

BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI
TRƯỜNG CAO ĐẲNG NGHỀ KỸ THUẬT CÔNG NGHỆ

----- Ỡ > & Ỡ > -----



GIÁO TRÌNH

MÔN HỌC: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN
NGHỀ: LẬP TRÌNH VIÊN MÁY TÍNH
TRÌNH ĐỘ: CAO ĐẲNG

*Ban hành kèm theo Quyết định số: 13A/QĐ-CDNKTCN ngày 10 tháng 01 năm 2019
của Hiệu trưởng Trường Cao đẳng nghề Kỹ thuật Công nghệ*

Hà Nội, năm 2021
(Lưu hành nội bộ)

TUYÊN BỐ BẢN QUYỀN:

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

MÃ TÀI LIỆU: MHLTV 19

Tailieu.vn

LỜI GIỚI THIỆU

Tổ chức xây dựng, quản lý vận hành hệ thống thông tin (HTTT) là một trong những ứng dụng quan trọng của ngành công nghệ thông tin (CNTT) và đến nay đã có nhiều HTTT được xây dựng và ứng dụng trong thực tiễn. Mặc dù hiện nay có khá nhiều ngôn ngữ lập trình và hệ quản trị cơ sở dữ liệu cũng như các phần mềm chuyên dụng áp dụng trong công tác quản lý, tuy nhiên đối với một hệ thống thông tin việc vận dụng ngay các phần mềm đó là một vấn đề gặp không ít khó khăn.

Các hệ thống thông tin chưa đáp ứng được yêu cầu của các nhà quản lý có nhiều nguyên nhân song nguyên nhân quan trọng đó là các nhà xây dựng hệ thống thông tin không được trang bị kiến thức cơ bản về phân tích và thiết kế, thiếu kinh nghiệm tham gia vào quá trình phân tích thiết kế dẫn đến giai đoạn cài đặt phải thay đổi nhiều, gây ra sự lãng phí trong việc xây dựng khai thác, bảo trì và phát triển hệ thống.

Phân tích thiết kế hệ thống thông tin là phương pháp luận để xây dựng và phát triển hệ thống thông tin bao gồm các lý thuyết, mô hình, phương pháp và các công cụ sử dụng trong quá trình phân tích và thiết kế hệ thống.

Mặc dầu có rất nhiều cố gắng, nhưng không tránh khỏi những thiếu sót, nhóm tác giả rất mong nhận được sự đóng góp ý kiến của độc giả để giáo trình được hoàn thiện hơn.

Xin chân thành cảm ơn!

Hà Nội, ngày 23 tháng 04 năm 2021

Tham gia biên soạn

1. Chủ biên **Trần Thị Vinh**
2. Tập thể Giảng viên Khoa CNTT

Mọi thông tin đóng góp chia sẻ xin gửi về hòm thư tranthivinhvnn@gmail.com, hoặc liên hệ số điện thoại 0978113529

MỤC LỤC

LỜI GIỚI THIỆU	3
CHƯƠNG 1: HỆ THỐNG THÔNG TIN	8
1.1. Hệ thống thông tin	8
1.2. Các cách tiếp cận phân tích hệ thống thông tin	8
1.2.1. Phương pháp hướng cấu trúc	8
1.2.2. Phương pháp hướng đối tượng.....	9
1.3. Các khái niệm cơ bản về hướng đối tượng	10
1.3.1. Đối tượng và trừu tượng hóa.....	10
1.3.2. Lớp và thể hiện.....	12
1.3.3. Sự trao đổi và thông điệp	12
1.3.4. Sự phân cấp	12
1.3.5. Tính bao bọc.....	13
1.3.6. Tính đa hình	13
1.4. Chu trình phát triển phần mềm và tiến trình RUP	13
1.4.1. Chu trình phát triển phần mềm.....	13
1.4.2. Các giai đoạn của chu trình phát triển phần mềm	13
1.4.3. Tiến trình phát triển phần mềm RUP	14
1.5. Các bước phân tích thiết kế hướng đối tượng	15
2.1. Giới thiệu UML	17
2.1.1. Lịch sử về UML	17
2.1.2. UML – ngôn ngữ mô hình hóa hướng đối tượng	17
2.2. Các khái niệm cơ bản trong UML	17
2.3. Các biểu đồ UML	18
2.3.1. Biểu đồ Use case	18
2.3.2. Biểu đồ lớp	19
2.3.4. Biểu đồ đối tượng.....	23
2.3.5. Biểu đồ trạng thái	23
2.3.6. Biểu đồ trình tự.....	25
2.3.7. Biểu đồ cộng tác	26
2.3.8. Biểu đồ hoạt động.....	27
2.3.9. Biểu đồ thành phần	30
2.3.10. Biểu đồ triển khai	30
2.4. Giới thiệu công cụ Rational Rose	31
CHƯƠNG 3: PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG.....	33
3.1. Phân tích yêu cầu hệ thống	33
3.1.1. Yêu cầu là gì?	33
3.1.2. Xác định yêu cầu hệ thống.....	33
3.1.3. Phân loại yêu cầu.....	34
1.4. Mô hình hoá nghiệp vụ	34
3.2. Mô hình hóa Use case	38
3.2.1. Giới thiệu về use case	38
3.2.2. Sơ đồ use case	39
3.2.3. Xác định các biến thể của use case.....	39

2.2.4. Thiết lập các mối quan hệ giữa các use case.....	39
3.2.5. Đặc tả actor và use case	39
3.3. Xây dựng đối tượng hệ thống 40	
3.3.1. Các khái niệm cơ bản về sơ đồ lớp.....	40
3.3.2. Xác định lớp đối tượng	40
3.3.3. Mô hình hóa liên kết giữa các lớp	41
3.3.4. Xác định thuộc tính, method của các lớp.....	41
3.3.5. Xây dựng mô hình khái niệm.....	41
3.3.6. Xây dựng biểu đồ tương tác	42
3.3.7. Xây dựng biểu đồ trạng thái.....	42
3.3.8. Xây dựng biểu đồ hoạt động	42
CHƯƠNG 4: THIẾT KẾ HƯỚNG ĐỐI TƯỢNG.....	46
4.1. Thiết kế các hệ thống con 46	
4.1.1. Hệ thống con	46
4.1.2. Phân chia hệ thống thành các hệ thống con	46
4.1.3. Kiến trúc phân tầng.....	46
4.2. Thiết kế giao diện người dùng và thiết kế lớp 47	
4.2.1. Thiết kế giao diện người dùng	47
4.2.2. Thiết kế lớp	48
4.3. Thiết kế việc lưu trữ các dữ liệu 50	
4.4. Mô hình hóa cài đặt hệ thống 53	
4.4.1. Giới thiệu.....	53
4.4.2. Xây dựng biểu đồ thành phần	53
4.4.3. Xây dựng biểu đồ triển khai.....	57
TÀI LIỆU THAM KHẢO	61

