

TRƯỜNG CAO ĐẲNG NGHỀ CÔNG NGHIỆP HÀ NỘI

Tác giả: Vũ Thị Kim Phụng

Bùi Quang Ngọc

GIÁO TRÌNH

Cơ sở dữ liệu hướng đối tượng

(Lưu hành nội bộ)

Hà Nội năm 2012

Tuyên bố bản quyền

Giáo trình này sử dụng làm tài liệu giảng dạy nội bộ trong trường cao đẳng nghề Công nghiệp Hà Nội

Trường Cao đẳng nghề Công nghiệp Hà Nội không sử dụng và không cho phép bất kỳ cá nhân hay tổ chức nào sử dụng giáo trình này với mục đích kinh doanh.

Mọi trích dẫn, sử dụng giáo trình này với mục đích khác hay ở nơi khác đều phải được sự đồng ý bằng văn bản của trường Cao đẳng nghề Công nghiệp Hà Nội

Mục lục

BÀI 1: GIỚI THIỆU LỊCH SỬ PHÁT TRIỂN	1
1. Lịch sử phát triển của Cơ sở dữ liệu hướng đối tượng.....	1
2. Nền tảng của dữ liệu hướng đối tượng.	3
BÀI 2: MÔ HÌNH DỮ LIỆU HƯỚNG ĐỐI TƯỢNG ODMG – CÁC THÀNH PHẦN ĐẶC TRƯNG	5
1. Khái niệm về mô hình dữ liệu.....	5
2. Mô hình dữ liệu hướng đối tượng – ODMG.....	8
2.1. Mô hình hóa các đối tượng	9
2.2. Mô hình hóa tính động	10
2.3. Các liên kết ngữ nghĩa giữa các lớp	11
2.4. Tổ chức các nhóm đối tượng	12
2.5. Lược đồ.....	13
3. Phương pháp xây dựng mô hình dữ liệu hướng đối tượng.....	14
3.1. Phương pháp chuyển đổi	14
3.2. Phương pháp phân tích và xây dựng trực tiếp	14
4. Các thành phần đặc trưng của kiểu dữ liệu hướng đối tượng	18
4.1. Kiểu dữ liệu hướng đối tượng.....	18
4.2. Tính chất của các đối tượng.....	20
4.3. Quản lý tính bền vững của các đối tượng.....	21
BÀI 3: NGÔN NGỮ TRUY VẤN DỮ LIỆU HƯỚNG ĐỐI TƯỢNG	22
1. Giới thiệu một số ngôn ngữ truy vấn dữ liệu đối tượng	22
2. Ngôn ngữ truy vấn dữ liệu đối tượng OQL tương thích với đa hệ quản trị.....	22
3. Cú pháp OQL	22
3.1. Quy ước	22
3.2. Ngữ pháp OQL.....	22
4. Bài tập ví dụ minh họa.....	23
BÀI 4: HỆ QUẢN TRỊ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG	26
1. Giới thiệu một số hệ quản trị dữ liệu đối tượng.....	26
1.1. Một số hệ quản trị cơ sở dữ liệu hướng đối tượng.....	26
1.2. Các tính năng bắt buộc của hệ quản trị cơ sở dữ liệu hướng đối tượng	26
1.3. Các thành phần của hệ quản trị cơ sở dữ liệu hướng đối tượng	29
1.4. Chuẩn của hệ quản trị cơ sở dữ liệu hướng đối tượng	29
2. Cài đặt, cấu hình tích hợp với môi trường phát triển ứng dụng.....	29
3. Khai thác công cụ (Versant, DB4o)	29

