

OOAD

Các khái niệm chính

Nguyễn Anh Hào

0913609730 – nahao@ptithcm.edu.vn

Nội dung

2

- 1 Hệ thống ?
- 2 Tư duy hệ thống
- 3 Mô hình hóa & Phương pháp tiếp cận
 - Hướng cấu trúc & SADT
 - Hướng đối tượng & OOAD

1) Hệ thống là gì?

3

- Có nhiều thứ được gọi là hệ thống: hệ thống điện, hệ thống giao thông, hệ thống giáo dục, v.v. **Vậy những hệ thống này có những đặc điểm chung nào?**
 - Tất cả đều được con người tạo ra một cách có chủ đích và có mục đích.
 - Có nhiều bộ phận hợp thành hệ thống theo một cách nào đó. Không giống như một chiếc túi chứa nhiều đồ vật: Nếu tùy ý tháo rời hoặc sửa chữa một bộ phận nào đó, hệ thống sẽ bị hỏng.
- Tủ lạnh có phải là một hệ thống không?
 - a) Không, nếu người ta chỉ sử dụng chức năng của nó để đạt được một số lợi ích → nó là một công cụ.
 - b) Có, nếu nó bị hỏng và cần sửa chữa → chúng ta phải tìm ra bộ phận nào bị hỏng để sửa chữa → nó là một hệ thống.

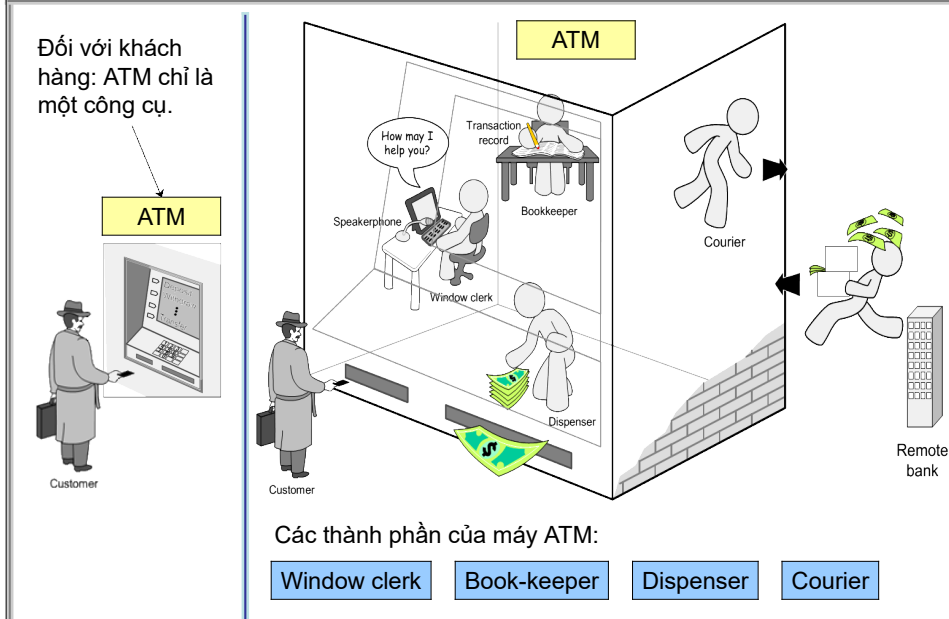
Định nghĩa hệ thống

4

- Hệ thống là một tập hợp gồm nhiều thành phần hoạt động cùng nhau để thực hiện một (hoặc nhiều) chức năng của hệ thống.
 - Mỗi thành phần của hệ thống không có khả năng tự thực hiện tất cả các chức năng của hệ thống.
 - Sự hợp tác (phối hợp) giữa các thành phần giúp hệ thống thực hiện được chức năng của mình: Đây là mối quan hệ tất yếu giữa các thành phần, trong đó mỗi thành phần có khả năng riêng, hình thành nên khái niệm "hợp tác" hoặc "hệ thống".