GIÁO TRÌNH MÔN HỌC

Tên môn học: Phân tích và thiết kế hệ thống thông tin

Mã môn học: MHLTV 19

Vị trí, tính chất, ý nghĩa và vai trò của môn học:

- Vị trí: Môn học được bố trí sau khi sinh viên học xong các môn học chung và môn kiến thức kỹ thuật cơ sở, thuộc về khối kiến thức chuyên môn nghề và trước các môn học, mô đun đào tạo nghề chuyên sâu khác;
- Tính chất: Là môn học chuyên ngành.
- Ý nghĩa và vai trò của môn học: Đây là mô đun tự chọn trong chuyên môn nghề, cung cấp cho sinh viên các kỹ năng cơ bản nhất về phân tích và thiết kế hệ thống thông tin, xây dựng các hệ thống thông tin cơ bản để phục vụ trong thực tiễn.

Mục tiêu của môn học:

- Về kiến thức
 - + Nhắc lại được các khái niệm: dữ liệu, thông tin, hệ thống, hệ thống thông tin.
 - + Phân tích thiết kế hệ thống thông tin theo hướng đối tượng bằng UML và sử dụng thành thạo một công cụ làm tài liệu như Rational Rose;
 - + Trình bày được những kiến thức cơ bản về phân tích và thiết kế hệ thống thông tin hướng đối tượng bằng UML(Unified Modeling Language), có kỹ năng sử dụng công cụ Rational Rose cho việc phát triển các phần mềm hướng đối tượng
- Về Kỹ năng
 - + Áp dụng các phương pháp phân tích và thiết kế hệ thống thông tin vào việc xây dựng ứng dụng thực tế;
 - + Xây dựng được một bản phân tích thiết kế hoàn chỉnh cho các hệ thống vừa và nhỏ
- Về năng lực tự chủ và trách nhiệm
 - + Bố trí làm việc khoa học đảm bảo an toàn cho người và phương tiện học tập.
 - + Đảm bảo được tính chính xác trong sử dụng các kĩ pháp

Nội dung của môn học:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra* (LT hoặc TH)
1	Chương 1: Tổng quan	6	3	3	
	1.1. Hệ thống thông tin		0.5		
	1.2. Các cách tiếp cận phân tích hệ thống thông tin		0.5		
	1.3. Các khái niệm cơ bản về hướng đối tượng		0.5		
	1.4. Chu trình phát triển phần mềm		0.5	1	
	1.5. Các bước phân tích thiết kế hệ thống		1	2	
2	Chương 2: UML và công cụ phát triển hệ thống	15	9	6	
	2.1. Giới thiệu về UML (Unified Modeling Language)		2		

	2.2. Các khái niệm cơ bản trong UML		3	1	
	2.3. Các biểu đồ UML		3	3	
	2.4. Giới thiệu công cụ Rational Rose		1	2	
3	Chương 3: Phân tích hướng đối tượng	18	6	12	
	3.1. Phân tích yêu cầu hệ thống		2	4	
	3.2. Mô hình hóa Use case		2	4	
	3.3. Xây dựng đối tượng hệ thống		2	4	
4	Chương 4: Thiết kế hướng đối tượng	15	6	9	
	4.1. Thiết kế các hệ thống con		1	2	
	4.2. Thiết kế giao diện người dùng và thiết kế lớp		2	3	
	4.3. Thiết kế việc lưu trữ các dữ liệu		2	2	
	4.4. Mô hình hóa cài đặt hệ thống		1	2	
5	Kiểm tra	1			1
6	Thi hết môn	2			2
Cộng		60	27	30	3

* Ghi chú: Thời gian kiểm tra lý thuyết được tính vào giờ lý thuyết, kiểm tra thực hành được tính bằng giờ thực hành.

CHƯƠNG 1: HỆ THỐNG THÔNG TIN

Mã chương: MHLTV 19.01

Giới thiệu:

Chương này sẽ trình bày các khái niệm cơ bản về thông tin và hệ thống thông tin (HTTT). Tiếp sau các khái niệm khởi đầu, chương này trình bày các đặc trưng cơ bản của HTTT, khái niệm về hệ thống xử lý tác nghiệp, hệ thống thông tin quản lý và hệ hỗ trợ ra quyết định. Trình bày khái niệm về HTTT tổng thể trong tổ chức hoạt động và các phương pháp cơ bản xây dựng HTTT.

Mục tiêu:

- Hiểu được ý nghĩa, vai trò của thông tin trong thực tiễn;
- Nhận thức cơ bản về hệ thống thông tin nhằm định hướng cho quá trình phân tích và thiết kế hệ thống thông tin;
- Thực hiện các thao tác an toàn với máy tính.

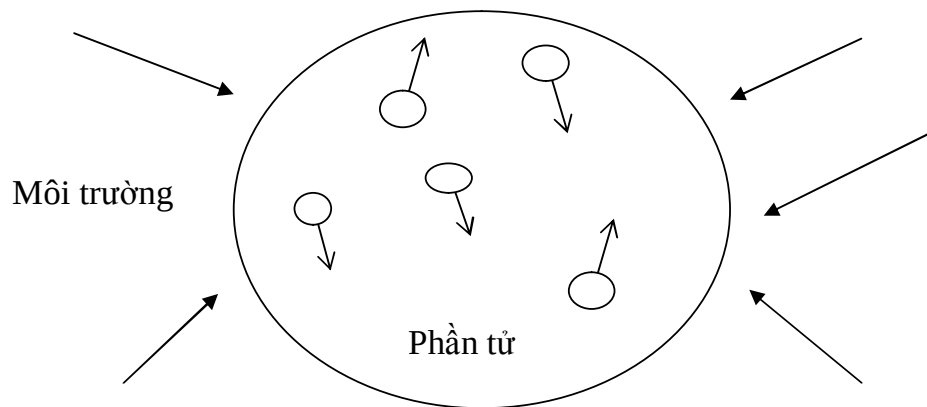
Nội dung:

1.1. Hệ thống thông tin

Mục tiêu:

- Làm cho sinh viên nhận biết được các yếu tố của một hệ thống: phần tử, mục đích, môi trường;
- Nhận thức cơ bản về hệ thống thông tin, nhằm định hướng cho quá trình phân tích và thiết kế hệ thống thông tin;
- Trình bày được các đặc trưng của HTTT;
- Hiểu và trình bày được các HTTT được phân loại theo chức năng. Nêu ra được các giai đoạn phát triển hệ thống.

