

Chương 1: Tổng quan về phân tích thiết kế hệ thống

5



Nội dung

- 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

6



Những khó khăn khi phát triển phần mềm

- Chưa thấu hiểu yêu cầu, hoặc hiểu lầm
- Yêu cầu thường thay đổi trong thời gian phát triển.
- Yêu cầu thường được miêu tả bằng văn bản, dài dòng, khó hiểu.
- Đội quân phát triển phần mềm rất khó nhận thức thấu đáo các mối quan hệ tiềm ẩn và phức tạp cần được thể hiện chính xác trong các ứng dụng lớn.
- Khả năng nắm bắt các dữ liệu phức tạp của con người (tại cùng một thời điểm) là có hạn.

7



Tiến trình phần mềm

- 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
 - Những người trong nhóm phát triển phần mềm.

8



Software Development Life Cycle –SDLC

- Analysis**
 - *Planning*
 - *Requirements Determination*
- Design**
 - *Conceptual Design*
 - *Physical Design*
 - *Construction and/or Purchase (prototype)*
- Implementation**
 - *Conversion - old to new*
 - *Implementation*
- Deployment**
 - *Training*
 - *Evolution - maintenance & enhancements*



Nghiên cứu sơ bộ (Planning)

- Lập kế hoạch cho dự án
 - Khởi động dự án
 - Thiết lập đội ngũ nhân viên cho việc khởi động dự án
 - Thiết lập quan hệ với khách hàng
 - Lập kế hoạch cho dự án
 - Mô tả phạm vi của dự án
 - Chia dự án thành các tác vụ có thể thực hiện được
 - Ước lượng tài nguyên, lập kế hoạch tài nguyên
 - Đánh giá rủi ro
 - Lập kế hoạch hội họp
 - Ước lượng sơ bộ chi phí
- Thực thi dự án: giám sát tiến trình trong kế hoạch

10



Nghiên cứu sơ bộ

- Nhận biết các chức năng chính của hệ thống
- Đánh giá sơ bộ rủi ro
- Xem xét các yêu cầu, các nguồn tài nguyên có thể sử dụng
- Thảo luận, xem xét các khía cạnh thương mại, phân tích khả năng lời lỗ, đánh giá mức độ rủi ro
- Phải trả lời được câu hỏi: “Có nên phát triển hệ thống này hay không?”

11



Phân tích (Analysis)

- Xác định yêu cầu (Determining Requirement)
 - Phỏng vấn và lắng nghe
 - Đưa ra các câu hỏi phỏng vấn
 - Câu hỏi Open-ended
 - Câu hỏi closed-ended
- Xác định tổng thể hệ thống cần làm những gì
- Hệ thống lại các yêu cầu: bằng cách xây dựng mô hình xử lý
- Xây dựng mô hình dữ liệu quan niệm

12



Thiết kế (Design)

- Phải trả lời câu hỏi: “Hệ thống làm cách nào để thỏa mãn các yêu cầu”
- Thiết kế menu, form nhập, báo cáo (report)
- Thiết kế logic và vật lý các thành phần dữ liệu của hệ thống để tạo database
- Thiết kế các chức năng xử lý của hệ thống


13



Thực hiện (Implementation)

- Xây dựng phần mềm
 - Viết code theo yêu cầu quy định sẵn
 - Viết tài liệu giải thích rõ các chương trình, thủ tục, hàm
 - Kiểm nghiệm chương trình chạy đúng
- Thử nghiệm hệ thống
 - Thử nghiệm đơn vị
 - Thử nghiệm tích hợp (tích hợp nhiều thành phần)
 - Thử nghiệm toàn bộ hệ thống
 - Khách hàng thử nghiệm (về mức độ chấp nhận)