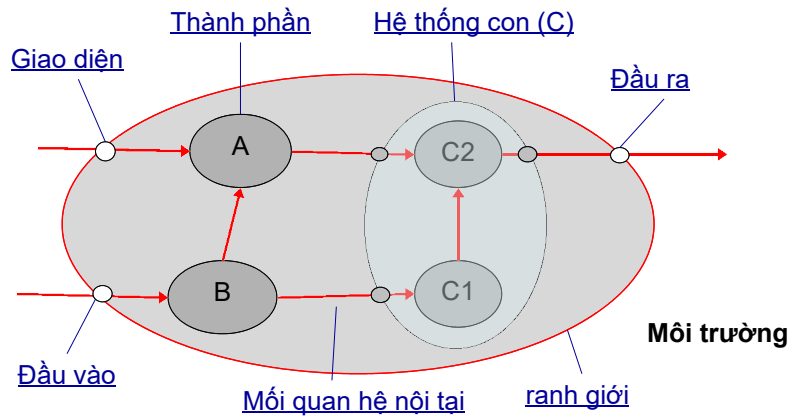
Ví dụ : máy ATM

5



Các thuộc tính của hệ thống

6



Mỗi thành phần là một "công cụ hệ thống" cho hệ thống (cụ thể là A, B, C,...). Một thành phần hệ thống có thể là một hệ thống con (C bao gồm C1, C2). Các chức năng mà C thực hiện cho hệ thống sẽ được thực hiện bởi hai thành phần C1 và C2 hoạt động cùng nhau.

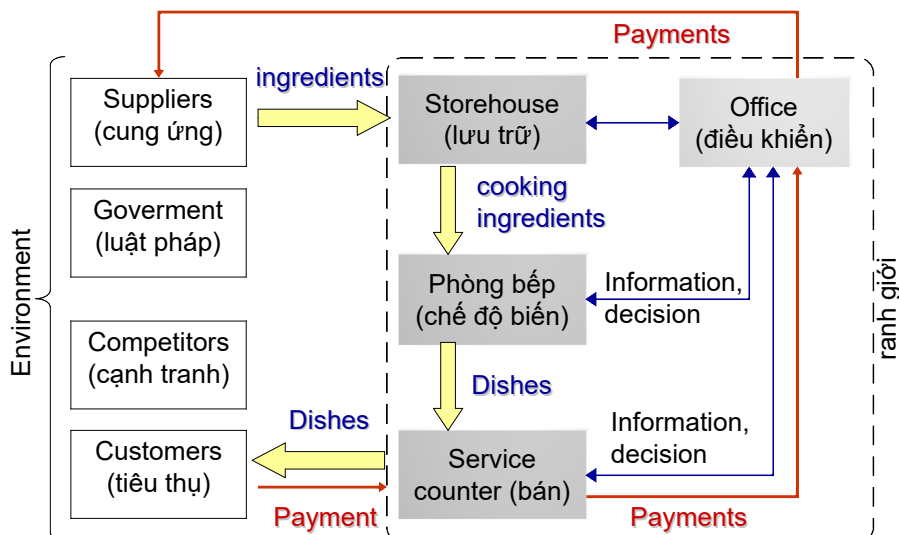
Các thuộc tính của hệ thống

7

- **Môi trường** là những gì nằm ngoài ranh giới của hệ thống và tương tác với hệ thống (ví dụ: người dùng). Bất kỳ hệ thống nào do con người tạo ra đều phải có lợi ích sử dụng (thông qua các chức năng của nó); đó là **nhiệm vụ của hệ thống** trong môi trường làm việc của nó.
- **Các chức năng** của hệ thống chuyển đổi đầu vào thành đầu ra mong đợi; chúng được thực hiện bằng **sự hợp tác giữa các thành phần** trong hệ thống.
 - **Đầu vào** = những thứ mà môi trường cung cấp thông qua **giao diện**
 - **đầu ra** = những gì mà môi trường muốn nhận được thông qua **giao diện**
- Sự hợp tác là một **mối quan hệ nội tại**.

Ví dụ : Nhà hàng là một hệ thống

8



Sub-systems

9

- Nếu chúng ta coi nhà hàng như một hệ thống thì **kho, bếp, quầy phục vụ và văn phòng** đều là các hệ thống con của nhà hàng.
- Nếu kho có hệ thống phần mềm quản lý kho cho thủ kho thì **kho là một hệ thống con** gồm 2 thành phần: **hệ thống quản lý kho** (tự động) và **thủ kho** (nhân công).
- Hệ thống quản lý kho cũng là một hệ thống con bao gồm **máy tính** (phần cứng) và **phần mềm quản lý kho** (chương trình, cơ sở dữ liệu) để cung cấp thông tin cho thủ kho.
- PM QL kho (module SW) cũng là một hệ thống con.

