

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN
TRƯỜNG CAO ĐẲNG CƠ GIỚI NINH BÌNH

GIÁO TRÌNH

MÔN HỌC: Phân tích thiết kế hệ thống thông tin

NGHỀ: Lập trình máy tính

TRÌNH ĐỘ: Cao đẳng

*Ban hành kèm theo Quyết định số: /QĐ-... ngày.....tháng....năm
..... của.....*

Hình minh họa

(tùy thuộc vào từng môn học lựa chọn hình minh họa cho thích hợp)

Ninh Bình, năm 2016

--

Tailieu.vn

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.

TaiLieu.vn

LỜI GIỚI THIỆU

Phương pháp luận phát triển các hệ thống thông tin luôn là một trong những chủ đề quan trọng nhất của công nghệ thông tin. Trải qua một giai đoạn tiến hoá lâu dài, phát triển theo cách tiếp cận hướng đối tượng đã dần dần chiếm ưu thế và ngày càng trở nên phổ biến và đã được chuẩn hoá trong công nghiệp phần mềm.

Cùng với sự ra đời của ngôn ngữ mô hình hoá thống nhất UML và nhiều công cụ hỗ trợ như Rational Rose, AgroUML... phương pháp luận phát triển phần mềm hướng đối tượng đã được áp dụng rộng rãi trong công nghiệp phần mềm trên khắp thế giới. Ngôn ngữ UML hiện thời vẫn đang được phát triển để đáp ứng cho nhiều yêu cầu và nhiều dạng hệ thống khác nhau như hệ phân tán, hệ nhúng...

Tài liệu này nhằm giới thiệu cho sinh viên các khái niệm cơ bản của hướng đối tượng và UML, sau đó trình bày các bước phân tích thiết kế hệ thống thông tin dựa trên UML và công cụ Rational Rose. Nội dung của tài liệu gồm 4 chương và phần Phụ lục:

Chương 1: Mở đầu. Giới thiệu các dạng hệ thống thông tin và các khái niệm cơ bản của cách tiếp cận hướng đối tượng; vòng đời phát triển hệ thống và so sánh các cách tiếp cận phát triển hệ thống.

Chương 2: UML và Công cụ phát triển hệ thống. Trình bày các khái niệm cơ bản của UML, các biểu đồ, các ký hiệu UML và các bước phát triển hệ thống sử dụng các biểu đồ đó. Chương này cũng giới thiệu công cụ Rational Rose cho phân tích thiết kế hệ thống thông tin.

Chương 3: Phân tích hướng đối tượng. Trình bày các bước phân tích hệ thống theo các biểu đồ UML bao gồm: xây dựng mô hình use case, xây dựng mô hình lớp và biểu đồ trạng thái. Tài liệu cũng đưa ra những gợi ý cho từng bước và hướng dẫn sử dụng công cụ Rational Rose cho các bước đó.

Chương 4: Tổng quan về thiết kế hướng đối tượng. Trình bày các bước thiết kế hệ thống bao gồm: xây dựng các biểu đồ tương tác, biểu đồ lớp chi tiết, thiết kế chi tiết và xây dựng biểu đồ triển khai hệ thống. Tài liệu cũng có những gợi ý cho từng bước của pha thiết kế.

Phần Phụ lục. Trình bày toàn bộ quá trình phân tích thiết kế hệ thống quản lý thư viện và phát sinh mã cho hệ thống này.

Mỗi chương đều có phần câu hỏi, bài tập để giúp sinh viên hiểu rõ hơn kiến thức được học và kiểm tra khả năng áp dụng kiến thức của sinh viên vào các bài toán thực tế.

Tài liệu này được xây dựng nhằm đáp ứng nhu cầu học tập của sinh viên từ xa của Học viện Công nghệ Bưu chính Viễn thông. Do thời gian có hạn nên phiên bản đầu tiên này chắc chắn còn nhiều hạn chế và thiếu sót. Các tác giả rất mong nhận được những đóng góp ý kiến của các đồng nghiệp và các bạn sinh viên..

Trân thành cảm ơn các thầy, cô giáo trong ban giám hiệu, lãnh đạo phòng khoa và đặc biệt là các thầy, cô giáo trong tổ bộ môn đã đóng góp ý kiến để giáo trình được hoàn thành.

