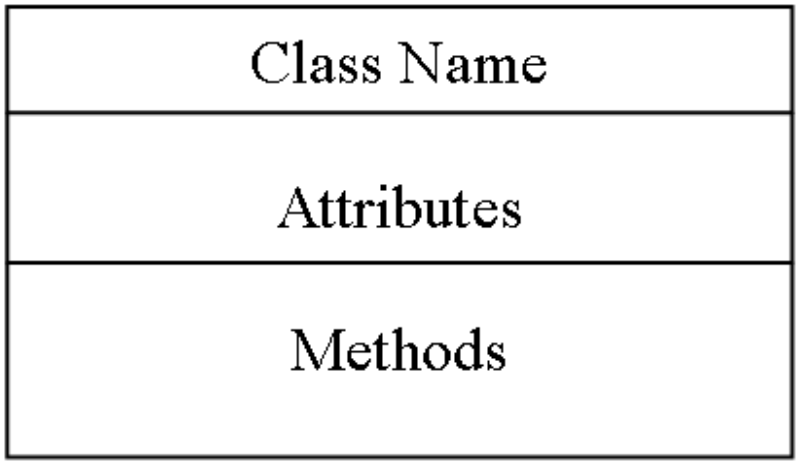
- Ôn tập OOP
- Phân tích Use-case
- Tìm các class từ use-case
- Mô tả các lớp và quan hệ giữa chúng.
- Phân bổ use-case về các class (Sequence diagram)
- Thể hiện khác của sequence diagram (colaboration diagram).

“4+1” góc nhìn

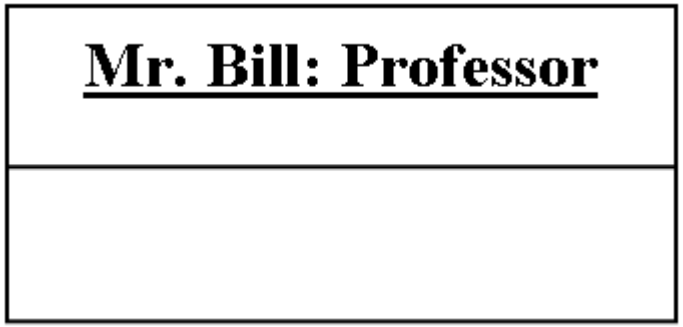

Logical View
Analysts/Designers
Structure
Implementation View
Programmers
Software management
End User
Functionality
Use-Case View
Process View
System Integrators
Performance
Scalability
Throughput
Deployment View
System engineering
System topology
Delivery, installation
communication

Lớp và Đối tượng

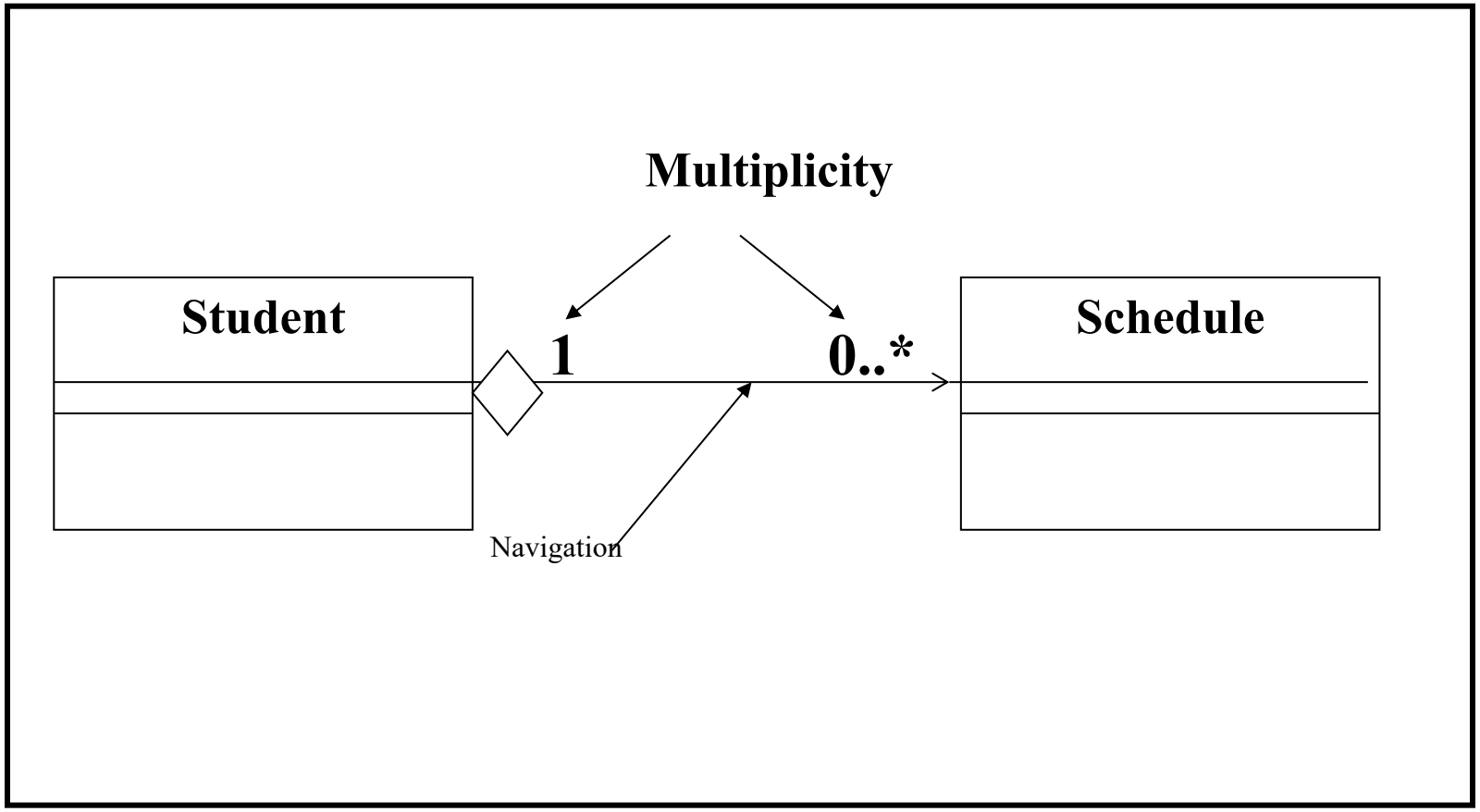
Class



Object

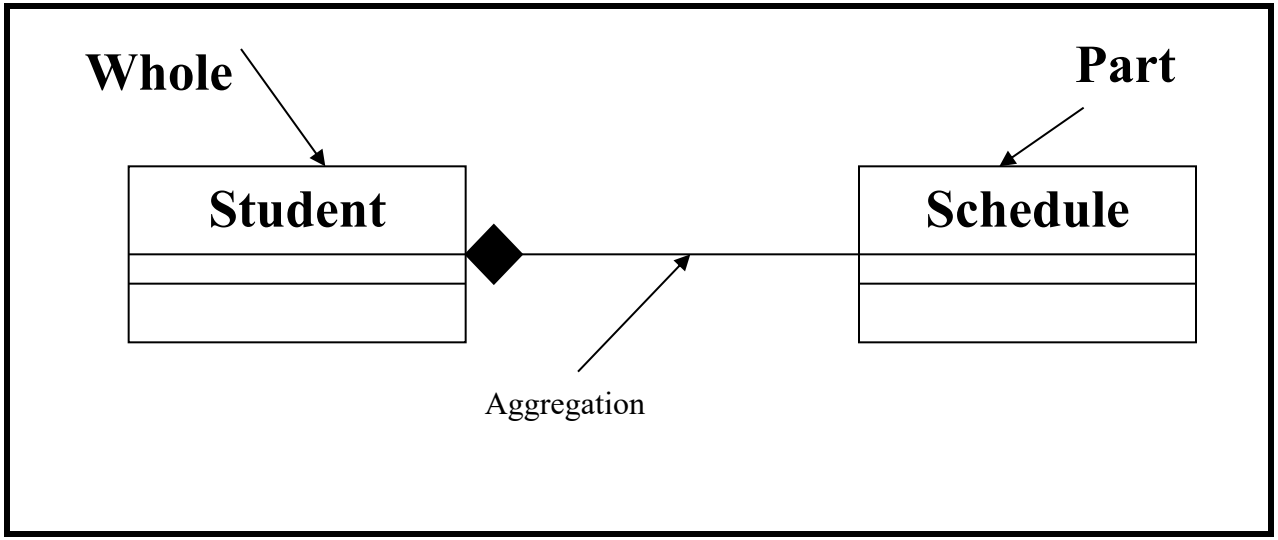


Quan hệ tập hợp



Quan hệ tổng hợp

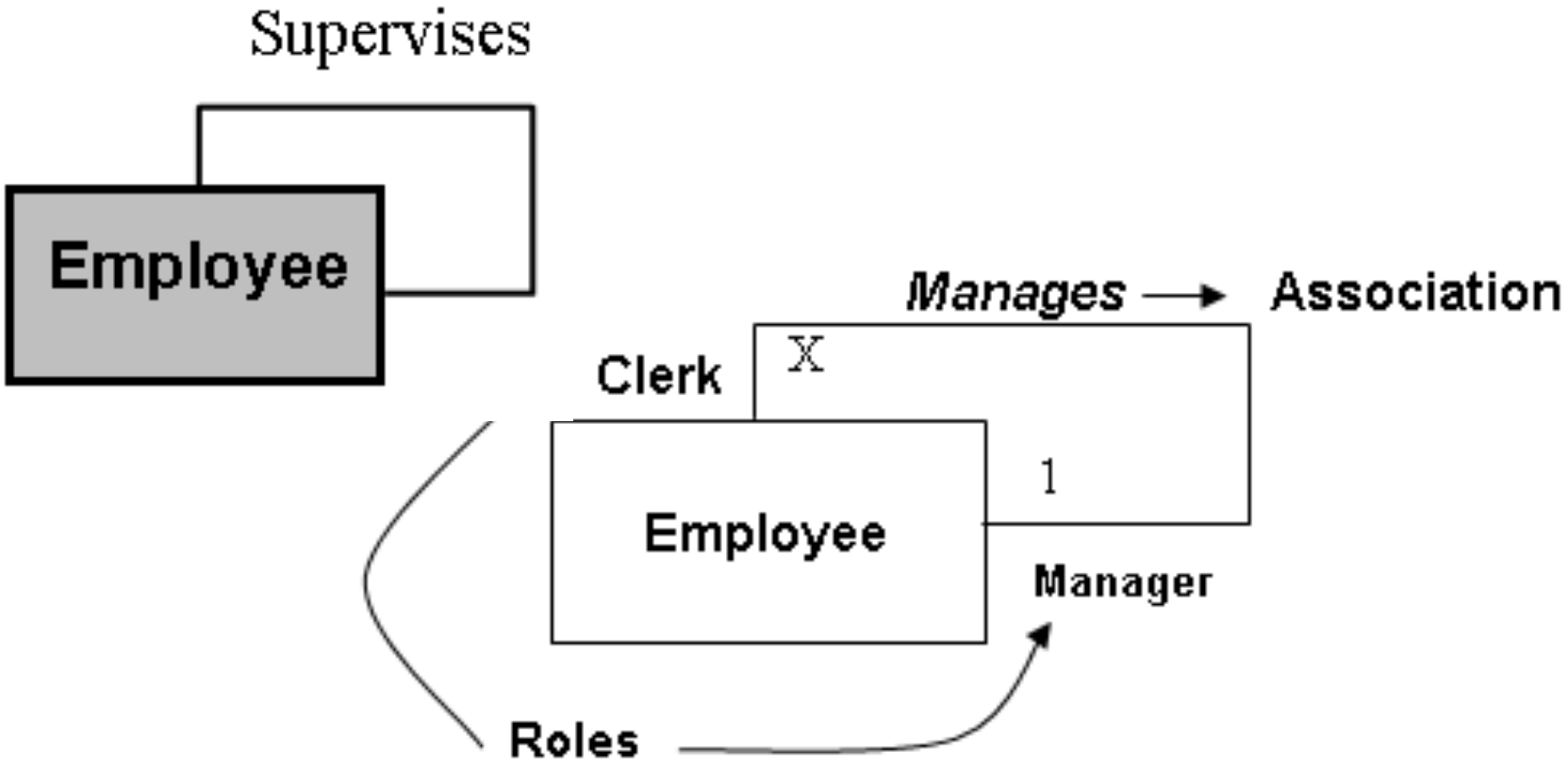
- Quan hệ tổng hợp, lớp Student được tạo thành từ các đối tượng của lớp Schedule.



