

# Công nghệ phần mềm

## Pha phân tích

# Mục tiêu

---

- Biểu diễn đặc tả yêu cầu bằng mô hình phân tích
- Mô hình phân tích đặc tả toàn bộ nội dung: chức năng, dữ liệu nhập/ xuất, các hoạt động của hệ thống cần phát triển
- Xây dựng các từ điển dữ liệu định nghĩa các k/n đặc thù của hệ thống, ý nghĩa, cấu trúc...
- Thống nhất với KH về mô hình và tính năng của hệ thống.

# Phân tích hệ thống

---

- Là pha rất quan trọng trong dự án PM
- Công việc phân tích hệ thống bao gồm:
  - Phân tích và xác lập các quy trình sẽ được phát triển/thay thế bằng máy tính
  - Xác thực các yc, tính năng của hệ thống

# Phân tích hệ thống

---

- Kết quả: là các tài liệu đặc tả tính năng hệ thống (thường ở dạng các sơ đồ, biểu đồ...)
- Kết quả này dùng cho việc xác thực các tính năng của hệ thống với khách hàng
- Kết quả này là đầu vào của quá trình tiếp theo là thiết kế hệ thống.
- Tùy thuộc vào công nghệ phát triển mà sử dụng các phương pháp phân tích phù hợp : cấu trúc hay OOP

# Những vấn đề trong phân tích hệ thống

---

- Cách biệt về chuyên môn của lĩnh vực cần phân tích.
- Sự hiểu biết của những người end user về quy trình làm việc và khả năng ứng dụng phần mềm cho công việc của họ.
- Những vấn đề về điều kiện hạ tầng hỗ trợ hoạt động của hệ thống

# Những vấn đề trong phân tích hệ thống

---

- Tính sẵn sàng thông tin của các hệ thống đang có sẽ tương tác với hệ thống cần xây dựng.
- Định hướng ứng dụng lâu dài chưa có/ chưa rõ ràng.
- Công cụ/ngôn ngữ sử dụng để đặc tả hệ thống / kết quả phân tích

# Quy trình phân tích hệ thống

---

- Phân tích các yc theo quy trình xử lý
- Bổ sung các quy trình cho phù hợp với máy tính
- Yc bổ sung các thông tin
- Xác lập các chức năng hệ thống (điều kiện và môi trường hoạt động)
- Xác thực với người dùng về tính hợp lý và đầy đủ của các tính năng (xác thực quy trình nghiệp vụ, các ràng buộc)

# Quy trình phân tích hệ thống

---

- Mô hình nghiệp vụ (chức năng)
  - Biểu đồ phân rã chức năng
  - Ma trận thực thể - chức năng
  - Mô hình ca sử dụng
  - Biểu đồ hoạt động
- Biểu đồ chuyển trạng thái
- Mô hình luồng dữ liệu
- Biểu đồ thực thể + mối quan hệ (dữ liệu)



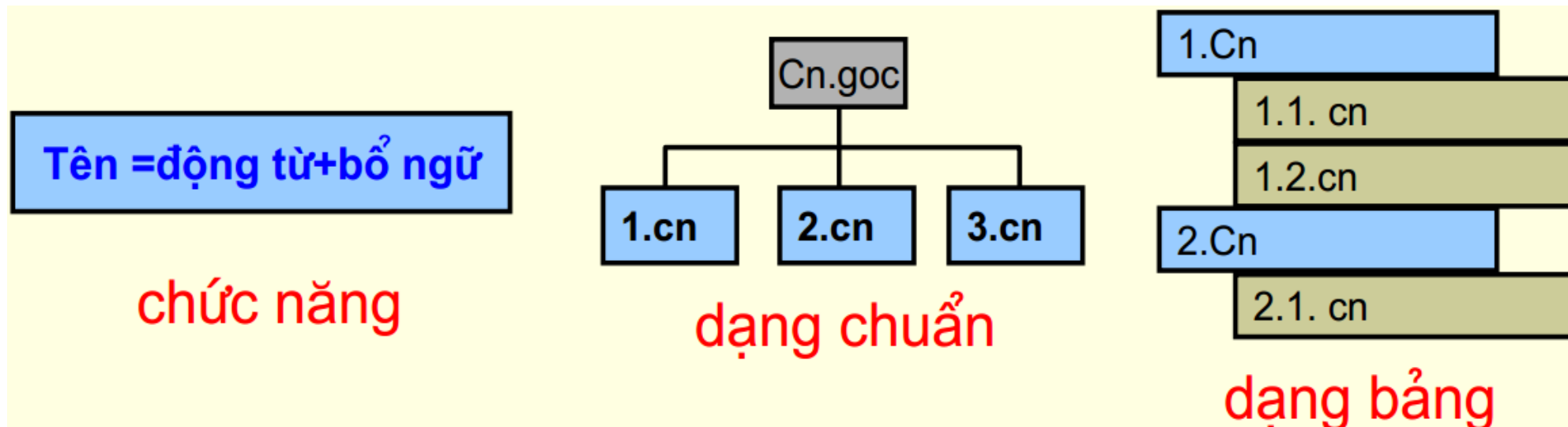
# Phương pháp và Công cụ

---

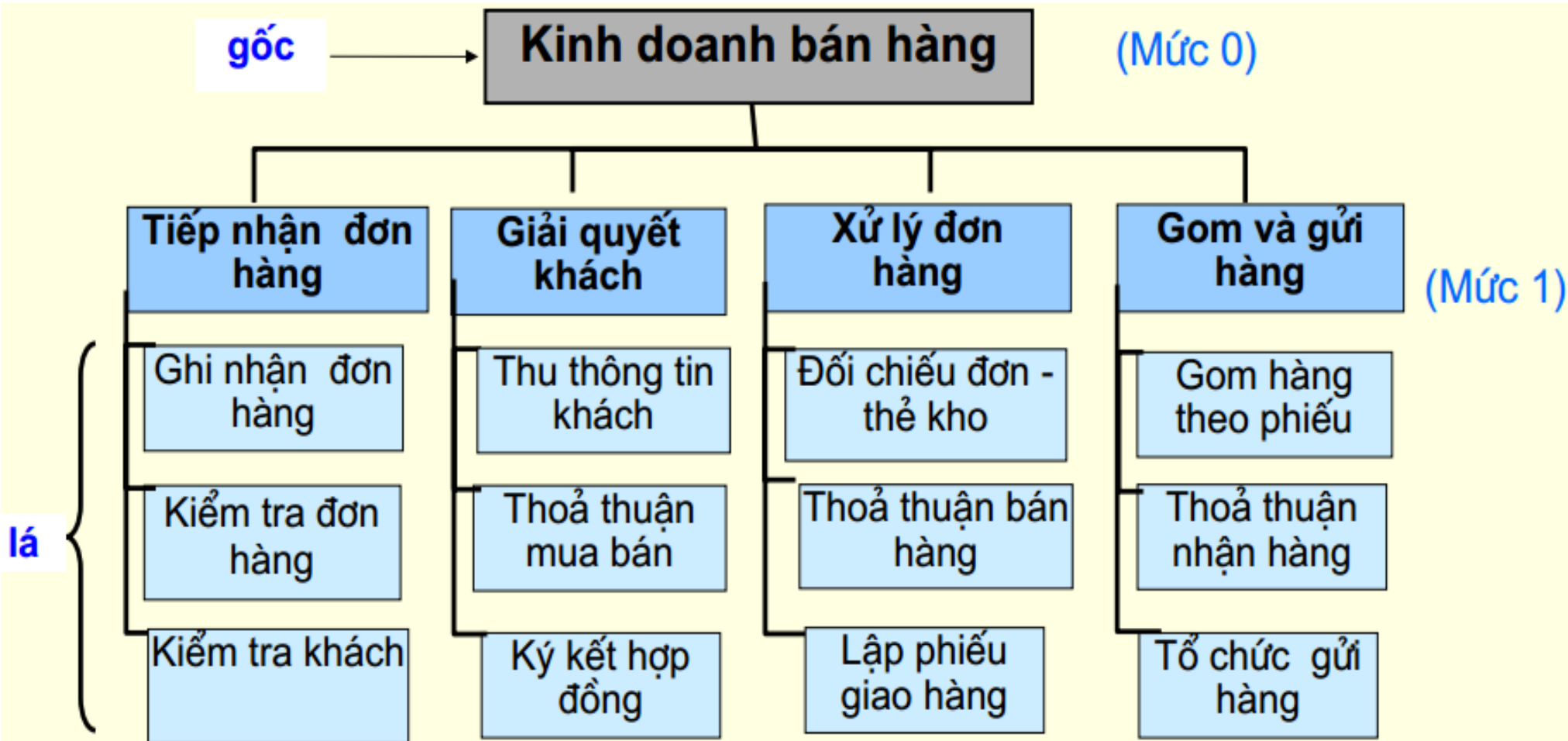
- Phương pháp cấu trúc:
  - Các bước được thực hiện đồng thời và xen kẽ nhau
  - Thường sử dụng lược đồ: DFD, ERD, STD
- Phương pháp HĐT:
  - Sử dụng UML: lược đồ use case,