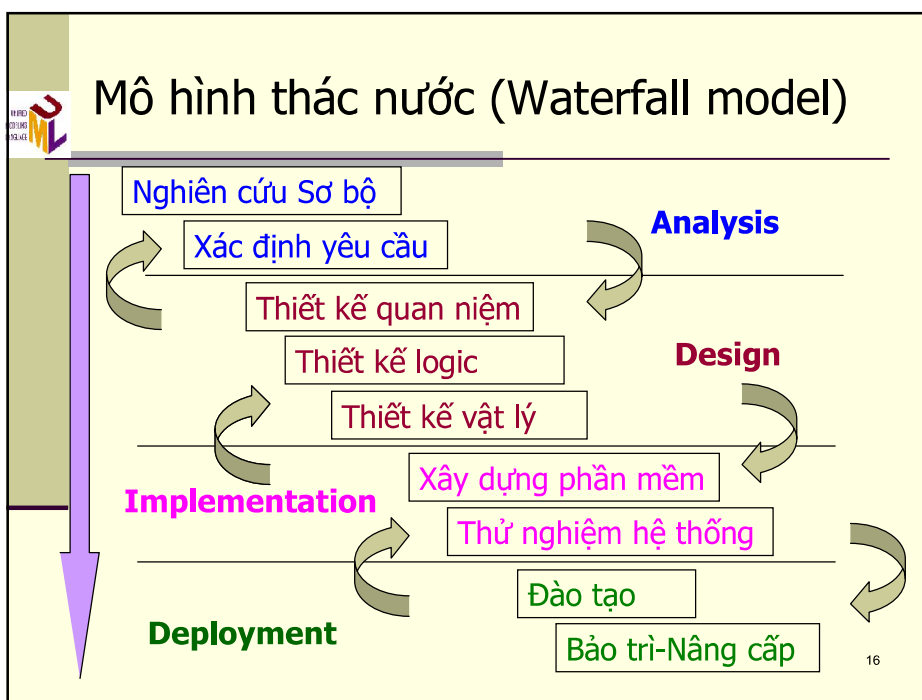
14



Triển khai (Deployment)

- Thực hiện triển khai
 - Huấn luyện người sử dụng
 - Tạo các file dữ liệu cần thiết
- Bảo trì nâng cấp
 - Hoạt động bảo trì tùy thuộc vào sự thay đổi của môi trường sử dụng
 - Các hệ thống khác nhau có những mức độ bảo trì khác nhau

15





Mô hình thác nước

■ Ưu điểm

- Có vòng lặp, cho phép trở về pha trước để sửa chữa khi phát hiện lỗi hoặc khi có thay đổi.
- Dễ bảo trì: tất cả các pha trong chu trình đều được viết tài liệu cẩn thận và được kiểm tra trước khi chuyển sang pha tiếp theo.

■ Khuyết điểm

- Sản phẩm phần mềm cuối cùng có thể không thỏa mãn nhu cầu thực sự của khách hàng

17



Những thành viên trong nhóm phát triển PM

■ Nhà phân tích (Analyst)

- Nghiên cứu yêu cầu của khách hàng để định nghĩa 1 phạm vi bài toán
- Nhận dạng nhu cầu tổ chức
- Xác định nhu cầu của nhân lực
- Xác định phương pháp và công nghệ
- Thiết kế cấu trúc Database

■ Nhà thiết kế (Designer)

- Giao diện chương trình (menu, form, report)
- Quyết định các yêu cầu về phần cứng và phần mềm

18



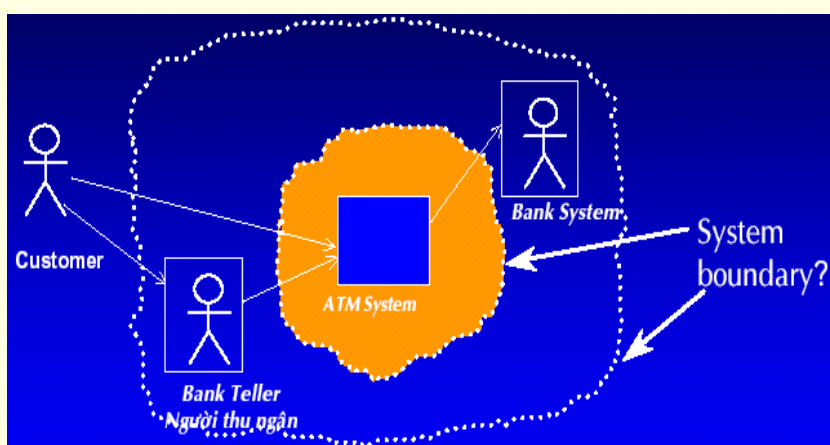
Những thành viên trong nhóm phát triển PM

- Chuyên gia lĩnh vực (Domain Expert)
- Lập trình viên (Programmer)
- Người dùng (User).