Hệ thống: Là một tập hợp có tổ chức của nhiều phần tử thường xuyên tương tác với nhau, có những mối quan hệ ràng buộc lẫn nhau và cùng hoạt động chung cho một mục đích nào đó. Môi trường là phần nằm ngoài hệ thống đang xét và thực chất nó là một hệ thống nào đó có giao tiếp với hệ thống đang xét. Giữa hệ thống và môi trường là đường giới hạn xác định phạm vi của hệ thống.



Hình 1.1 Mô hình tổng quát của một hệ thống

Ví dụ: Hệ mặt trời, hệ thống triết học, hệ thống thủy lực, hệ thống pháp luật, hệ thống cơ khí v.v...

1.2. Các cách tiếp cận phân tích hệ thống thông tin

1.2.1. Phương pháp hướng cấu trúc

Phương pháp phân tích và thiết kế hướng cấu trúc bao gồm các hoạt động : khảo sát, phân tích, thiết kế, xây dựng, kiểm thử và cài đặt, vận hành. Đặc trưng của phương pháp này là các hoạt động có thể thực hiện một cách song song. Mỗi hoạt động có thể cung cấp những sửa đổi phù hợp cho một hoặc nhiều hệ thống trước đó.

Ba công cụ quan trọng để mô hình hóa hệ thống theo phương pháp phân tích và thiết kế hướng cấu trúc là :

- Mô hình chức năng
- Mô hình dữ liệu
- Mô hình luồng dữ liệu

Trong đó mỗi mô hình thể hiện một cách nhìn ở góc độ khác nhau vào hệ thống.

Mô hình chức năng : Mô hình mô tả các chức năng chính của hệ thống thông tin, thông thường được biểu diễn bằng sơ đồ chức năng nghiệp vụ, thể hiện hệ thống từ khía cạnh chức năng, trả lời cho câu hỏi :

Hệ thống thực hiện những công việc gì ?

Mô hình được sử dụng cho mục đích này là sơ đồ phân rã chức năng BFD (Business Functional Diagram). Nội dung chính của BFD là sơ đồ phân cấp chức năng của hệ thống.

Mô hình dữ liệu : Mô tả các dữ liệu chính sẽ có trong hệ thống và mối quan hệ ràng buộc giữa chúng, thông thường được mô tả bằng sơ đồ quan hệ thực thể, các bảng thuộc tính các ràng buộc dữ liệu v.v..., thể hiện hệ thống từ khía cạnh dữ liệu hay trả lời cho câu hỏi :

Hệ thống sử dụng dữ liệu gì để phục vụ cho hoạt động của mình ?

Mô hình dữ liệu ERD (Entity Relationship Diagram) là một trong các công cụ phản ánh hệ thống từ một khía cạnh khác, bổ sung với BFD để tạo nên một tổ hợp trọn vẹn của quá trình phân tích.

Mô hình luồng dữ liệu : Mô tả luồng dữ liệu trong hệ thống. Có thể biểu diễn bằng nhiều sơ đồ: sơ đồ ngữ cảnh, sơ đồ phân rã các xử lý, sơ đồ dòng dữ liệu mức đỉnh và sơ đồ dòng dữ liệu các mức dưới đỉnh.

Một trong các mô hình kinh điển được sử dụng cho mục đích mô tả luồng dữ liệu là sơ đồ dòng dữ liệu DFD (Data Flow Diagram). DFD thể hiện một mô hình hệ thống với quan niệm bình đẳng cho cả dữ liệu và chức năng, là một trong những công cụ quan trọng nhất của phân tích hệ thống hướng cấu trúc. Sơ đồ chỉ cách thông tin chuyển vận từ chức năng này hoặc từ quá trình này sang chức năng hoặc quá trình khác. Một điều khá quan trọng là sơ đồ chỉ ra được những thông tin nào cần phải có trước khi thực hiện một chức năng hay một quá trình.

1.2.2. Phương pháp hướng đối tượng

Trong những năm gần đây phương thức lập trình hướng đối tượng đã thống lĩnh thị trường lập trình phần mềm và UML cũng đã trở thành ngôn ngữ mô hình hóa phổ biến trong sản xuất phần mềm. Hầu hết các trường đại học, cao đẳng đã đưa hai môn này vào đào tạo chính khóa và cũng có không ít tài liệu viết về những vấn đề này. Tuy nhiên, nó vẫn còn rất khó hiểu và khó áp dụng với sinh viên, và những bạn trẻ đang làm về Công nghệ thông tin.

Trong loạt bài này, chúng tôi sẽ cố gắng trình bày một cách đơn giản và dễ hiểu nhất các kiến thức về Phân tích và thiết kế hướng đối tượng và UML để giúp các bạn hiểu và áp dụng vào thực tế. Các bài viết sẽ hướng dẫn các bạn phân tích và thiết kế một ứng dụng cụ thể để từ đó tự rút ra bài học kinh nghiệm cho mình và tiếp tục nghiên cứu sâu hơn.

Bài đầu tiên sẽ bàn một cách cơ bản về Phân tích thiết kế hướng đối tượng và UML.

Lưu ý: Để hiểu được loạt bài này bạn phải có kiến thức cơ bản về lập trình hướng đối tượng, vì chúng tôi sẽ không đi chi tiết vào các định nghĩa về lập trình hướng đối tượng.