Ví dụ : Hệ thống đặt món ăn

10

- **Các loại IS** : TPS = ghi lại đơn hàng & thanh toán, MIS = thống kê, DSS = Gợi ý chọn món
- **Các thành phần cơ bản** : **Phần cứng** = máy chủ, **Phần mềm** = ứng dụng, **Dữ liệu** = chi tiết đơn hàng, **Con người** = khách hàng và nhân viên, **Quy trình** = quy trình đơn hàng → quy trình thực phẩm → giao đồ ăn.
- **Được cấu trúc theo OOP** : Lớp đơn hàng chứa thông tin mục (thuộc tính) + xác nhận (phương thức).
- **Đóng gói** : Khách hàng không biết thông tin chi tiết về nhà bếp, **chỉ cần gọi món và nhận đồ ăn**.
- **Kế thừa** : Lớp Pizza kế thừa từ lớp Food.
- **Mối quan hệ** : lớp Order “có” danh sách MenuItem, lớp Customer “đặt” Order...
- **Tái sử dụng** : Tích hợp Google Maps API,...

2) Nguyên tắc xây dựng hệ thống

11

1. **Keep it Simple** – tránh sự phức tạp không cần thiết
2. **Modularity** – chia thành các phần riêng biệt
3. **DRY** – Đừng lặp lại chính mình
4. **Maintainability** – dễ dàng cập nhật và sửa chữa
5. **Scalability** – có thể tăng trưởng

Tư duy hệ thống

12

Xem hệ thống như một hệ thống con của một hệ thống lớn hơn để xác định các chức năng cần thiết của nó cho hệ thống lớn hơn đó là gì.

- 1 **Vai trò** của hệ thống trong môi trường là gì ?
 - Hãy xem xét hệ thống như một công cụ được sử dụng trong môi trường của nó
- 2 Hệ thống **cần phải làm gì** để thực hiện vai trò này?
 - Xác định chức năng của hệ thống trong môi trường của nó
- 3 Mỗi chức năng này **được thực hiện như thế nào** ?
 - Xác định các thành phần, vai trò và sự hợp tác của chúng
- 4 **Mỗi thành phần cần phải làm gì** cho hệ thống ?
 - Xác định chức năng của các thành phần trong hệ thống

*** một thành phần cũng là một hệ thống con (đây là một quá trình suy nghĩ đệ quy).*

Môi trường của hệ thống

13

- Hệ thống là một hệ thống con của một hệ thống lớn hơn, được gọi là môi trường.
- Môi trường quyết định sự tồn tại của hệ thống. Ví dụ: hệ thống = nhà hàng/công ty/bệnh viện:
 - Hệ thống có **vai trò được chấp nhận** trong môi trường làm việc của nó (cung cấp thực phẩm/hàng hóa/điều trị bệnh).
 - Hệ thống **cần nhiều thứ** từ môi trường để tồn tại (tiền, điện, nước, nhân lực,...)
 - Hệ thống **cần phải thích nghi** với sự thay đổi của môi trường.

Khởi tạo và sửa đổi hệ thống

14

- Lý do cần tạo và sửa đổi
 - Mọi người cần công cụ mới
 - Mọi người cần cải thiện công cụ hiện có
- Để tạo ra một công cụ tốt, chúng ta nên:
 - Xem xét chức năng của công cụ, xem chúng có đáp ứng được nhu cầu của môi trường làm việc hay không.
 - Coi công cụ như một hệ thống (xem xét cấu trúc bên trong của nó), để chế tạo hoặc sửa chữa.

Phân tích và thiết kế hệ thống

15

- Phân tích và thiết kế hệ thống là một chuỗi tiến trình tạo ra hoặc sửa đổi hệ thống một cách có kiểm soát.
- **Phân tích hệ thống** : Là quá trình dựa trên bằng chứng (dữ liệu thu được từ thực tế) để xác định chính xác các yêu cầu cho hệ thống
- **Thiết kế hệ thống** : Là quá trình xác định & phối hợp các thành phần cần thiết trong hệ thống để giải quyết các yêu cầu đặt ra cho hệ thống.
- Nghĩa là: PTTK hệ thống để nó tác động tích cực (có lợi) vào môi trường (làm cho môi trường tốt hơn).

Hiểu hệ thống

16

- Hiểu một hệ thống là quá trình thu thập thông tin ("biết") và hệ thống hóa thông tin để giải thích cấu trúc cần thiết của hệ thống ("hiểu").
- Từ đó, chúng ta có thể xác định chính xác vấn đề của hệ thống : cần sửa chữa, bổ sung hay loại bỏ những gì. Đây là công việc quan trọng nhất và khó khăn nhất, bởi vì chúng ta phải suy nghĩ về cách hệ thống được tạo ra.
- Có rất nhiều điều cần phải biết và hiểu về hệ thống, ví dụ: chức năng, thành phần, mối quan hệ giữa chúng,... Vậy, chúng ta nên bắt đầu từ đâu và làm như thế nào?