19



Giới hạn của hệ thống



20



Các cách tiếp cận phân tích hệ thống

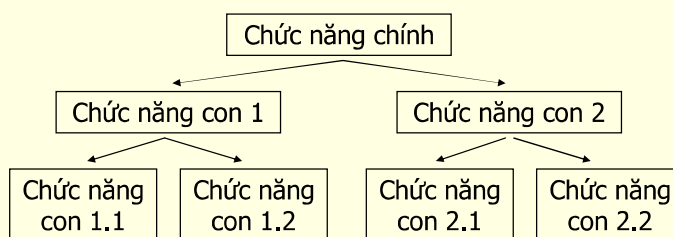
- Phương pháp hướng chức năng
- Phương pháp hướng đối tượng

21



Phương pháp hướng chức năng

- Phân rã các chức năng của người dùng thành các chức năng nhỏ hơn
- Đơn vị nhỏ nhất gọi là thủ tục hay hàm



- Hệ thống hoàn chỉnh khi các chức năng cũng hoàn chỉnh và không được thay đổi theo thời gian

22



Phương pháp hướng chức năng

■ Ưu điểm

- Phân tích thiết kế rõ ràng
- Chương trình sáng sửa dễ hiểu

■ Nhược điểm

- Chỉ quan tâm đến việc **quản lý dữ liệu** mà hệ thống cần lưu trữ (nhập xuất thông tin)
- Tập trung vào thông tin mà không quan tâm đến những gì xảy ra với thông tin và cách hoạt động của hệ thống
- Gặp nhiều khó khăn khi hệ thống thường xuyên thay đổi về các nguyên tắc nghiệp vụ.
- Không phù hợp cho phát triển các phần mềm lớn

23



Phương pháp hướng đối tượng

- Xem các thành phần trong hệ thống như các đối tượng ngoài đời thực
- Ứng dụng có thể chia thành các thành phần nhỏ gọi là đối tượng, sau đó chúng được tích hợp lại thành ứng dụng

24



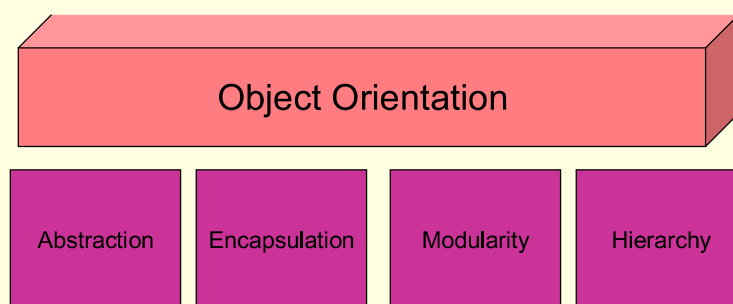
Phương pháp hướng đối tượng

- Ưu điểm
 - Hỗ trợ sử dụng lại mã nguồn
 - Phù hợp với các hệ thống lớn: Các đối tượng hoạt động độc lập và chỉ thực hiện hành động khi nhận được yêu cầu từ các đối tượng khác
 - Giảm lỗi, tăng tốc độ thiết kế và phát triển phần mềm


25



Các nguyên tắc cơ bản của hướng đối tượng




26



Các nguyên tắc cơ bản của hướng đối tượng

- Trừu tượng hóa (Abstraction)
 - Các lớp sẽ được trừu tượng hóa ở nhiều mức để tạo thành một sơ đồ các lớp được kế thừa lẫn nhau
- Tính đóng gói (Encapsulation) và ẩn dấu thông tin
 - Các đối tượng có thể có những phương thức hoặc thuộc tính riêng mà các đối tượng khác không thể sử dụng được
- 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