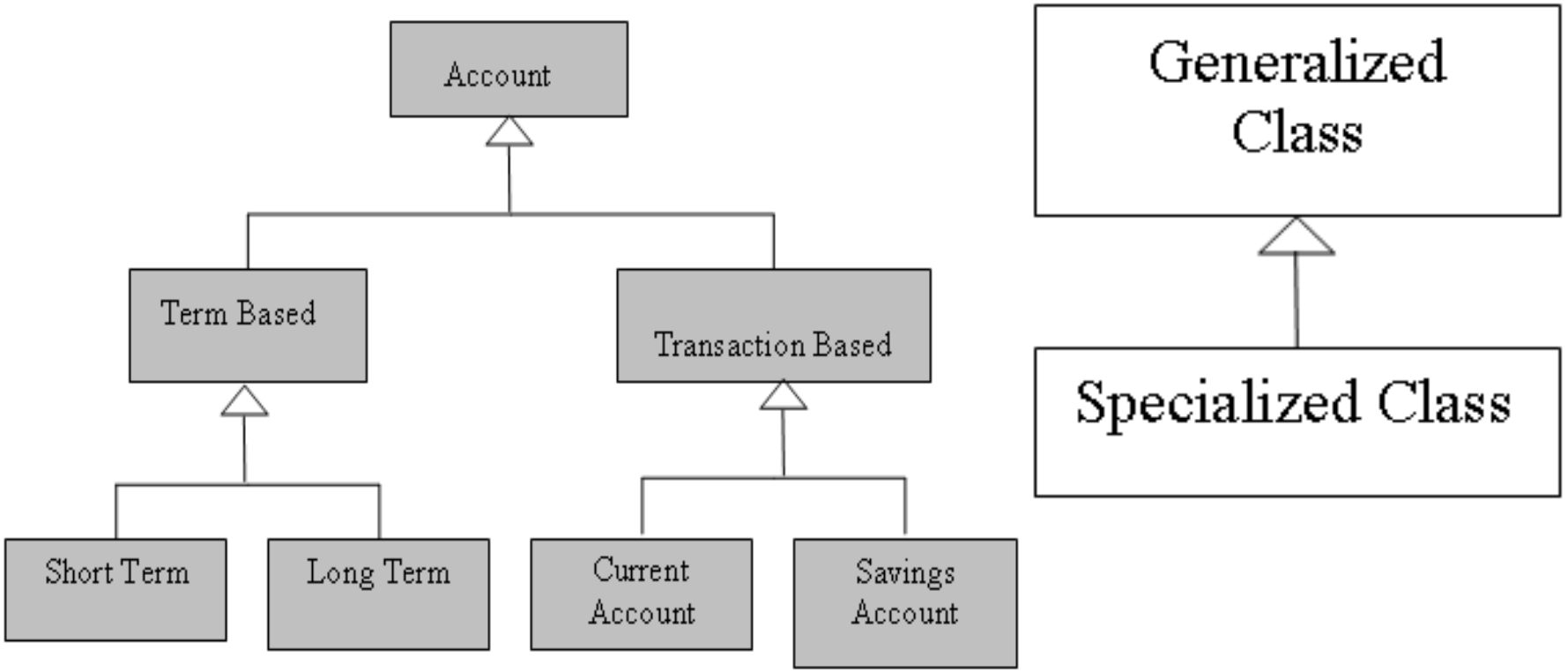
Quan hệ phụ thuộc

- Quan hệ phụ thuộc là một dạng yếu của quan hệ

Quan hệ đệ qui

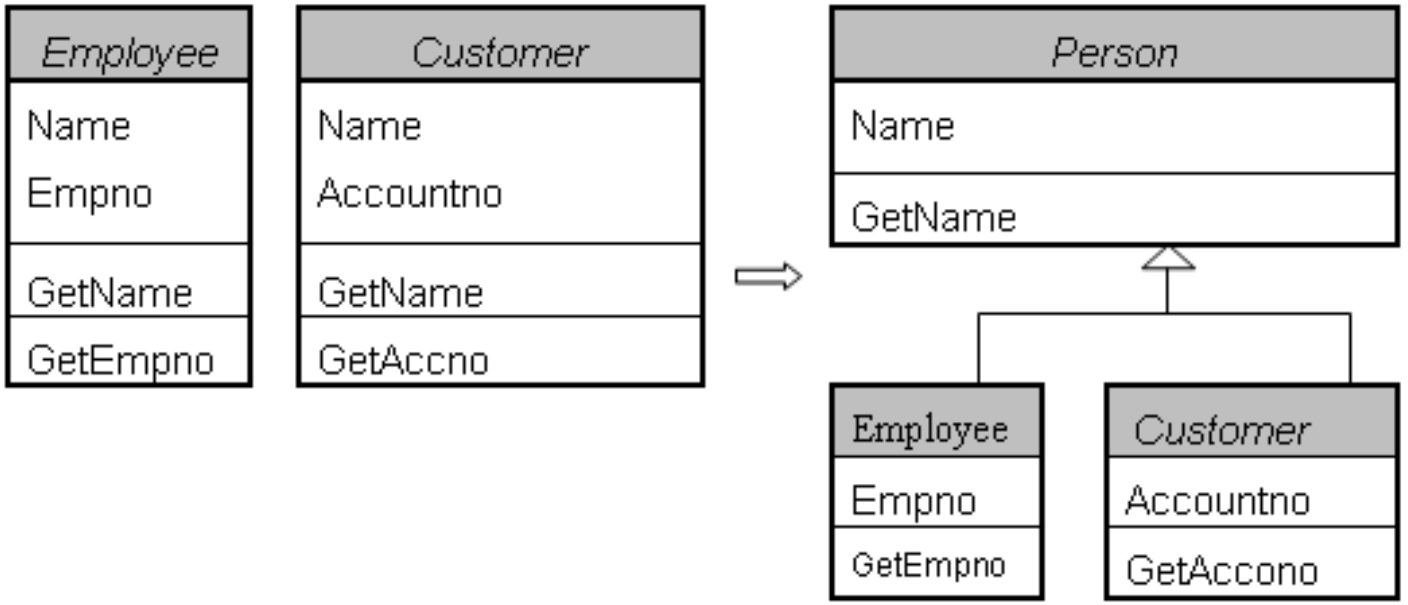


Kế thừa và Tổng quát

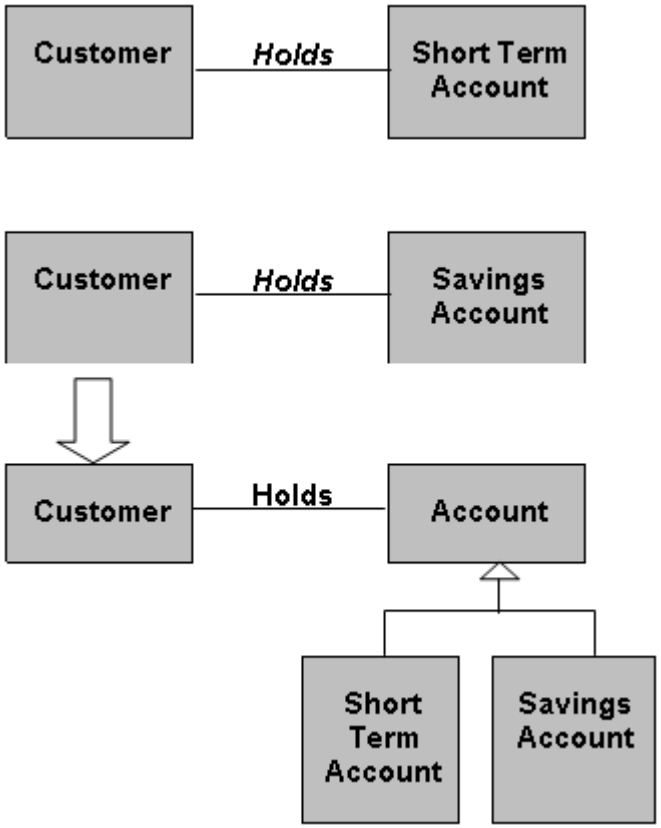


Kế thừa và Tổng quát

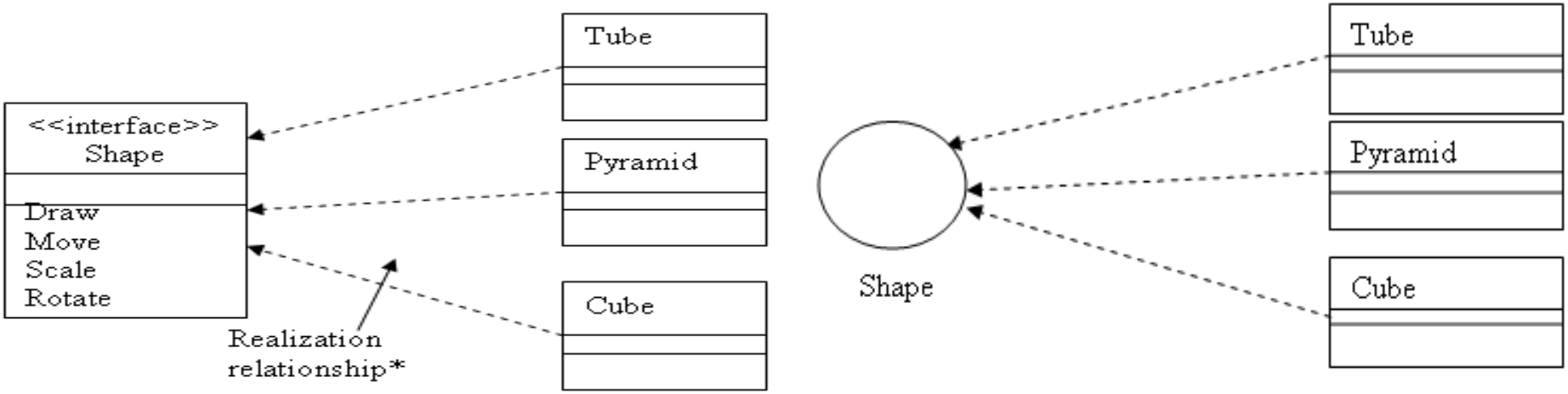
- Cơ chế chia sẻ các thuộc tính và phương thức dùng phương thức tổng quát được coi như kế thừa.



