

# *PHÂN TÍCH THIẾT KẾ HỆ THỐNG HƯỚNG ĐỐI TƯỢNG*

## *Chương 2: Giới thiệu UML*

1

Unified Modeling Language



## *Nội dung*

- Mô hình hoá hệ thống
- Giới thiệu về UML
- Khả năng sử dụng của UML
- UML và các giai đoạn phát triển hệ thống
- Các thành phần cơ bản của UML

2



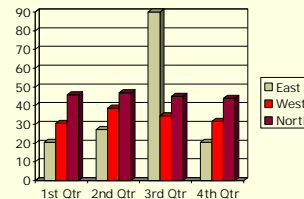
## Mô hình hoá hệ thống

3



## Mô hình (model)

- Mô hình là một biểu diễn của sự vật hay một tập các sự vật trong một lĩnh vực áp dụng nào đó theo một cách khác
- Mô hình nhằm nắm bắt các khía cạnh quan trọng của sự vật, bỏ qua các khía cạnh không quan trọng và biểu diễn theo một tập ký hiệu và quy tắc nào đó



4



## Ví dụ mô hình

- Các mô hình khác
  - Sơ đồ nhà
  - Bản vẽ thiết kế điện
  - Bản đồ địa lý
  - Mô hình giải thuật toán học,...

5



## Mô hình hóa hệ thống

- Các hệ thống thực tiễn ngày càng phức tạp
- Nhiều dự án quá tập trung vào việc viết code... thay vì xây dựng các mô hình trừu tượng cho hệ thống cần tạo dựng
  - Mô hình hoá 1 hệ thống tức là thực hiện trừu tượng hoá hệ thống đó theo 1 dạng nhất định
  - Trong mô hình chỉ xây dựng các chức năng cần thiết và không đi sâu vào chi tiết giảm độ phức tạp và nâng cao tính tổng quát, bỏ qua những vấn đề không cần thiết

6



## Mô hình hóa hệ thống (tt)

### ■ Lợi ích của việc sử dụng mô hình:

- Hiểu được hệ thống dưới góc độ tổng thể, và hiểu được cấu trúc, hành vi của hệ thống
- Dễ giao tiếp, soạn tài liệu, dễ bảo trì
- Là ngôn ngữ giao tiếp của các thành viên liên quan đến hệ thống.
- Là mẫu hướng dẫn trong việc xây dựng hệ thống
- Làm tài liệu cho các quyết định khi phân tích thiết kế hệ thống

- **Ví dụ:** việc xây 1 căn nhà nhỏ có thể không cần bản vẽ nhưng việc xây 1 cao ốc thì khác

7



## Nguyên tắc mô hình hóa

- Việc chọn mô hình nào để tạo lập có ảnh hưởng đến cách giải quyết vấn đề và cách hình thành các giải pháp
- Mỗi mô hình biểu diễn hệ thống với mức độ chính xác khác nhau
- Mô hình phù hợp với thế giới thực là mô hình tốt nhất
- Không có mô hình nào đầy đủ cả. Mỗi hệ thống thường được tiếp cận thông qua tập mô hình gần như độc lập nhau.

8



## Mục đích việc mô hình hóa hệ thống (tt)

- Mục tiêu của giai đoạn phân tích
  - Mô hình được trình bày theo hướng nhìn của người sử dụng
  - Là một mô hình tổng thể miêu tả được tất cả các khía cạnh của sản phẩm
  - Giúp xác định yêu cầu của người dùng đối với hệ thống
  - Giúp đánh giá tính khả thi của dự án

9



## Mục đích việc mô hình hóa hệ thống

- Nhà thiết kế cần phải tạo ra các mô hình mô tả tất cả các khía cạnh khác nhau của sản phẩm
- Ngoài ra, một mô hình có thể được chia thành nhiều hướng nhìn (view), mỗi hướng nhìn trong số chúng sẽ mô tả một khía cạnh riêng biệt của sản phẩm hay hệ thống cần được xây dựng
- Một mô hình cũng có thể được xây dựng trong nhiều giai đoạn và ở mỗi giai đoạn, mô hình sẽ được bổ sung thêm một số chi tiết nhất định