3.1.	Giới thiệu hệ cơ sở dữ liệu hướng đối tượng DB4O	29
3.2.	Các loại lớp trong hệ thống đối tượng của DB4o	33
3.3.	Object Identity.....	34
3.4.	Lược đồ cơ sở dữ liệu trong db4o (database schema).....	36
3.5.	Object Relationships.....	37
3.6.	Các kiểu quan hệ của đối tượng	40
4.	Truy vấn dữ liệu đối tượng trực tiếp bằng tool hoặc dos	45
BÀI 5: TÍCH HỢP DỮ LIỆU HƯỚNG ĐỐI TƯỢNG TRÊN MÔI TRƯỜNG PHÁT TRIỂN		
ỨNG DỤNG .NET		46
1.	Cài đặt tích hợp môi trường	46
2.	Cài đặt mô hình dữ liệu hướng đối tượng bằng ngôn ngữ .Net (Visual, C#)	48
3.	Biên dịch lược đồ (diagram)	50
4.	Xây dựng ứng dụng cho phép truy xuất dữ liệu đối tượng.....	50

BÀI 1: GIỚI THIỆU LỊCH SỬ PHÁT TRIỂN

1. Lịch sử phát triển của Cơ sở dữ liệu hướng đối tượng

Các loại cấu trúc cơ sở dữ liệu và mối liên hệ giữa chúng đóng vai trò rất lớn trong việc xác định tính hiệu quả của hệ quản trị cơ sở dữ liệu. Vì vậy, thiết kế cơ sở dữ liệu trở thành hoạt động chính trong môi trường cơ sở dữ liệu.

Việc thiết kế cơ sở dữ liệu được thực hiện đơn giản hơn nhiều khi ta sử dụng các mô hình. Các mô hình là sự trừu tượng đơn giản của các sự kiện trong thế giới thực. Các trừu tượng như vậy cho phép ta khảo sát các đặc điểm của các thực thể và các mối liên hệ được tạo ra giữa các thực thể đó. Việc thiết kế các mô hình tốt sẽ đưa ra các cơ sở dữ liệu tốt và trên cơ sở đó sẽ có các ứng dụng tốt. Ngược lại, mô hình không tốt sẽ đưa đến thiết kế cơ sở dữ liệu tồi và dẫn đến các ứng dụng không đúng.

Một mô hình cơ sở dữ liệu là một tập hợp các khái niệm dùng để biểu diễn các cấu trúc của cơ sở dữ liệu. Cấu trúc của một cơ sở dữ liệu là các kiểu dữ liệu, các mối liên kết và các ràng buộc phải tuân theo trên các dữ liệu. Nhiều mô hình còn có thêm một tập hợp các phép toán cơ bản để đặc tả các thao tác trên cơ sở dữ liệu.

Các loại mô hình cơ sở dữ liệu:

- Các mô hình dữ liệu bậc cao hoặc mô hình dữ liệu mức quan niệm cung cấp các khái niệm gắn liền với cách cảm nhận dữ liệu của nhiều người sử dụng.
- Các mô hình dữ liệu bậc thấp hoặc các mô hình dữ liệu vật lý cung cấp các khái niệm mô tả chi tiết về việc dữ liệu được lưu trữ trong máy tính như thế nào.
- Các mô hình dữ liệu thể hiện (mô hình dữ liệu mức logic), chúng cung cấp những khái niệm mà người sử dụng có thể hiểu được và không xa với cách tổ chức dữ liệu bên trong máy tính.

Trong một mô hình dữ liệu cần phải phân biệt rõ giữa mô tả của cơ sở dữ liệu và bản thân cơ sở dữ liệu.

Sau đây, chúng ta sẽ đi qua lịch sử phát triển của các mô hình cơ sở dữ liệu:

- Vào những năm sáu mươi, thế hệ đầu tiên của cơ sở dữ liệu ra đời dưới dạng mô hình thực thể kết hợp (*Entity Relationship Model*), mô hình mạng (*Network Model*) và mô hình phân cấp (*Hierarchical Model*).
- Vào những năm bảy mươi, thế hệ thứ hai của cơ sở dữ liệu ra đời. Đó là mô hình dữ liệu quan hệ (*Relational Data Model*) do EF. Codd phát minh. Mô hình này có cấu trúc logic chặt chẽ. Đây là mô hình đã và đang được sử dụng rộng khắp trong công tác quản lý trên phạm vi toàn cầu. Việc nghiên cứu mô hình dữ liệu quan hệ nhằm vào lý thuyết chuẩn hoá các quan hệ và là một công cụ quan trọng trong việc phân tích thiết kế các hệ cơ sở dữ liệu hiện nay. Mục đích của nghiên cứu này nhằm bỏ đi các phần tử không bình thường của quan hệ khi thực hiện các phép cập nhật, loại bỏ các phần tử dư thừa.
- Sang thập kỷ tám mươi, mô hình cơ sở dữ liệu thứ ba ra đời, đó là mô hình cơ sở dữ liệu hướng đối tượng (*Object Oriented Data Model*), mô hình cơ sở dữ liệu phân tán, mô hình cơ sở dữ liệu suy diễn,...

Thực tế chưa có mô hình dữ liệu nào là tốt nhất. Tốt nhất phụ thuộc vào yêu cầu truy xuất và khai thác thông tin của đơn vị quản lý nó. Nó được sử dụng ở đâu và vào lúc nào là tốt nhất. Tuy nhiên, người ta thường dựa vào các tiêu chí sau để nói rằng mô hình dữ liệu tốt nhất:

- Mục đích: Phần lớn các mô hình dữ liệu sử dụng hệ thống ký hiệu để biểu diễn dữ liệu và làm nền tảng cho các hệ ứng dụng và ngôn ngữ thao tác dữ liệu. Các mô hình thực thể quan hệ không có hệ thống ký hiệu để xây dựng các phép toán thao tác dữ liệu, mà sử dụng để thiết kế lược đồ khái niệm, cài đặt trong một mô hình dữ liệu với một hệ quản trị cơ sở dữ liệu nào đó.

- Hướng giá trị hay hướng đối tượng: Các mô hình dữ liệu quan hệ và mô hình logic là các mô hình dữ liệu hướng giá trị. Trong các mô hình dữ liệu hướng giá trị có tính khai báo (declarativeness) và có tác động đến các ngôn ngữ được nó hỗ trợ. Các mô hình mạng, phân cấp, mô hình dữ liệu hướng đối tượng cung cấp đặc tính nhận dạng đối tượng, nên có thể xem chúng là các mô hình hướng đối tượng. Mô hình thực thể quan hệ cũng được có đặc tính nhận dạng hướng đối tượng.
- Tính dư thừa: Tất cả các mô hình dữ liệu đều có khả năng hỗ trợ lưu trữ dữ liệu vật lý và hạn chế sự dư thừa dữ liệu. Tuy nhiên các mô hình dữ liệu hướng đối tượng giải quyết sự dư thừa tốt hơn, bằng cách tạo ra sử dụng con trỏ trỏ đến nhiều vị trí khác nhau.
- Giải quyết mối quan hệ nhiều – nhiều: Phần lớn trong các mô hình cơ sở dữ liệu có chứa các mối quan hệ nhiều – nhiều, một – nhiều hay quan hệ một – một. Một quan hệ có nhiều phần tử của các quan hệ khác và ngược lại. Tuy nhiên trong mô hình dữ liệu mạng không chấp nhận mối quan hệ nhiều – nhiều.

2. **Nền tảng của dữ liệu hướng đối tượng.**

Cơ sở dữ liệu hướng đối tượng và hệ quản trị hướng đối tượng (Object Oriented DataBase Management Systems – OO DBMS) mô tả các kiểu dữ liệu được xây dựng bằng phương pháp tạo bản ghi và tạo tập hợp. Các quan hệ được xây dựng từ các bộ bảng thao tác tạo một tập hợp các bản ghi có khuôn dạng thống nhất.

Che dấu dữ liệu (Encapsulation): Nghĩa là khi có yêu cầu truy xuất đến các đối tượng thuộc kiểu đặc biệt, phải qua các thủ tục đã được định nghĩa cho các đối tượng đó. Chẳng hạn định nghĩa stack như là một kiểu và định nghĩa các thao tác PUSH, POP áp dụng cho stack.