# Biểu đồ phân rã chức năng

- Function Decomposition Diagram - FDD
  - Xác định phạm vi của hệ thống
  - Phân hoạch chức năng
  - Tạo nền tảng cho thiết kế kiến trúc hệ thống



# Biểu đồ phân rã chức năng



**Biểu đồ phân rã chức năng nghiệp vụ kinh doanh bán hàng**

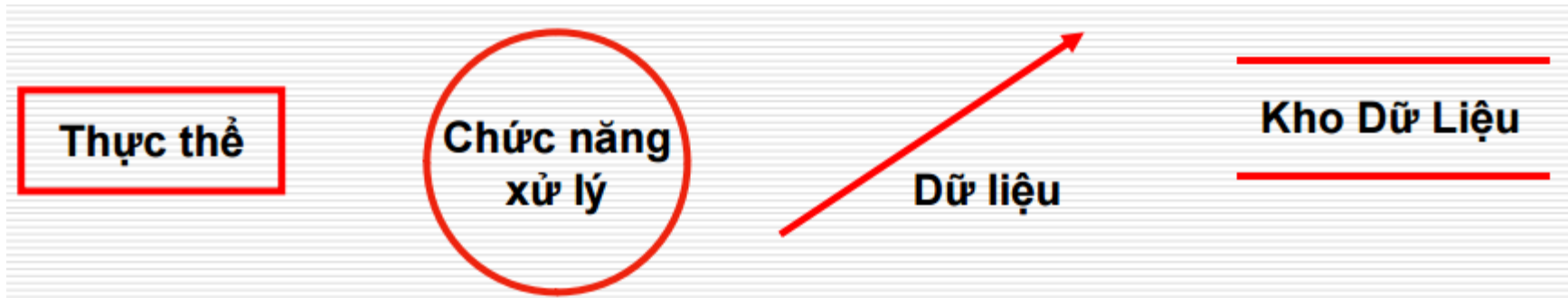
# Biểu đồ luồng dữ liệu

---

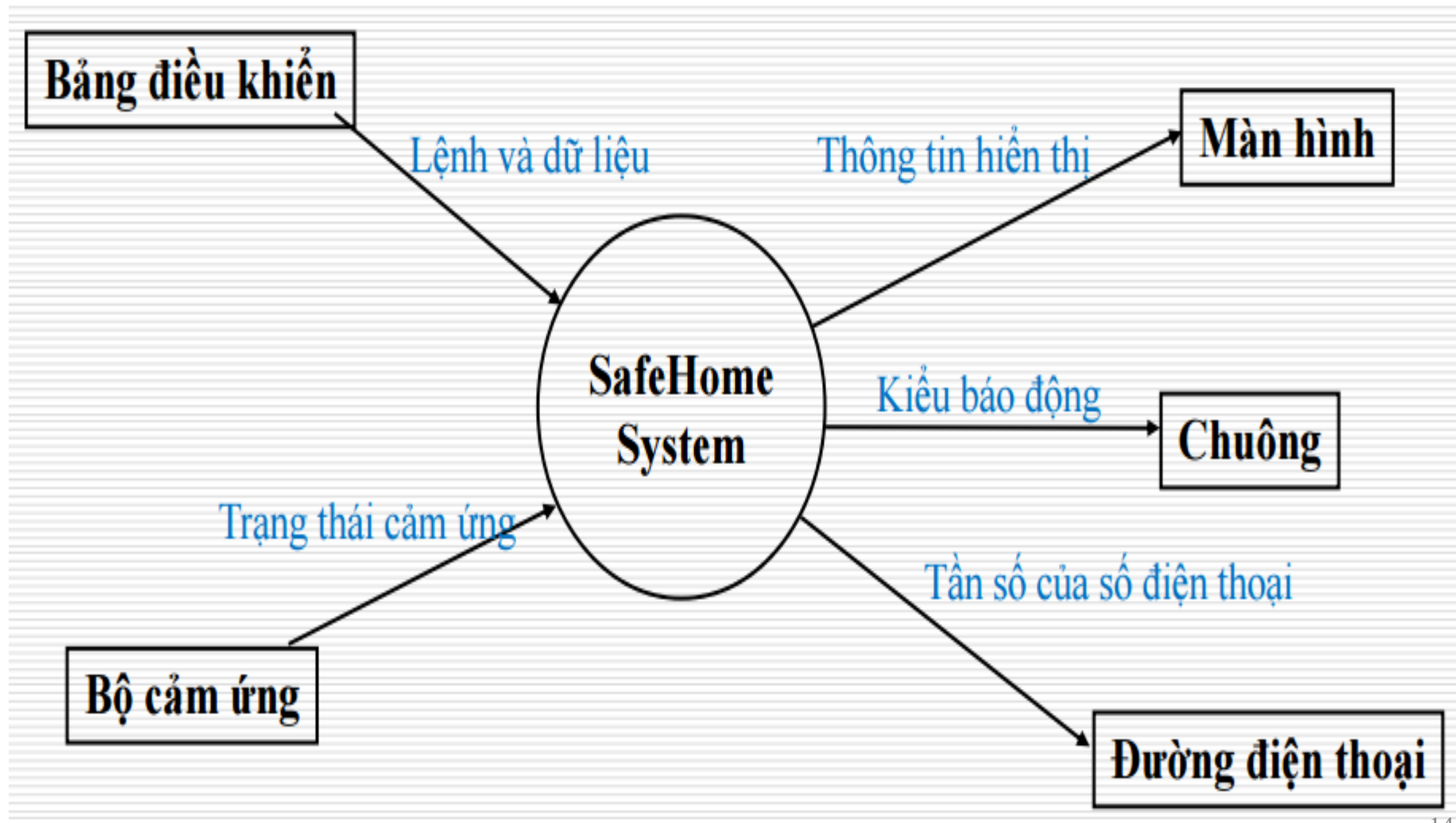
- **Data Flow Diagram - DFD**
- Mô tả quá trình hoạt động xử lý thông tin nghiệp vụ
- Biểu diễn cách thức dữ liệu di chuyển, được xử lý, lưu trữ trong hệ thống và trao đổi với môi trường
- Có nhiều mức trừu tượng khác nhau (phân tích có cấu trúc)
- Làm cơ sở cấu trúc hệ thống phần mềm

# Lược đồ luồng dữ liệu DFD

- **Thực thể:** tạo ra hoặc tiêu thụ thông tin, nằm bên ngoài biên giới của phạm vi thông tin hệ thống.
- **Chức năng xử lý:** thực hiện chức năng nào đó, tiêu thụ và tạo ra thông tin, nằm bên trong phạm vi thông tin hệ thống.
- **Thông tin** hay dữ liệu.
- **Kho dữ liệu:** lưu trữ dữ liệu mà được sử dụng bởi nhiều chức năng xử lý