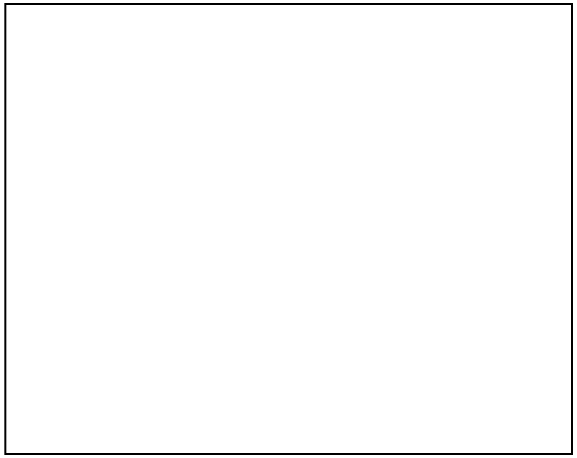
Kế thừa và Tổng quát



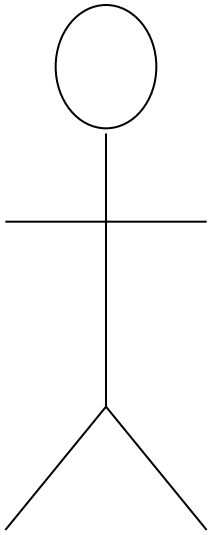
Giao diện - Interface



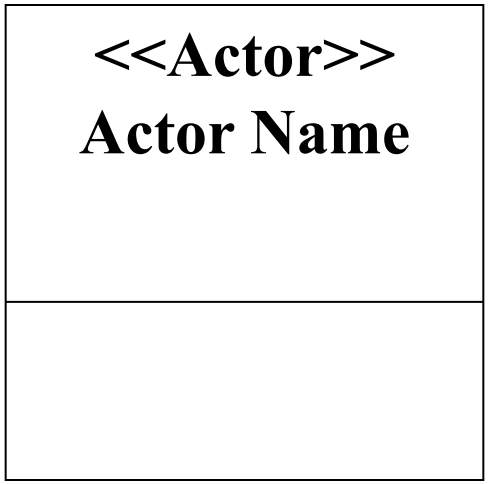
Lược đồ Use Case



Giới hạn hệ thống

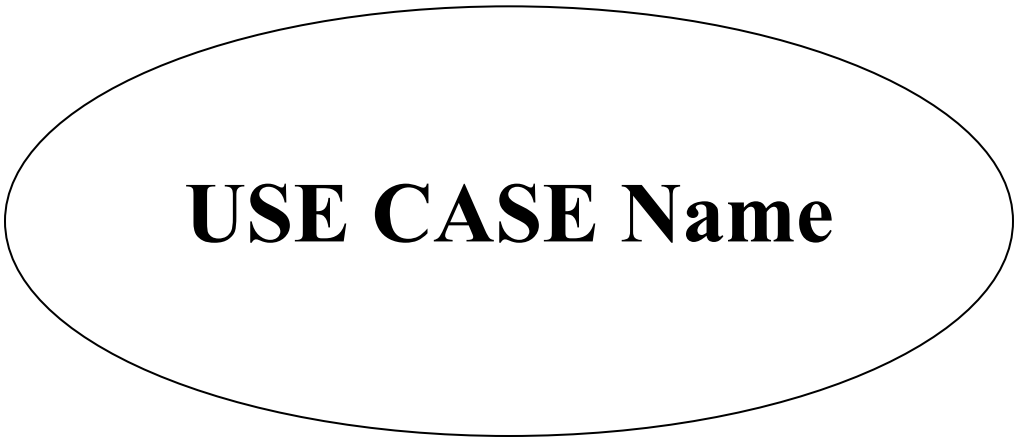


Tác nhân



Lớp tác nhân

Lược đồ Use Case



Use Case

<< >>

Stereotypes

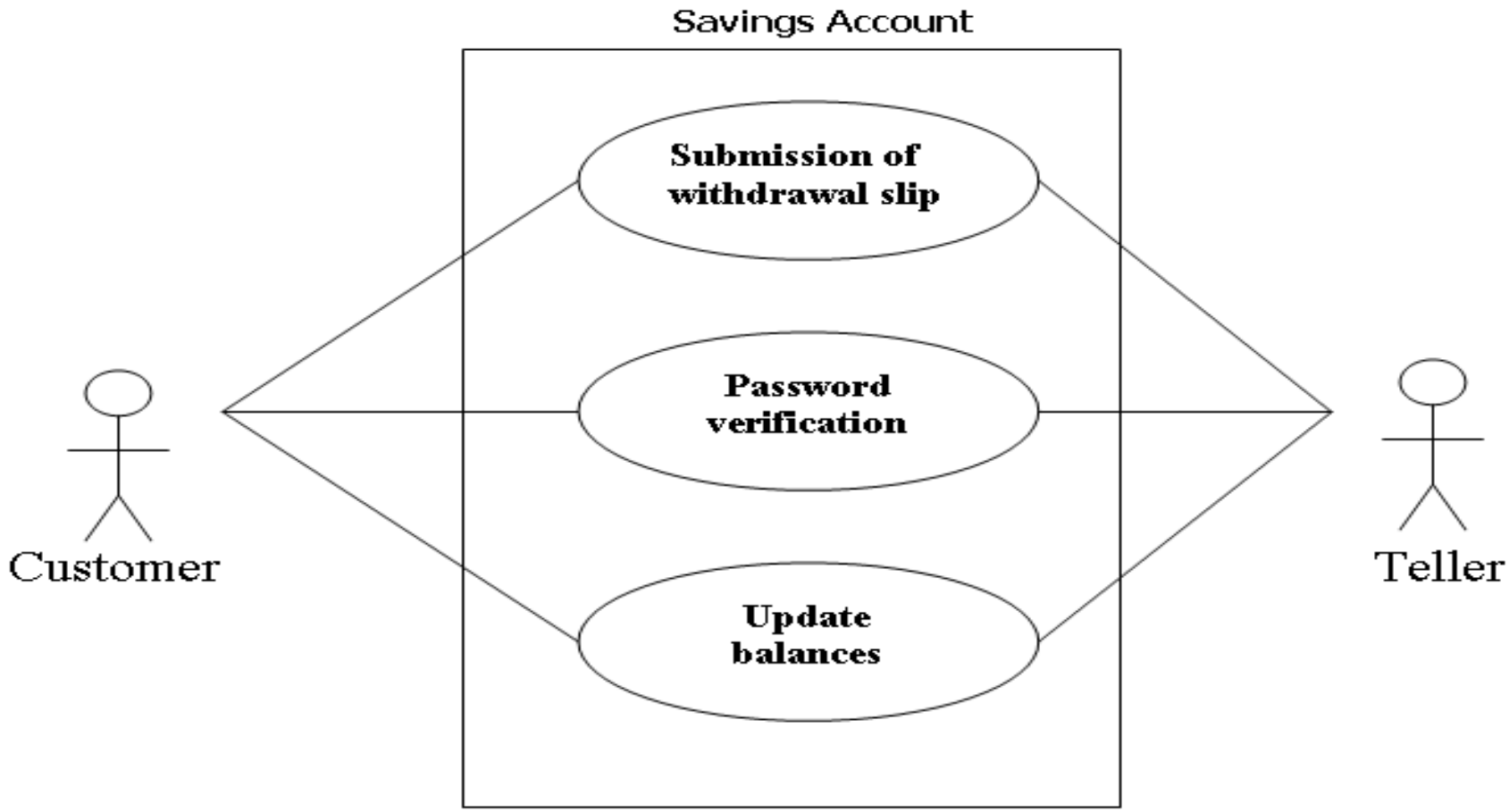
Lược đồ Use Case

Connects actors to use cases

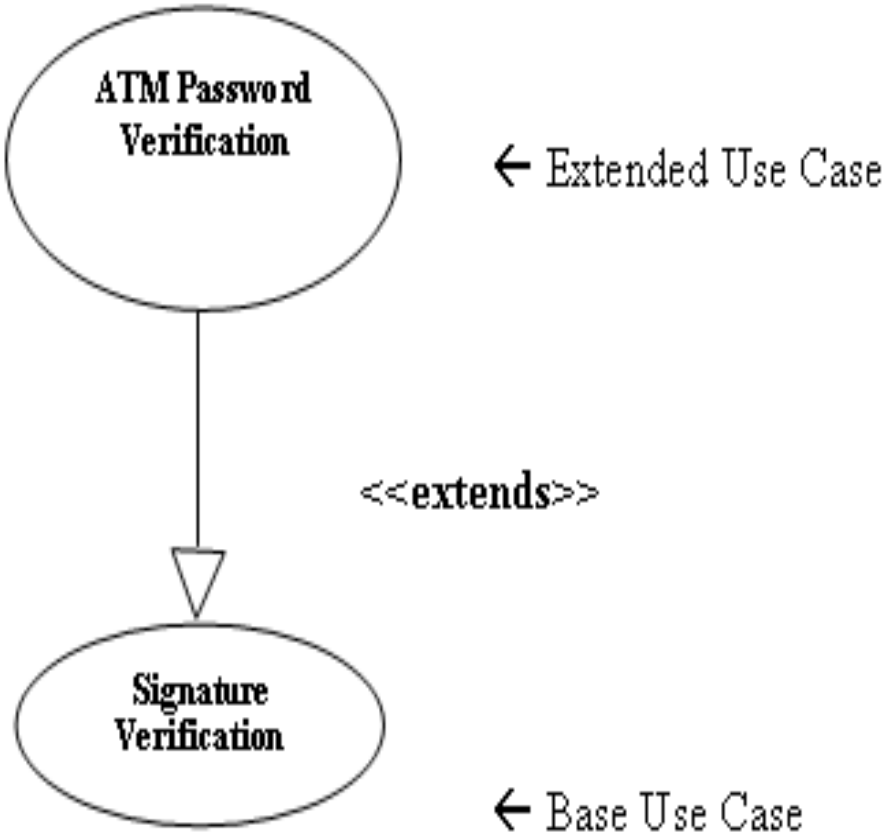
Shows dependencies between use cases

Relationship <<extends>> / <<uses>>

Lược đồ Use Case

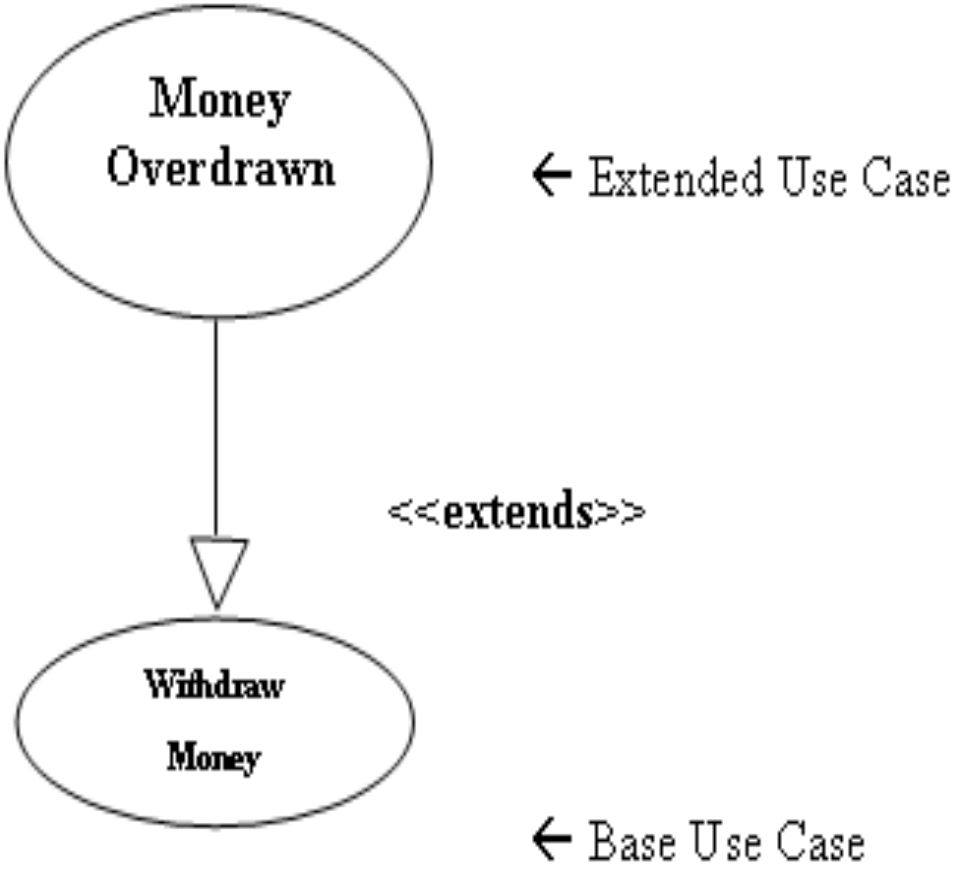


Quan hệ giữa các Use case



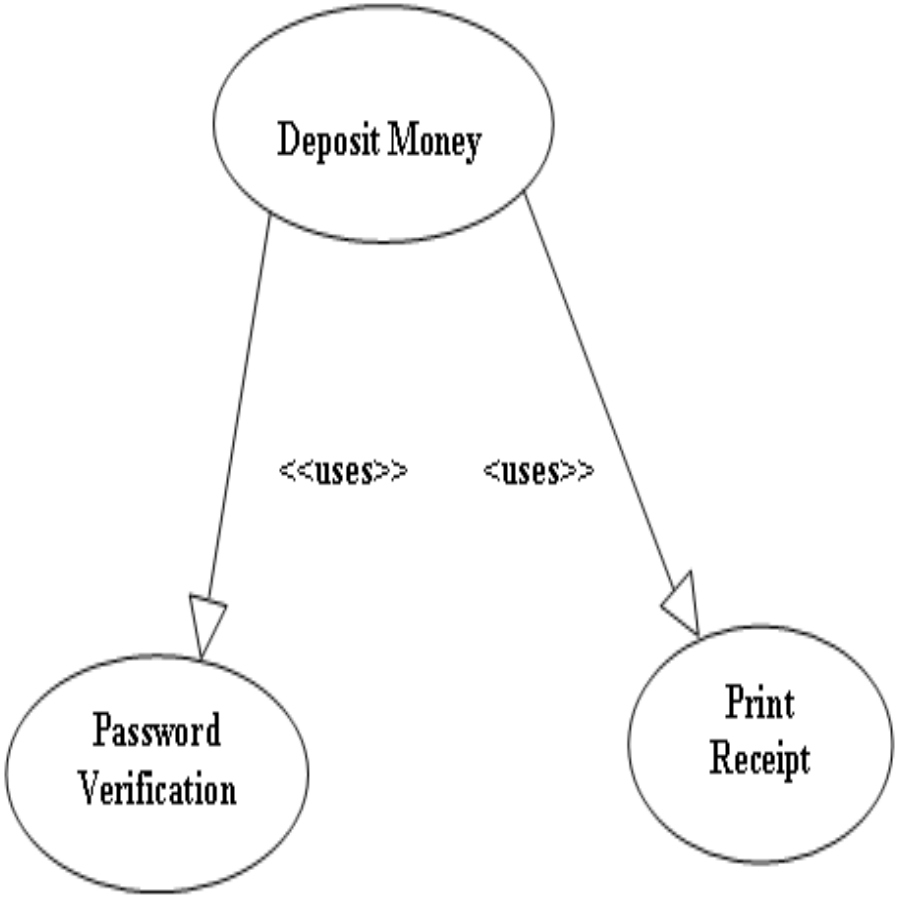
Extends

Quan hệ giữa các Use case



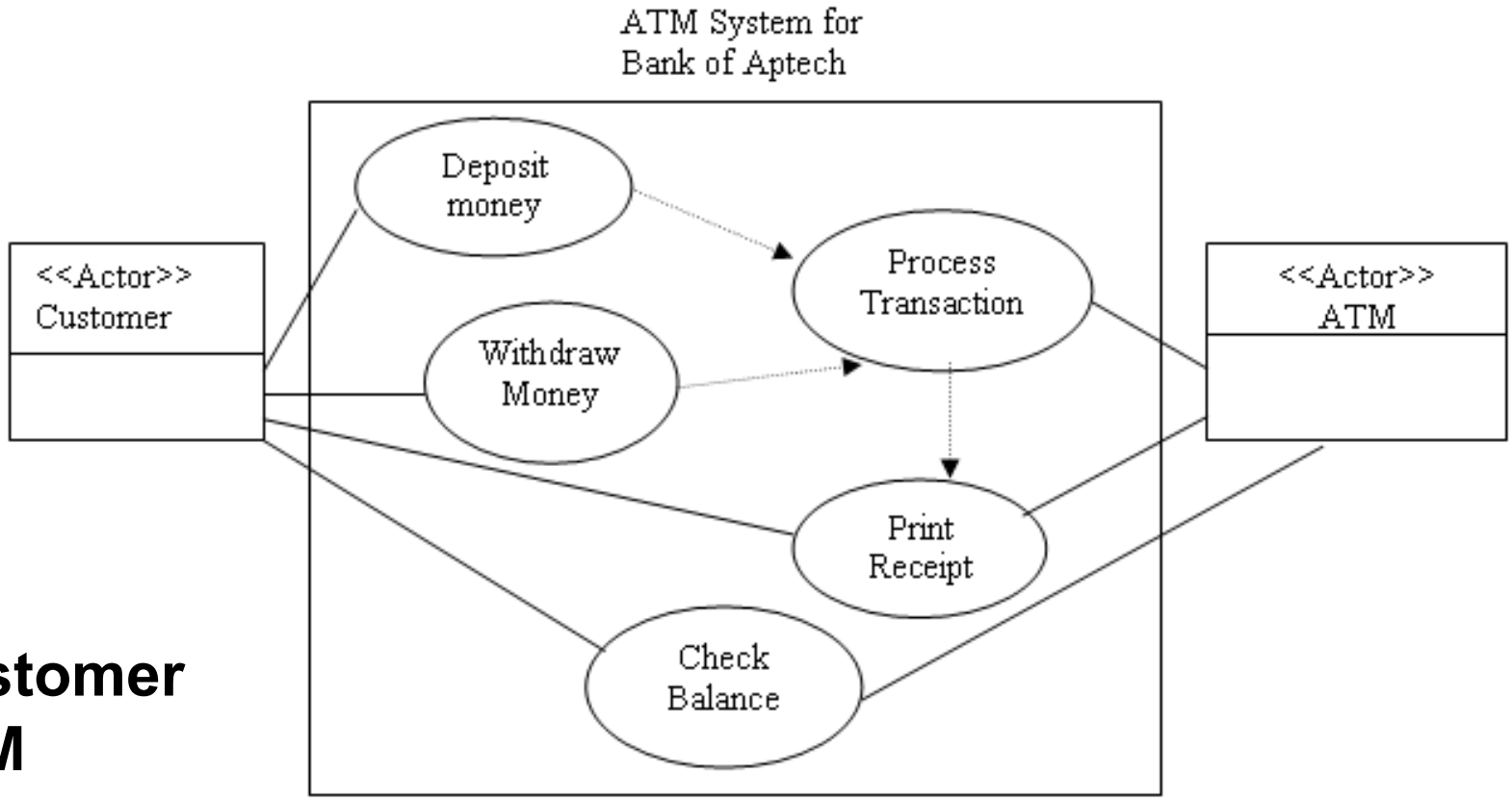
Extends

Quan hệ giữa các Use case



Use

Ví dụ về Use case



- Customer
- ATM

Xây dựng mô hình phân tích

- **Các công việc xây dựng lược đồ phân tích bao gồm**
  - Tìm kiếm các đối tượng / lớp trong hệ thống
    - Đối tượng / lớp thực thể
    - Đối tượng / lớp biên
    - Đối tượng / lớp control
  - Xác định các thuộc tính của đối tượng / lớp
  - Xác định các phương thức của đối tượng / lớp
  - Nhận diện các lớp trừu tượng qua mối quan hệ tổng quát hóa
  - Xác lập các mối quan hệ giữa các lớp:
    - Tổng quát hoá (generalization)
    - Tập hợp (association)
    - Tổng hợp (aggregation)
- **Biểu diễn thành lược đồ lớp phân tích**

Nhận diện đối tượng / lớp

- Dựa vào đặc tả của từng use-case để tìm kiếm các đối tượng
- Các đối tượng thường xuất hiện trong các danh từ hay nhóm danh từ
- Một số lưu ý
  - Không nên dùng đối tượng để biểu diễn một dữ liệu đơn (nên xem là thuộc tính của đối tượng khác)
  - Đối tượng/lớp phải thực sự cần thiết cho sự hoạt động của hệ thống
  - Đối tượng/lớp >< bảng cơ sở dữ liệu
  - Đối tượng/lớp >< actor

Nhận diện và biểu diễn đối tượng / lớp

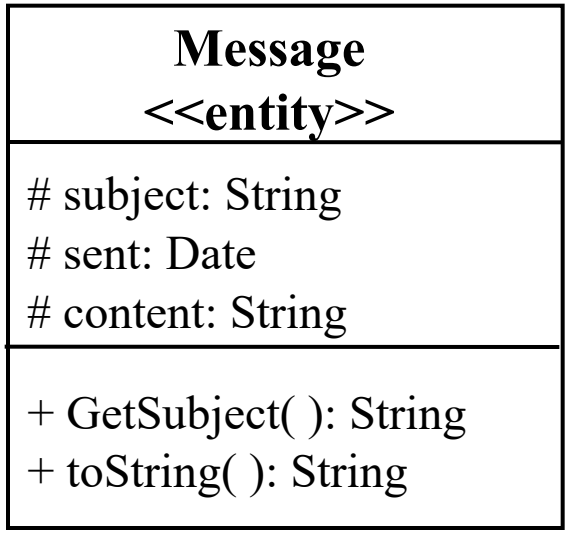
- Phân loại đối tượng/lớp
  - Đối tượng thực thể (entity)
  - Đối tượng biên (boundary)
  - Đối tượng điều khiển (control)
- Trong UML, lớp được biểu diễn bằng một hình chữ nhật gồm 3 phần: tên, các thuộc tính và các tác vụ
- Có thể áp dụng stereotype cho lớp: <<entity>>, <<boundary>>, <<control>>...
- Đối tượng cũng được biểu diễn bằng hình chữ nhật, thông thường gồm 2 phần: tên đối tượng + tên lớp (được gạch chân), giá trị các thuộc tính (trạng thái của đối tượng)

Đối tượng / lớp thực thể

- Biểu diễn cho các thực thể xuất hiện một cách tự nhiên trong hệ thống
- Thông tin về các đối tượng thực thể có thể phải được lưu trữ lâu dài (database, file...)
- Trong UML, được gán stereotype <<entity>>
- Dễ nhận diện các thuộc tính của chúng

Ví dụ:

- Đối với hệ thống đăng ký môn học hệ tín chỉ qua WEB, nhận diện các đối tượng thực thể như: thông tin SV, thông tin GV, nhóm lớp học, đăng ký nhóm, sổ tay sinh viên …
- Đối với hệ thống mail, nhận diện các đối tượng thực thể như: hộp thư, thông điệp mail…



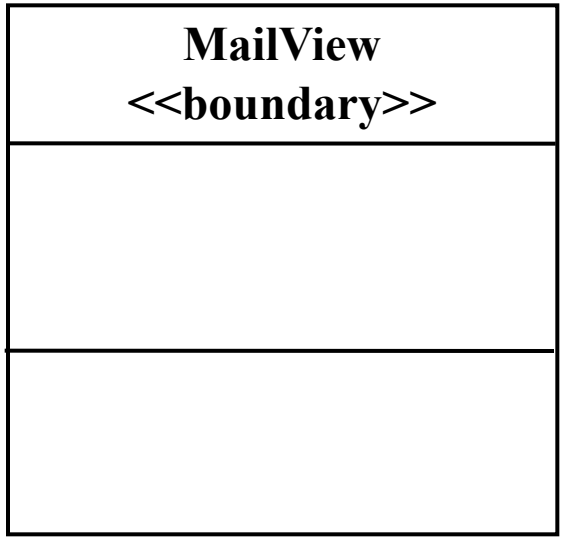
Đối tượng / lớp biên

- Thực hiện chức năng giao tiếp với actor
- Thường chứa các phần tử hoặc điều khiển giao diện người dùng (nút nhấn, hộp danh sách, tuỳ chọn, menu...)
- Trong UML, được gán stereotype <<boundary>>
- Khó nhận biết các thuộc tính và tác vụ trong mô hình phân tích

Ví dụ:

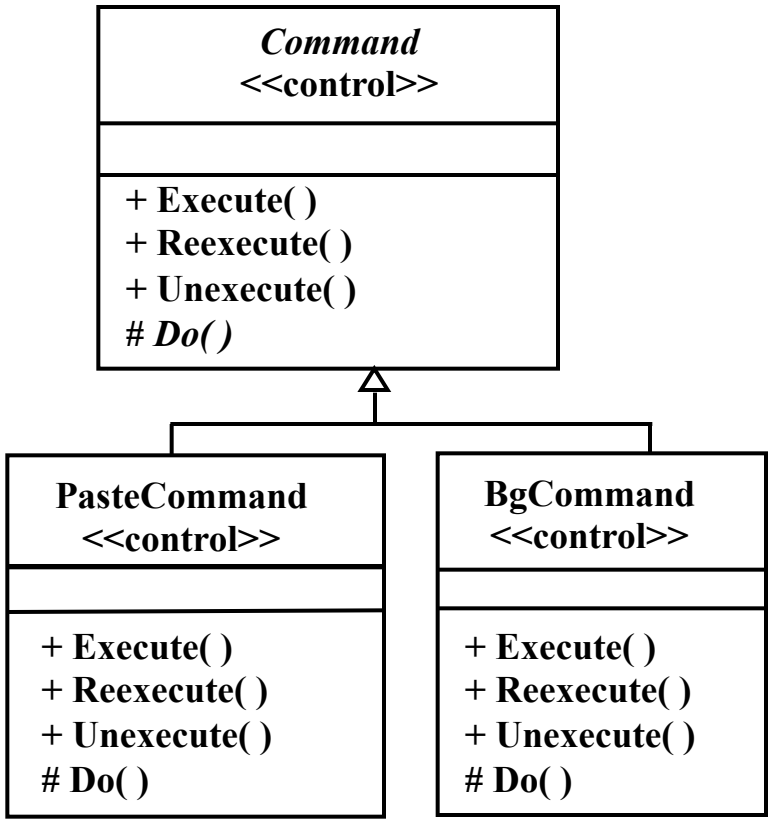
• Đối với hệ thống đăng ký môn học hệ tín chỉ qua WEB, nhận diện các đối tượng biên như: RegisterForm, StudentForm…

• Đối với hệ thống mail, nhận diện các đối tượng biên như: MailView, MailCompose...