Ninh Bình, ngày 23 tháng 6 năm 2016

Tham gia biên soạn

1. Chủ biên Phạm Thị Thoa
2. Phan Huy Thành
3. Nguyễn Anh Văn

MỤC LỤC

	TRANG
Chương 1: Mở đầu	4
1.1. <u>Khái quát vòng đời phát triển hệ thống thông tin</u>	6
1.2. <u>Các cách tiếp cận phân tích thiết kế hệ thống</u>	8
1.2.1. <u>Phương pháp hướng cấu trúc</u>	8
1.2.2. <u>Phương pháp hướng đối tượng</u>	9
1.3. <u>Các khái niệm cơ bản của hướng đối tượng</u>	10
1.4. <u>Các bước phân tích thiết kế hướng đối tượng</u>	12
Chương 2: UML và công cụ phát triển hệ thống	20
2.1 <u>Giới thiệu về UML</u>	20
2.1.1 <u>Lịch sử ra đời của UML</u>	20
2.1.2 <u>UML – Ngôn ngữ mô hình hoá hướng đối tượng</u>	22
2.1.3 <u>Các khái niệm cơ bản trong UML</u>	25
2.2 <u>Các biểu đồ UML</u>	29
2.2.1 <u>Biểu đồ use case</u>	31
2.2.2 <u>Biểu đồ lớp</u>	35
2.2.3 <u>Biểu đồ trạng thái</u>	39
2.2.4 <u>Biểu đồ tương tác dạng tuần tự</u>	42
2.2.5 <u>Biểu đồ tương tác dạng cộng tác</u>	45
2.2.6 <u>Biểu đồ hoạt động</u>	48
2.2.7 <u>Biểu đồ thành phần</u>	53
2.2.8 <u>Biểu đồ triển khai hệ thống</u>	54
2.3 <u>Giới thiệu công cụ Rational Rose</u>	57
Chương 3: Phân tích hướng đối tượng	65

<u>3.1 Tổng quan về phân tích hướng đối tượng</u>	65
<u>3.1.1 Vai trò của pha phân tích</u>	65
<u>3.1.2 Các bước phân tích hướng đối tượng</u>	69
<u>3.1.3 Ví dụ</u>	75
<u>3.2 Mô hình Use Case và kịch bản</u>	83
<u>3.2.1 Vai trò của mô hình use case</u>	83
<u>3.2.2 Xây dựng biểu đồ use case</u>	86
<u>3.2.3 Xây dựng biểu đồ use case trong Rational Rose</u>	89
<u>3.3 Mô hình lớp</u>	91
<u>3.3.1 Vấn đề xác định lớp</u>	91
<u>3.3.2 Xây dựng biểu đồ lớp trong pha phân tích</u>	
<u>3.3.3 Biểu diễn biểu đồ lớp trong Rational Rose</u>	93
<u>3.4 Mô hình động dựa trên biểu đồ trạng thái</u>	95
<u>3.4.1 Khái quát về mô hình động</u>	95
<u>3.4.3 Xây dựng biểu đồ trạng thái</u>	97
<u>3.4.3 Biểu diễn biểu đồ trạng thái trong Rational Rose</u>	99
Chương 4: Tổng quan về thiết kế hướng đối tượng	101
<u>4.1 Tổng quan về thiết kế hướng đối tượng</u>	101
<u>4.1.1 Vai trò của pha thiết kế</u>	101
<u>4.1.2 Các bước thiết kế hướng đối tượng</u>	103
<u>4.2 Các biểu đồ tương tác</u>	105
<u>4.2.2 Xây dựng biểu đồ tuần tự</u>	105
<u>4.2.3 Xây dựng biểu đồ cộng tác</u>	107
<u>4.2.4 Biểu diễn các biểu đồ tương tác trong Rational Rose</u>	108
<u>4.3 Biểu đồ lớp chi tiết</u>	109

<u>4.3.1 Xác định các phương thức cho mỗi lớp</u>	109
<u>4.3.2 Xác định mối quan hệ giữa các lớp</u>	111
<u>4.3.3 Hoàn chỉnh biểu đồ lớp chi tiết</u>	112
<u>4.4 Thiết kế chi tiết</u>	113
<u>4.4.1 Xây dựng biểu đồ hoạt động cho các phương thức</u>	113
<u>4.4.2 Xây dựng bảng thiết kế chi tiết</u>	114
<u>4.5 Biểu đồ thành phần và biểu đồ triển khai</u>	114
<u>4.5.1 Xây dựng biểu đồ thành phần</u>	108
<u>4.5.2 Xây dựng biểu đồ triển khai</u>	109
<u>4.5.3 Biểu diễn biểu đồ thành phần và triển khai trong Rational Rose</u>	111

MÔN HỌC: PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Mã môn học: MH20

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

- Vị trí của môn học: Môn học được bố trí sau khi học xong các môn cơ sở dữ liệu và lập trình cơ bản;
- Tính chất của môn học: là môn học lý thuyết chuyên ngành bắt buộc.

Mục tiêu của môn học:

- Nhằm đào tạo cho học viên nắm được vòng đời phát triển hệ thống thông tin, các biểu đồ UML và thiết kế hướng đối tượng;
- Đảm bảo an toàn cho người và trang thiết bị.

Nội dung của môn học:

Số TT	Tên chương mục	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra
1	Mở đầu	5	5	0	
2	UML và công cụ phát triển hệ thống	25	22	2	1
3	Phân tích hướng đối tượng	30	19	10	1
4	Pha thiết kế hướng đối tượng	30	14	15	1
	Cộng	90	60	27	3

CHƯƠNG I

MỞ ĐẦU

Mã chương: PTTKHT01

Giới thiệu: Phần mở đầu là phần giới thiệu sơ lược về lịch sử môn học và khái quát một số khái niệm cơ bản.