# Lược đồ luồng dữ liệu DFD

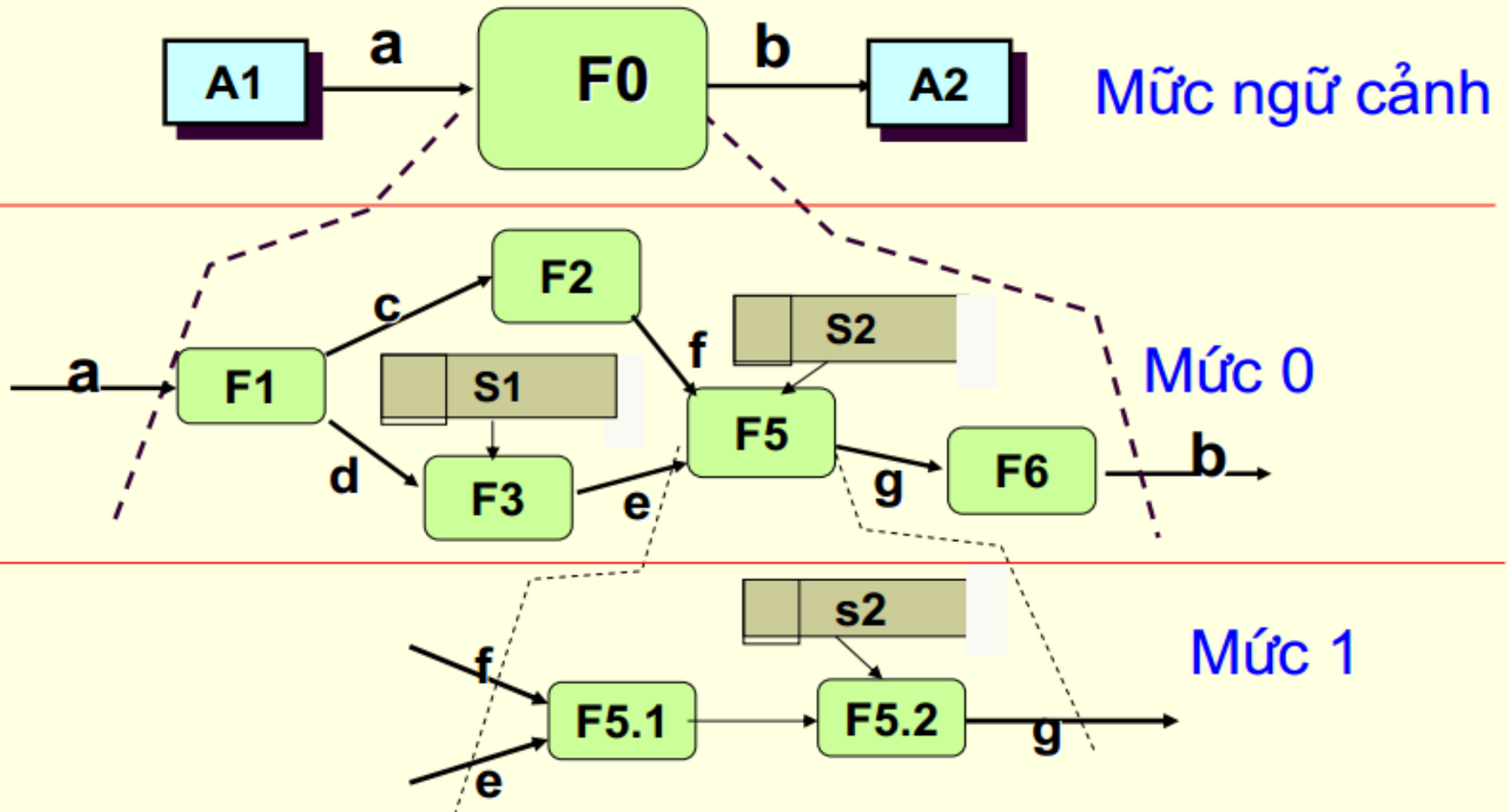


# Lược đồ luồng dữ liệu DFD

---

- DFD được xây dựng qua nhiều mức khác nhau: mức 0, 1, 2...
- DFD mức 0 còn được gọi là “Fundamental System Model” hay “Context Model” , đại diện cho toàn bộ hệ thống với một hình tròn duy nhất với các đường input và output data.
- DFD mức sau chi tiết hơn mức trước

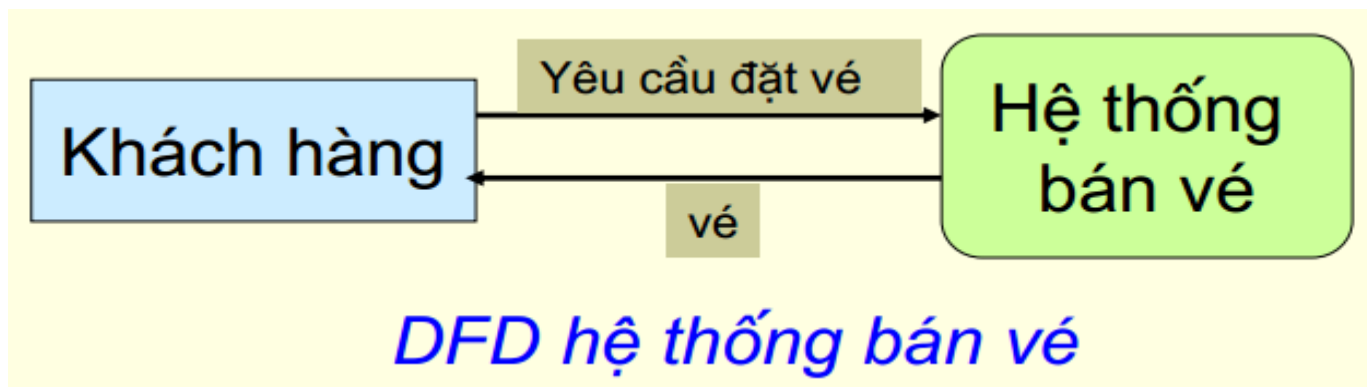
# Lược đồ luồng dữ liệu DFD



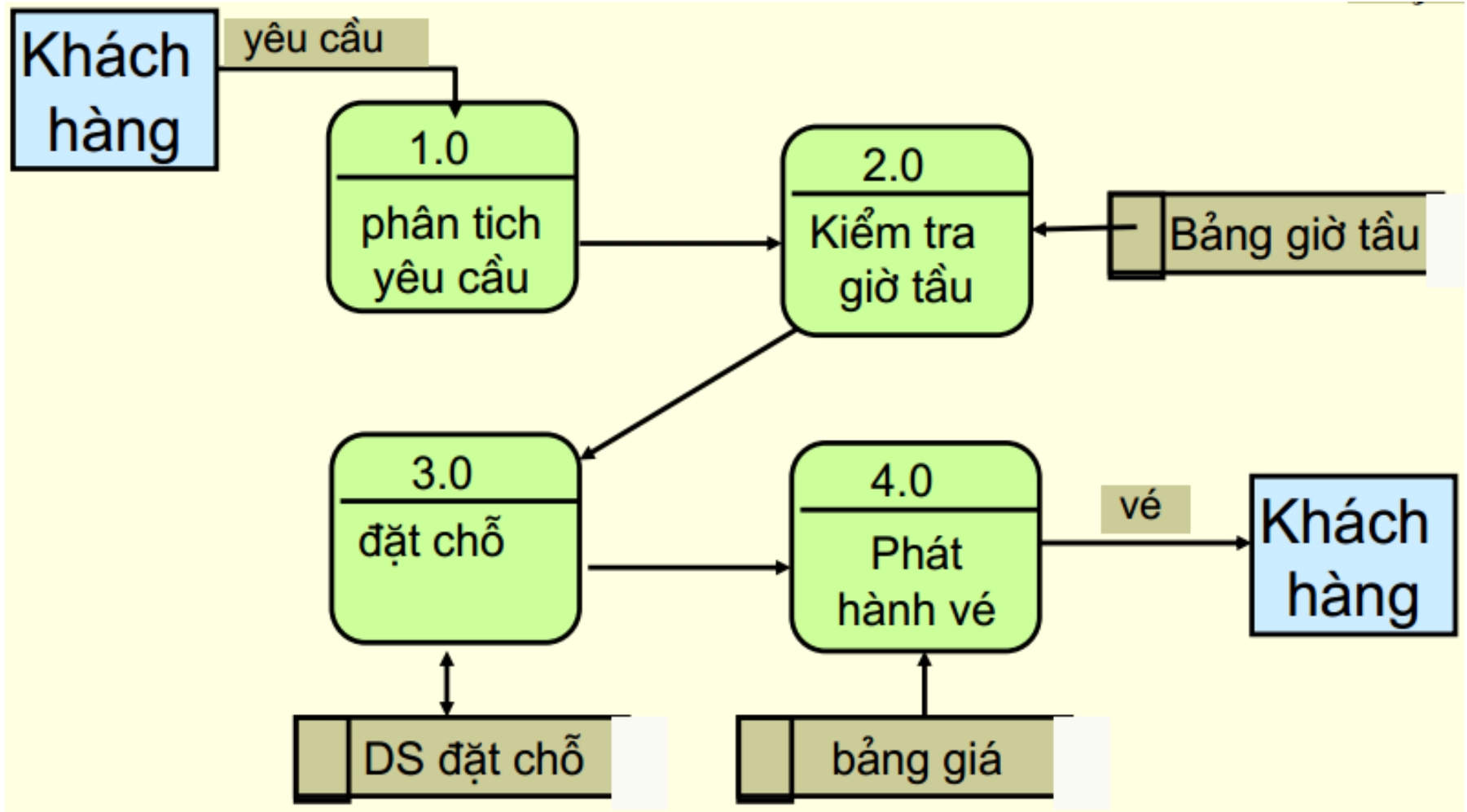


# Biểu đồ ngữ cảnh hệ thống

- Mô tả hệ thống trong môi trường
- **Hệ thống:** là 1 tiến trình duy nhất
- **Môi trường:** các tác nhân có tương tác với hệ thống
- **Tương tác:** các luồng DL giữa các tác nhân – hệ thống