10



## Mục đích việc mô hình hóa hệ thống

- Mô hình thường được mô tả trong ngôn ngữ trực quan:
  - Đa phần các thông tin được thể hiện bằng các ký hiệu đồ họa và các kết nối giữa chúng
  - Chỉ khi cần thiết một số thông tin mới được biểu diễn ở dạng văn bản
- Việc biểu diễn mô hình phải thỏa mãn:
  - Chính xác, đồng nhất, dễ hiểu, dễ thay đổi, dễ liên lạc với các mô hình khác

11



## Mục đích việc mô hình hóa hệ thống

- Tóm lại, mô hình dùng để:
  - Hình dung một hệ thống theo thực tế hay theo mong muốn của chúng ta
  - Chỉ rõ cấu trúc hoặc ứng xử của hệ thống
  - Tạo một khuôn mẫu hướng dẫn nhà phát triển trong suốt quá trình xây dựng hệ thống
  - Ghi lại các quyết định của nhà phát triển để sử dụng sau này
  - Làm công cụ cho phép mọi thành viên phát triển dự án có thể hiểu và làm việc với nhau

12

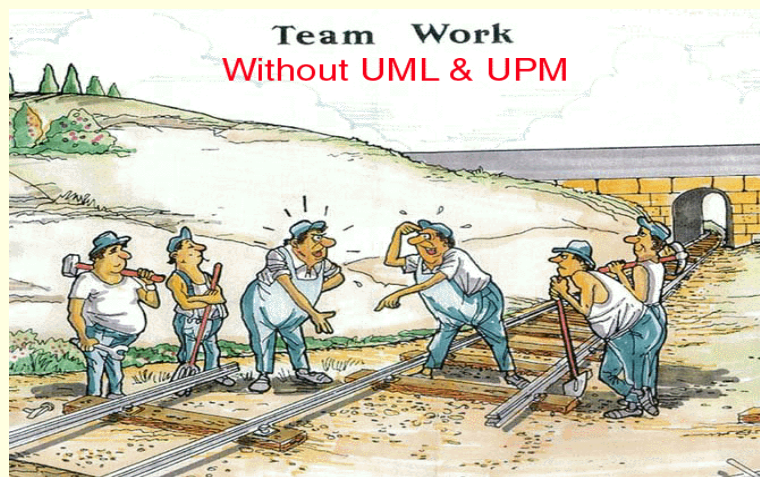


## *Giới thiệu về UML*

13



## *Tại sao phải dùng UML?*



14



## UML là gì?

- UML là ngôn ngữ chuẩn để viết kế hoạch chi tiết phần mềm
- Đặc trưng của UML
  - UML là 1 ngôn ngữ biểu diễn mô hình, hiển thị, đặc tả, làm tài liệu
  - UML là công cụ giao tiếp giữa User, Analyst, Designer, Software Developer
- UML kết hợp các điểm mạnh của các mô hình:
  - Mô hình dữ liệu (ERD)
  - Mô hình dòng dữ liệu (DFD)
  - Mô hình đối tượng (Object Model) & Mô hình thành phần (Component Model)

15



## Sử dụng UML cho các ứng dụng

- Các hệ thống thông tin
  - Lưu trữ, truy vấn, biểu diễn thông tin trong CSDL quan hệ hay CSDL hướng đối tượng
- Các hệ thống kỹ thuật
  - Điều khiển các thiết bị kỹ thuật trong công nghiệp, quân sự
- Các hệ thống nhúng thời gian thực
  - Nhúng trong các thiết bị khác như điện thoại di động, xe ô tô
  - Các hệ thống này thường thiếu thiết bị như màn hình, ổ đĩa
- Các hệ thống phân tán
  - Phân tán trên nhiều máy
  - Cần cơ chế đồng bộ để toàn vẹn dữ liệu
  - Xây dựng trên cơ sở đối tượng như CORBA, COM/DCOM
- Các hệ thống thương mại