Mục tiêu: Chương này tập trung trình bày các nội dung sau đây:

- Các hệ thống thông tin và vấn đề phát triển hệ thống thông tin
- Khái quát vòng đời phát triển hệ thống thông tin
- Các cách tiếp cận phân tích và thiết kế hệ thống
- Các khái niệm cơ bản của hướng đối tượng

Nội dung chính:

1.1. Khái quát vòng đời phát triển hệ thống thông tin

1.2. Các cách tiếp cận phân tích thiết kế hệ thống

1.2.1. Phương pháp hướng cấu trúc

1.2.2. Phương pháp hướng đối tượng

1.3. Các khái niệm cơ bản của hướng đối tượng

1.4. Các bước phân tích thiết kế hướng đối tượng

1.1 Các hệ thống thông tin

Mục tiêu: làm được một số khái niệm về hệ thống thông tin.

Ngày nay, hệ thống thông tin đã được ứng dụng trong mọi lĩnh vực khác nhau của đời sống xã hội. Tùy theo quan điểm mà có thể phân loại các hệ thống thông tin theo các tiêu chí khác nhau. Xét về mặt ứng dụng, hệ thống thông tin có thể được phân chia thành một số dạng như sau:

Hệ thống thông tin quản lý: Bao gồm các hệ thống thông tin hỗ trợ các hoạt động nghiệp vụ và quản lý của các doanh nghiệp, các tổ chức. Ví dụ các hệ thống quản lý nhân sự, hệ thống kế toán, hệ thống tính cước và chăm sóc khách hàng, hệ thống quản lý thư viện, hệ thống đào tạo trực tuyến ...

Các hệ thống Website: là các hệ thống có nhiệm vụ cung cấp thông tin cho người dùng trên môi trường mạng Internet. Các hệ thống Website có đặc điểm là thông tin cung cấp cho người dùng có tính đa dạng (có thể là tin tức hoặc các

dạng file đa phương tiện) và được cập nhật thường xuyên.

Hệ thống thương mại điện tử: Là các hệ thống website đặc biệt phục vụ việc trao đổi mua bán hàng hoá, dịch vụ trên môi trường Internet. Hệ thống thương mại điện tử bao gồm cả các nền tảng hỗ trợ các giao thức mua bán, các hình thức thanh toán, chuyển giao hàng hoá ...

Hệ thống điều khiển: là các hệ thống phần mềm gắn với các thiết bị phần cứng hoặc các hệ thống khác nhằm mục đích điều khiển và giám sát hoạt động của thiết bị hay hệ thống đó.

Mỗi loại hệ thống thông tin có những đặc trưng riêng và cũng đặt ra những yêu cầu riêng cho việc phát triển hệ thống. Ví dụ, các hệ thống điều khiển đòi hỏi những yêu cầu về môi trường phát triển, hệ điều hành và ngôn ngữ lập trình riêng; các hệ website thực thi các chức năng trên môi trường mạng phân tán đòi hỏi cách phát triển riêng...Do vậy, không có một phương pháp luận chung cho tất cả các dạng hệ thống thông tin.

Phạm vi của tài liệu này nhằm giới thiệu một số khái niệm cơ bản của UML cho phát triển các hệ thống và để dễ dàng minh họa chúng ta sẽ xem xét vấn đề phát triển dạng hệ thống thông tin phổ biến nhất là hệ thống thông tin quản lý.

1.2 Khái quát vòng đời phát triển hệ thống thông tin

Việc phát triển các hệ thống thông tin không chỉ đơn giản là lập trình mà luôn được xem như một tiến trình hoàn chỉnh.

Tiến trình phần mềm là phương cách sản xuất ra phần mềm với các thành phần chủ yếu bao gồm: mô hình vòng đời phát triển phần mềm, các công cụ hỗ trợ cho phát triển phần mềm và những người trong nhóm phát triển phần mềm.

Như vậy, tiến trình phát triển phần mềm nói chung là sự kết hợp cả hai khía cạnh kỹ thuật (vòng đời phát triển, phương pháp phát triển, các công cụ và ngôn ngữ sử dụng, ...) và khía cạnh quản lý (quản lý dự án phần mềm).

Mô hình vòng đời phần mềm là các bước phát triển một sản phẩm phần mềm cụ thể. Một vòng đời phát triển phần mềm thường có các pha cơ bản sau:

Pha xác định yêu cầu: khám phá các khái niệm liên quan đến việc phát triển phần mềm, xác định chính xác yêu cầu và các ràng buộc của khách hàng với sản phẩm phần mềm đó.

Pha phân tích: mô tả chức năng của sản phẩm, các input của sản phẩm và các output được yêu cầu; khám phá các khái niệm trong miền quan tâm của sản phẩm và bước đầu đưa ra giải pháp xây dựng hệ thống.