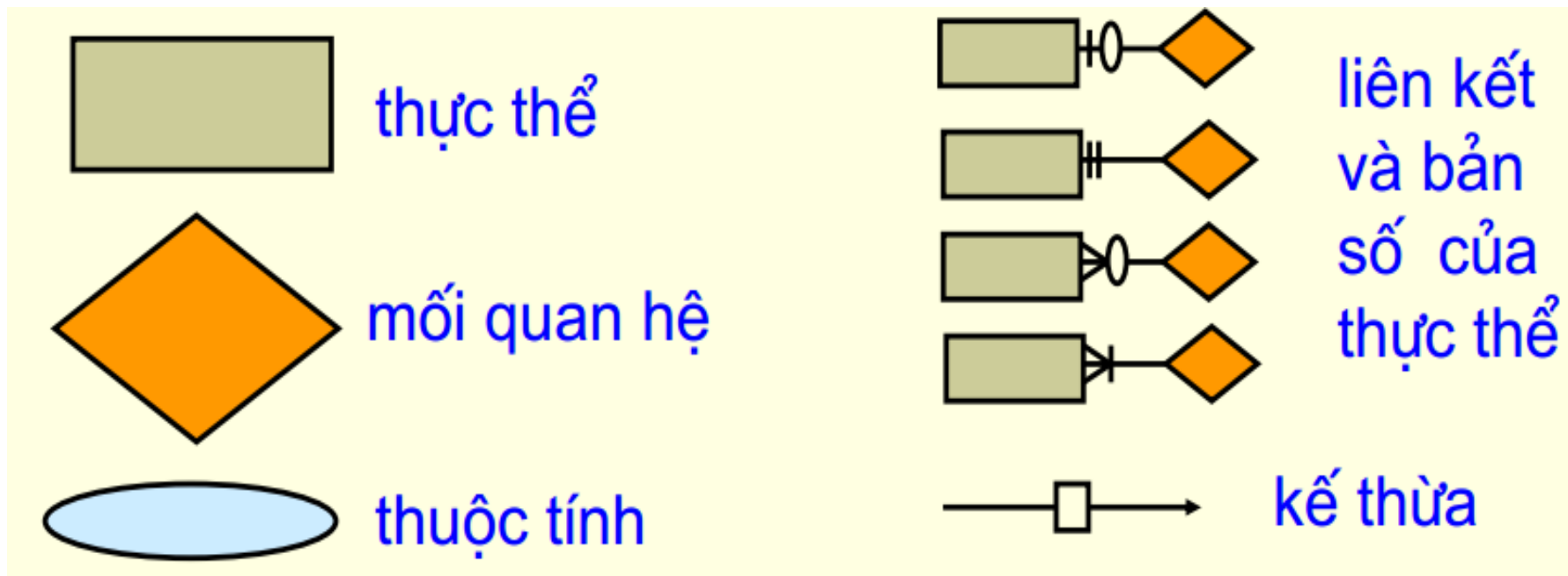


# Biểu đồ DFD mức 0

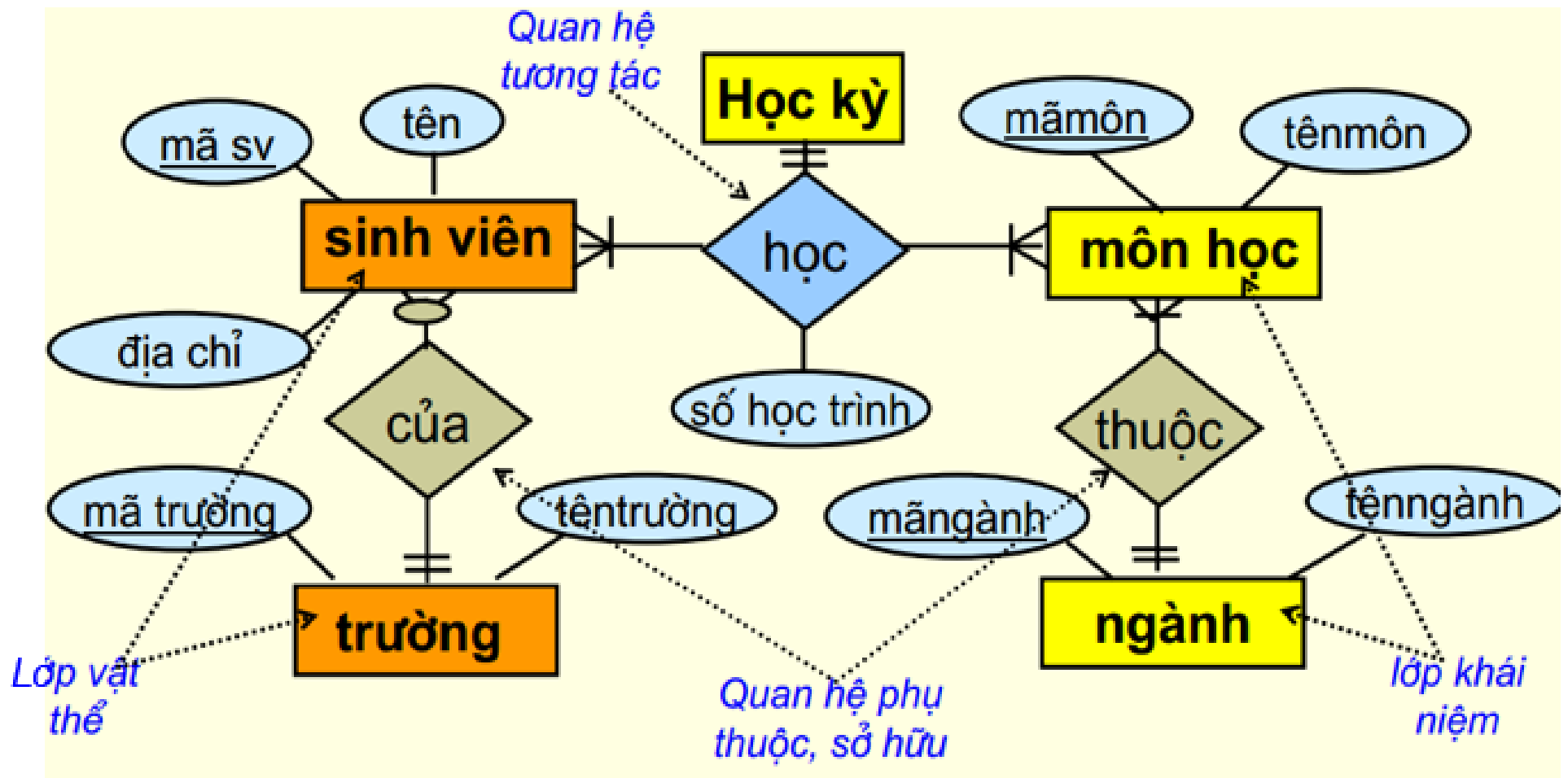


# Mô hình thực thể - mối quan hệ

- Entity Relationship Model – ER/ ERM
- Các phần tử trong mô hình:



# Mô hình thực thể - mối quan hệ



# Tầm quan trọng của ER

---

- Phân tích DL độc lập với việc lưu trữ, xử lý
- Nghiên cứu phạm vi miền thông tin
- Tạo ra mô hình trừu tượng hướng KH (mô hình khái niệm) trực quan: làm công cụ giao tiếp
- Xác định mqh mang tính cấu trúc giữa các DL.
- Dễ dàng chuyển sang mô hình thiết kế

# Các bước xây dựng

---

## 1. Đầu vào:

- Các hồ sơ sử dụng (đã, sẽ dùng)
- Từ điển dữ liệu (mô tả chi tiết mỗi mục tin)

## 2. Các bước:

- Liệt kê, chính xác hóa, chọn mục tin từ hồ sơ
- Xác định các thực thể và thuộc tính
- Xác định các mqh và thuộc tính
- Vẽ biểu đồ mô hình
- Rút gọn, hoàn thiện mô hình

# Từ điển dữ liệu

---

- Nhiều phần tử được tạo ra trong mô hình phân tích: DL, chức năng, điều khiển...
- Phải có 1 cách thức quản lý các phần tử đó sao cho hiệu quả → ***Từ điển dữ liệu***

# Từ điển dữ liệu

---

- **Đ/n:**

Từ điển DL là 1 ds có tổ chức của tất cả các phần tử DL cần thiết cho hệ thống. Các phần tử được định nghĩa chính xác và chặt chẽ, sao cho cả người phân tích và KH cùng chia sẻ 1 suy nghĩ về chúng.