Khái niệm về Phân tích và thiết kế hướng đối tượng (Object Oriented Analysis and Design: OOAD)

Trong kỹ nghệ phần mềm để sản xuất được một sản phẩm phần mềm người ta chia quá trình phát triển sản phẩm ra nhiều giai đoạn như thu thập và phân tích yêu cầu, phân tích và thiết kế hệ thống, phát triển (coding), kiểm thử, triển khai và bảo trì. Trong đó, giai đoạn phân tích, thiết kế bao giờ cũng là giai đoạn khó khăn và phức tạp nhất. Giai đoạn này giúp chúng ta hiểu rõ yêu cầu đặt ra, xác định giải pháp, mô tả chi tiết giải pháp. Nó trả lời 2 câu hỏi What (phần mềm này làm cái gì?) và How (làm nó như thế nào?).

Để phân tích và thiết kế một phần mềm thì có nhiều cách làm, một trong những cách làm đó là xem hệ thống gồm những đối tượng sống trong đó và tương tác với nhau. Việc mô tả được tất cả các đối tượng và sự tương tác của chúng sẽ giúp chúng ta hiểu rõ hệ thống và cài đặt được nó. Phương thức này gọi là Phân tích thiết kế hướng đối tượng (OOAD)

Khái niệm về UML (Unified Modeling Language)

UML là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau cũng như dùng để thi công hệ thống (phát triển), thuyết phục khách hàng, các nhà đầu tư v.v.. (Giống như trong xây dựng người ta dùng các bản vẽ thiết kế để hướng dẫn và kiểm soát thi công, bán hàng căn hộ v.v..)

Tại sao lại là OOAD và UML?

OOAD cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung thể hiện. Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó nên chúng thường đi đôi với nhau.

OOAD sử dụng UML

UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng v... trong phạm vi các bài viết này chúng ta chỉ nghiên cứu cách sử dụng UML cho phân tích và thiết kế hướng đối tượng trong ngành phần mềm. OOAD sử dụng UML bao gồm các thành phần sau:

- View (góc nhìn)
- Diagram (bản vẽ)
- Notations (ký hiệu)
- Mechanisms (qui tắc, cơ chế)

Chúng ta sẽ tìm hiểu kỹ hơn các thành phần trên.

View (góc nhìn)

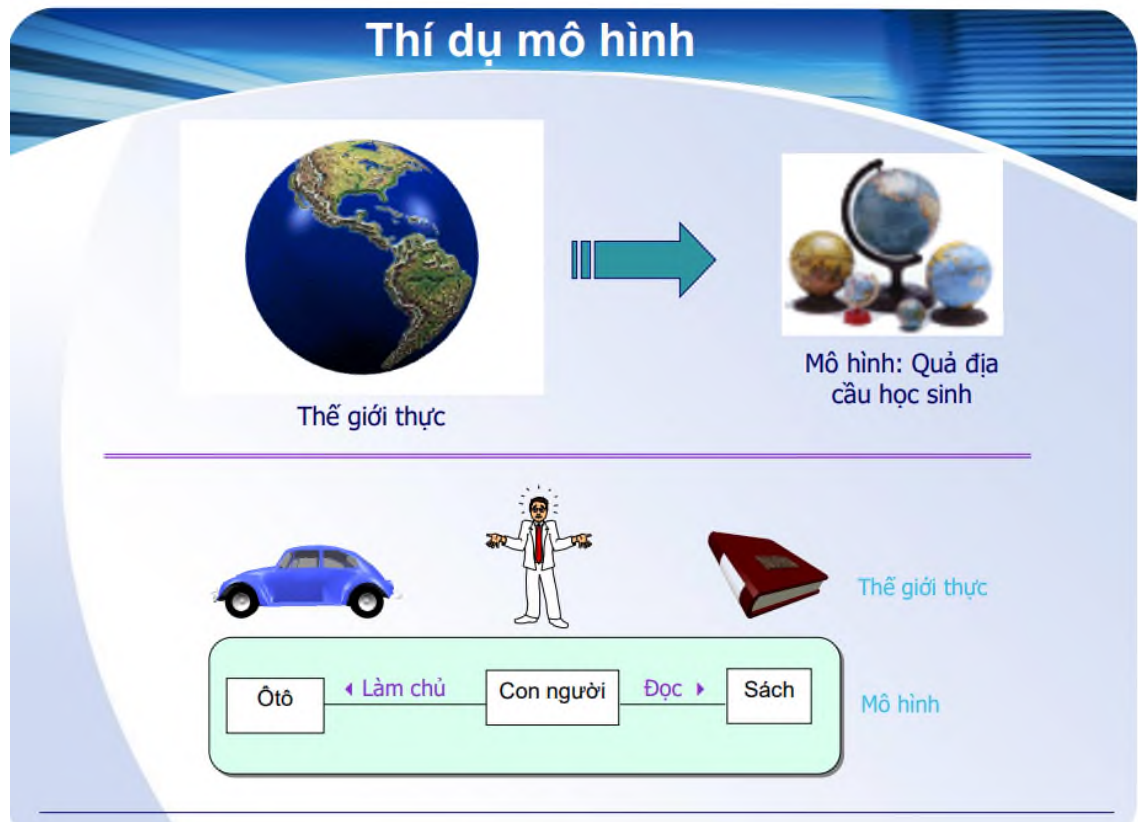
Mỗi góc nhìn như thầy bói xem voi, nó không thể hiện hết hệ thống nhưng thể hiện rõ hệ thống ở một khía cạnh. Chính vì thế trong xây dựng có bản vẽ kiến trúc (nhìn về mặt kiến trúc), bản vẽ kết cấu (nhìn về mặt kết cấu), bản vẽ thi công (nhìn về mặt thi công). Trong phần mềm cũng như vậy, OOAD sử dụng UML có các góc nhìn sau:

1.3. Các khái niệm cơ bản về hướng đối tượng

1.3.1. Đối tượng và trừu tượng hóa

Mô hình hóa hướng đối tượng (Object-Oriented Modeling – OOM): Phương pháp dùng để mô hình hóa các chương trình máy tính theo phương pháp hướng đối tượng (OO Programming – OOP).

Phân tích thiết kế hướng đối tượng (OO analysis and design – OOAD): là một kỹ thuật phần mềm tiếp cận việc mô hình hóa các vấn đề thực tế thành các hệ thống phần mềm như các đối tượng tương tác lẫn nhau



Mô hình hóa trực quan?