16





## Lịch sử phát triển hướng đối tượng & UML

- 1970-1980: Simula → Smaltalk và C++
- Đầu 1990:
  - Grady Booch's Booch Modeling Methodology
  - James Rumbaugh's Object Modeling Technique – OMT
  - Ivar Jacobson's OOSE Methodology
  - Hewlett- Packard's Fusion
  - Coad and Yordon's OOA and OOD
- 1995: Các phương pháp trên được kết hợp thống nhất thành UML cho OOAD (James Rumbaugh, Grady Booch và Ivar Jacobson)

17



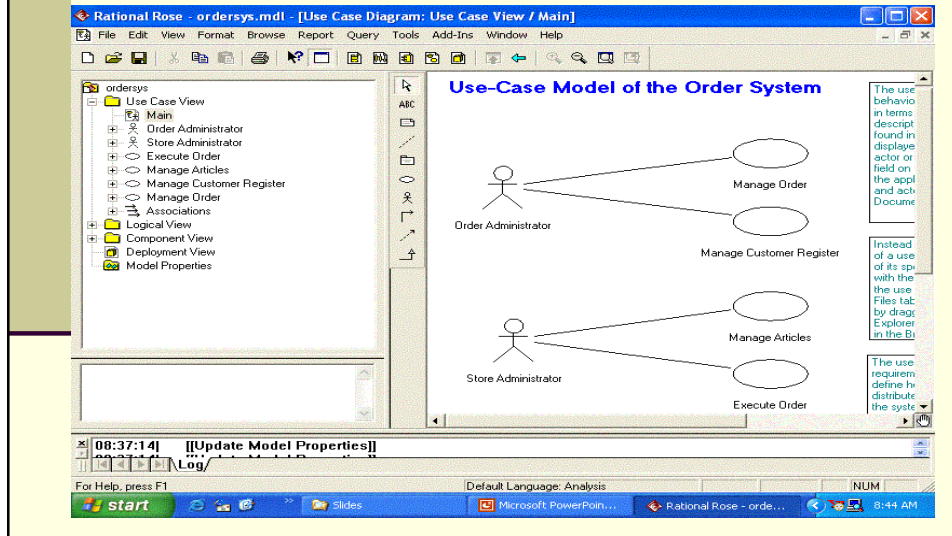
## Lịch sử phát triển hướng đối tượng & UML

- UML 1.1 được OMG (Object Management Group) công nhận vào 1997
- UML 1.3 được công nhận vào 1999
- UML 1.4 được công nhận vào 2000
- Hiện nay chúng ta sử dụng UML 2.0

18



## Phần mềm Rational Rose



## Phương pháp và ngôn ngữ mô hình hóa

### ■ Phương pháp:

- Cấu trúc hóa hành động và suy nghĩ của con người
- Cho biết phải làm gì, làm như thế nào, bao giờ và tại sao (mục đích và hành động)

### ■ Ngôn ngữ mô hình hóa:

- Bao gồm các ký hiệu và 1 tập hợp các quy tắc chỉ cách sử dụng chúng
- Các quy tắc gồm:
  - Cú pháp (Syntax) cho biết hình dạng các phần tử và cách kết nối chúng trong mô hình
  - Ngữ nghĩa (Semantic) ý nghĩa của mỗi 1 phần tử cấu trúc



## UML và các giai đoạn của chu trình phát triển phần mềm

21



### Giai đoạn phân tích

- **Xây dựng Biểu đồ use case:**
  - 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
  - Xây dựng 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