Tư duy hệ thống trong PT&TK

17

- 1 Hãy coi hệ thống như một công cụ, tìm hiểu về **những tình huống thực tế cần sự trợ giúp từ công cụ này**.
- 2 Hệ thống hóa (liên kết) mọi thứ đã biết, tóm tắt bằng **mô hình** (ví dụ vẽ sơ đồ mô hình).
- 3 Phân tích các mô hình (sơ đồ) để xác định cấu trúc cần thiết bên trong của hệ thống.
 - a. **những yêu cầu gì đặt ra cho hệ thống** (từ nhiệm vụ của nó trong môi trường)
 - b. **Hệ thống tương tác với môi trường như thế nào để thực hiện nhiệm vụ của mình.**
 - c. **Những thành phần cần thiết trong hệ thống là gì ?**
 - d. **Cách phối hợp các thành phần này để thực hiện các tương tác của hệ thống như thế nào.**

3) Mô hình hóa & Phương pháp tiếp cận

18

- **Mô hình** là phương tiện để 'tóm tắt' những đặc điểm quan trọng của một hệ thống (hình ảnh, công thức, v.v.).
 - Ví dụ: bản đồ, sơ đồ, biểu đồ dòng chảy...
- Mô hình giúp chúng ta khái quát mọi thứ để dễ tập trung và hiểu.
- Mô hình dựa trên ngữ pháp, ngữ nghĩa và ngữ cảnh.
- **Mô hình hóa** là việc tạo ra các mô hình từ thế giới thực.
 - Ví dụ: vẽ lược đồ
- **Cách tiếp cận** là một cách để hiểu và mô hình hóa một hệ thống. Mỗi cách tiếp cận có phương pháp mô hình hóa riêng. Cách tiếp cận phổ biến cho PTTK:
 - Tiếp cận hướng cấu trúc
 - Tiếp cận hướng đối tượng



Tiếp cận hướng cấu trúc

19

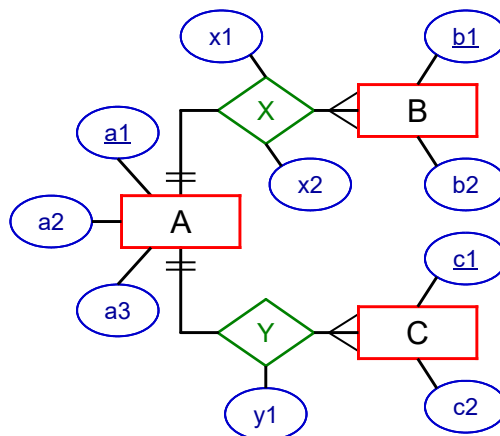
- Phương pháp tiếp cận cấu trúc là cách phân tích và giải quyết vấn đề dựa trên tư duy logic.
 - Hiểu rõ vấn đề (phân tích) để tìm ra giải pháp (thiết kế) đáp ứng mọi yêu cầu và ràng buộc đã biết.
- Mô hình hóa hệ thống: dữ liệu và xử lý.
 - Dữ liệu: Khái quát các đối tượng trong thế giới thực thành các khái niệm về thực thể và mối quan hệ (lược đồ ERD) có các thuộc tính cần thiết cho việc thu thập, lưu trữ và truy xuất dữ liệu của hệ thống
 - Xử lý: Xác định và phân tích các xử lý của hệ thống đến mức đủ đơn giản để hiểu và làm đúng (lược đồ DFD)