Pha thiết kế: xác định cụ thể phần mềm sẽ được xây dựng như thế nào. Pha thiết kế bao gồm hai mức là thiết kế kiến trúc và thiết kế chi tiết.

Pha cài đặt tích hợp: cài đặt chi tiết và tích hợp hệ thống phần mềm dựa

trên kết quả của pha thiết kế.

Pha bảo trì: tiến hành sửa chữa phần mềm khi có các thay đổi. Đây là pha rất quan trọng, tiêu tốn nhiều thời gian và chi phí nhất trong tiến trình phát triển phần mềm.

Pha loại bỏ: thực hiện loại bỏ phần mềm hoặc thay thế phần mềm bởi một phần mềm hoàn toàn mới.

Thông thường hai quá trình không thể thiếu được trong vòng đời phát triển phần mềm là *viết tài liệu* và *kiểm thử*. Các quá trình này không trở thành một pha riêng biệt mà được tiến hành song song với tất cả các pha khác trong tiến trình phần mềm nghĩa là tất cả các pha đều phải viết tài liệu và kiểm thử với các mức độ khác nhau.

Có rất nhiều mô hình vòng đời phần mềm nhưng hai mô hình đơn giản và được sử dụng rộng rãi nhất là mô hình thác nước và mô hình làm bản mẫu nhanh.

1.2.1 Mô hình thác nước

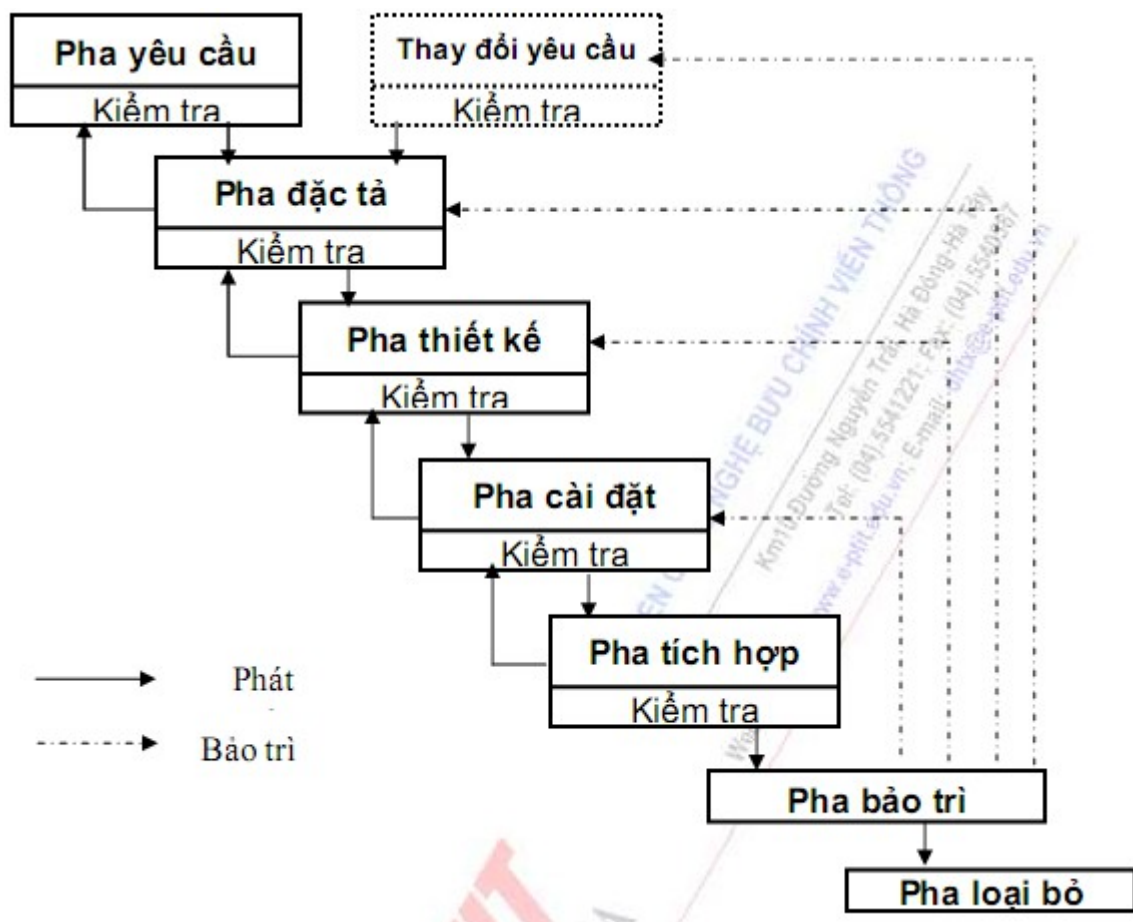
Theo mô hình thác nước, sau khi yêu cầu của hệ thống đã được xác định và kiểm tra bởi nhóm SQA, pha phân tích sẽ được tiến hành để xây dựng tài liệu. Sau khi tài liệu phân tích được khách hàng chấp nhận, nhóm phát triển sẽ tiến hành lập kế hoạch và lịch biểu cho các quá trình phát triển tiếp theo. Sau đó, các pha thiết kế, cài đặt và tích hợp sẽ lần lượt được tiến hành; mỗi pha này đều có phần kiểm tra để khi cần có thể quay lại sửa đổi tài liệu của pha trước đó. Khi phần mềm đã được triển khai và chuyển sang pha bảo trì; nếu có lỗi hoặc thay đổi xảy ra, nhóm thiết kế sẽ phải quay trở lại sửa đổi tài liệu cho một trong các pha trước đó và nếu cần có thể quay trở lại thay đổi một số yêu cầu ban đầu của hệ thống.