Đối tượng / lớp điều khiển

- Có nhiệm vụ điều khiển các lớp khác hoặc
- Những lớp không phải là lớp thực thể và lớp biên
- Trong UML, được gán stereotype <<control>>
- Lớp biên thường có quan hệ liên kết hoặc phụ thuộc với các lớp khác
- Ví dụ:
- Đối tượng biểu diễn một số lệnh thông thường như cắt, dán, thay đổi thông số nhìn trong hiển thị đồ hoạ …



Nhận diện các thuộc tính

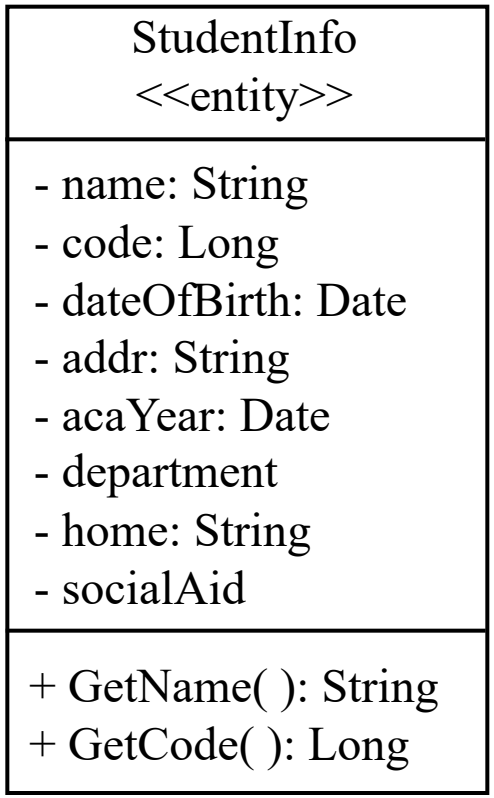
- Dựa vào đặc tả của từng use-case, tìm kiếm các danh từ hoặc nhóm danh từ liên quan đến đối tượng đang xét
- Trả lời câu hỏi: những thành phần nào cấu thành đối tượng đang xét ?
  - Lưu ý: cùng một đối tượng trong các ngữ cảnh khác nhau chúng ta có thể tìm được các thuộc tính khác nhau
- Nên xác định (tuy nhiên không bắt buộc) trong mô hình phân tích
  - Kiểu của thuộc tính: một số kiểu cơ bản
  - Bậc của thuộc tính: số ít hoặc số nhiều
  - Visibility của thuộc tính: mức độ cho phép truy xuất thuộc tính từ bên ngoài
- UML: thuộc tính được miêu tả tường minh hoặc thông qua quan hệ với các lớp khác

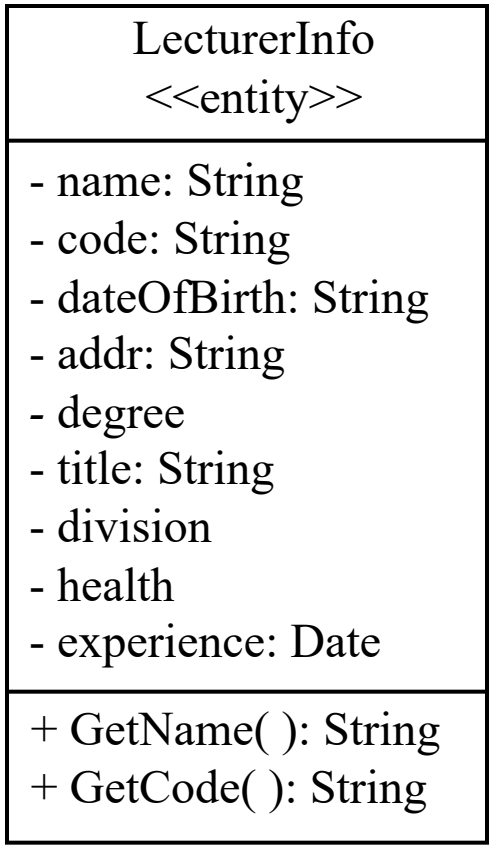
Xác định mức độ truy cập của thuộc tính

- Mức độ truy cập và phạm vi mà thuộc tính đó có thể được tham khảo đến trực tiếp
- UML định nghĩa 3 mức độ truy xuất thuộc tính (visibility)
  - public (+): có thể truy xuất thuộc tính từ tất cả các vị trí khác nhau
  - protected (#): bản thân lớp đang xét và các lớp con của nó có thể truy xuất thuộc tính
  - private (-): chỉ có lớp đang xét có thể truy xuất thuộc tính
- Thông thường nên đặt mức độ truy xuất thuộc tính là private hoặc protected (cho các lớp cơ sở), không nên là public.
- Thuộc tính nên được truy xuất thông qua tác vụ get/set

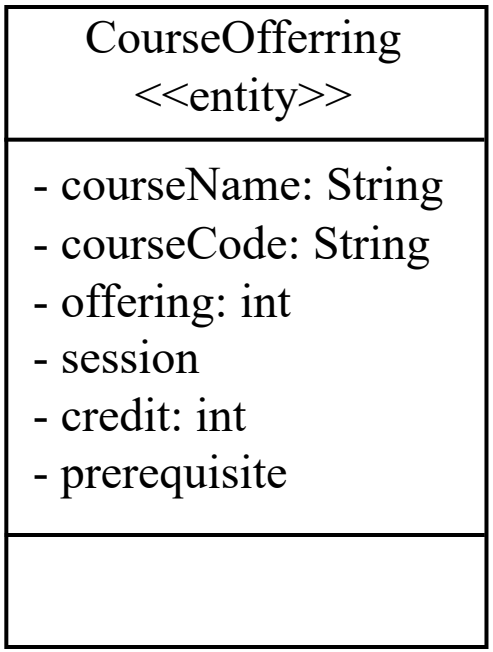
Ví dụ về nhận diện các thuộc tính

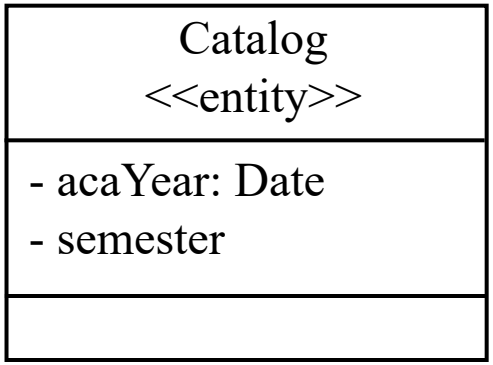
- Hệ thống đăng ký môn học hệ tín chỉ qua WEB - Nhận diện các thuộc tính cho các đối tượng: StudentInfo, LecturerInfo
- Chú ý các mức độ truy cập của các thuộc tính
- Các tác vụ phát sinh trong khi nhận diện các thuộc tính → như Get/Set





Ví dụ về nhận diện các thuộc tính





- Hệ thống đăng ký môn học hệ tín chỉ qua WEB
- Nhận diện các thuộc tính cho các đối tượng:

  CourseOffering,

  Catalog

Nhận diện các tác vụ

- Dựa vào đặc tả của từng use-case, tìm kiếm các *động từ* hoặc *nhóm động từ* liên quan đến đối tượng đang xét
- Chú ý xem đối tượng được tạo ra và bị huỷ bỏ đi như thế nào ? Trong thời gian đó nó gửi/nhận thông điệp ra sao ?
- Các đối tượng biên có các tác vụ nhận lệnh từ actor.
- Xem xét mức độ truy xuất của tác vụ tương tự như đối với các thuộc tính; các tác vụ thường có visibility là + hoặc #
- Một số tác vụ không xuất hiện một cách tự nhiên trong mô hình phân tích → mô hình thiết kế sẽ nghiên cứu kỹ trách nhiệm và hành vi của từng đối tượng

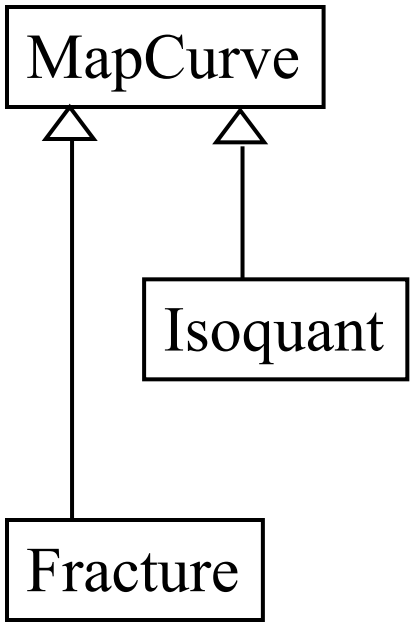
Nhận diện lớp cơ sở

- Lớp cơ sở (base class) được nhận diện sau khi đã nhận diện các lớp cụ thể
- Sự xuất hiện của lớp cơ sở làm cho mô hình phân tích có tính tái sử dụng lại cao (reusability) và dễ mở rộng (scalability)
- UML hỗ trợ quan hệ tổng quát hoá (generalization)
- Lớp cơ sở trừu tượng (không thể cụ thể hoá tạo ra đối tượng) có tên in nghiêng
- Lớp cơ sở được hình thành bằng cách xác lập các quan hệ tổng quát hóa của các lớp cụ thể có chung một số thuộc tính và/hay một số tác vụ

Nhận diện lớp cơ sở (tt)

- **Đối với các đối tượng/lớp thực thể, tìm các thuộc tính chung để hình thành lớp cơ sở**
- **Ví dụ**
  - **Trong hệ thống quản lý thư viện qua WEB: các đối tượng Book, Magazine có một số thuộc tính chung hình thành lớp LibraryItem**
  - **Đối với hệ thống đăng ký môn học tín chỉ qua WEB: lớp PeopleInfo là lớp cơ sở của StudentInfo và LecturerInfo**
  - **Chương trình vẽ bề mặt địa hình: lớp MapCurve là lớp cơ sở của đường đồng mức Isoquant và đứt gãy Fracture**
- **Giữa lớp cơ sở và các lớp cụ thể có mối quan hệ tổng quát hóa có thể biểu diễn được trong UML**

Biểu diễn lớp cơ sở và quan hệ tổng quát hóa

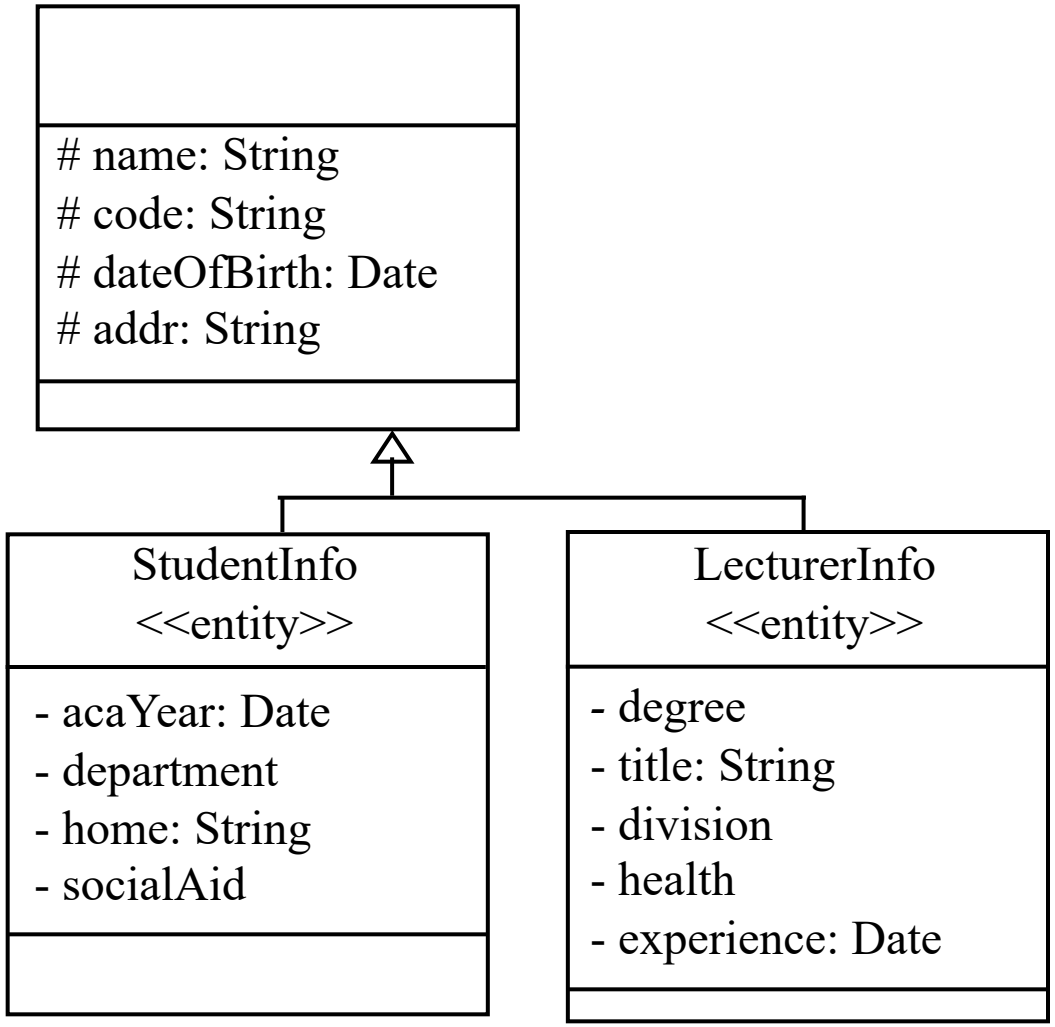


- UML định nghĩa quan hệ tổng quát hoá giữa một lớp tổng quá hơn với một lớp cụ thể hơn: lớp cụ thể hơn có tất cả thuộc tính, tác vụ và quan hệ của lớp kia + những thuộc tính/tác vụ riêng của nó
- Ký hiệu: mũi tên có đầu là một tam giác nhỏ
- Lớp tổng quát hơn nằm về phía mũi tên

Ví dụ về nhận diện lớp cơ sở

Ví dụ:

- Trong hệ thống đăng ký môn học tín chỉ qua WEB, lớp PeopleInfo là tổng quát hoá của StudentInfo và LecturerInfo



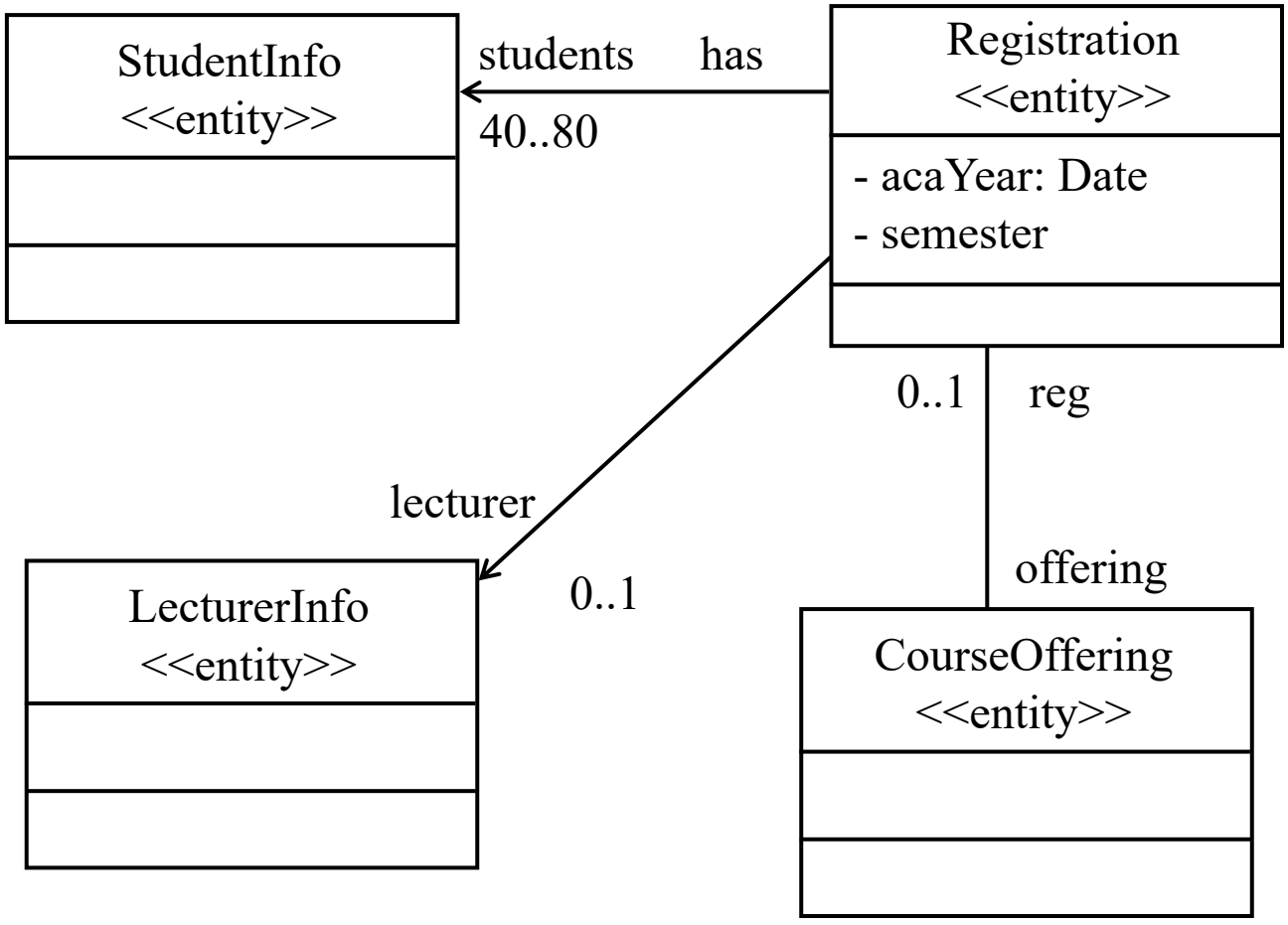
Nhận diện các mối quan hệ

- Sau khi xác định các lớp/đối tương kể cả các đối tượng cơ sở, các quan hệ giữa các lớp cần được xác lập
- Trong mô hình phân tích các đối tượng/lớp có quan hệ với nhau
- Một số quan hệ mà UML hỗ trợ
  - Tổng quát hoá (generalization)
  - Liên kết (association)
  - Bao gộp (aggregation)
- Các quan hệ khác được áp dụng cho mô hình thiết kế
  - Phụ thuộc (dependency)
  - Cụ thể hoá (realization)

Quan hệ liên kết

- Quan hệ liên kết là mối quan hệ giữa 2 đối tượng/lớp
- Về ý nghĩa và ký hiệu giống như quan hệ liên kết trong mô hình nghiệp vụ
- Áp dụng cho 2 lớp có mối tương quan mang ý nghĩa nhất định
- Chú ý ghi rõ (nếu có thể được)
  - Bậc và tên vai trò của mỗi lớp trong quan hệ
  - Tên của chính quan hệ liên kết
- Dựa vào mô hình nhgiệp vụ xác định các mối quan hệ liên kết

Ví dụ - mối quan hệ liên kết



Ví dụ:

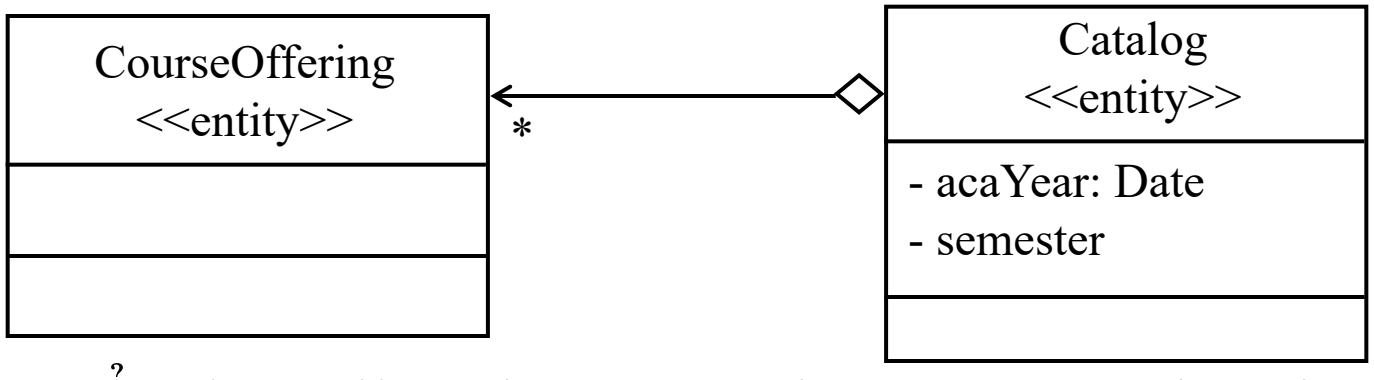
Lớp Registration liên kết với lớp StudentInfo LecturerInfo và CourseOffering

Quan hệ bao gộp

- **UML định nghĩa quan hệ bao gộp là trường hợp đặc biệt của quan hệ liên kết, khi mà một đầu nối liên kết trở thành đầu nối bao gộp (aggregation)**
- **Lớp ở đầu nối bao gộp sẽ bao hàm lớp kia**
- **Ký hiệu của đầu nối bao gộp là một hình thoi tô hoặc không tô đen**
- **Có hai dạng bao gộp**
  - **Chia sẻ (shared): chia xẻ giữa các bao gộp khác nhau**
  - **Hoàn toàn (composite): sở hữu đầy đủ**
- **Xác lập các mối quan hệ bao gộm và biểu diễn chúng lên lược đồ lớp**

Quan hệ bao gộp – ví dụ

- Đối với hệ thống đăng ký môn học tín chỉ qua WEB, lớp Catalog bao gộp lớp CourseOffering



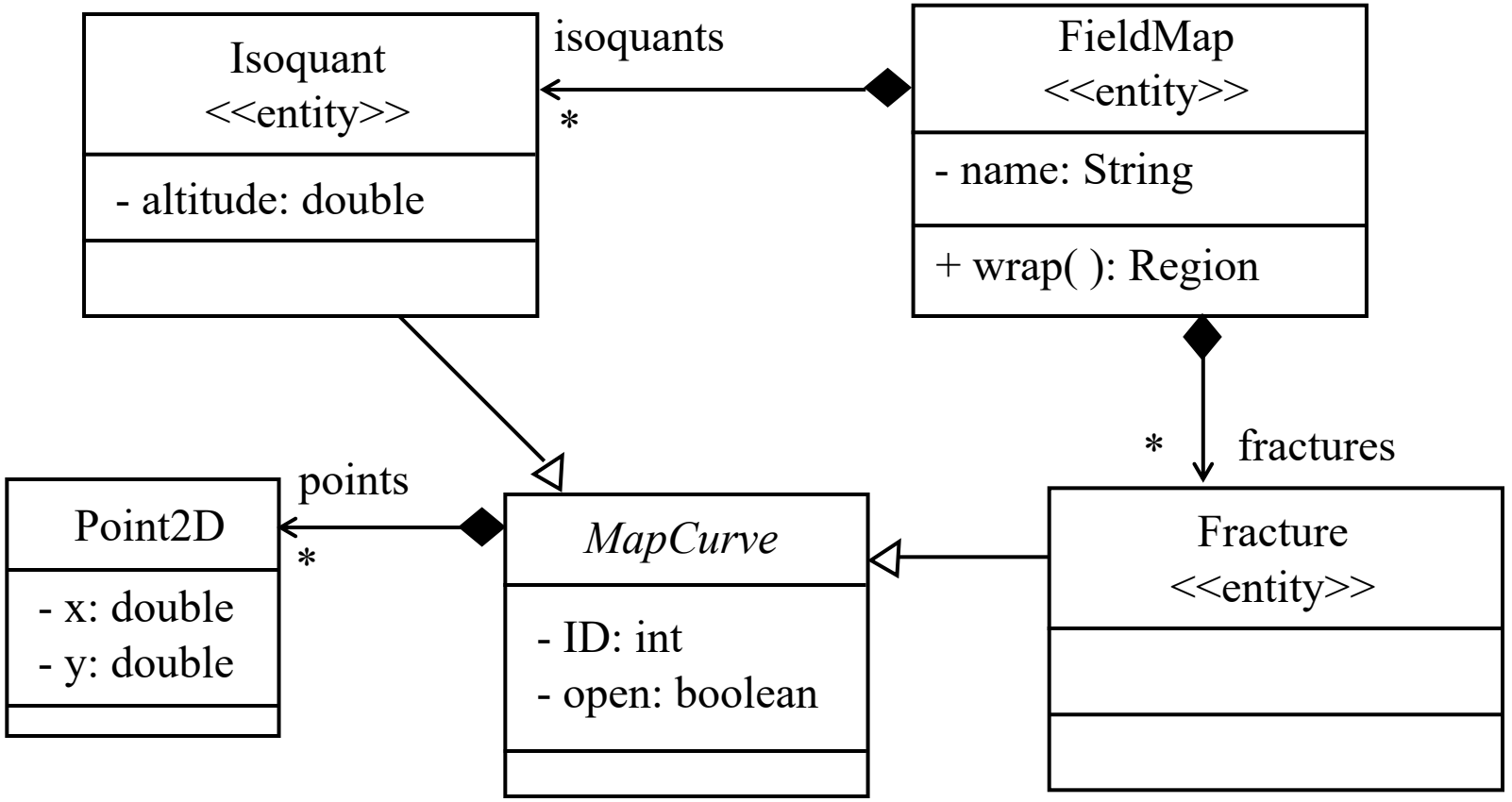
- Cửa sổ giao diện bao gộp hoàn toàn thanh cuộn và menu

Xây dựng lược đồ lớp

- Lược đồ lớp biểu diễn cấu trúc của một số lớp và quan hệ giữa chúng → mô tả khía cạnh tĩnh (static) của hệ thống
- Hệ thống phức tạp có nhiều lớp → cần xây dựng nhiều lược đồ lớp, mỗi lược đồ mô tả một phần của hệ thống
- Lược đồ lớp được bổ sung và hoàn thiện trong mô hình thiết kế (thêm một số lớp, chi tiết các thuộc tính và tác vụ, làm rõ các quan hệ)
- Lược đồ lớp được xây dựng qua các bước
  - Xác định các lớp
  - Xác định thuộc tính và tác vụ của các lớp
  - Xác định các lớp cơ sở và quan hệ tổng quát hoá
  - Xác định các quan hệ liên kết và bao gộp

Ví dụ một lược đồ lớp phân tích

- một lược đồ lớp của chương trình hiển thị bề mặt địa hình



Lược đồ tuần tự

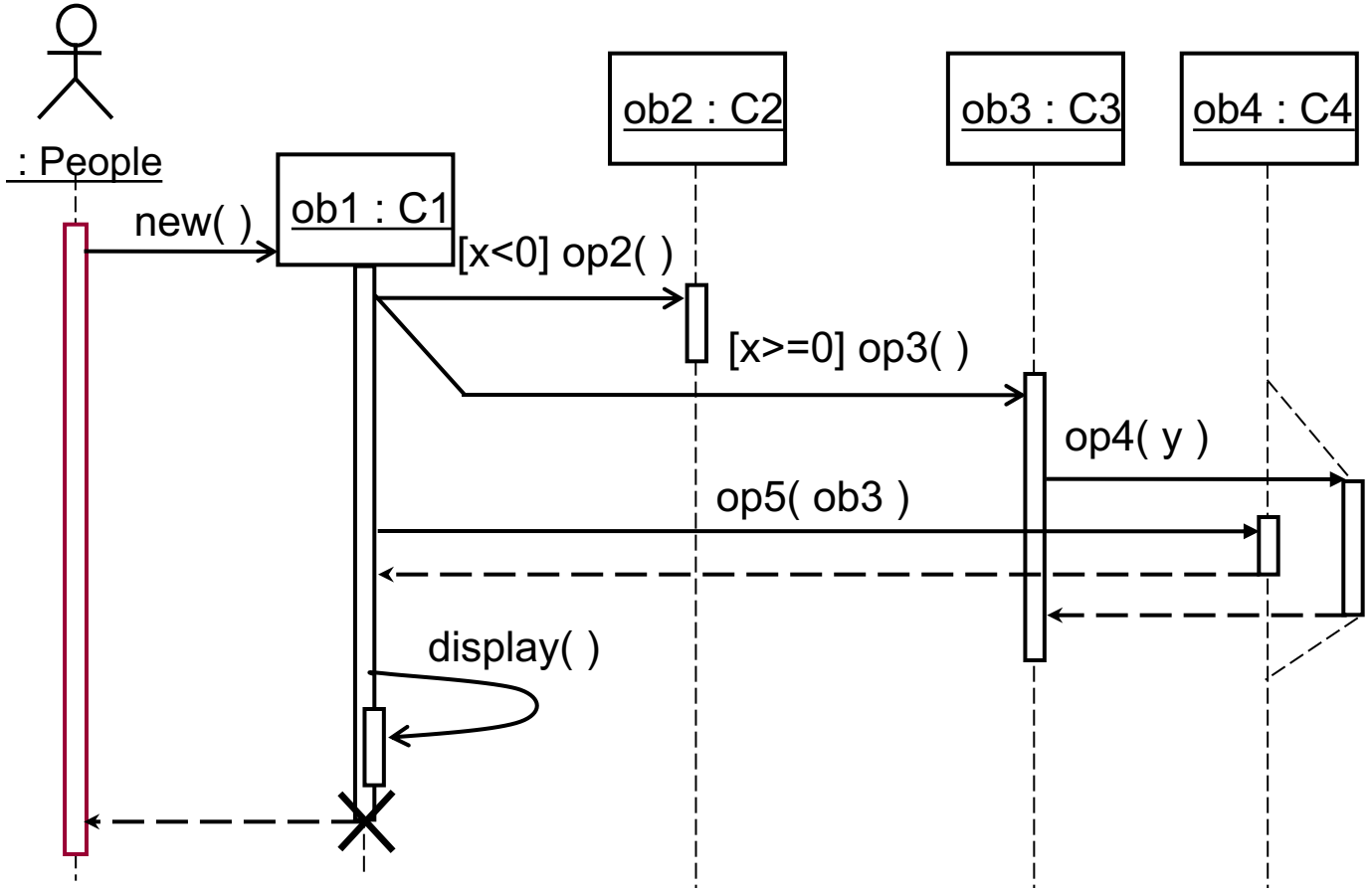
- Lược đồ tuần tự có 2 dạng

  Dạng tổng quát: thể hiện cả vòng lặp và rẽ nhánh

  Dạng cụ thể: miêu tả một kịch bản cụ thể
- Thời gian sống của mỗi đối tượng được mô tả theo một đường thẳng đứng
- Thông thường thời gian trôi theo chiều từ trên xuống dưới
- Ít khi quan tâm đến khoảng thời gian, thường chỉ quan tâm đến trình tự mà thôi
- Thanh hình chữ nhật mô tả sự thực thi của một tác vụ để đáp ứng lại thông điệp gửi đến. Độ dài của thanh chữ nhật phản ánh thời gian thực thi của tác vụ và tính chất lồng nhau (nested) giữa chúng
- Các dòng text phụ trợ (mô tả tác vụ, ràng buộc thời gian...) được viết ở lề trái