"Mô hình hóa nắm bắt các phần thiết yếu của hệ thống."

Dr. James Rumbaugh

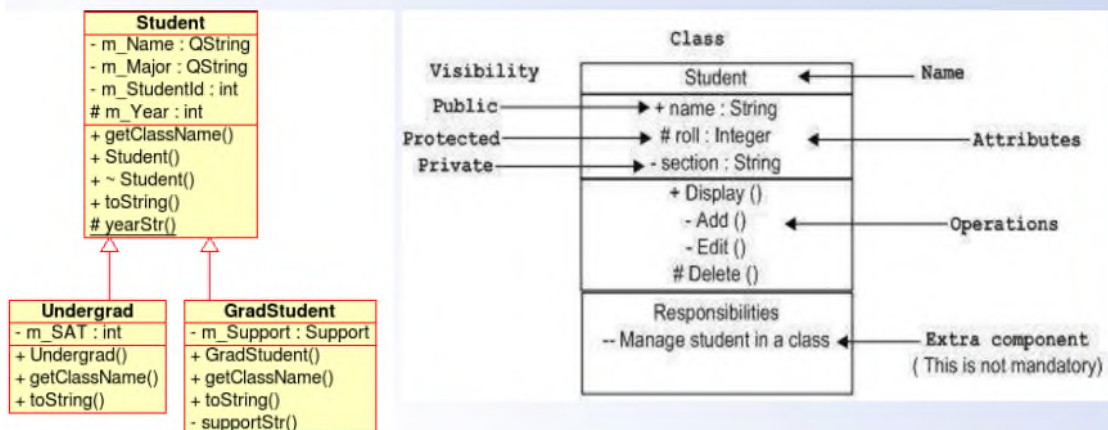


Computer System

Mô hình hóa trực quan đang tạo mô hình bằng cách sử dụng ký hiệu đồ họa tiêu chuẩn

1.3.2. Lớp và thể hiện

Lớp đối tượng (Class) là một khung chứa các cấu trúc và hành vi chung của một đối tượng.



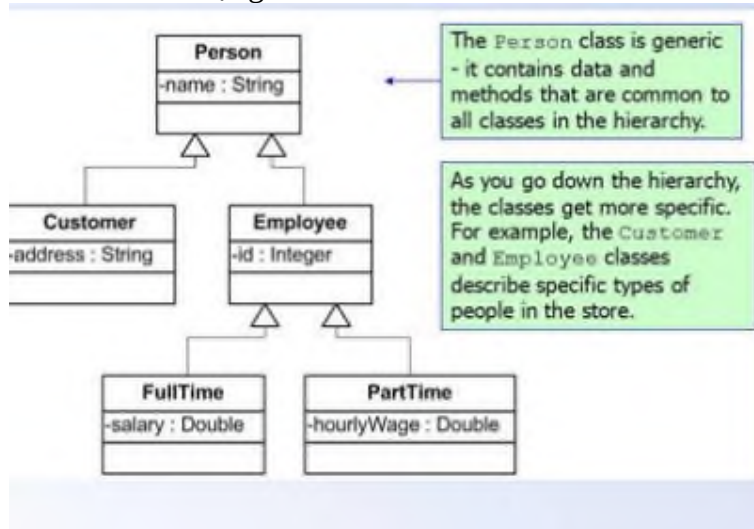
1.3.3. Sự trừu tượng và thông điệp

Sự trừu tượng hóa (Abstraction) là việc xây dựng mô hình chỉ bao gồm các đặc điểm quan trọng, cần thiết và phân biệt với các đặc điểm của mô hình khác. Cũng có khái niệm nói về sự trừu tượng hóa là cách cơ chế biểu diễn thực tế phức tạp bằng mô hình đơn giản

1.3.4. Sự phân cấp

Sự phân cấp (Hierarchy) có thể coi như sắp xếp thứ tự hoặc phân hạng sự trừu tượng hóa của các đối tượng theo một cấu trúc cây. Trong đó: ▪ Phân cấp toàn thể - bộ phận ▪

Phân cấp các đối tượng chứa nhau ▪ Phân cấp lớp ▪ Phân cấp sự kế thừa ▪ Phân cấp theo kiểu đối tượng

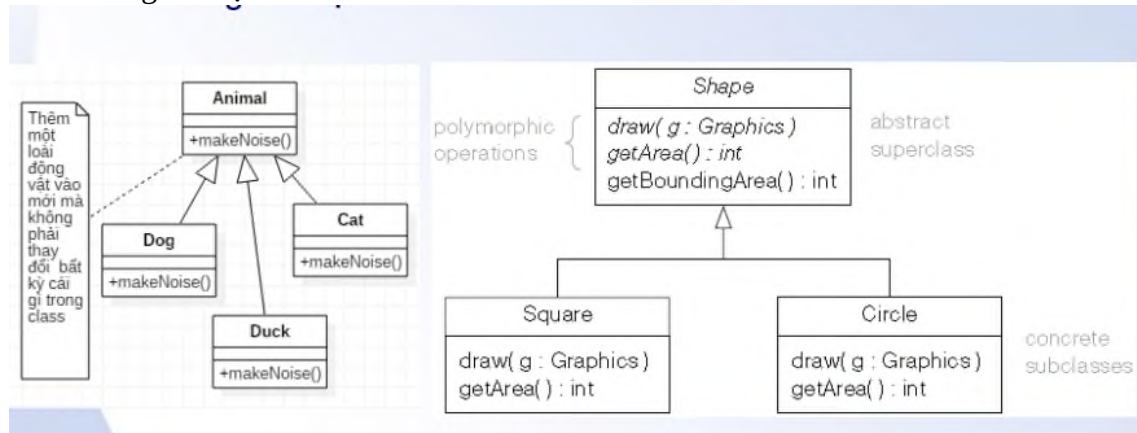


1.3.5. Tính bao bọc

Tính đóng gói (Encapsulation) là quy tụ các tính chất (thuộc tính, hành vi) vào trong hộp đen của sự trừu tượng hóa – cho phép ẩn đi cài đặt ở phía sau các giao diện

1.3.6. Tính đa hình

Đa hình (Polymorphos - Nhiều hình thái khác nhau) ▪ Mọi sự thực thi của giao diện phải bao gồm ít nhất một giao diện. Trong có có các thực thi có thể áp dụng cho nhiều hình thức giao diện



1.4. Chu trình phát triển phần mềm và tiến trình RUP

1.4.1. Chu trình phát triển phần mềm

RUP – Rational Unified Process – Quy trình hợp nhất

Quy trình nghiệp vụ cho kỹ thuật phần mềm hướng đối tượng, dùng mô tả một họ các tiến trình kỹ thuật phần mềm cùng chia sẻ cấu trúc và kiến trúc tiến trình chung.