ERD: Sơ đồ quan hệ thực thể

20

❖ ERD được sử dụng để xác định nội dung dữ liệu hệ thống

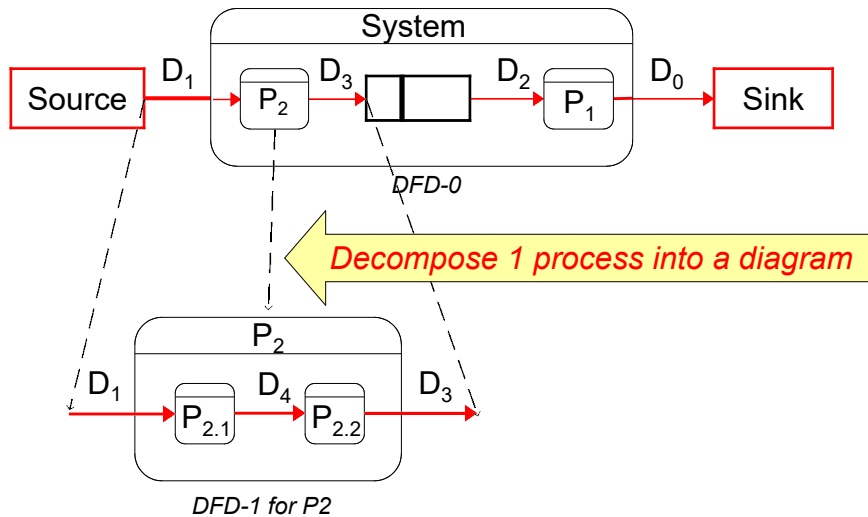
1. Các thực thể
2. Mối quan hệ
3. Thuộc tính
4. Khóa
5. Số liên hệ
6. Loại quan hệ



DFD: Data Flow Diagram

21

❖ DFD được sử dụng để xác định các xử lý của hệ thống .



Tiếp cận hướng đối tượng

22

- Triết lý chung của OOAD là định nghĩa một hệ thống như một tập hợp **các đối tượng từ trong thế giới thực có tương tác** với nhau thông qua **các giao diện được xác định rõ**
 - Tìm các đối tượng từ 'thế giới thực' có khả năng hoạt động như các thành phần cần thiết của hệ thống;
 - Chỉ định trách nhiệm cho từng thành phần đối tượng
 - Xác định cách các đối tượng thành phần tương tác/hợp tác để thực hiện các chức năng hệ thống

Hướng đối tượng so với hướng cấu trúc ²³

- **Hướng cấu trúc** dựa trên việc phân tích vấn đề của hệ thống thành các vấn đề nhỏ cần giải quyết, sau đó tìm giải pháp (thiết kế chức năng xử lý) cho vấn đề từ tư duy.
- **Hướng đối tượng** giải quyết các vấn đề của hệ thống bằng cách tìm kiếm (hoặc mô phỏng) các đối tượng từ thế giới thực có thể tham gia vào hệ thống và phối hợp chúng lại với nhau để giải quyết vấn đề.

Đối tượng trong tiếp cận OO

24

- Một đối tượng là thứ gì đó trong thực tế mà con người có thể nhận biết **tên**, **thuộc tính** và **hành vi**.
 - **Phân lớp đối tượng** làm nổi bật khả năng của đối tượng trong hệ thống
 - **Quan hệ kế thừa** tạo ra sự linh hoạt trong thiết kế.
- Việc mô tả (hoặc **mô hình hóa**) đối tượng dựa trên các nguyên tắc của tiếp cận hướng đối tượng

OO: Nguyên tắc 1

25

Một đối tượng được nhận biết bởi lớp của nó

- Phân loại đối tượng là cách đơn giản nhất để hiểu biết về đối tượng.
 - Ví dụ: Bob là một con chó, nó có thể sủa, vì chó có thể sủa
- Trong phân loại, các đối tượng thuộc một lớp đều có chung các thuộc tính và hành vi của lớp đó.
- Lớp đối tượng cũng được phân loại thành một lớp tổng quát hơn (siêu lớp); Đây là khái niệm tổng quát hóa.
 - Ví dụ: bác sĩ là nhân viên của bệnh viện, nhân viên là công dân (do đó bác sĩ cũng là công dân).

OO: Nguyên tắc 2

26

Đối tượng có quyền thừa kế

- Mỗi lớp con đều kế thừa mọi thứ từ lớp cha; bao gồm cả các mối quan hệ của lớp cha.
- Một lớp con có thể kế thừa từ nhiều lớp cha: đó chính là tính kế thừa đa lớp.
 - Ví dụ, một lớp lập trình viên kế thừa từ hai lớp: nhân viên (tên, tuổi) và nghề lập trình (có thể viết mã).
- Hành vi được kế thừa có thể được thay đổi trong lớp con để làm cho hành vi phức tạp hơn, đây chính là tính đa hình trong kế thừa.