22



## Giai đoạn 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 kịch bản (scenario) đã có và các lớp đã xác định trong pha phân tích.
- *Xây dựng biểu đồ lớp chi tiết:*
  - 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

23



## Giai đoạn thiết kế (tt)

- *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*
  - Xác định các thành phần và các thiết bị cần thiết để triển khai hệ thống, các giao thức và dịch vụ hỗ trợ

24



## *Giai đoạn lập trình*

- Các lớp của giai đoạn thiết kế sẽ được biến thành những dòng code cụ thể trong một ngôn ngữ lập trình hướng đối tượng cụ thể

25



## *Giai đoạn thử nghiệm*

- Một hệ thống phần mềm thường được thử nghiệm qua nhiều giai đoạn và với nhiều nhóm thử nghiệm khác nhau
- Nhiều loại biểu đồ UML được sử dụng để làm công việc thử nghiệm:
  - Thử nghiệm đơn vị sử dụng biểu đồ lớp
  - Thử nghiệm tích hợp sử dụng biểu đồ thành phần (component diagram) và biểu đồ cộng tác (collaboration diagram)
  - Thử nghiệm hệ thống sử dụng biểu đồ Use case (use case diagram)

26

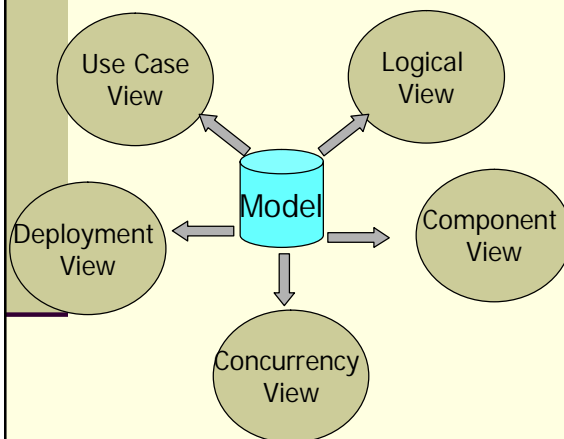


## Các thành phần cơ bản của UML

27



## Hướng nhìn (view)



- Là sự trừu tượng hóa bao gồm các biểu đồ khác nhau
- Chỉ ra các khía cạnh khác nhau của hệ thống cần mô hình hóa
- Kết hợp các hướng nhìn khác nhau tạo ra 1 bức tranh tổng thể

28



## Hướng nhìn (tt)

- Hướng nhìn **Use case** (use case view): đây là hướng nhìn chỉ ra khía cạnh chức năng của một hệ thống, nhìn từ hướng tác nhân bên ngoài.
- Hướng nhìn **logic** (logical view): chỉ ra chức năng sẽ được thiết kế bên trong hệ thống như thế nào, qua các khái niệm về cấu trúc tính cũng như ứng xử động của hệ thống.
- Hướng nhìn **thành phần** (component view): chỉ ra khía cạnh tổ chức của các thành phần code.
- Hướng nhìn **song song** (concurrency view): chỉ ra sự tồn tại song song/ trùng hợp trong hệ thống, hướng đến vấn đề giao tiếp và đồng bộ hóa trong hệ thống.
- Hướng nhìn **triển khai** (deployment view): chỉ ra khía cạnh triển khai hệ thống vào các kiến trúc vật lý (các máy tính hay trang thiết bị được coi là trạm công tác)