Có 4 pha (phases):

- Khởi tạo
- Chi tiết
- Xây dựng
- Chuyển giao

1.4.2. Các giai đoạn của chu trình phát triển phần mềm

Quy trình phát triển của một phần mềm có thể được chia thành các giai đoạn như sau:
Nghiên cứu sơ bộ (Preliminary Investigation hay còn gọi là Feasibility Study)

Phân tích yêu cầu (Analysis)
Thiết kế hệ thống (Design of the System)
Xây dựng phần mềm (Software Construction)
Thử nghiệm hệ thống (System Testing)
Thực hiện, triển khai (System Implementation)
Bảo trì, nâng cấp (System Maintenance)



1.4.3. Tiến trình phát triển phần mềm RUP

Tiến trình thống nhất (Rational Unified Process - RUP)

- Là Software Engineering process
- Là sản phẩm tiến trình (process product) do Rational Software phát triển và bảo trì RUP nâng cao team productivity
- Các hoạt động RUP tạo lập và quản lý models
- Là hướng dẫn cách sử dụng hiệu quả UML

Các nguyên tắc cơ bản của RUP

Lập và tăng trưởng Dự án được cắt thành những vòng lặp hoặc giai đoạn ngắn. Cuối mỗi vòng lặp thì một phần thi hành được của hệ thống được sản sinh theo cách tăng trưởng (thêm vào) dần dần.

Tập trung vào kiến trúc Toàn bộ hệ thống phức tạp phải được chia thành từng phần (các modun) để có thể dễ dàng triển khai và bảo trì, tạo nên một kiến trúc. (Theo 5 góc nhìn)

Dẫn dắt các ca sử dụng RUP nhấn mạnh sự đáp ứng nhu cầu người dùng, thể hiện bởi các ca sử dụng. Các ca sử dụng ảnh hưởng và dẫn đường cho mọi giai đoạn phát triển của hệ thống.

Không chế bởi các nguy cơ Các nguy cơ chính đối với dự án phải phát hiện sớm và loại bỏ càng sớm càng tốt. Yêu cầu này cũng là căn cứ để xác định thứ tự trước sau của các vòng lặp.

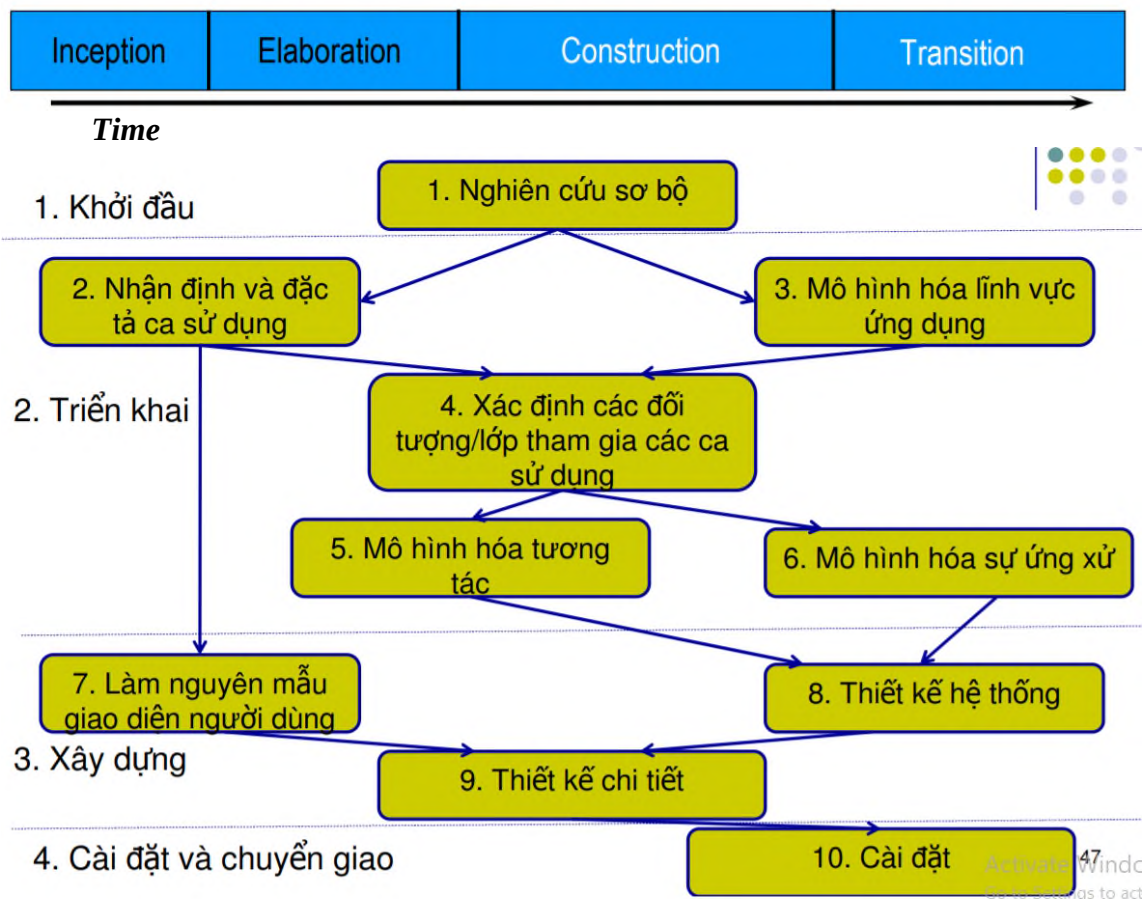
Các pha và công đoạn của tiến trình RUP Có 4 pha

Khởi đầu (inception) Cho một cái nhìn tổng quát về hệ thống sẽ xây dựng và về dự án sẽ triển khai.