Vì các pha cứ nối tiếp nhau một cách liên tục như một thác nước nên mô hình này được gọi là mô hình thác nước. Tiến trình phần mềm theo mô hình thác nước được biểu diễn như trong Hình 1.1. Mô hình thác nước có một số ưu điểm như sau:

Có vòng lặp, cho phép trở về pha trước trong vòng đời phần mềm để sửa chữa khi phát hiện lỗi hoặc khi có thay đổi.

- Hướng tài liệu: tất cả các pha trong vòng đời phần mềm theo mô hình thác nước đều được viết tài liệu cẩn thận và được kiểm tra bởi nhóm SQA trước khi chuyển sang pha tiếp theo. Do vậy, hệ thống sẽ dễ dàng bảo trì khi có những thay đổi.

Tuy nhiên, mô hình thác nước cũng có nhược điểm là sản phẩm phần mềm cuối cùng có thể không thỏa mãn nhu cầu thực sự của khách hàng. Lý do là khách hàng chỉ được trao đổi một lần duy nhất và chưa được hình dung sản phẩm nên rất có thể các pha tiếp theo sẽ không thực hiện đúng những gì khách hàng cần.



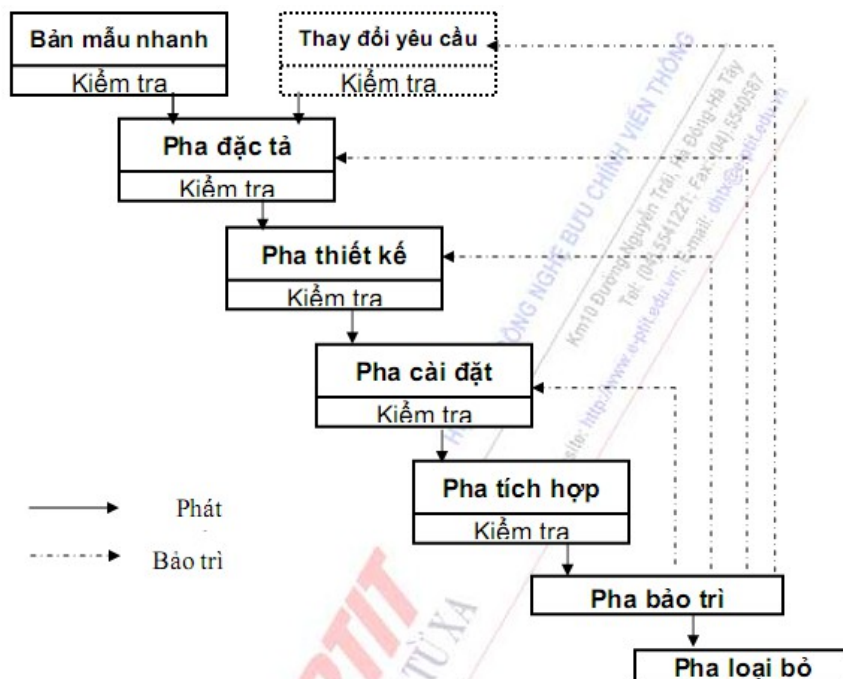
Hình 1.1: Tiến trình phần mềm theo mô hình thác nước

1.2.2 Mô hình làm bản mẫu nhanh

Trong mô hình làm bản mẫu nhanh, bước đầu tiên là nhóm phát triển sẽ xây dựng một bản mẫu và giao cho khách hàng và người sử dụng hệ thống dùng thử. Khi khách hàng đồng ý chấp nhận bản mẫu thì nhóm phát triển mới tiếp tục tiến hành các pha khác của vòng đời phần mềm. Trong các pha tiếp theo, do đã có bản mẫu nên các pha sẽ được tiến hành liên tục và không có bước quay về pha trước đó.

Chỉ khi hệ thống đã triển khai và chuyển sang pha bảo trì, nếu có thay đổi hay phát hiện lỗi thì nhóm phát triển mới quay lại một trong những pha trước đó, nhưng không quay lại pha làm bản mẫu vì bản mẫu đã được chấp nhận.