29



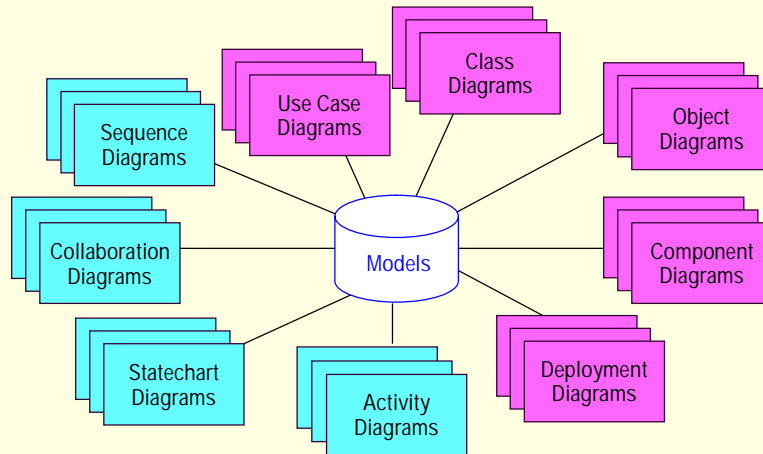
## Biểu đồ (diagram)

- Biểu đồ là các hình vẽ bao gồm các ký hiệu phần tử mô hình hóa được sắp xếp để minh họa một thành phần cụ thể hay một khía cạnh cụ thể của hệ thống.
- Một mô hình hệ thống thường có nhiều loại biểu đồ, mỗi loại có nhiều biểu đồ khác nhau.
- Một biểu đồ là một thành phần của một hướng nhìn cụ thể, mặt khác, một số loại biểu đồ có thể là thành phần của nhiều hướng nhìn khác nhau, tùy thuộc vào nội dung của biểu đồ

30



## Biểu đồ (tt)

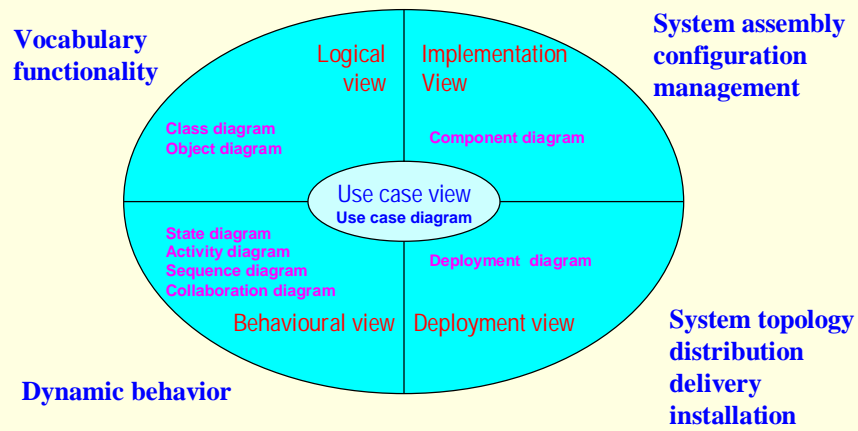


Là các hình vẽ miêu tả nội dung trong một hướng nhìn

31



## View & Diagram



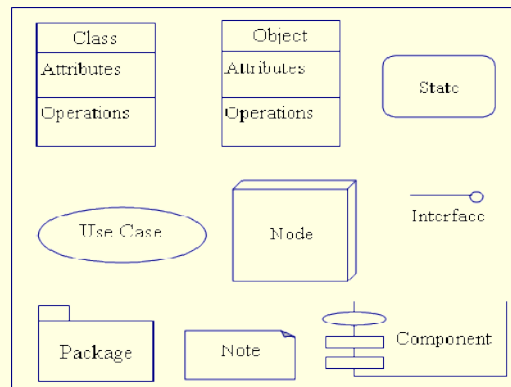
32





## Các thành phần mô hình

- Mô tả lớp, đối tượng, thông điệp, mối quan hệ...