27



Các giai đoạn PTTK hướng đối tượng

- Phân tích hướng đối tượng (Object Oriented Analysis)
- Thiết kế hướng đối tượng (Object Oriented Design)
- Lập trình hướng đối tượng (Object Oriented Programming)


28



Giai đoạn phân tích hướng đối tượng (OOA)

- Giai đoạn phát triển một **mô hình** chính xác của vấn đề, có thành phần là các đối tượng có thực và dễ hiểu đối với người sử dụng
- Trong giai đoạn này, vấn đề được trình bày bằng các thuật ngữ tương ứng với các đối tượng có thực
- Mô hình thiết kế sẽ chứa các thực thể trong một vấn đề có thực và giữ nguyên các mẫu hình về cấu trúc, quan hệ cũng như hành vi của chúng

29



Giai đoạn phân tích hướng đối tượng (OOA)

- Ví dụ: Đối với ví dụ nhà băng lẻ, giai đoạn phân tích sẽ nhận biết được các thực thể như:
 - Loại tài khoản: ATM (rút tiền tự động), Savings (tiết kiệm), Current (bình thường), Fixed (đầu tư), ...
 - Khách hàng
 - Nhân viên
 - Phòng máy tính
- Tương tác và quan hệ giữa các đối tượng trên:
 - Một khách hàng mới mở một tài khoản tiết kiệm
 - Chuyển tiền từ tài khoản tiết kiệm sang tài khoản đầu tư
 - Chuyển tiền từ tài khoản tiết kiệm sang tài khoản ATM

30



Giai đoạn thiết kế hướng đối tượng (OOD)

- Trong giai đoạn OOD, nhà thiết kế định nghĩa các chức năng, thủ tục (operations), thuộc tính (attributes) cũng như mối quan hệ của một hay nhiều lớp (class) và quyết định chúng cần phải được điều chỉnh sao cho phù hợp với môi trường phát triển
- OOD cũng là giai đoạn để thiết kế ngân hàng dữ liệu và áp dụng các kỹ thuật tiêu chuẩn hóa

31



Giai đoạn lập trình hướng đối tượng (OOP)

- Giai đoạn xây dựng phần mềm có thể được thực hiện sử dụng kỹ thuật lập trình hướng đối tượng
- Một vài ngôn ngữ hướng đối tượng thường được nhắc tới là C++, C# và Java

32



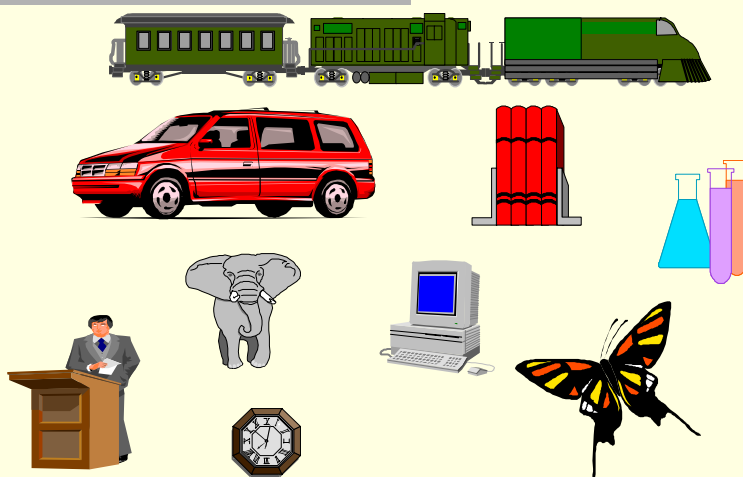
Các khái niệm cơ bản

- Đối tượng
- Lớp
- Thuộc tính
- Phương thức
- Thiết kế lớp đối tượng

33



Đối tượng là gì?

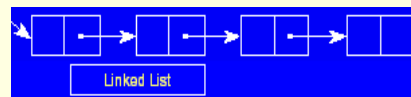
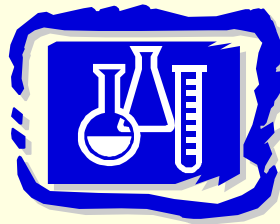


34



Đối tượng là gì?