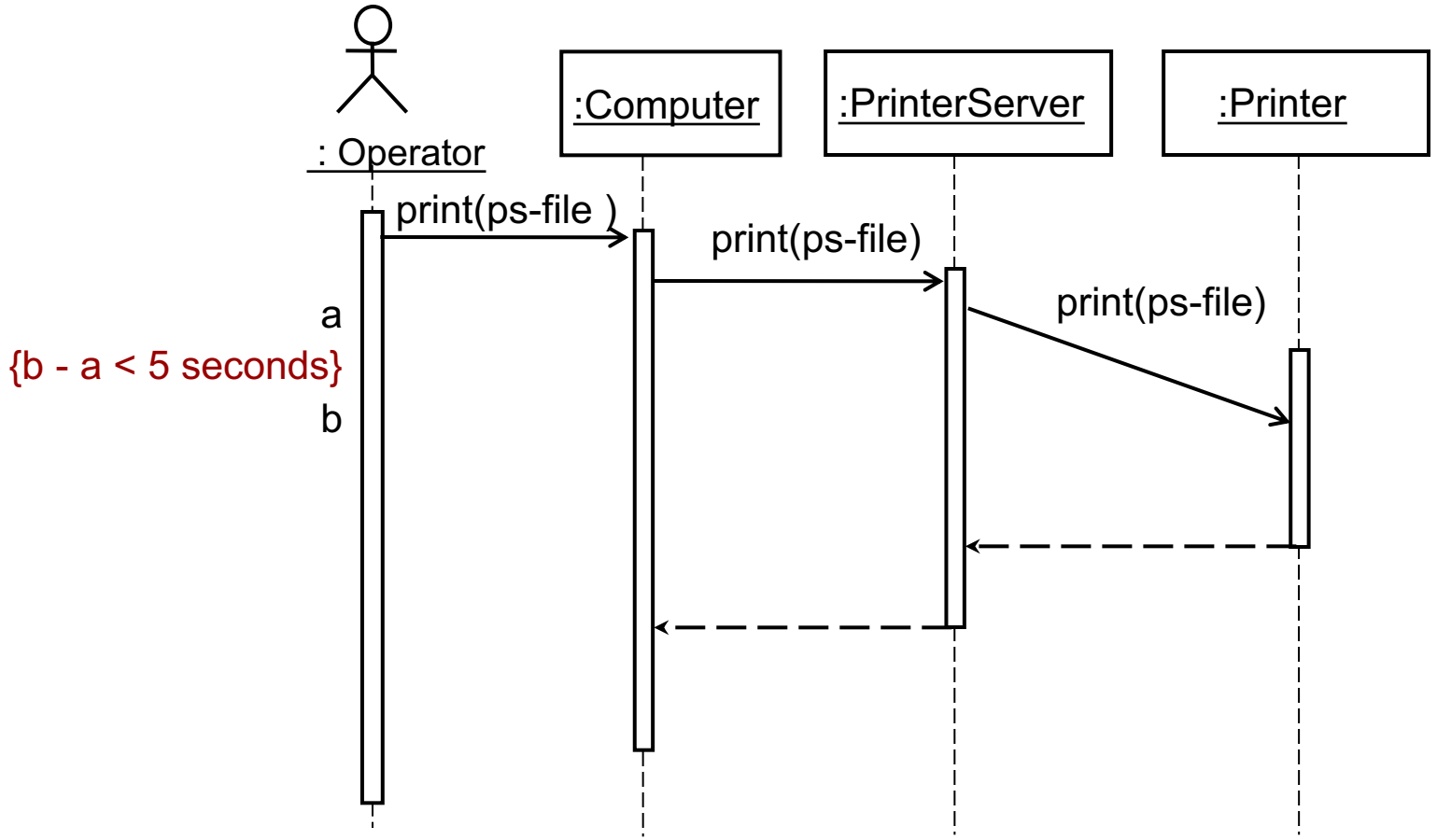
Lược đồ tuần tự - Ví dụ

- Ví dụ: lược đồ tuần tự dạng tổng quát



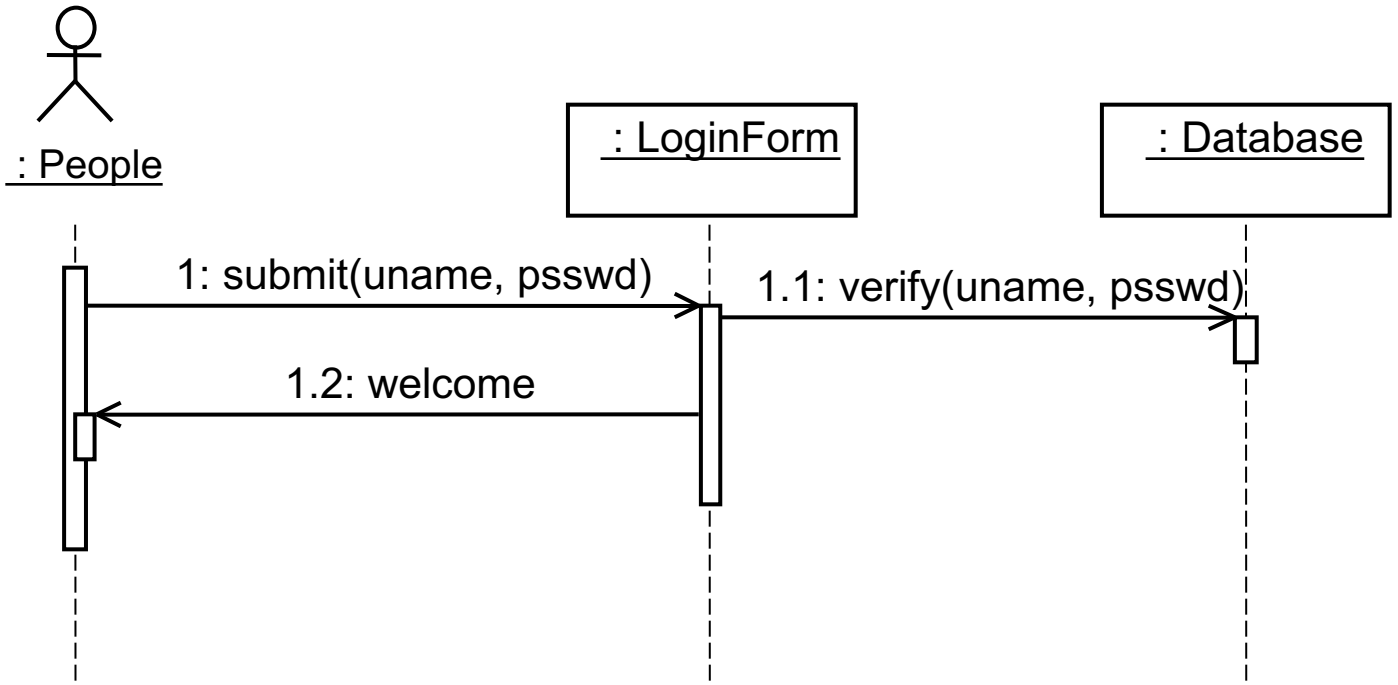
Lược đồ tuần tự - Ví dụ

- Lược đồ tuần tự với các ghi chú ràng buộc thời gian

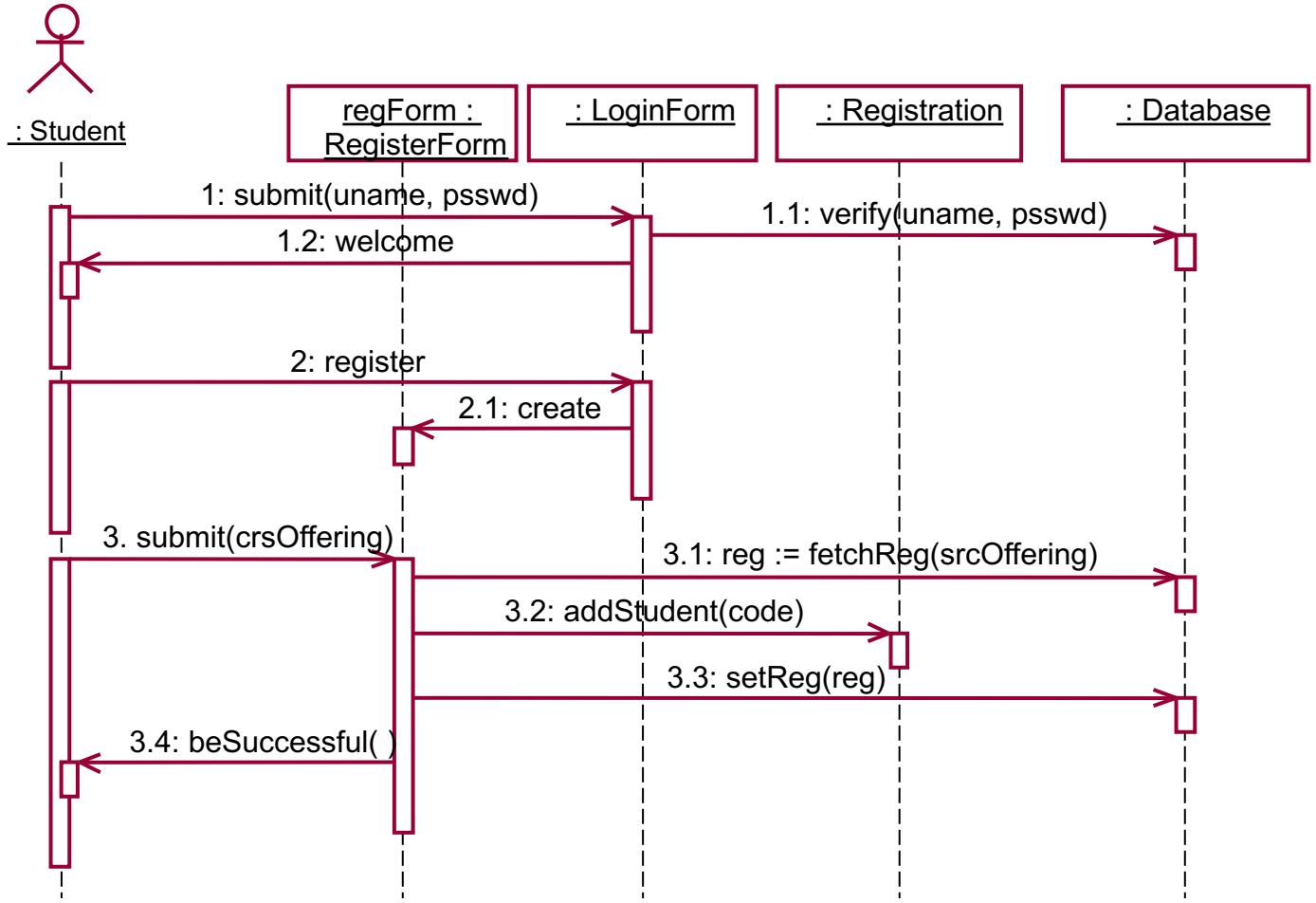


Lược đồ tuần tự - Ví dụ

- Ví dụ: lược đồ tuần tự dạng cụ thể cho use-case Login của hệ thống đăng ký môn học tín chỉ qua WEB



Lược đồ tuần tự - Ví dụ



Lược đồ tuần tự dạng cụ thể cho *use-case* Register courses

Lược đồ cộng tác

- Lược đồ cộng tác là một đồ thị liên kết các vai trò của các đối tượng hình thành nên các chức năng của hệ thống (các usecase)
- Mỗi cạnh liên kết 2 vai trò được biểu diễn bằng một đoạn thẳng
- Tương tác được thể hiện bằng gửi/nhận thông điệp
- Hai vai trò liên kết với nhau khi có trao đổi thông điệp
- Mỗi thông điệp được thể hiện bằng mũi tên (như đã miêu tả) cộng với phần đặc tả

Lược đồ cộng tác (tt)

- Các thông điệp được đánh số theo kiểu phân cấp
  - 3.4.2 xảy ra sau 3.4.1 và cả hai được lồng (*nested*) trong 3.4
  - 3.4.3a và 3.4.3b xảy ra đồng thời và được lồng trong 3.4
- Cú pháp tổng quát của thông điệp

precedessor guard-condition sequence-expression return-value := message-name argument-list

- Ví dụ: 2/ 1.3.1: p := find(specs)

Lược đồ cộng tác (tt)

- Lược đồ cộng tác có thể được thiết lập ở một trong 2 dạng:

Dạng cụ thể: mỗi vai trò được biểu diễn bằng một ký hiệu của đối tượng cụ thể, các thông điệp được trao đổi trên các đường liên kết

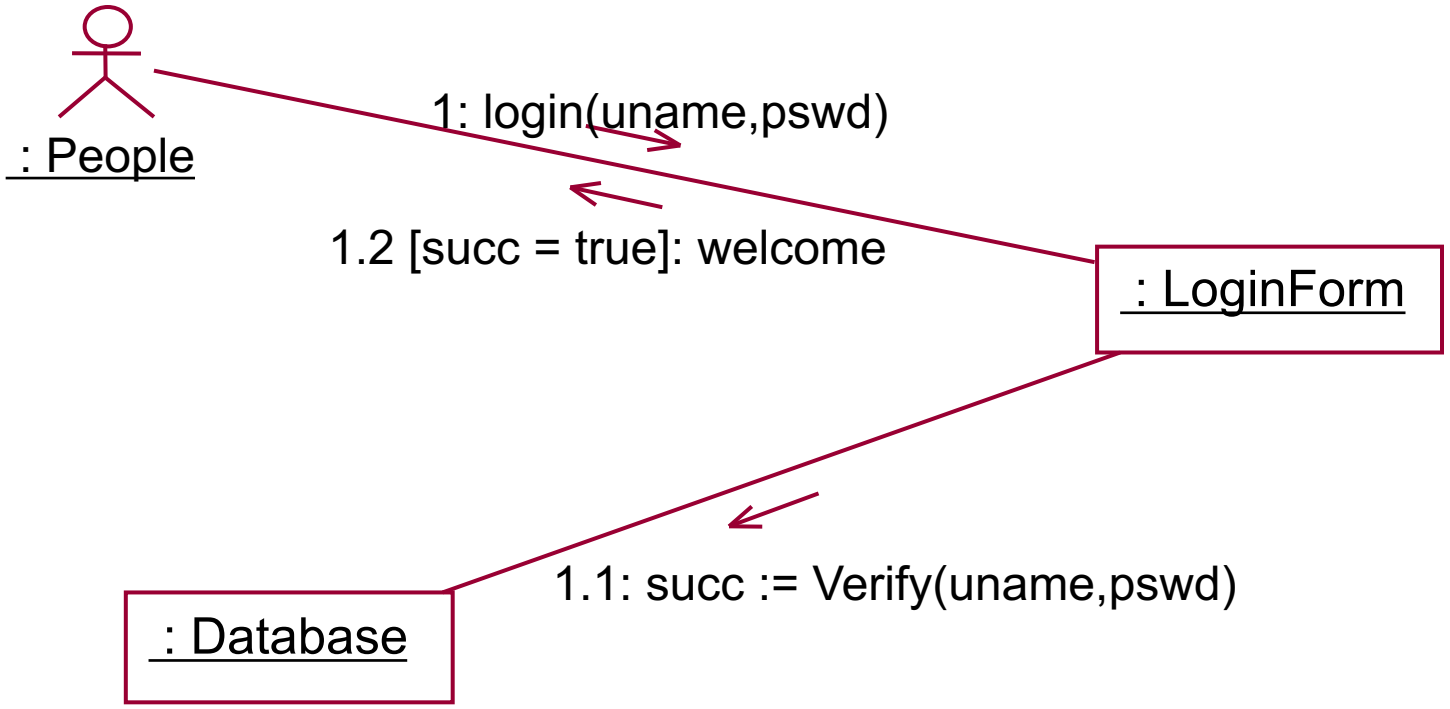
Dạng đặc tả: mô tả các lớp; các đường liên kết được ánh xạ vào các thông điệp

- Thiết lập lược đồ cộng tác giúp cụ thể hoá (realize) các use-case và nhận diện thêm một số tác vụ của các đối tượng/lớp phân tích

Lược đồ cộng tác – Ví dụ

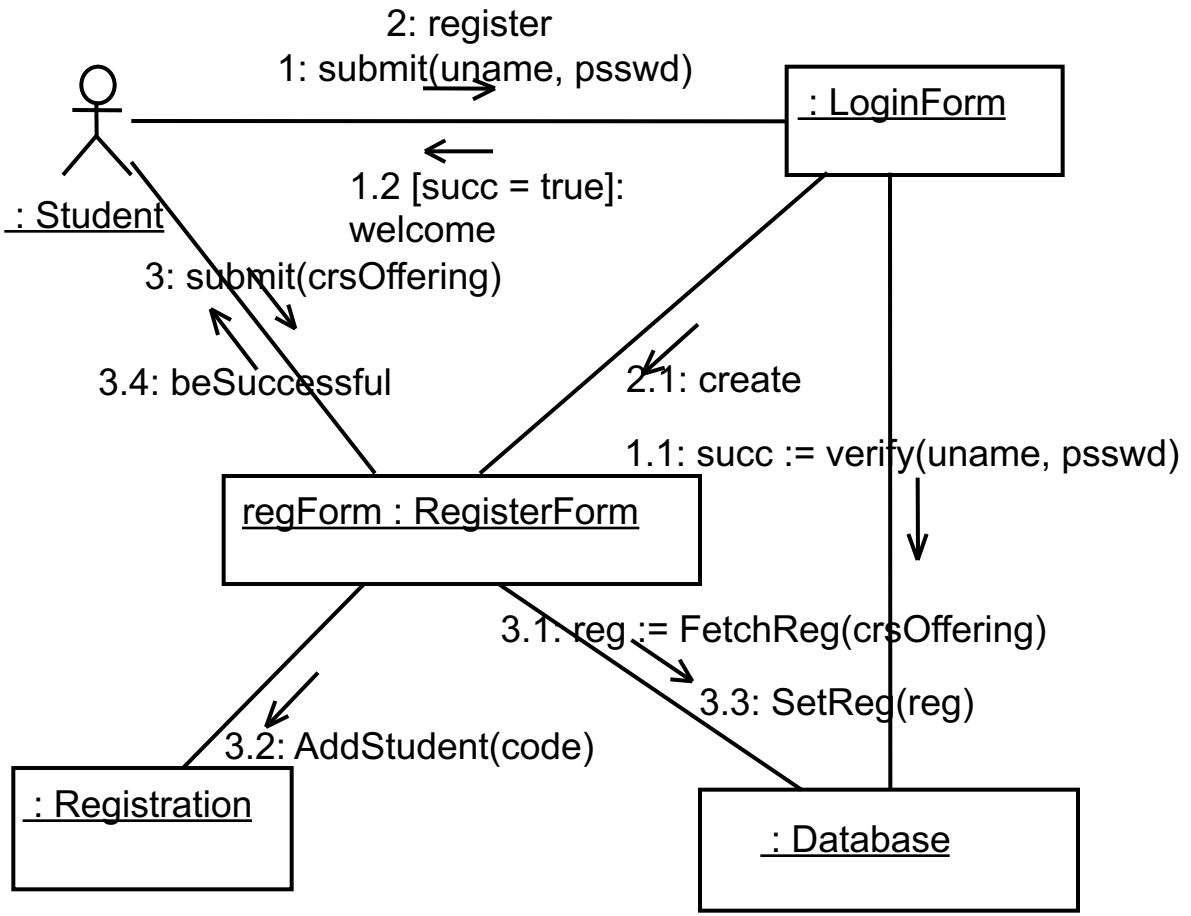
- Ví dụ: lược đồ cộng tác mức cụ thể cho *use-case*

  Login của hệ thống đăng ký môn học tín chỉ qua WEB



Lược đồ cộng tác – ví dụ 2

Ví dụ: lược đồ cộng tác mức cụ thể cho use-case Registers course của hệ thống đăng ký môn học tín chỉ qua WEB



II. Phân tích dữ liệu hệ thống

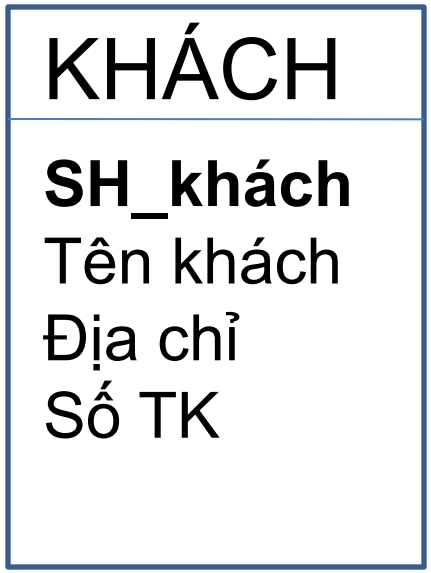
- Dùng lược đồ thực thể kết hợp (ERD) để mô hình hóa dữ liệu.
- Thiết kế dữ liệu.

Mô hình Thực thể - Kết hợp

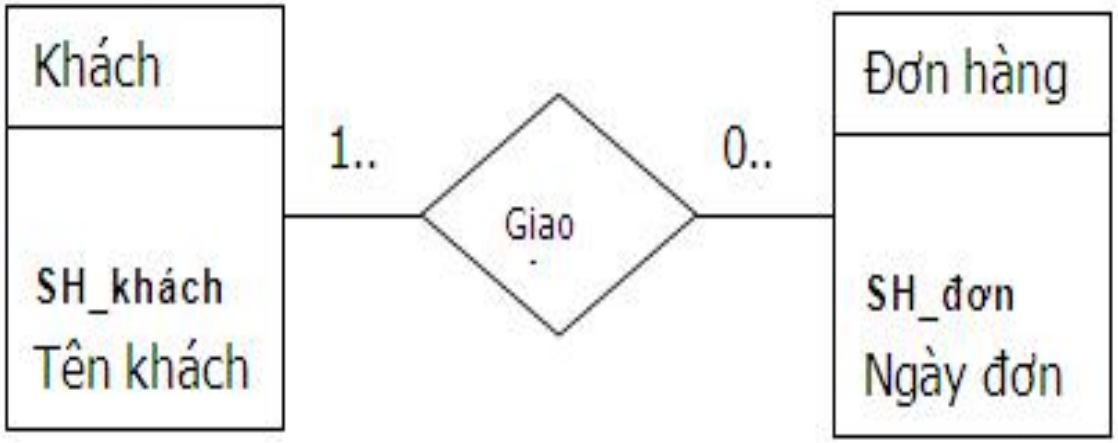
- Dùng để mô tả Dữ liệu ở mức quan niệm
- Dữ liệu là thông tin cần lưu lâu dài trong phần mềm.
- Là cơ sở tiến tới mức vật lý (xác định được có bao nhiêu bảng, và quan hệ giữa các bảng trong CSDL quan hệ)
- Giữa mức quan niệm và mức vật lý là mức logic.