# Từ điển dữ liệu

---

- Phương pháp (văn phạm) có tính hình thức để mô tả DL mà hệ thống xử lý
- Ký pháp để mô tả các dữ liệu điều khiển và miền giá trị của chúng (vd: on/off)
- Chứa đựng các thông tin về nơi (mô đun) và cách thức xử lý DL.
- Có các ràng buộc nghiệp vụ lên nó
- Thường được tạo bằng các công cụ trợ giúp

# Khoản mục từ điển DL

---

- Tên (Name): tên DL
- Bí danh (Aliases): tên gọi khác
- Vị trí (Where): nơi (modul) xử lý/sử dụng
- Cách thức (How): cách thức xử lý, vai trò của DL
- Ký pháp (Description): ký pháp mô tả DL
- Định dạng (Format): kiểu, quy mô, giá trị mặc định...
- Quy tắc nghiệp vụ (rules): ràng buộc liên quan đến DL khác.

# VD: từ điển DL

---

- Tên: Số điện thoại
- Bí danh: không
- Được dùng ở đâu/ ntn:
  - Output của Thiết lập điều kiện báo động
  - Input của quay số.
- Đặc tả nội dung:
  - Số điện thoại = [mở rộng địa phương | số bên ngoài]
  - Mở rộng địa phương = [ 2010 | 2011 | ... |2016]
  - Số bên ngoài = 9 + [số địa phương | số đường dài]
  - Số địa phương = tiền tố + <chuỗi 4 ký số>
  - Số đường dài = (1) + mã vùng + số địa phương
  - Tiền tố = [795 | 799 | 850 | 999]

# Làm bản mẫu trong phân tích

---

- Trong nhiều trường hợp, mô hình mẫu là phương pháp duy nhất xác định yêu cầu PM lớn, phức tạp

 bản mẫu = trực quan.

- ☐ Bản mẫu PM: phát triển yc
- ☐ Bản mẫu phần cứng: kiểm tra thiết kế

# Các bước làm bản mẫu

---

- Bước 1: Đánh giá yc và xác định có nên làm bản mẫu không?
  - Độ phức tạp, chủng loại, khách hàng.
- Bước 2: Biểu diễn vắn tắt yc
  - Yc chức năng/ phi chức năng
- Bước 3: Thiết kế nhanh
  - Kiến trúc, cấu trúc DL

# Các bước làm bản mẫu

---

- Bước 4: Phát triển, kiểm thử
  - Sử dụng các thành phần sẵn có
  - Dùng các ngôn ngữ bậc cao
  - Các thuật toán dễ cài đặt
- Bước 5: Người dùng đánh giá
  - Bước quan trọng nhất???
- Bước 6: Lặp lại từ B2 -> B5 cho đến khi đủ yc

# Ưu nhược điểm của làm bản mẫu

---

- Loại bớt hiểu lầm
- Phát hiện thiếu hụt chức năng
  - Vd: soạn thảo kéo theo kiểm tra chính tả
- Phát hiện các điểm yếu:
  - Khó sử dụng
  - Thao tác không an toàn.
- Kiểm tra tính khả thi/hữu ích
  - Có thể xây dựng được không?
  - Có thực sự cần không?

# Ưu nhược điểm của làm bản mẫu

---

- Làm cơ sở cho đặc tả
  - Làm phần mềm giống thế này nhưng tốt hơn
- Huấn luyện người sử dụng
- Hỗ trợ kiểm thử ( so sánh kết quả)

 **Làm bản mẫu là kỹ thuật tránh rủi ro**



# Ưu nhược điểm của làm bản mẫu

---

- Các yc phi chức năng thường không được thể hiện đầy đủ
- Làm tăng chi phí: cần phải ước lượng chính xác chi phí của bản mẫu
- Các yc thay đổi quá nhanh → Bản mẫu trở nên vô nghĩa
- Người dùng không dùng bản mẫu theo cách thông thường (kết quả đánh giá không có giá trị)

# Ngôn ngữ dùng làm bản mẫu

---

- Java
- Visual Basic
- Prolog, Lisp, Python
- Các ngôn ngữ script khác
  - Perl, php, shell script

# Bản mẫu trực quan

---

- Sử dụng các ngôn ngữ lập trình trực quan:
  - VB
  - Delphi, Jbuilder
  - FrontPage
- Sử dụng lại 1 lượng lớn thư viện sẵn có
- Tính cấu trúc không cao, khó tích hợp kết quả của nhiều nhóm, bảo trì.

# Các phương pháp đặc tả

---

- Có nhiều phương pháp đặc tả:
  - Phi hình thức
  - Bán hình thức
  - Hình thức

# Đặc tả phi hình thức

---

- Sử dụng ngôn ngữ tự nhiên
- Bảng bảng, sơ đồ, hình vẽ tùy chọn

- Không chặt chẽ, dễ hiểu lầm



- Nhiều cách biểu diễn cho 1 k/n

- Khó phân hoạch, khó sửa đổi

# Đặc tả bán hình thức

---

- Đặc tả dựa trên mô hình:
  - Khái niệm
  - Ký pháp đồ họa
  - Quy tắc biểu diễn
  - Các bước xây dựng
  - Giải thích
- ❖ Mức hình thức hóa khác nhau, tùy mức độ hình thức hóa các k/n & quy tắc biểu diễn
- ❖ Xử lý mô hình cần sự tham gia của con người
- ❖ Nhiều loại mô hình có ứng dụng rộng rãi

# Đặc tả bán hình thức

---

- ❖ Mô hình DFD
- ❖ Mô hình ER
- ❖ Mô hình DL quan hệ
- ❖ Các mô hình nghiệp vụ
- ❖ Mô hình UML

# Đặc tả hình thức

---

- **Biểu diễn mô hình toán học**
  - K/n và ký pháp toán học
  - Quy tắc khai báo và biểu diễn biểu thức
  - Các luật biến đổi biểu thức
- ❖ Có thể tự động hóa giải mô hình
- ❖ Sử dụng mô hình cho chứng minh, kiểm chứng
- ❖ Khó hiểu, khó sử dụng

**“không ai biết chắc phải làm gì khi chưa có đặc tả”**



# Đặc tả hình thức

---

- **Ưu điểm**
  - Tính duy nhất của định nghĩa
- **Nhược điểm**
  - Khó hiểu, tốn thời gian mô tả
  - Không áp dụng được với mọi bài toán (các yc phi chức năng)

**“Cần phối hợp giữa đặc tả hình thức & phi hình thức”**

# Ngôn ngữ đặc tả hình thức

---

- **Đặc tả yc:**
  - Z: mô tả biến đổi trạng thái
  - Lưới Petri
  - Đặc tả đại số
  - Phương pháp máy trạng thái
  - VDM
- **Đặc tả thiết kế:**
  - RAISE
  - Flowchart, pseudo code

# Review – Phân tích hệ thống theo cấu trúc

---

- **Phân tích yc theo phương pháp cổ điển bao gồm:**
  - Mô hình chức năng DFD
  - Mô hình dữ liệu ER
  - Mô hình hành vi STD
- DFD cơ bản có 4 ký hiệu và nó được mở rộng để biểu diễn được các hệ thống thời gian thực
- Xây dựng DFD mức 0 rồi đến các mức cao hơn; chú ý bảo toàn tính liên tục của DL.
- Từ điển DL giúp quản lý và tra cứu các phần tử DL