Ưu điểm chính của mô hình này là “nhanh” và hơn nữa do sản phẩm phần mềm được tạo ra từ mô hình làm bản mẫu nên có khả năng cao là đảm bảo thỏa mãn yêu cầu thực sự của khách hàng. Tuy nhiên, mô hình làm bản mẫu nhanh cũng có nhược điểm do các pha được tiến hành liên tục mà không được viết tài liệu. Mô hình làm bản mẫu nhanh được biểu diễn như trong Hình 1.2.



Hình 1.2: Vòng đời phát triển phần mềm theo mô hình làm bản mẫu nhanh

1.3 Các cách tiếp cận phân tích thiết kế hệ thống

Mục tiêu:

- Phương pháp hướng cấu trúc

- Phương pháp hướng đối tượng

Trong những năm 70 - 80, phương pháp hướng cấu trúc được coi là phương pháp chuẩn để phát triển phần mềm. Tuy nhiên, phương pháp này tỏ ra không phù hợp trong phát triển các hệ phần mềm lớn và đặc biệt là kém hiệu quả trong sử dụng lại - một yêu cầu quan trọng trong công nghiệp phần mềm. Thập niên 90 chứng kiến sự nở rộ trong nghiên cứu và xây dựng phương pháp luận phát triển phần mềm hướng đối tượng và nhanh chóng trở thành phổ biến trong công nghiệp phần mềm ngày nay. Để hiểu rõ phần nào sự khác biệt này phần này dành so sánh một số khác biệt giữa hai phương pháp này.

1.3.1 Phương pháp hướng cấu trúc

Đặc trưng của phương pháp hướng cấu trúc là phân chia chương trình chính thành nhiều chương trình con, mỗi chương trình con nhằm đến thực hiện một công việc xác định.

Trong phương pháp hướng cấu trúc, phần mềm được thiết kế dựa trên một trong hai hướng : hướng dữ liệu và hướng hành động.

- Cách tiếp cận hướng dữ liệu xây dựng phần mềm dựa trên việc phân rã phần mềm theo các chức năng cần đáp ứng và dữ liệu cho các chức năng đó. Cách tiếp cận hướng dữ liệu sẽ giúp cho những người phát triển hệ thống dễ dàng xây dựng ngân hàng dữ liệu.

- Cách tiếp cận hướng hành động lại tập trung phân tích hệ phần mềm dựa trên các hoạt động thực thi các chức năng của phần mềm đó.

Cách thức thực hiện của phương pháp hướng cấu trúc là **phương pháp thiết kế từ trên xuống (top-down)**. Phương pháp này tiến hành phân rã bài toán thành các bài toán nhỏ hơn, rồi tiếp tục phân rã các bài toán con cho đến khi nhận được các bài toán có thể cài đặt được ngay sử dụng các hàm của ngôn ngữ lập trình hướng cấu trúc.

Phương pháp hướng cấu trúc có ưu điểm là tư duy phân tích thiết kế rõ ràng, chương trình sáng sửa dễ hiểu. Tuy nhiên, phương pháp này có một số nhược điểm sau:

- Không hỗ trợ việc sử dụng lại. Các chương trình hướng cấu trúc phụ thuộc chặt chẽ vào cấu trúc dữ liệu và bài toán cụ thể, do đó không thể dùng lại một modul nào đó trong phần mềm này cho phần mềm mới với các yêu cầu về dữ liệu khác.

- Không phù hợp cho phát triển các phần mềm lớn. Nếu hệ thống thông tin lớn, việc phân ra thành các bài toán con cũng như phân các bài toán con thành các modul và quản lý mối quan hệ giữa các modul đó sẽ là không phải là dễ dàng và dễ gây ra các lỗi trong phân tích và thiết kế hệ thống, cũng như khó kiểm thử và bảo trì.

1.3.2 Phương pháp hướng đối tượng

Khác với phương pháp hướng cấu trúc chỉ tập trung hoặc vào dữ liệu hoặc vào hành động, phương pháp hướng đối tượng tập trung vào cả hai khía cạnh của hệ thống là dữ liệu và hành động.

Cách tiếp cận hướng đối tượng là một lối tư duy theo cách ánh xạ các thành phần trong bài toán vào các đối tượng ngoài đời thực. Với cách tiếp cận này, một hệ thống được chia tương ứng thành các thành phần nhỏ gọi là các *đối tượng*, mỗi đối tượng bao gồm đầy đủ cả dữ liệu và hành động liên quan đến đối tượng đó.

Các đối tượng trong một hệ thống tương đối độc lập với nhau và phần mềm sẽ được xây dựng bằng cách kết hợp các đối tượng đó lại với nhau thông qua các mối quan hệ và tương tác giữa chúng. Các nguyên tắc cơ bản của phương pháp hướng đối tượng bao gồm :

- **Trừu tượng hóa (abstraction):** trong phương pháp hướng đối tượng, các thực thể phần mềm được mô hình hóa dưới dạng các đối tượng. Các đối tượng này được trừu tượng hóa ở mức cao hơn dựa trên thuộc tính và phương thức mô tả đối tượng để tạo thành các lớp. Các lớp cũng sẽ được trừu tượng hóa ở mức cao hơn nữa để tạo thành một sơ đồ các lớp được kế thừa lẫn nhau. Trong phương pháp hướng đối tượng có thể tồn tại những lớp không có đối tượng tương ứng, gọi là *lớp trừu tượng*. Như vậy, nguyên tắc cơ bản để xây dựng các khái niệm trong hướng đối tượng là sự trừu tượng hóa theo các mức độ khác nhau.