- Đối tượng là 1 thực thể có thuộc tính và hành vi (phương thức)
- Đối tượng là:
 - Thực thể cụ thể (**tangible**) hay
 - Thực thể trừu tượng
 - Danh sách liên kết
 - Mảng 1 hay 2 chiều (ma trận), chuỗi ký tự,...
 - Một khái niệm,...



Các đặc trưng của đối tượng

- Một đối tượng có chứa:
 - Tên đối tượng
 - Các thuộc tính → mô tả trạng thái của đối tượng
 - Các thao tác hay phương thức → là các chức năng mà đối tượng đó có

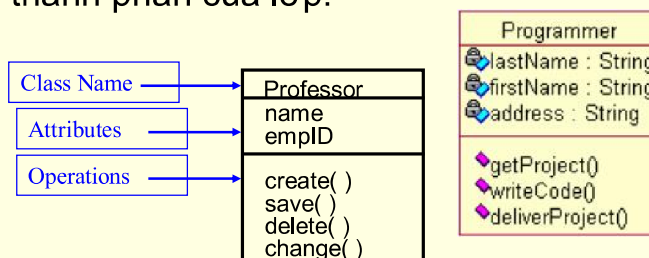


36



Lớp (Class)

- Lớp là 1 tập hợp các đối tượng có cùng tập thuộc tính (attributes), tập phương thức (operations), các quan hệ (relationships), và ngữ nghĩa (semantics)
- Các thành phần của lớp:

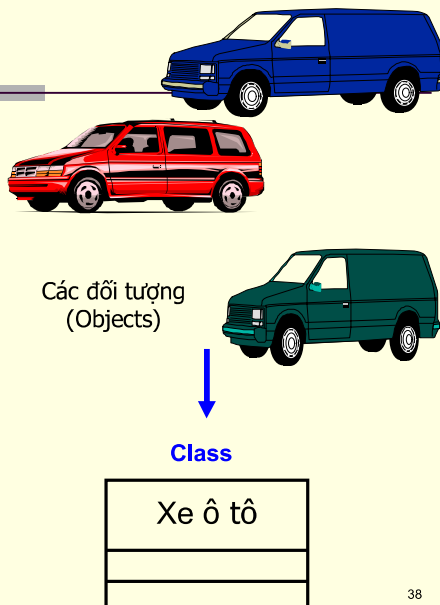


37



Lớp và đối tượng

- Lớp là định nghĩa từ đối tượng
 - Lớp định nghĩa cấu trúc và hành vi của các đối tượng trong lớp
 - Lớp được xem như 1 “**khuôn mẫu**” để đúc ra đối tượng
- Các đối tượng có chung kiểu thuộc tính và các phương thức được nhóm lại tạo thành lớp
- Đối tượng là thể hiện (**instance**) của lớp



38



Thuộc tính của lớp là gì?

- Tập các **thành phần dữ liệu** mô tả và phân biệt các đặc trưng của đối tượng thuộc về lớp
 - Thể hiện dưới dạng dữ liệu
- Được tham gia xử lý tính toán bởi các phương thức của lớp
- Thuộc tính cũng có thể là đối tượng
- Kiểu thuộc tính
 - Mỗi thuộc tính có 1 tên và thuộc về 1 kiểu dữ liệu duy nhất

39



Thuộc tính hay đối tượng?

- Luôn là danh từ và có tên
- Tuy nhiên
 - Thuộc tính là đặc trưng của đối tượng
 - Thuộc tính mô tả đối tượng
- Nếu gặp 1 danh từ mà chưa phân biệt được thuộc tính hay đối tượng
 - Xác định các tính chất mô tả cho danh từ đó!!!