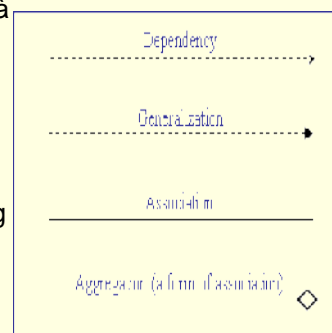


33



## Các thành phần mô hình – quan hệ

- Nối kết (Association): nối các phần tử và các thực thể nối (link).
- Khái quát hóa (Generalization): còn được gọi là tính thừa kế, có ý nghĩa rằng một phần tử này có thể là một sự chuyên biệt hóa của một phần tử khác.
- Sự phụ thuộc (Dependency): chỉ ra rằng một phần tử này phụ thuộc trong một phương thức nào đó vào một phần tử khác.
- Kết tập (Aggregation): Một dạng của nối kết, trong đó một phần tử này chứa các phần tử khác



34



## Các loại biểu đồ UML

35



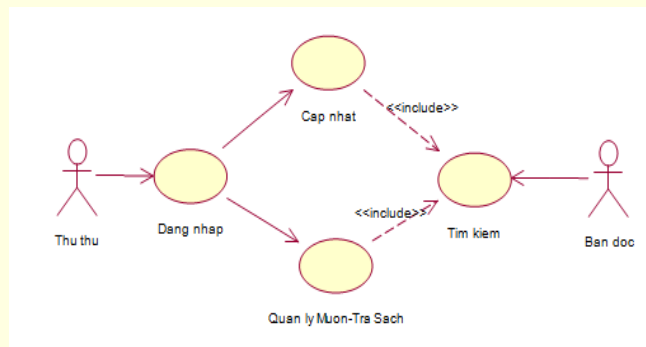
### *Biểu đồ UC (Use Case diagram)*

- Chỉ ra tương tác giữa các UC và các tác nhân
- UC là các chức năng của hệ thống
- Tác nhân hay hệ thống khác cung cấp hay thu nhận thông tin từ hệ thống.
- Biểu đồ UC chỉ ra chức năng tổng thể của hệ thống đang phát triển.

36



## Ví dụ Biểu đồ UC

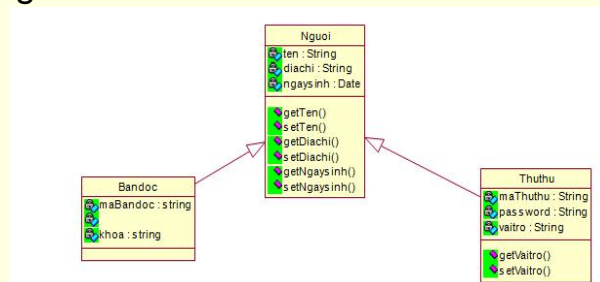


37



## Biểu đồ lớp (class diagram)

- Biểu đồ lớp là một biểu đồ dạng mô hình tĩnh nhằm mô tả hướng 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



38



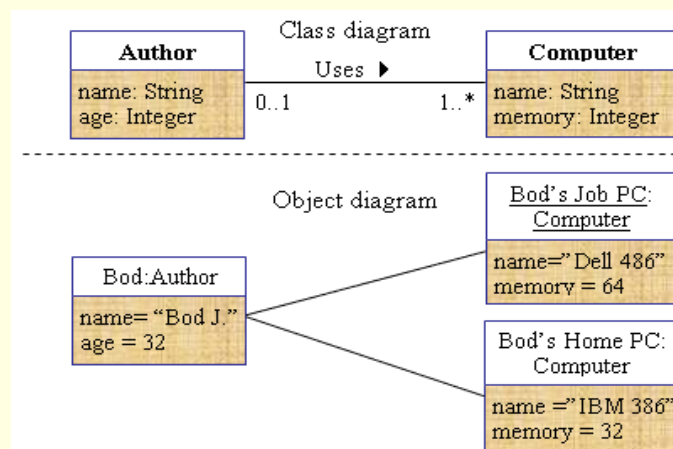
## Biểu đồ đối tượng (Object Diagram)