# Phân tích hệ thống theo OOP

---

- **Tiếp cận của phương pháp phát triển OOP cho bước phân tích hệ thống:**

**Ai/ Vị trí như thế nào/ Làm gì/ Khi nào**

- **Các lược đồ:**
  - Lược đồ Use case: thu thập yc – mô hình nghiệp vụ
  - Lược đồ lớp: phân tích hệ yc – mô hình phân tích

# Mô hình nghiệp vụ - Thu thập yc

- Quan điểm thu thập/ phân tích yc của mô hình nghiệp vụ: hệ thống gồm có **Ai/ Làm những gì/ Khi nào**
- Lược đồ Use-case
  - Actor và use-case
  - Các mqh: Actor – Actor, Actor – Use-case, Use-case – Use-case.
- Actor xác định 1 bộ vai trò mà người hoặc vật sẽ đóng vai khi tương tác với hệ thống PM
  - Actor nằm ngoài phạm vi của hệ thống
  - Chỉ quan tâm các thông điệp mà actor gửi hay nhận
  - Không quan tâm cấu trúc bên trong của actor
- Phân loại actor:
  - Chủ yếu/ thứ yếu
  - Tích cực/ thụ động

# Mô hình nghiệp vụ - Thu thập ỹc

---

- Actor chủ yếu sử dụng các chức năng chính của hệ thống
  - Thủ quỹ của ngân hàng
- Actor thứ yếu sử dụng các chức năng phụ của hệ thống
  - Người quản trị hệ thống, quản lý ngân hàng
- Một actor chủ động khởi tạo 1 use-case
- Một actor thụ động chỉ tham gia vào 1 hoặc nhiều use-cases.

# Mô hình nghiệp vụ - Thu thập yc

---

- VD: Xét ngữ cảnh là 1 trường PTTH với phần mềm quản lý trường cấp 3:
- Danh sách đề nghị:

– Học sinh	Tổ bộ môn	Số tiết
– Giáo viên	BGH	TKB
– Môn học	Khối	Bảng điểm
– Lớp	Phụ huynh	Phòng
– Học kỳ	ĐTB	Học phí
– Năm học	Diện HS	...

**Đối tượng/ không phải đối tượng?**

# Mô hình nghiệp vụ - Thu thập yc

---

## Các PM:

- PM quản lý học sinh:
  - Học sinh, Giáo viên, Môn học, Lớp, Khối, Phụ huynh, Diện HS, Học kỳ, Năm học...
- PM quản lý giáo viên:
  - Giáo viên, Tổ bộ môn, Môn học, Khối, Lớp, Học kỳ, Năm học...
- PM xếp TKB:
  - Giáo viên, môn học, lớp, phòng, học kỳ, Năm học...



# Mô hình nghiệp vụ - Thu thập yc

---

## Đối tượng chính/ đối tượng phụ?

- PM quản lý học sinh:
  - **Học sinh**, Giáo viên, **Môn học**, **Lớp**, Khối, Phụ huynh, Học kỳ, Năm học...
- PM quản lý giáo viên:
  - **Giáo viên**, Tổ bộ môn, **Môn học**, Khối, **Lớp**, Học kỳ, Năm học...
- PM xếp TKB:
  - **Giáo viên**, môn học, lớp, phòng, học kỳ, Năm học...

# Nhận diện các Actor

---

- Trả lời một số câu hỏi như:
  - Ai là người sử dụng chức năng chính của hệ thống?
  - Ai cần sự hỗ trợ của hệ thống để thực hiện công việc thường nhật của họ?
  - Ai phải thực hiện công việc bảo dưỡng, quản trị và giữ cho hệ thống hoạt động?
  - Hệ thống sẽ kiểm soát những thiết bị phần cứng nào?
  - Hệ thống đang xây dựng cần tương tác với các hệ thống khác hay không?
  - Ai hoặc vật thể nào quan tâm đến hay chịu ảnh hưởng bởi kết quả mà hệ thống PM tạo ra?

# Mô hình phân tích – Phân tích yc

---

- Mô hình nghiệp vụ biểu diễn các chức năng PM cần xây dựng dưới dạng các use-case.
- Mô hình phân tích sẽ tìm kiếm các đối tượng “sống” trong ngữ cảnh của PM
- Các đối tượng sẽ tương tác với nhau để tạo nên các chức năng mô tả bởi use-case
- Lược đồ lớp phân tích diễn tả cấu trúc, mqh giữa các đối tượng/lớp trong hệ thống
- Chưa quan tâm đến hành vi cụ thể và nhiệm vụ chi tiết của chúng trong ngữ cảnh của hệ thống.
- Nguyên tắc: mô hình phân tích phải **độc lập với OS, ngôn ngữ lập trình, công cụ phát triển**

# Xây dựng mô hình phân tích

---

- Mô hình phân tích được diễn đạt trong UML bằng lược đồ lớp phân tích (Class diagram)
- Các công việc xây dựng lược đồ lớp phân tích:
  - Tìm kiếm các đối tượng/ lớp trong hệ thống
  - Xác định các thuộc tính của đối tượng/ lớp
  - Xác định các tác vụ của đối tượng/ lớp
  - Nhận diện các lớp trừu tượng hóa qua mqh tổng quát hóa
  - Xác lập các mqh giữa các lớp:
    - Tổng quát hóa (generalization)
    - Liên kết (association)
    - Bao gộp (aggregation)
- Biểu diễn thành lược đồ lớp phân tích

# Xây dựng lược đồ lớp

---

- Lược đồ lớp biểu diễn cấu trúc của một số lớp và mqh giữa chúng → mô tả khía cạnh tĩnh của hệ thống
- Hệ thống phức tạp có nhiều lớp → cần xây dựng nhiều lược đồ lớp, mỗi lược đồ mô tả 1 phần của hệ thống
- Lược đồ lớp được bổ sung & hoàn thiện trong mô hình thiết kế (*thêm 1 số lớp, chi tiết các thuộc tính và tác vụ, làm rõ các quan hệ*)

# Xây dựng lược đồ lớp

---

- Lược đồ lớp được xây dựng qua các bước:
  - Xác định các lớp
  - Xác định thuộc tính và tác vụ của lớp
  - Xác định các lớp cơ sở và quan hệ tổng quát hóa
  - Xác định các quan hệ liên kết và bao gộp

# Xây dựng lược đồ lớp

---

- Lược đồ lớp được xây dựng qua các bước:
  - Xác định các lớp
  - Xác định thuộc tính và tác vụ của lớp
  - Xác định các lớp cơ sở và quan hệ tổng quát hóa
  - Xác định các quan hệ liên kết và bao gộp

# Kết quả: Xây dựng lược đồ lớp

- Sơ đồ lớp
- Danh sách các đối tượng và quan hệ

STT	Tên lớp / quan hệ	Loại	Ý nghĩa/ ghi chú
...	...	...	...

- Mô tả chi tiết từng lớp đối tượng và quan hệ

- Với mỗi lớp đối tượng:

- Mô tả các thuộc tính

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ ghi chú
...	...	...	...	...

- Danh sách các trách nhiệm chính

- Với mỗi quan hệ

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ ghi chú
...	...	...	...	...