40



Phương thức

- Là chức năng hoặc hành động mà đối tượng có thể thực hiện
- Xử lý các thành phần dữ liệu che dấu bên trong lớp

Programmer	
lastName : String = Bardot	
firstName : String = Brigitte	
address : String = Cote d'Azur, France	
getProject() writeCode() deliverProject() setFirstName() setLastName() opname()	

41



Cách xác định đối tượng

- Tìm hiểu **yêu cầu**
 - Tìm hiểu, quán triệt các yêu cầu đặt ra của bài toán
- Phân tích bài toán để xác định được danh sách đối tượng
 - Phân tích **nội dung mô tả bài toán** gồm những **danh từ** chỉ người, chỉ vật, chỉ sự kiện
 - Phân tích các hệ thống thực và **các tài liệu hồ sơ mô tả hệ thống** gần với hệ thống cần xây dựng
 - Dựa vào **kinh nghiệm và kiến thức** của người lập trình để xác định các đối tượng
- Chú ý:
 - Ở bước này không nên quan tâm đến nhiệm vụ bài toán mà chỉ quan tâm đến yếu tố tạo thành đối tượng

42



Phân tích chi tiết

- Phân tích bài toán
 - Phân tích trong nội dung bài toán, cố gắng **xác định các danh từ**.
 - Từ các danh từ này có thể xác định các lớp trong bài toán.
 - tất nhiên chỉ cần các danh từ cần thiết
 - Từ các động từ, xác định các tác vụ cần xử lý trên lớp này.

43



Ví dụ 1

- Ví dụ: Cho 1 đường tròn bán kính R . Tính chu vi và diện tích của hình tròn.
 - Bài toán này có 2 danh từ là “**Đường tròn**” và “**Bán kính**”. Đường tròn chính là class, còn bán kính chỉ là 1 thuộc tính của class đường tròn bởi vì bán kính chỉ là 1 đại lượng về giá trị chiều dài, không phải là class.
 - Hai động từ hay hai tác vụ cần xử lý là “**Tính diện tích**” và “**Tính chu vi của hình tròn**” chính là 2 thành phần phương thức của class đường tròn

44



Thiết kế lớp đối tượng

- Xác định sơ bộ ban đầu **các lớp và mối quan hệ của các lớp** trong sự thừa kế.
 - Ví dụ: mô phỏng sự hoạt động của đồng hồ có 3 kim: Đồng hồ bao gồm có mặt đồng hồ (class FACE) và các kim đồng hồ (class KIM) và class CLOCK thừa kế từ 2 class trên
- Xác định thành phần dữ liệu và thành phần phương thức của lớp.
- Mỗi 1 thành phần dữ liệu hay hàm, phải xác định kiểu dữ liệu của thành phần dữ liệu và tham số + kiểu dữ liệu trả về của các thành phần hàm
- Mỗi thành phần có quy định: private, public, protected....

45



Thiết kế lớp đối tượng (tt1)

- Không quan tâm đến chi tiết của từng hàm, chỉ xác định đầu vào và đầu ra của từng phương thức
- Đối với các thành phần dữ liệu của lớp, chỉ chọn các thành phần (thuộc tính) cần thiết cho class
 - 1 đối tượng có thể có nhiều thuộc tính tùy nhiên tùy theo từng chương trình mà số thuộc tính phải chọn cần thiết cho lớp đó
 - Lớp PTB2 có các thành phần dữ liệu là: a, b, c, x1, x2, delta, x.
→ dễ dàng thấy rằng delta và x bị dư thừa.
- **Lưu ý:**
 - Lớp nên được tìm trong phạm vi bài toán cần giải quyết
 - Tên lớp phải gợi nhớ

46



Thiết kế lớp đối tượng (tt2)

- Thành phần dữ liệu phải được sử dụng bởi 1 trong các phương thức của lớp.
 - Ví dụ đối tượng SINHVIEN có nhiều thuộc tính như hoten, namsinh, gioitinh,...
- Thiết kế sự thừa kế của các lớp: xem các lớp nào thừa kế từ lớp nào tạo thành mô hình phân cấp thừa kế.
- Xem xét áp dụng tính đa hình (**polymorphism**) nếu có của ứng dụng.

47



Thiết kế lớp đối tượng (tt3)

- Xác định cách thực hiện hoặc thuật giải của từng phương thức bằng mã giả (peseudo code)
- Thiết kế constructor và destructor nếu cần thiết

48