OO: Nguyên tắc 3

27

Các đối tượng có quyền tự do phát triển riêng của chúng

- **Việc đóng gói** bảo vệ quyền này; Nó tách biệt cách hiểu biết về đối tượng thành 2 hướng:
 - 1 **Nhìn từ bên ngoài** : các đối tượng khác chỉ biết các dịch vụ và thuộc tính công khai mà đối tượng cung cấp (những gì đối tượng có thể làm) và không thể biết đối tượng thực hiện các dịch vụ đó như thế nào.
 - 2 **Nhìn từ bên trong** : các đối tượng có thể tự do thay đổi các thuộc tính và hành vi riêng tư của chúng mà không làm ảnh hưởng đến các dịch vụ công khai cho bên ngoài.

OO: Nguyên tắc 4

28

Đối tượng tự quyết định cách phản hồi các yêu cầu

Tom và Mary đang ăn tối trong một nhà hàng. Tom cần nhờ Mary giúp anh ta lấy lọ muối bên cạnh cô ấy.

Tom sẽ nói gì với Mary?

- (1) “ **Đưa giúp tôi lọ muối** ”
- (2) “ **Hãy bỏ tay ra khỏi ly và với lấy lọ muối, cầm lấy và đưa về phía tôi cho đến khi tôi có thể giữ được nó** ”

(1) là một yêu cầu (một thông điệp mang yêu cầu dịch vụ) được sử dụng trong hướng đối tượng. Mọi thứ tiếp theo đều do người nhận quyết định: Mary có quyền từ chối hoặc nhờ người khác làm thay (→ cơ chế ủy thác).

(2) là một yêu cầu chi tiết và chính xác về hành động cần phải thực hiện (và không thể thực hiện theo cách khác), được sử dụng trong phương pháp định hướng cấu trúc (→ cơ chế mệnh lệnh).

Hành vi của một đối tượng phụ thuộc vào trạng thái của nó

- Giá trị cụ thể của thuộc tính quyết định trạng thái của đối tượng. Trạng thái của một đối tượng là tập hợp các giá trị thuộc tính của nó, ví dụ: một đối tượng có 2 thuộc tính A và B, a và b là 2 giá trị dữ liệu của A và B thì 1 trạng thái của đối tượng này là (a,b). Đối tượng có nhiều trạng thái khác nhau. Ví dụ: cột đèn giao thông có 3 trạng thái: Xanh, Vàng, Đỏ.
- Sự thay đổi trạng thái của đối tượng là do bản thân đối tượng phản ứng với các sự kiện kích hoạt (trong cột sáng là tín hiệu hết thời gian của mỗi màu). Sự chuyển đổi sang trạng thái mới (S_2) được xác định bởi hai yếu tố: trạng thái hiện tại (S_1) và sự kiện kích hoạt e: $S_2 = \delta(S_1, e)$, δ được gọi là hàm chuyển trạng thái (hàm chuyển trạng thái).

- 1 Lớp đối tượng phản ánh các thực thể thực tế vào trong hệ thống. Điều này giúp nhà thiết kế dễ dàng tạo ra các lớp **dễ hiểu hơn**.
- 2 Có thể phân tách các phần khác nhau của hệ thống thành các lớp và chỉ xem xét những thay đổi trong từng lớp này (phạm vi thay đổi nhỏ). Điều này **giúp giảm thiểu rủi ro trong quá trình phát triển hệ thống**.
- 3 Giúp giảm bớt áp lực tạo ra (và sửa lỗi) hệ thống thông qua **việc tái sử dụng các thành phần đã có sẵn**, sự kế thừa giúp mở rộng và sửa đổi phương thức của lớp tương đối dễ dàng

1 Phân tích OO

- Xem hệ thống như một công cụ, xác định các tình huống trong môi trường mà hệ thống được nhiều đối tượng khác sử dụng (usecase) và hợp tác với hệ thống để giải quyết tình huống đó.
- Mô tả sự hợp tác của hệ thống trong từng usecase bằng các lược đồ UML (kịch bản của usecase).
- Tìm các đối tượng thành phần cần thiết để hệ thống tham gia vào quá trình xử lý các usecase và mô tả chúng thành lược đồ lớp
- Dựa trên các lược đồ đã vẽ cho usecase, xác định các phương thức và thuộc tính của đối tượng (chi tiết lớp)

2 Thiết kế OO

- Cụ thể hóa các đối tượng khái niệm cho hệ thống thành các đối tượng thiết kế (giao diện, xử lý, thực thể)
- Xác định các tương tác giữa các đối tượng thiết kế để cộng tác và giảm thiểu sự phụ thuộc, dựa trên các nguyên tắc (ví dụ: SOLID, tái sử dụng, chuẩn,...)

OOAD UML diagram supports