- **Tính đóng gói (encapsulation) và ẩn giấu thông tin:** các đối tượng có thể có những phương thức hoặc thuộc tính riêng (*kiểu private*) mà các đối tượng khác không thể sử dụng được. Dựa trên nguyên tắc ẩn giấu thông tin này, cài đặt của các đối tượng sẽ hoàn toàn độc lập với các đối tượng khác, các lớp độc lập với nhau và cao hơn nữa là cài đặt của hệ thống hoàn toàn độc lập với người sử dụng cũng như các hệ thống khác sử dụng kết quả của nó.

- **Tính modul hóa (modularity):** các bài toán sẽ được phân chia thành những vấn đề nhỏ hơn, đơn giản và quản lý được.

- **Tính phân cấp (hierarchy):** cấu trúc chung của một hệ thống hướng đối tượng là dạng phân cấp theo các mức độ trừu tượng từ cao đến thấp.

Ưu điểm nổi bật của phương pháp hướng đối tượng là đã giải quyết được các vấn đề nảy sinh với phương pháp hướng cấu trúc:

- **Hỗ trợ sử dụng lại mã nguồn :** Chương trình lập trình theo phương

pháp hướng đối tượng thường được chia thành các gói là các nhóm của các lớp đối tượng khác nhau. Các gói này hoạt động tương đối độc lập và hoàn toàn có thể sử dụng lại trong các hệ thống thông tin tương tự.

- **Phù hợp với các hệ thống lớn:** Phương pháp hướng đối tượng không chia bài toán thành các bài toán nhỏ mà tập trung vào việc xác định các đối tượng, dữ liệu và hành động gắn với đối tượng và mối quan hệ giữa các đối tượng. Các đối tượng hoạt động độc lập và chỉ thực hiện hành động khi nhận được yêu cầu từ các đối tượng khác. Vì vậy, phương pháp này hỗ trợ phân tích, thiết kế và quản lý một hệ thống lớn, có thể mô tả các hoạt động nghiệp vụ phức tạp bởi quá trình phân tích thiết kế không phụ thuộc vào số biến dữ liệu hay số lượng thao tác cần thực hiện mà chỉ quan tâm đến các đối tượng tồn tại trong hệ thống đó.

1.4. Các khái niệm cơ bản của hướng đối tượng

Mục tiêu: Làm được khái niệm cơ bản trong hướng đối tượng

Một số khái niệm cơ bản trong hướng đối tượng bao gồm:

- **Đối tượng (object):** một đối tượng biểu diễn một thực thể vật lý, một thực thể khái niệm hoặc một thực thể phần mềm. Có thể định nghĩa một đối tượng là một khái niệm, sự trừu tượng hoặc một vật với giới hạn rõ ràng và có ý nghĩa với một ứng dụng cụ thể.

- **Lớp (Class):** là mô tả của một nhóm đối tượng có chung các thuộc tính, hành vi và các mối quan hệ. Như vậy, một đối tượng là thể hiện của một lớp và một lớp là một định nghĩa trừu tượng của đối tượng.

- **Thành phần (component):** là một phần của hệ thống hoạt động độc lập và giữ một chức năng nhất định trong hệ thống.

- **Gói (package):** là một cách tổ chức các thành phần, phần tử trong hệ thống thành các nhóm. Nhiều gói có thể được kết hợp với nhau để trở thành một hệ thống con (subsystem).

- **Kế thừa:** Trong phương pháp hướng đối tượng, một lớp có thể có sử dụng lại các thuộc tính và phương thức của một hoặc nhiều lớp khác. Kiểu quan hệ này gọi là quan hệ kế thừa, được xây dựng dựa trên mối quan hệ kế thừa trong bài toán thực tế. Ví dụ, giải sử ta có lớp *Người* gồm các thuộc tính : *tên, ngày sinh, quê quán, giới tính* ; Lớp *Nhân Viên* có quan hệ kế thừa từ lớp *Người* sẽ có tất cả các thuộc tính trên và bổ sung thêm các thuộc tính mới gồm : *chức vụ, lương*.

Vòng đời phát triển phần mềm hướng đối tượng cũng có các pha tương tự như các vòng đời phát triển phần mềm nói chung. Các pha cơ bản đặc trưng trong phát triển phần mềm hướng đối tượng bao gồm:

- **Phân tích hướng đối tượng:** xây dựng một mô hình chính xác để mô tả hệ thống cần xây dựng là gì. Thành phần của mô hình này là các đối tượng gắn với hệ thống thực.