Phác thảo (elaboration) Bao gồm sự phân tích chi tiết hơn về hệ thống, cả về chức năng lẫn cấu trúc tĩnh. Đồng thời một kiến trúc hệ thống cũng được đề xuất. Kiến trúc này có thể dựng thành nguyên mẫu, trên đó thể hiện nhiều ý đồ đối với hệ thống

Xây dựng (construction) Tập trung vào việc thiết kế và thực thi hệ thống.

Chuyển giao (transition) Nhằm chuyển hệ thống đã xây dựng cho người dùng cuối



1.5. Các bước phân tích thiết kế hướng đối tượng

Quá trình phân tích nhìn chung là hệ quả của việc trả lời câu hỏi "Hệ thống cần phải làm gì?".

Những mục tiêu cụ thể của giai đoạn phân tích là:

- Xác định hệ thống cần phải làm gì.
- Nghiên cứu thấu đáo tất cả các chức năng cần cung cấp và những yếu tố liên quan
- Xây dựng một mô hình nêu bật bản chất vấn đề từ một hướng nhìn có thực (trong đời sống thực).
- Trao định nghĩa vấn đề cho chuyên gia lĩnh vực để nhận sự đánh giá, góp ý.
- Kết quả của giai đoạn phân tích là bản Đặc tả yêu cầu (Requirements Specifications).

Một số các công việc thường được thực hiện trong giai đoạn thiết kế:

- Nhận biết form nhập liệu tùy theo các thành phần dữ liệu cần nhập.
- Nhận biết reports và những output mà hệ thống mới phải sản sinh
- Thiết kế forms (vẽ trên giấy hay máy tính, sử dụng công cụ thiết kế)
- Nhận biết các thành phần dữ liệu và bảng để tạo database

Ước tính các thủ tục giải thích quá trình xử lý từ input đến output.

CHƯƠNG 2: UML VÀ CÔNG CỤ PHÁT TRIỂN HỆ THỐNG

Mã chương: MHLTV 19.02

Giới thiệu

UML (Unified Modeling Language) là ngôn ngữ dành cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm.

UML tạo cơ hội để viết thiết kế hệ thống, bao gồm những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống.

Cụ thể, nó hữu dụng cho những ngôn ngữ khai báo, giản đồ cơ sở dữ liệu, thành phần phần mềm có khả năng tái sử dụng.

UML được phát triển bởi Rational Rose và một số nhóm cộng tác, nó nhanh chóng trở thành một trong những ngôn ngữ chuẩn để xây dựng hệ thống phần mềm hướng đối tượng (Object-Oriented).

Đây là ngôn ngữ kế vị xứng đáng cho những ngôn ngữ mô hình hoá như Booch, OOSE/Jacobson, OMT và một số các phương thức khác.

Mục tiêu

- Nêu được lịch sử phát triển của UML
- Trình bày được khái niệm cơ bản trong UML, các biểu đồ UML
- Vận dụng linh hoạt các biểu đồ UML xác định yêu cầu hệ thống trong thực tế.
- Vận dụng thành thạo công cụ Rational Rose.

Nội dung

2.1. Giới thiệu UML

Mô hình khái niệm – mô hình được tạo lên bởi các khái niệm và các quan hệ giữa chúng.

Mô hình khái niệm là bước đầu tiên vẽ các biểu đồ UML. Nó giúp hiểu được các thực thể trong thế giới thực và chúng tương tác lẫn nhau như thế nào.

Để UML mô tả thế giới thực, nó cần phải tạo ra mô hình khái niệm và sau đó xử lý dần dần từng bước.

Mô hình khái niệm của UML có thể mô tả 3 khái niệm chính:

Các khối xây dựng UML (UML Building Blocks)

Luật liên kết các khối xây dựng UML

Cơ chế chung của UML

2.1.1. Lịch sử về UML

A picture is worth a thousand words” – UML.

Mục đích chính là UML như là cơ chế mô hình hóa đơn giản để mô hình tất cả các hệ thống tiện ích của môi trường thực tế phức tạp hiện nay

2.1.2. UML – ngôn ngữ mô hình hóa hướng đối tượng

UML – Ngôn ngữ chuẩn cho đặc tả, hình ảnh hóa, xây dựng, và mô tả tài liệu cho các tác nhân của hệ thống phần mềm.

UML – được tạo bởi Nhóm Quản lý Đối tượng (Object Management Group – OMG), phiên bản UML 1.0 được đưa ra vào tháng 01/1997.

OMG tiếp tục phát triển UML như một tiêu chuẩn công nghiệp phần mềm thực sự.

2.2 Các khái niệm cơ bản trong UML

Đặc điểm UML:

UML viết tắt của Unified Modeling Language

UML khác với ngôn ngữ lập trình: C/C++, Java, PHP, Cobol,...

UML là ngôn ngữ hình ảnh sử dụng tạo ra các bản kế hoạch thiết kế.

UML có thể được mô tả như ngôn ngữ mô hình hóa hình ảnh đa năng để hình ảnh hóa, đặc tả, xây dựng và văn bản hóa hệ thống phần mềm. Mặc dù UML chuyên dành cho thiết kế, mô hình hóa phần mềm, nhưng nó có thể dùng để mô hình hóa các hệ thống không phải phần mềm. Ví dụ: luồng nghiệp vụ cho một đơn vị sản xuất.

2.3. Các biểu đồ UML

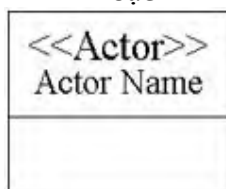
2.3.1. Biểu đồ Use case

Một biểu đồ Use case chỉ ra một số lượng các tác nhân ngoại cảnh và mối liên kết của chúng đối với Use case mà hệ thống cung cấp. Một Use case là một lời miêu tả của một chức năng mà hệ thống cung cấp. Lời miêu tả Use case thường là một văn bản tài liệu, nhưng kèm theo đó cũng có thể là một biểu đồ hoạt động. Các Use case được miêu tả duy nhất theo hướng nhìn từ ngoài vào của các tác nhân (hành vi của hệ thống theo như sự mong đợi của người sử dụng), không miêu tả chức năng được cung cấp sẽ hoạt động nội bộ bên trong hệ thống ra sao. Các Use case định nghĩa các yêu cầu về mặt chức năng đối với hệ thống.