Đặc tính nhận dạng đối tượng (Object Identity) là khả năng phân biệt các đối tượng. Nghĩa là cấu trúc các kiểu cơ bản như nhau. Các kiểu cơ bản là chuỗi ký tự, số.

Thực tế cho thấy cơ sở dữ liệu hướng đối tượng có các ưu điểm:

- Cho phép xét các liên kết đối tượng dưới dạng các phép lưu trữ với các đối tượng.
- Các đối tượng dùng chung giữa nhiều người sử dụng.
- Khả năng phát triển kho tri thức bằng cách thêm các đối tượng mới và các phép xử lý kèm theo.
- Phát triển hệ quản trị cơ sở dữ liệu dựa trên việc xử lý các đối tượng phức tạp, giao diện chương trình, đối tượng động và trùu tượng.

TaiLieu.vn

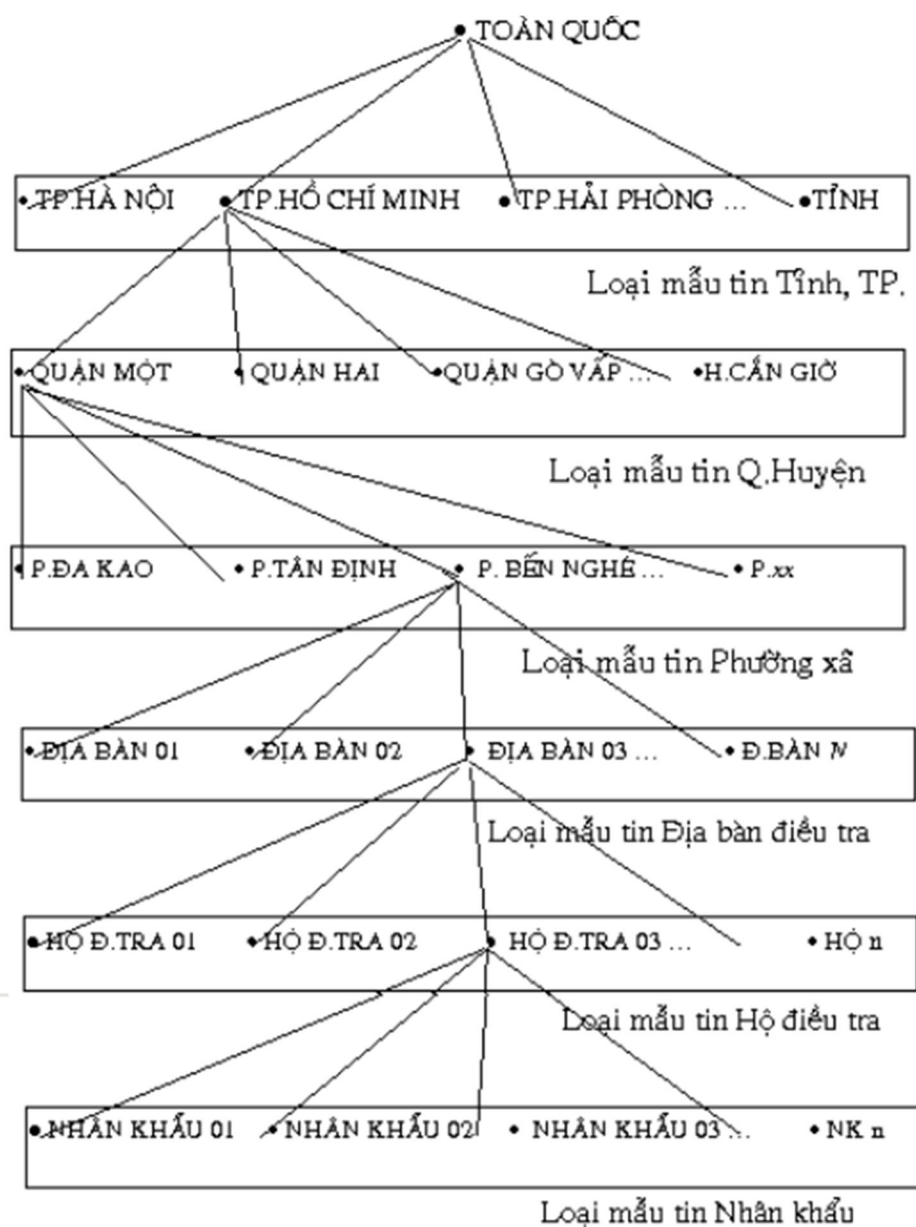
BÀI 2: MÔ HÌNH DỮ LIỆU HƯỚNG ĐỐI TƯỢNG ODMG – CÁC THÀNH PHẦN ĐẶC TRƯNG

1. Khái niệm về mô hình dữ liệu

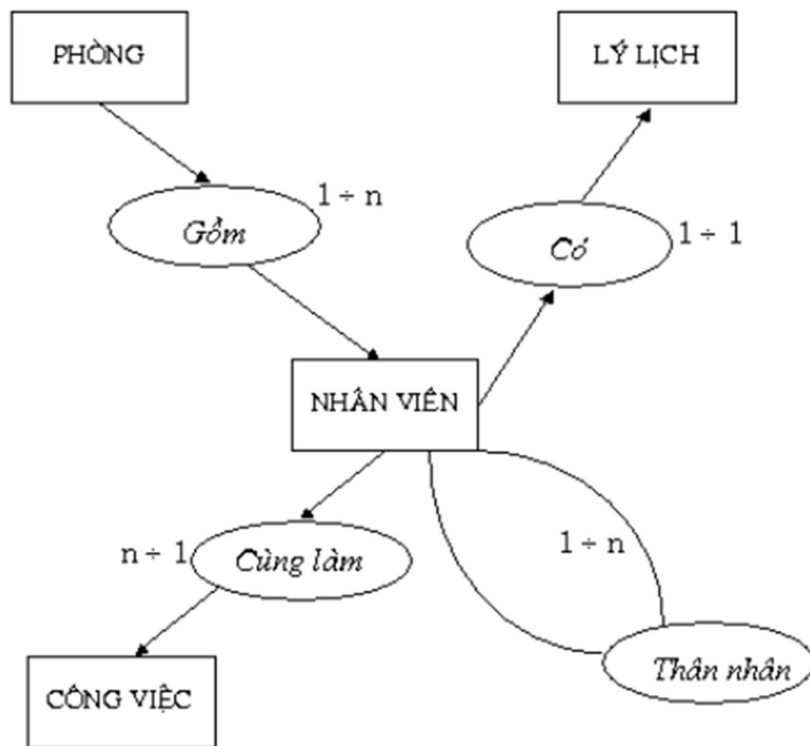
Mô hình dữ liệu là một hệ thống hình thức toán học, bao gồm:

- Hệ thống các ký hiệu biểu diễn dữ liệu.
- Tập hợp các phép toán thao tác trên cơ sở dữ liệu.