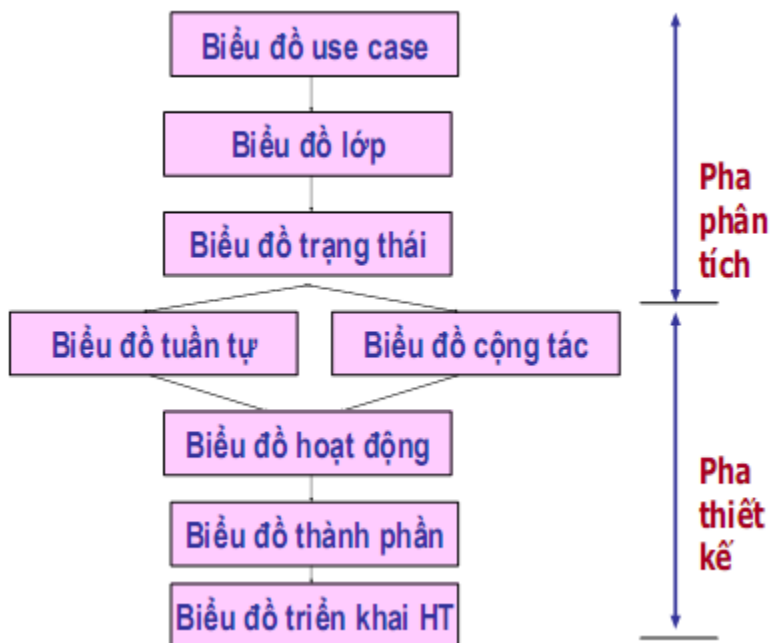
- **Thiết kế hướng đối tượng:** Là giai đoạn tổ chức chương trình thành các tập hợp đối tượng cộng tác, mỗi đối tượng trong đó là thực thể của một lớp. Kết quả của pha thiết kế cho biết hệ thống sẽ được xây dựng như thế nào qua các bản thiết kế kiến trúc và thiết kế chi tiết.

- **Lập trình và tích hợp:** Thực hiện bản thiết kế hướng đối tượng bằng cách sử dụng các ngôn ngữ lập trình hướng đối tượng (C++, Java, ...).

1.5. Các bước phân tích thiết kế hướng đối tượng

Mục tiêu: Làm được các bước phân tích hướng đối tượng

Các bước phân tích thiết kế hướng đối tượng được xây dựng dựa trên biểu đồ các ký hiệu UML. Đó là ngôn ngữ mô hình hoá thống nhất được xây dựng để mô hình hoá quá trình phát triển hệ thống phần mềm hướng đối tượng. Các vấn đề cơ bản về UML sẽ được giới thiệu chi tiết trong Chương 2. Phần này chỉ nhằm giới thiệu một cách khái quát các bước trong phân tích và thiết kế hướng đối tượng.



Hình 1.3: Các bước phát triển hệ thống hướng đối tượng

Pha phân tích

Xây dựng Biểu đồ use case: Dựa trên tập yêu cầu ban đầu, người phân tích tiến hành xác định các tác nhân, use case và các quan hệ giữa các use case để mô tả lại các chức năng của hệ thống. Một thành phần quan trọng trong biểu đồ use case

là các kịch bản mô tả hoạt động của hệ thống trong mỗi use case cụ thể.

Xây dựng Biểu đồ lớp: Xác định tên các lớp, các thuộc tính của lớp, một số phương thức và mối quan hệ cơ bản trong sơ đồ lớp.

Xây dựng biểu đồ trạng thái: Mô tả các trạng thái và chuyển tiếp trạng thái trong hoạt động của một đối tượng thuộc một lớp nào đó.

Trong Pha thiết kế

Xây dựng các biểu đồ tương tác (gồm biểu đồ cộng tác và biểu đồ tuần tự): mô tả chi tiết hoạt động của các use case dựa trên các scenario đã có và các lớp đã xác định trong pha phân tích.

Xây dựng biểu đồ lớp chi tiết: tiếp tục hoàn thiện biểu đồ lớp bao gồm bổ sung các lớp còn thiếu, dựa trên biểu đồ trạng thái để bổ sung các thuộc tính, dựa trên biểu đồ tương tác để xác định các phương thức và mối quan hệ giữa các lớp.

Xây dựng biểu đồ hoạt động: mô tả hoạt động của các phương thức phức tạp trong mỗi lớp hoặc các hoạt động hệ thống có sự liên quan của nhiều lớp. Biểu đồ hoạt động là cơ sở để cài đặt các phương thức trong các lớp.

Xây dựng biểu đồ thành phần: xác định các gói, các thành phần và tổ chức phần mềm theo các thành phần đó.

Xây dựng biểu đồ triển khai hệ thống: xác định các thành phần và các thiết bị cần thiết để triển khai hệ thống, các giao thức và dịch vụ hỗ trợ.

Tailieu.vn