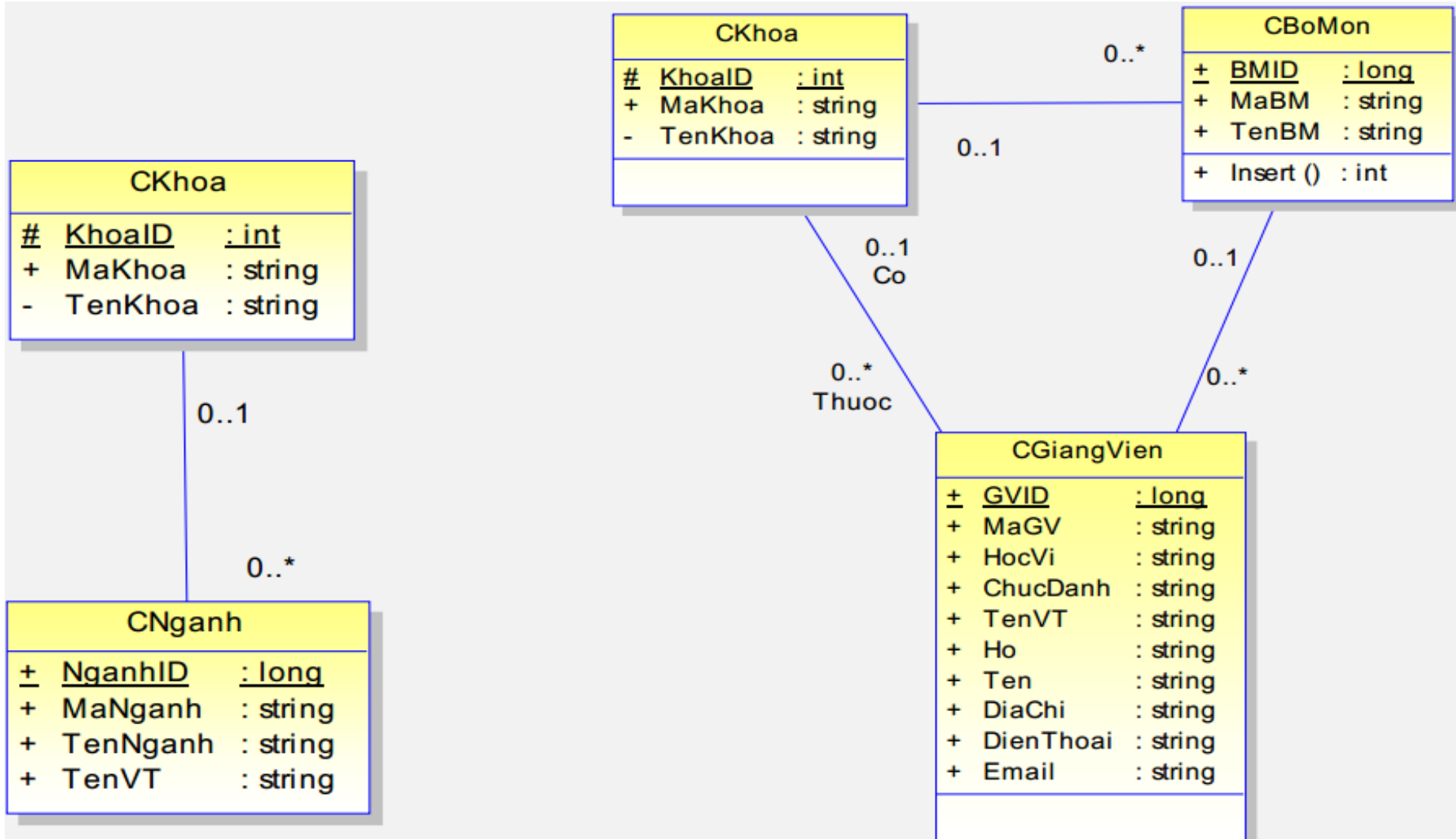


# Áp dụng: Xây dựng lược đồ lớp

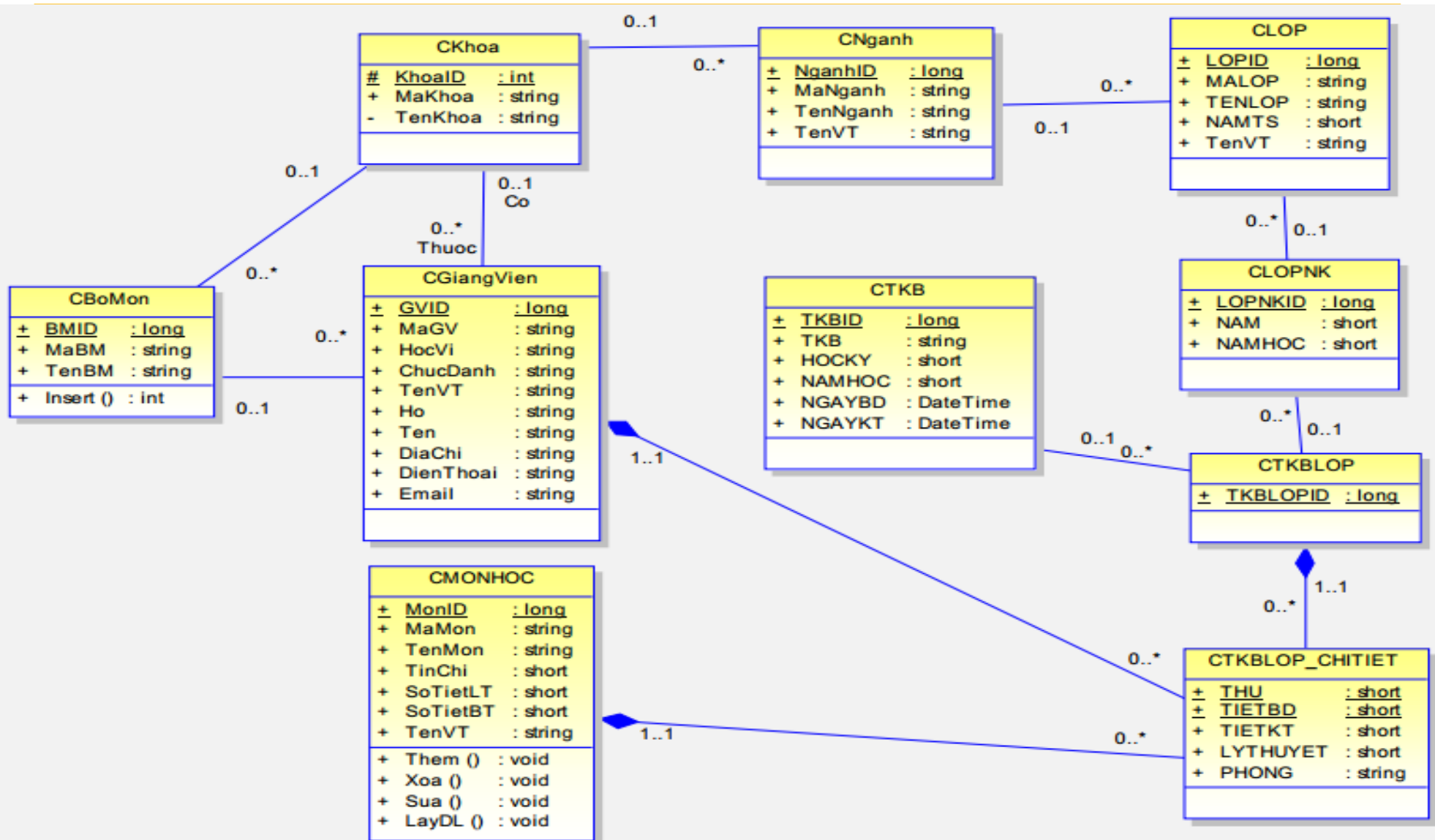
---

- Áp dụng thực tế vào các bài tập
  - Xác định các lớp đối tượng chính
  - Xác định các thông tin và hành động/ trách nhiệm của mỗi lớp đối tượng chính
  - Xác định các quan hệ chính
  - Xác định các lớp đối tượng phụ, các danh mục

# VD: Xây dựng lược đồ lớp



# VD: Xây dựng lược đồ lớp



# Thiết lập Package

---

- Package là 1 cơ chế để tổ chức các phần tử vào các nhóm có liên hệ về ngữ nghĩa với nhau
- Package có thể import các phần tử từ 1 package khác
- Có thể chỉ ra quan hệ giữa các package:
  - Phụ thuộc
  - Tổng quát hóa