- Một biểu đồ đối tượng là một phiên bản của biểu đồ lớp và thường cũng sử dụng các ký hiệu như biểu đồ lớp
- Biểu đồ đối tượng sử dụng chung các ký hiệu của biểu đồ lớp, chỉ trừ hai ngoại lệ: đối tượng được viết với tên được gạch dưới và tất cả các thực thể trong một mối quan hệ đều được chỉ ra

39



## Biểu đồ đối tượng (Object Diagram)

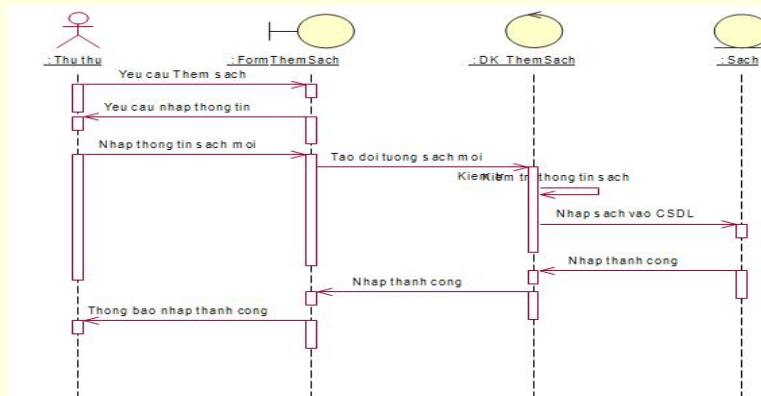


40



## Biểu đồ trình tự (Sequence Diagram)

- Là các luồng chức năng xuyên qua các use case
- Biểu diễn mối quan hệ giữa các đối tượng và giữa các đối tượng và tác nhân theo thứ tự thời gian

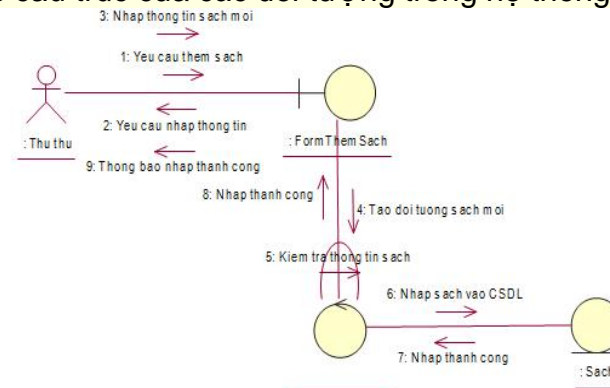


41



## Biểu đồ cộng tác (Collaboration Diagram)

- Gần giống như biểu đồ trình tự
- Không quan tâm đến thời gian mà chỉ quan tâm đến tổ chức cấu trúc của các đối tượng trong hệ thống

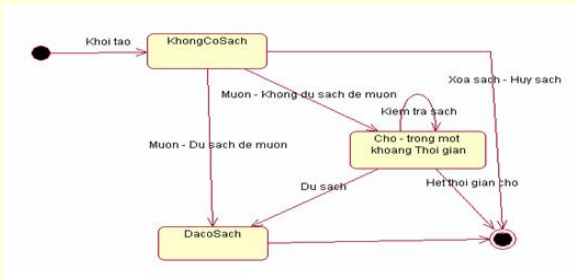


42



## Biểu đồ trạng thái (State Diagram)

- Một biểu đồ trạng thái thường là một sự bổ sung cho lời miêu tả một lớp, nó chỉ ra tất cả các trạng thái mà đối tượng của lớp này có thể có, và những sự kiện nào sẽ gây ra sự thay đổi trạng thái
- Mô tả vòng đời của đối tượng từ khi được sinh ra đến khi phá hủy