Biểu tượng kí hiệu

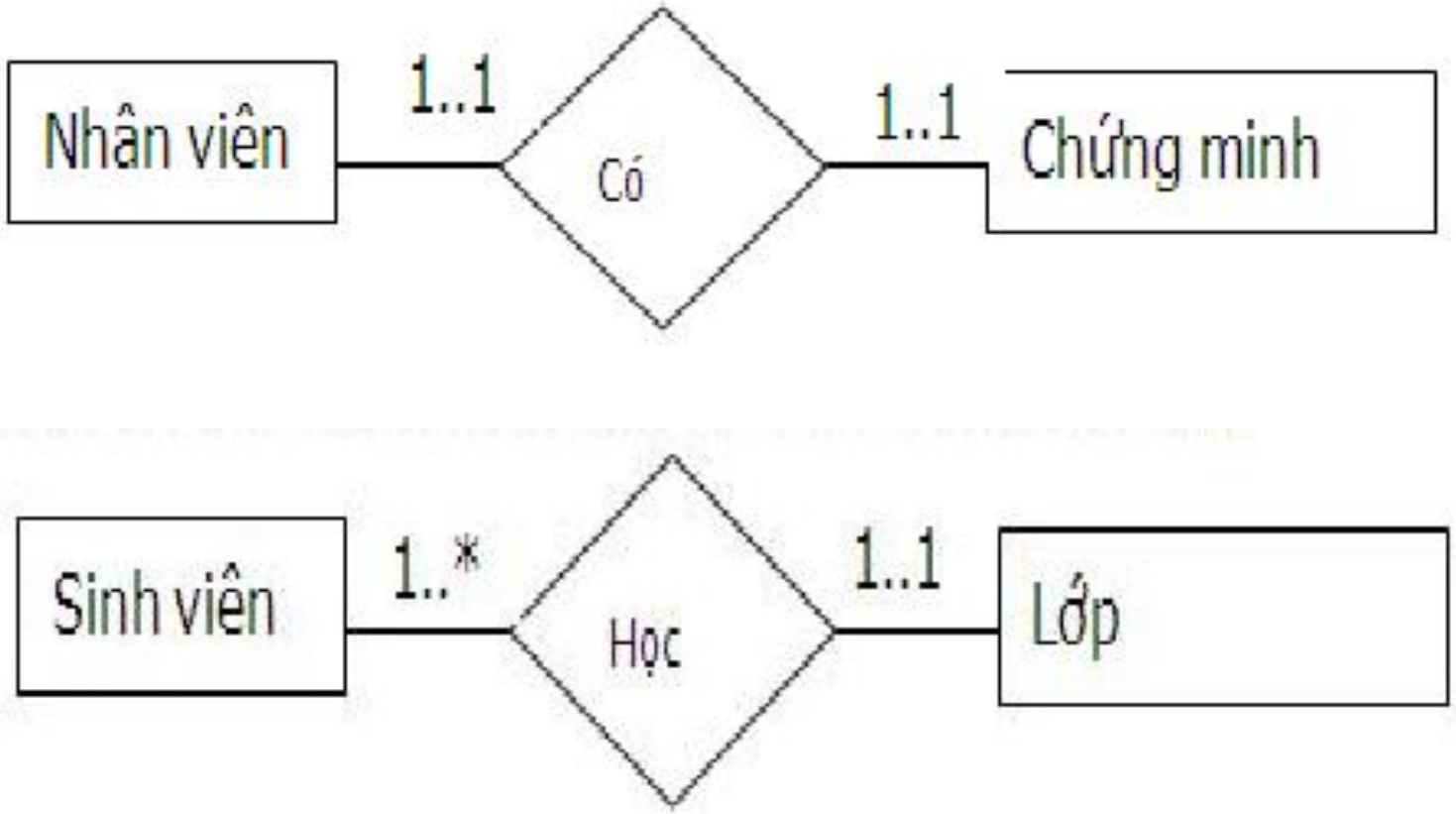
Thực thể



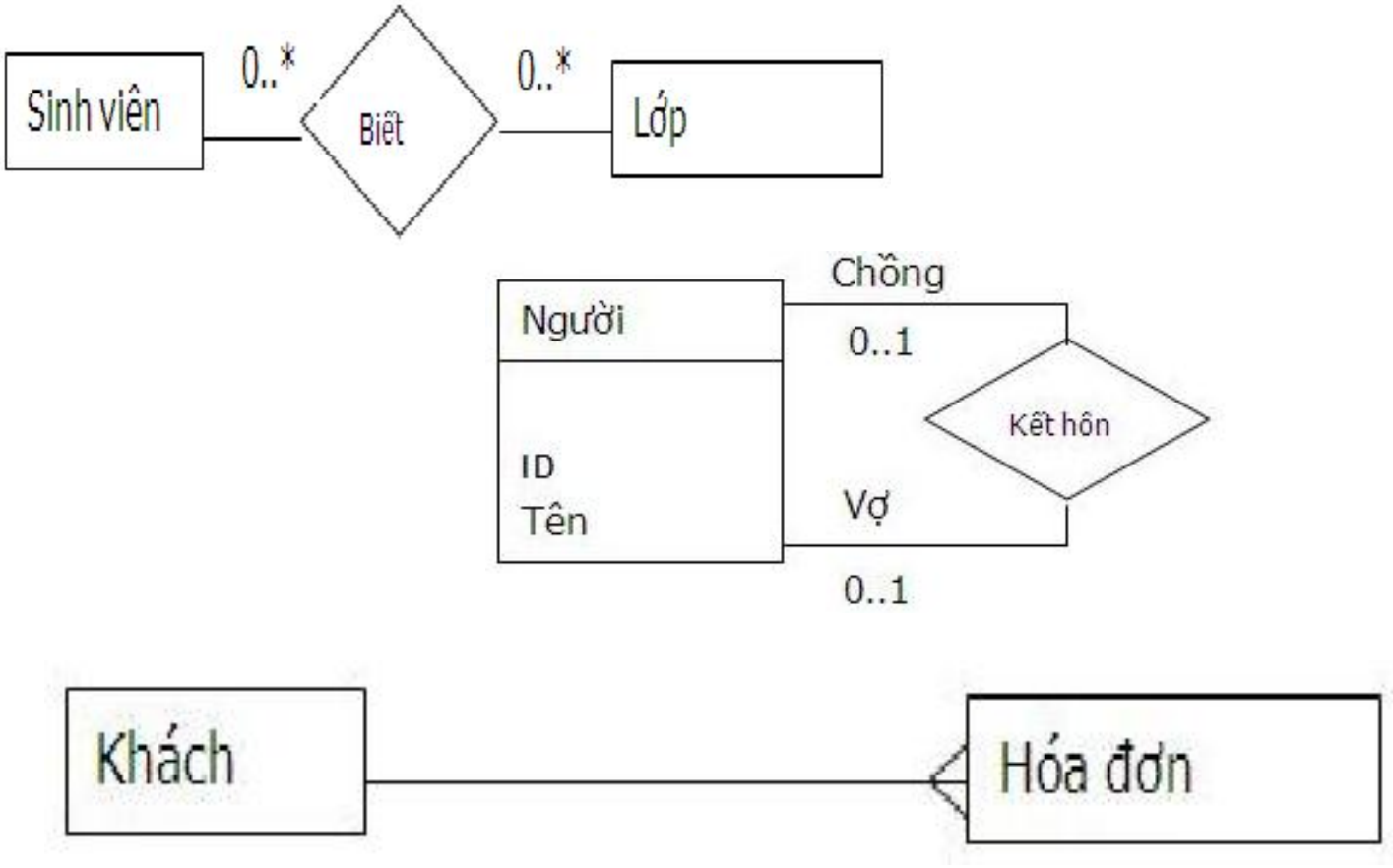
Kết hợp



Ví dụ Thực thể - Kết hợp



Ví dụ Thực thể - Kết hợp



Các dạng chuẩn

- **Dạng chuẩn 1:**

- Là quan hệ không chứa thuộc tính lặp. Tách thành 2 QH

+ Quan hệ 1: Gồm các thuộc tính lặp và phần khóa xác định chúng.

+ Quan hệ 2: Gồm các thuộc tính còn lại và khóa chính.

Ví dụ: Quan hệ **Đơn hàng**(Số đơn, Mã khách, Tên KH, Địa chỉ, Ngày đặt, Mã hàng*, Tên hàng*, Đơn vị*, Số lượng*) tách thành:

QH1 (Số đơn, Mã khách, Tên KH, Địa chỉ, Ngày đặt, Mã hàng)

QH2 (Mã hàng, Tên hàng, Đơn vị, Số lượng)

Các dạng chuẩn

- **Dạng chuẩn 2:**

- Là quan hệ:

+ Ở dạng chuẩn 1

+ Không tồn tại các thuộc tính không khóa phụ thuộc vào một phần khóa chính.

- Tách quan hệ ở dạng chuẩn 1 thành quan hệ ở dạng chuẩn 2

+ Quan hệ 1: Gồm các thuộc tính phụ thuộc vào 1 phần khóa chính và phần khóa xác định chúng.

+ Quan hệ 2: Gồm các thuộc tính còn lại và khóa chính.

Các dạng chuẩn

Ví dụ dạng chuẩn 2:

Điểm (Mã SV, Tên SV, Địa chỉ, Mã MH,Tên MH, Điểm)

Mã SV, Mã MH →Tên SV, Địa chỉ, Tên MH, Điểm

Mã MH → Tên MH

Mã SV → Tên SV, Địa chỉ

Tách thành 3 quan hệ ở dạng chuẩn 2 qua 2 phép tách

QH1(Mã MH,Tên MH)

QH2(Mã SV, Tên SV, Địa chỉ)

QH3(Mã SV, Mã MH, Điểm)

Các dạng chuẩn

Dạng chuẩn 3:

- Là quan hệ:

+ Ở dạng chuẩn 2

+ Không tồn tại các thuộc tính không khóa phụ thuộc bắc cầu vào khóa chính (qua một thuộc tính gọi là thuộc tính cầu).

- Tách quan hệ ở dạng chuẩn 2 thành quan hệ ở dạng chuẩn 3

+ Quan hệ 1: Gồm các thuộc tính phụ thuộc bắc cầu và thuộc tính cầu.

+ Quan hệ 2: Gồm các thuộc tính còn lại và thuộc tính cầu

Các dạng chuẩn

Ví dụ dạng chuẩn 3:

Đơn hàng (Số đơn, Mã khách, Tên khách, Địa chỉ, Ngày đặt)

Số đơn → Mã khách, Tên khách, Địa chỉ, Ngày đặt

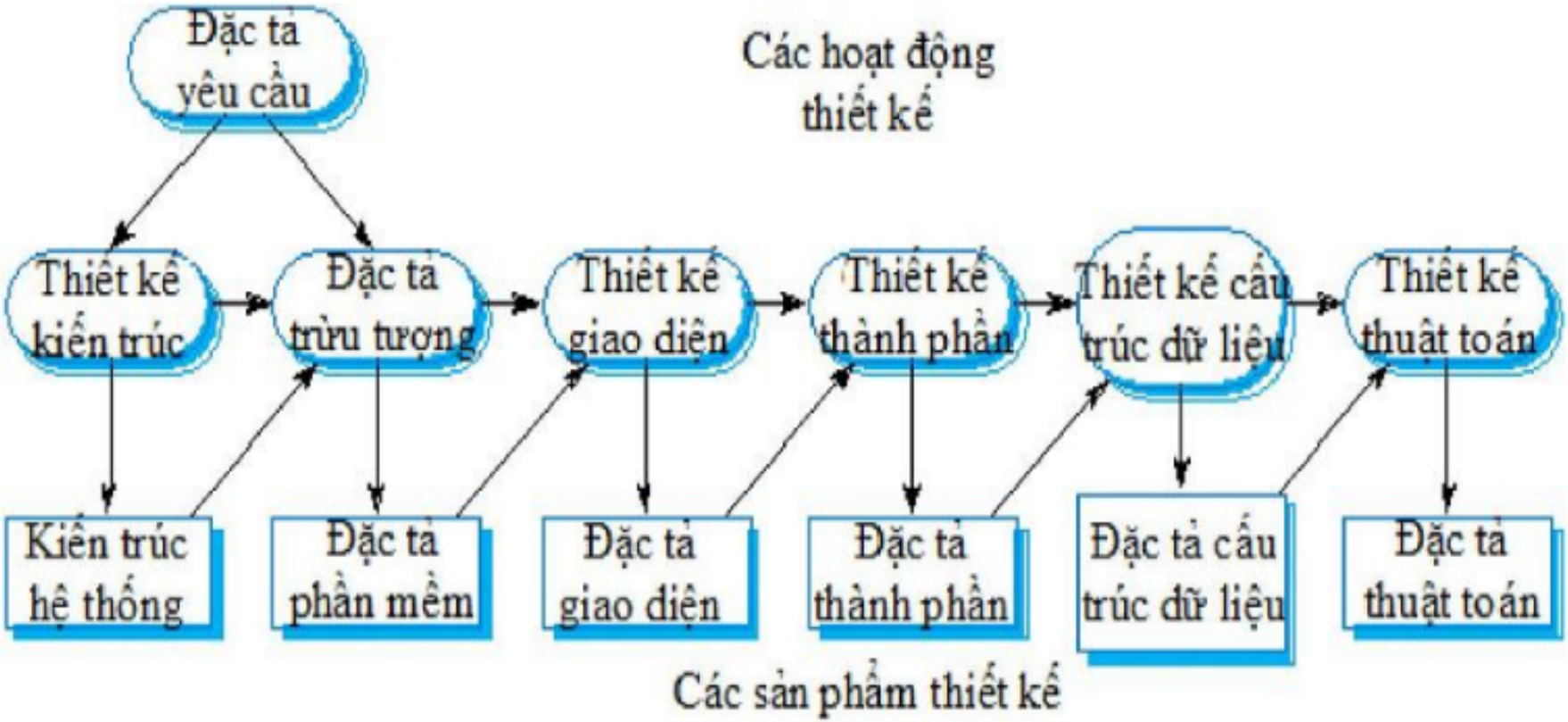
Mã khách → Tên khách, Địa chỉ

Tách thành 2 quan hệ

QH1(Mã khách, Tên khách, Địa chỉ)

QH2(Số đơn, Mã khách, Ngày đặt)

Các hoạt động chính của thiết kế



Các hoạt động chính của thiết kế

- Thiết kế kiến trúc:

  Xác định hệ tổng thể phần mềm bao gồm các hệ con và các quan hệ giữa chúng và ghi thành tài liệu
- Đặc tả trừu tượng:

  Các đặc tả trừu tượng cho mỗi hệ con về các dịch vụ mà nó cung cấp cũng như các ràng buộc chúng phải tuân thủ.

Các hoạt động chính của thiết kế

- Thiết kế giao diện:

  Giao diện của từng hệ con với các hệ con khác được thiết kế và ghi thành tài liệu; đặc tả giao diện không được mơ hồ và cho phép sử dụng hệ con đó mà không cần biết về thiết kế nội tại của nó
- Thiết kế các thành phần:

  Các dịch vụ mà một hệ con cung cấp được phân chia cho các thành phần hợp thành của nó

Các hoạt động chính của thiết kế

- Thiết kế cấu trúc dữ liệu:

  Các cấu trúc dữ liệu được dùng trong việc thực hiện hệ thống được thiết kế chi tiết và đặc tả
- Thiết kế thuật toán:

  Các thuật toán được dùng cho các dịch vụ được thiết kế chi tiết và được đặc tả.

III. Thiết kế giao diện

- Giao diện người dùng (User Interface)
  - Giao diện người dùng cần phải được thiết kế sao cho phù hợp với kỹ năng, kinh nghiệm và sự trông đợi của người sử dụng nó.
  - Người sử dụng hệ thống thường đánh giá hệ thống thông qua giao diện hơn là chức năng của nó.
  - Giao diện của hệ thống nghèo nàn có thể khiến người sử dụng tạo ra các lỗi hết sức nghiêm trọng.

Thiết kế giao diện

- Tác nhân con người trong thiết kế giao diện
  - Khả năng nhớ tức thời của con người hạn chế.
  - Nhu cầu của người dùng hệ thống
  - Kinh nghiệm, năng lực
    - □ khả năng dùng bàn phím, mouse,…
    - □ tốc độ phản ứng, khả năng nhớ thao
  - Sở thích, văn hóa, lứa tuổi
    - □ mầu sắc, ngôn ngữ, biểu tượng
    - các loại tương tác khác nhau: hình ảnh, văn bản, âm thanh,..

Thiết kế giao diện

- Các nguyên tắc thiết kế giao diên
  - Sự quen thuộc của người sử dụng:

    Giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính.

    Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy … mà không nên sử dụng những khái niệm như thư mục, danh mục …

Thiết kế giao diện

- Nhất quán:

  hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng …

- Tối thiểu hoá sự bất ngờ:

  Nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tư.

Thiết kế giao diện

- Khả năng phục hồi:

  hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi.
- Hướng dẫn người sử dụng:

  Như hệ thống trợ giúp, hướng dẫn trực tuyến …
- Tính đa dạng:

  Hỗ trợ nhiều loại tương tác cho nhiều loại người sử dung khác nhau.