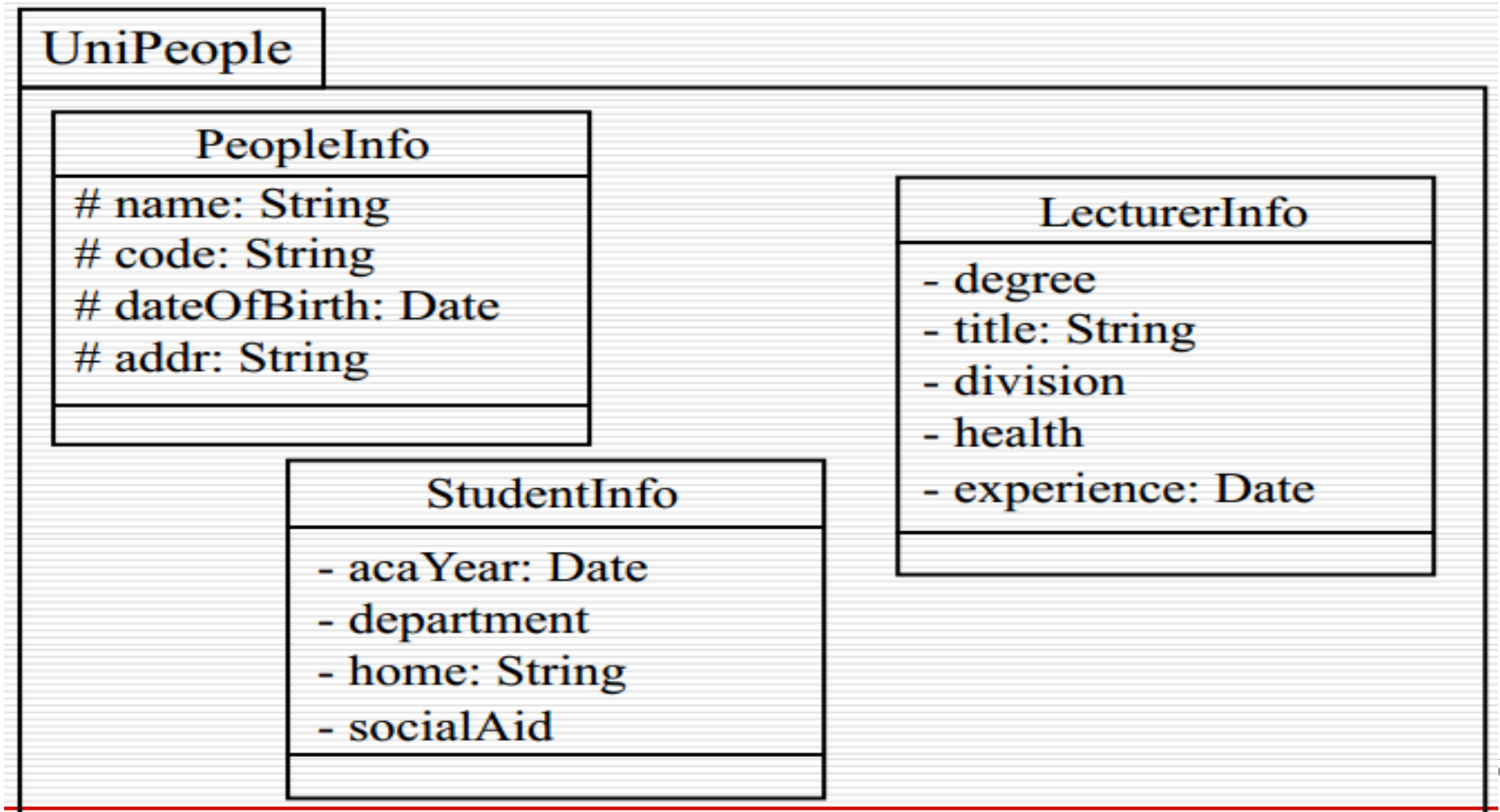
# Thiết lập Package

---

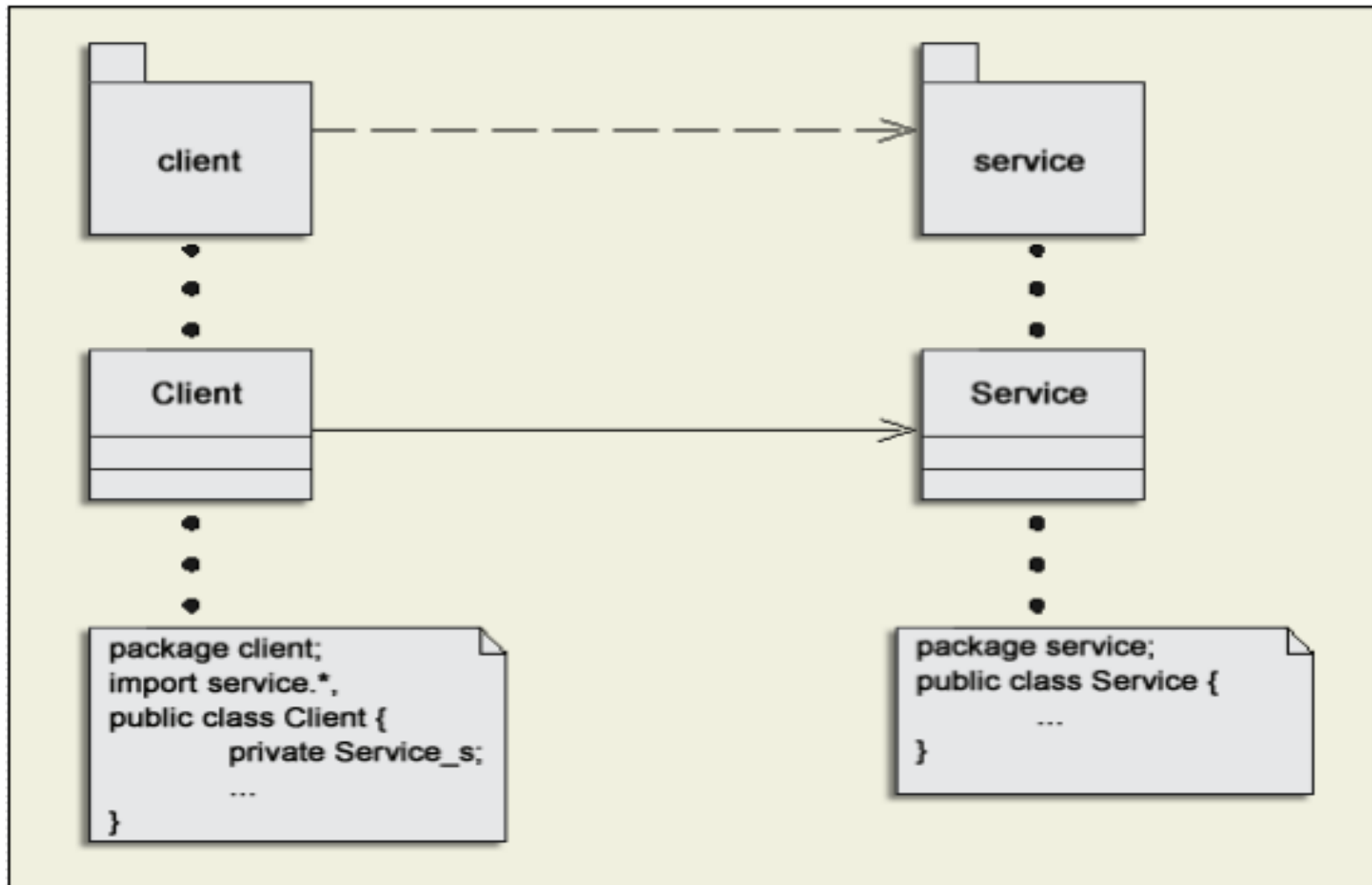
- Mức độ truy xuất của package
  - Private: chỉ nó và các package import nó có thể truy xuất
  - Protected: giống private + các package dẫn xuất
  - Public: Các package khác có thể truy xuất
  - Implementation: không cho phép import, có thể áp dụng cho các phần tử bên trong package
- UML cho phép biểu diễn các package và các mqh package

# VD: Package

Package UniPeople chứa các lớp liên quan đến thông tin con người



# VD: Package



# Review

---

- Phân tích hệ thống cho OOP theo UML chia làm 2 bước:
  - Thu thập yc bằng mô hình nghiệp vụ
  - Phân tích và xác định tính năng hệ thống bằng mô hình phân tích
- Mô hình phân tích nhận diện các đối tượng/lớp: thực thể, biên, điều khiển
- Nhận diện các thuộc tính và một số tác vụ, tuy nhiên chưa làm rõ hành vi của chúng
- UML hỗ trợ một số phần tử: lớp, đối tượng, lược đồ lớp, gói



# Bài tập – 17h00 20/4/2021

---

- Xét PM quản lý sinh viên
  - Liệt kê các actor
  - Cho biết actor chính/phụ, tích cực/thụ động.
  - Xây dựng các lớp có liên quan
  - Vẽ sơ đồ lớp
- Nộp bài (.doc) qua email
- Dùng Power Designer vẽ sơ đồ lớp