43



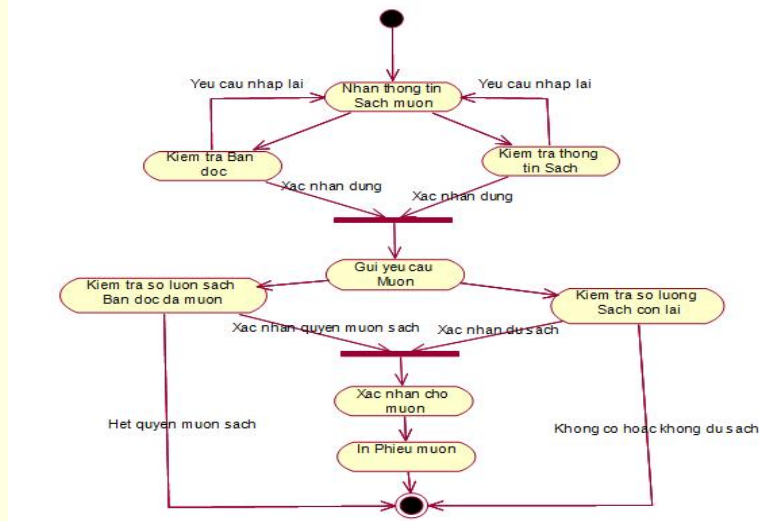
## Biểu đồ hoạt động (Activity Diagram)

- Một biểu đồ hoạt động chỉ ra một trình tự lần lượt của các hoạt động
- Thường được sử dụng để miêu tả các hoạt động được thực hiện trong **một thủ tục**
- Bao gồm các trạng thái hoạt động, chứa đặc tả của một hoạt động cần được thực hiện

44



## Ví dụ Biểu đồ hoạt động



45



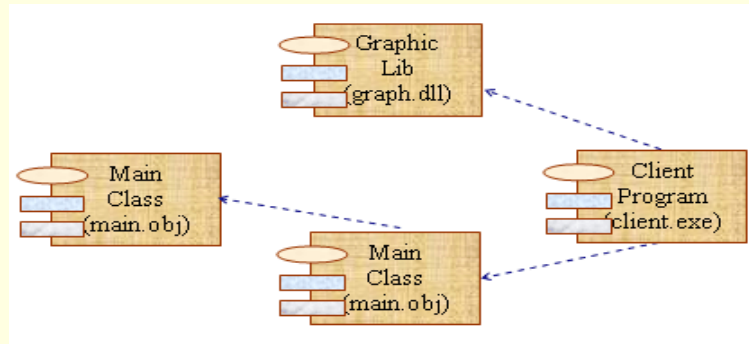
## Biểu đồ thành phần (Component Diagram)

- Một biểu đồ thành phần chỉ ra cấu trúc vật lý của các dòng lệnh (code) theo khái niệm thành phần code
- Một thành phần code có thể là một tập tin source code, một thành phần nhị phân (binary) hay một thành phần thực thi được (executable)
- Biểu đồ thành phần cũng chỉ ra những sự phụ thuộc giữa các thành phần với nhau, trợ giúp cho công việc phân tích hiệu ứng mà một thành phần được thay đổi sẽ gây ra đối với các thành phần khác

46



## Ví dụ Biểu đồ thành phần

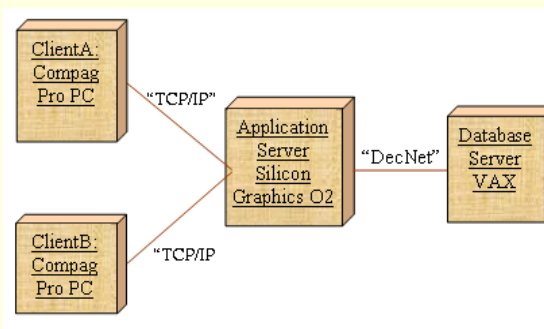


47



## Biểu đồ triển khai (Deployment Diagram)

- Biểu đồ triển khai mô tả hệ thống sẽ được triển khai như thế nào, thành phần nào được cài đặt ở đâu, các liên kết vật lý hoặc giao thức truyền thông nào được sử dụng



48