Thiết kế giao diện

- Kỹ thuật: hai vấn đề chính cần giải quyết
  - Tương tác người dùng: cách người dùng đưa thông tin vào cho hệ thống
  - Biểu diễn thông tin: cách hệ thống trình diễn thông tin cho người dùng
  - Giải pháp được xem xét theo góc độ
    - Thiết bị tương tác người dùng
    - Cách hệ thống trình diễn - chủng loại giao diện
    - Mô hình tương tác

Thiết kế giao diện

- Thiết bị tương tác
  - Màn hình
  - Bàn phím
  - Mouse, bút từ, ...
  - Màn hình cảm biến
  - Mic/Speaker
  - Smart cards,…
- Cả thiết bị lẫn phương thức đều đang tiến hóa:
  - Nhận dạng tiếng nói, chữ viết…

Thiết kế giao diện

- Các loại giao diện:
  - Giao diện dòng lệnh:
    - □ Là phương thức tương tác đầu tiên □ Nhập lệnh/dữ liệu từ bàn phím
    - □ Dễ cài đặt so với GUI
      - □ thực hiện thông qua hàm chuẩn của ngôn ngữ
      - □ không tốn tài nguyên hệ thống

Thiết kế giao diện

- □ Có khả năng tổ hợp lệnh để tạo các lệnh phức tạp
  - – □ phối hợp các filter, tạo các lô xử lý (batch)
  - – □ có thể lập trình bằng (Unix) shell
  - – □ có thể tự động hóa
- Thao tác thực hiện tuần tự
  - – □ khó sửa lỗi thao tác trước đó
- Không phù hợp với người dùng ít kinh nghiệm
  - – □ khó học, khó nhớ □ dễ nhầm
  - –đòi hỏi kỹ năng sử dụng bàn phím

Thiết kế giao diện

- Giao diện đồ họa (GUI)
  - □ Là giao diện thông dụng trên PC, Apple, Unix WS
  - □ Dễ học, dễ sử dụng, thuận tiện với người ít kinh nghiệm
  - □ Có nhiều cửa sổ, có thể tương tác song song trên nhiều cửa sổ mà không bị mất thông tin
  - □ Có thể hiển thị, tương tác dữ liệu trên nhiều vị trí trong cửa sổ

Thiết kế giao diện

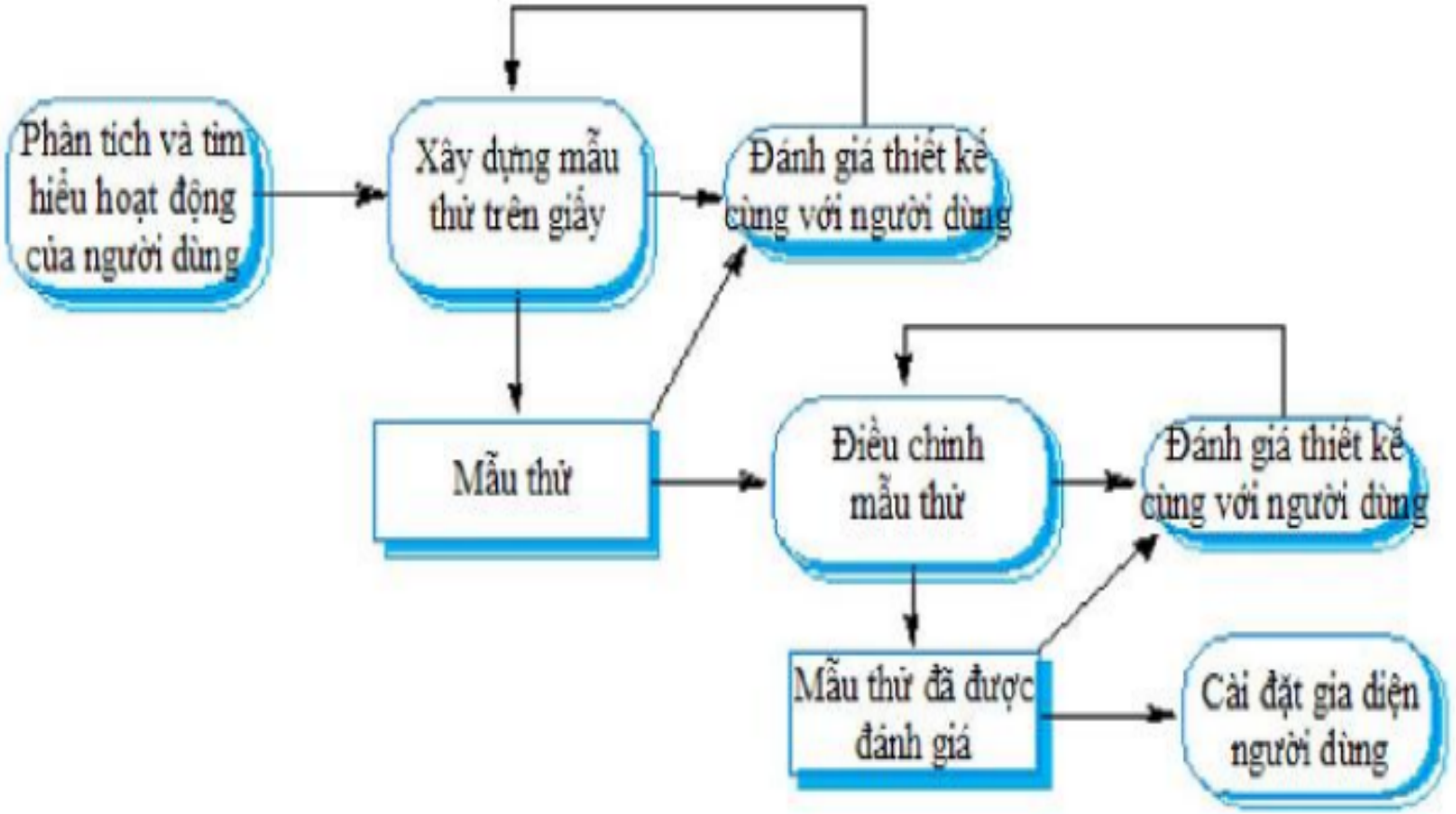
- Hình thức tương tác
  - Tương tác trực tiếp với thông tin ví dụ: soạn thảo; nhập dữ liệu vào các form…
    - dễ học, dễ sử dụng
    - nhận được tức thời kết quả thao tác
    - cài đặt phức tạp, tốn tài nguyên phần cứng
  - Tương tác gián tiếp ví dụ: chọn lệnh từ menu, giao diện dòng lệnh
    - kém trực quan
    - thuận tiện khi lặp lại thao tác phức tạp

Thiết kế giao diện

- Tính công nghệ
  - – □ Giao diện là phần tử dễ thay đổi
    - • □ thay đổi quy trình, phương thức thao tác
    - • □ thay đổi môi trường (phần cứng, hệ điều hành)
    - • □ nâng cấp (đẹp hơn, dễ sử dụng hơn…)
  - – Giao diện phải dễ sửa đổi
  - – Giao diện phải có tính khả chuyển
  - – Giao diện nên độc lập với xử lý

Thiết kế giao diện

- **Quy trình thiết kế giao diện**



Thiết kế giao diện

- Quy trình thiết kế giao diện
  - Phân tích người sử dụng:

    Tìm hiểu những gì người sử dụng sẽ làm với hệ thống.
  - Lập mẫu thử hệ thống:

    Xây dựng một tập các mẫu thử để thử nghiệm
  - Đánh giá giao diện:

    Thử nghiệm các mẫu thử cùng với người sử dụng.

Thiết kế giao diện

- Phân tích người sử dụng
  - Các kỹ thuật phân tích
    - Phân tích nhiệm vụ: mô hình hoá các bước cần thực hiện để hoàn thành một nhiệm vụ.
    - Phân tích nhiệm vụ phân cấp.
    - Phỏng vấn và trắc nghiệm: hỏi người sử dụng về những gì mà họ làm.
    - Mô tả: quan sát người sử dụng làm việc và hỏi họ về những cách mà không được biết tới

Thiết kế giao diện

- Lập mẫu thử giao diện người dùng
  - Mẫu thử cho phép người sử dụng có được những kinh nghiệm trực tiếp với giao diện.
  - Nếu không có những kinh nghiệm trực tiếp như vậy thì không thể đánh giá được khả năng có thể sử dụng được của giao diện.
  - Quy trình :
    - Lập các mẫu thử trên giấy.
    - Tinh chỉnh mẫu thử và xây dựng chúng

Thiết kế giao diện

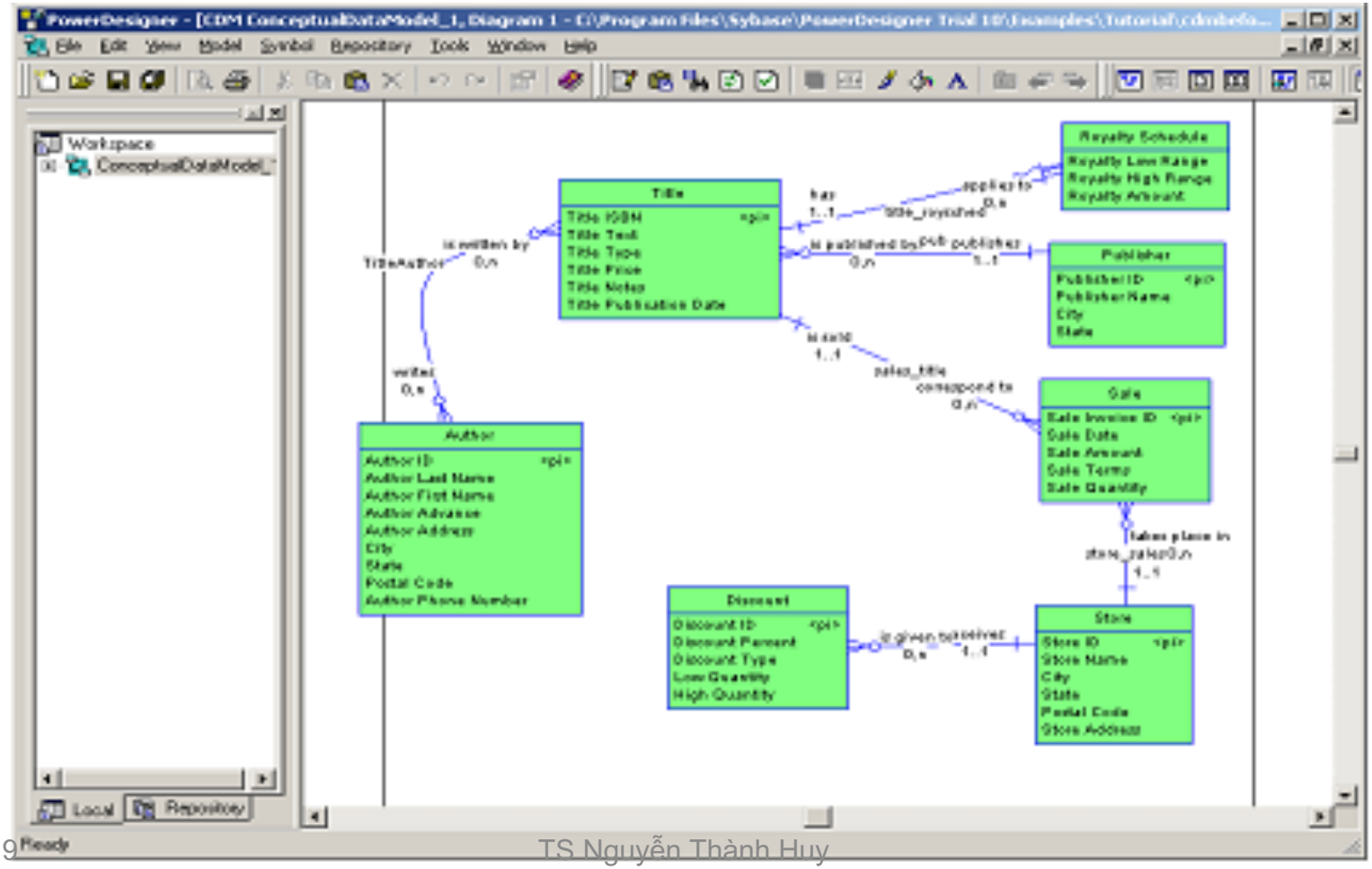
– Các kỹ thuật lập mẫu thử
  - Mẫu thử hướng nguyên mẫu:sử dụng công cụ như Macromedia Director để xây dựng một tập hợp các nguyên mẫu và màn hình. Khi người sử dụng tương tác với chúng thì màn hình sẽ thay đổi để hiển thị trạng thái kế tiếp.
  - Lập trình trực quan: sử dụng các ngôn ngữ được thiết kế cho việc phát triển nhanh như Visual Basic.
  - Mẫu thử dựa Internet: sử dụng web browser và script

Thiết kế giao diện

- Đánh giá giao diện người dùng
  - Các kỹ thuật:
    - Trắc nghiệm lại các phản hồi của người sử dụng
    - Ghi lại quá trình sử dụng mẫu thử của hệ thống và đánh giá nó.
    - Lựa chọn những thông tin về việc sử dụng dễ dàng và các lỗi của người sử dụng.
    - Cung cấp mã lệnh trong phần mềm để thu thập những phản hồi của người sử dụng một cách trực tuyến.

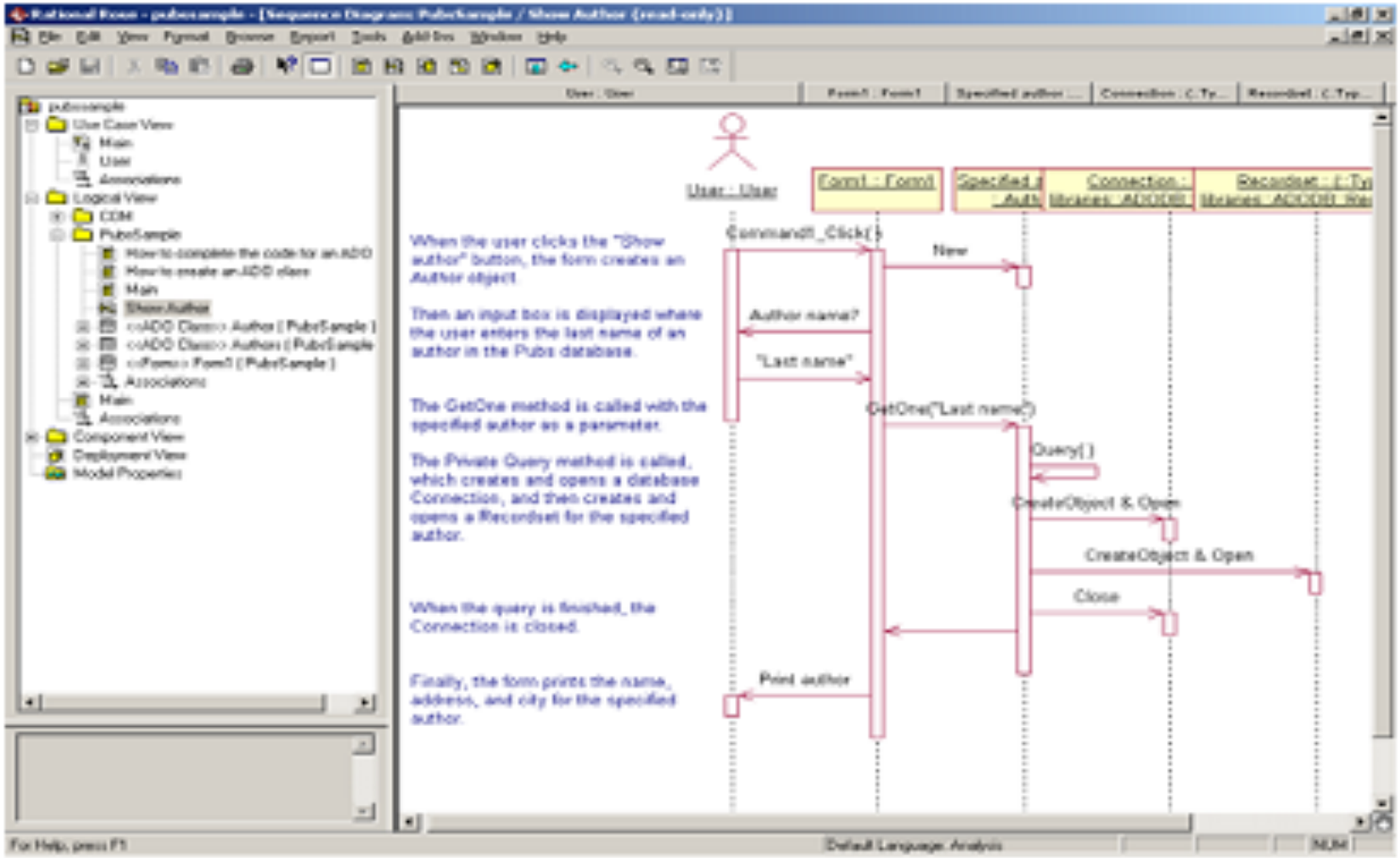
Công cụ thiết kế

– **Power Designer**



Công cụ thiết kế

- Rational Rose



Thảo luận

- Quan hệ giữa hướng chức năng và hướng đối tượng?
- Phân biệt mức quan niệm, logic, vật lý?
- Ai là người thực hiện công việc phân tích, thiết kế trong dự án phần mềm?
- Kết quả của giai đoạn phân tích, thiết kế để làm gì?