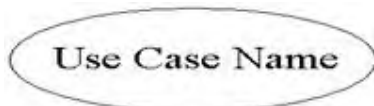
- Hệ thống: Với vai trò là thành phần của biểu đồ use case, hệ thống biểu diễn ranh giới giữa bên trong và bên ngoài của một chủ thể trong phần mềm chúng ta xây dựng. Một hệ thống ở trong biểu đồ use case không nhất thiết là một hệ phần mềm; nó có thể là một chiếc máy, hoặc là một hệ thống thực như một doanh nghiệp, một trường đại học,...
- Tác nhân(actor): là người dùng của hệ thống, một tác nhân có thể là một người dùng thực hoặc các hệ thống máy tính khác có vai trò nào đó trong hoạt động của hệ thống. Như vậy, tác nhân thực hiện các use case. Một tác nhân có thể thực hiện nhiều use case và ngược lại một use case cũng có thể được thực hiện bởi nhiều tác nhân



Tác nhân được kí hiệu:
hoặc



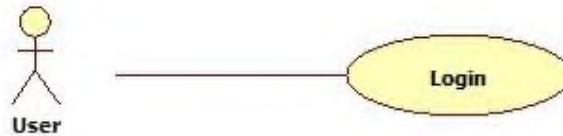
- Các use case: Đây là thành phần cơ bản của biểu đồ use case. Các use case được biểu diễn bởi các hình elip. Tên các use case thể hiện một chức năng xác định của hệ thống. Các Use case được kí hiệu bằng hình elips.



- Mối quan hệ giữa các use case:
 - o Association: thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau

Association

Ví dụ quan hệ association:

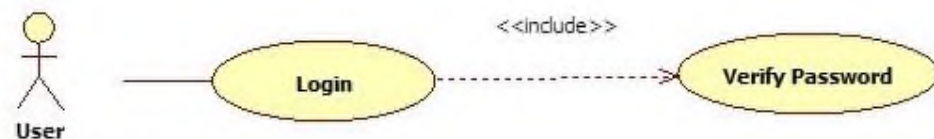


- o Include: là quan hệ giữa các Use Case với nhau, nó mô tả việc một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa) hoặc thể hiện sự dùng lại.

<<Include>>

→ Include

Ví dụ quan hệ include:



- o Extent: Extend dùng để mô tả quan hệ giữa 2 Use Case. Quan hệ Extend được sử dụng khi có một Use Case được tạo ra để bổ sung chức năng cho một Use Case có sẵn và được sử dụng trong một điều kiện nhất định nào đó.

2.3.2. Biểu đồ lớp

Classes (Các lớp)

Class là thành phần chính của bản vẽ Class Diagram. Class mô tả về một nhóm đối tượng có cùng tính chất, hành động trong hệ thống. Ví dụ mô tả về khách hàng chúng ta dùng lớp “Customer”. Class được mô tả gồm tên Class, thuộc tính và phương thức.

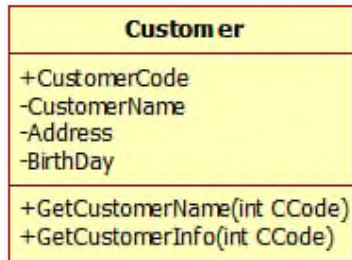
Class Name
Attributes
Methods

Ký hiệu về Class

Trong đó,

- Class Name: là tên của lớp.
- Attributes (thuộc tính): mô tả tính chất của các đối tượng. Ví dụ như khách hàng có Mã khách hàng, Tên khách hàng, Địa chỉ, Ngày sinh v.v...

- **Method (Phương thức):** chỉ các hành động mà đối tượng này có thể thực hiện trong hệ thống. Nó thể hiện hành vi của các đối tượng do lớp này tạo ra.



Ví dụ về một Class

Một số loại Class đặc biệt như Abstract Class (lớp không tạo ra đối tượng), Interface (lớp khai báo mà không cài đặt) v.v.. chúng ta xem thêm các tài liệu về lập trình hướng đối tượng để hiểu rõ hơn các vấn đề này.

Ý nghĩa

Trong phương pháp hướng đối tượng, một nhóm đối tượng có chung một số thuộc tính và phương thức tạo thành một lớp. Mỗi tương tác giữa các đối tượng trong hệ thống sẽ được biểu diễn thông qua mối quan hệ giữa các lớp.

Các lớp (bao gồm cả các thuộc tính và phương thức) cùng với các mối quan hệ sẽ tạo thành biểu đồ lớp (class diagram). Biểu đồ lớp là một biểu đồ dạng mô hình tĩnh nhằm mô tả hướng cách nhìn tĩnh về một hệ thống bằng các khái niệm lớp, các thuộc tính, phương thức của lớp và mối quan hệ giữa chúng với nhau.

Tập ký hiệu UML cho biểu đồ lớp

Trong phần này, chúng ta sẽ xem xét các vấn đề liên quan đến biểu diễn biểu đồ lớp trong UML.

Đăng nhập

Đăng ký

Mua sách

Kí hiệu lớp: trong UML, mỗi lớp được biểu diễn bởi hình chữ nhật gồm ba phần: tên lớp, các thuộc tính và các phương thức.

Thuộc tính: các thuộc tính trong biểu đồ lớp được biểu diễn theo cấu trúc chung như sau:

phạmvi tên: kiểu sốĐỐItượng = mặcđịnh (Giá trịGiớihạn)

Trong đó:

phạmvi cho biết phạm vi truy nhập của thuộc tính. Có bốn kiểu xác định phạm vi thuộc tính là:

+: thuộc tính kiểu public

#: thuộc tính kiểu protected

-: thuộc tính kiểu private.

~: thuộc tính được phép truy nhập tới từ các lớp trong cùng package

Các phạm vi của thuộc tính có thể được biểu diễn dưới dạng ký hiệu (+, #, -, ~) như trong UML hoặc biểu diễn dưới dạng các từ khoá (public, protected, private) như trong các ngôn ngữ lập trình.

Tên: là xâu ký tự biểu diễn tên thuộc tính.

kiểu: là kiểu dữ liệu của thuộc tính.