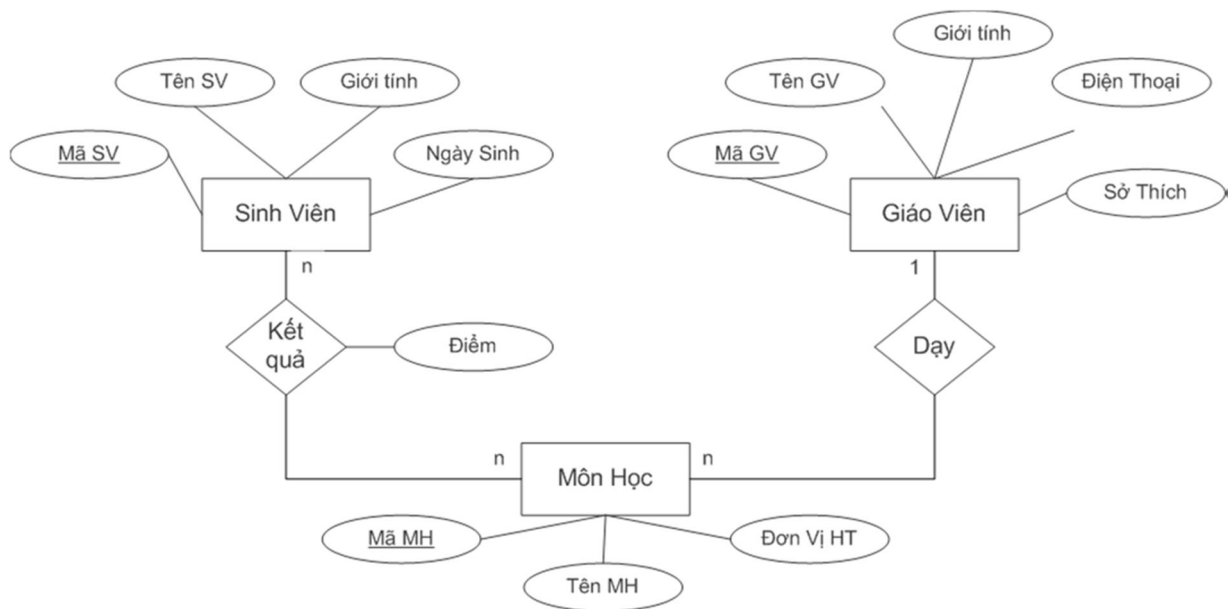
Ví dụ: Các mô hình dữ liệu:



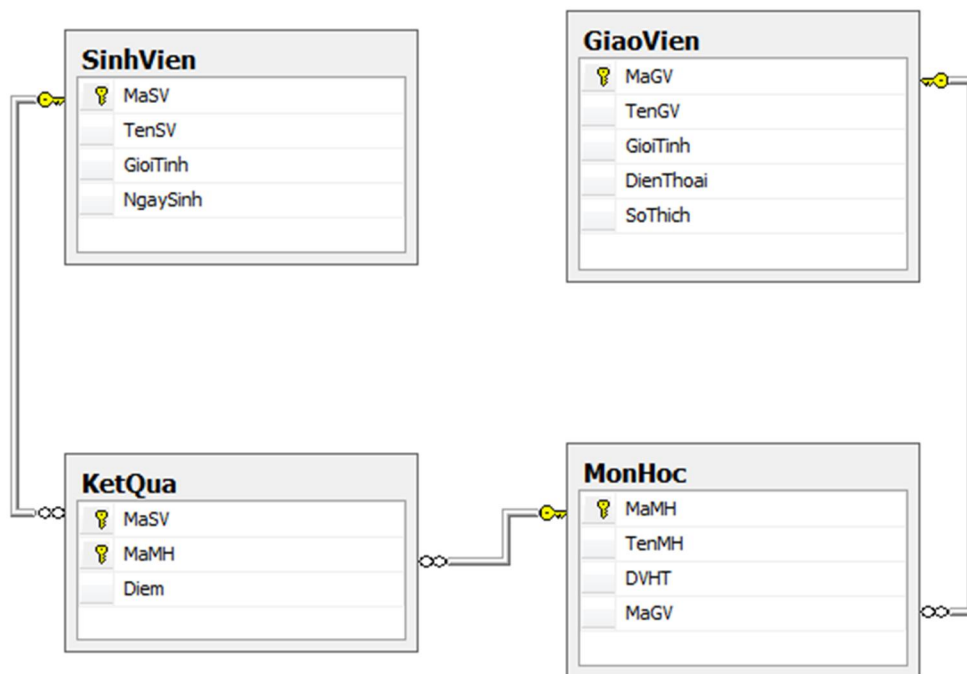
Hình 2.1: Mô hình phân cấp (Hierarchical Model)



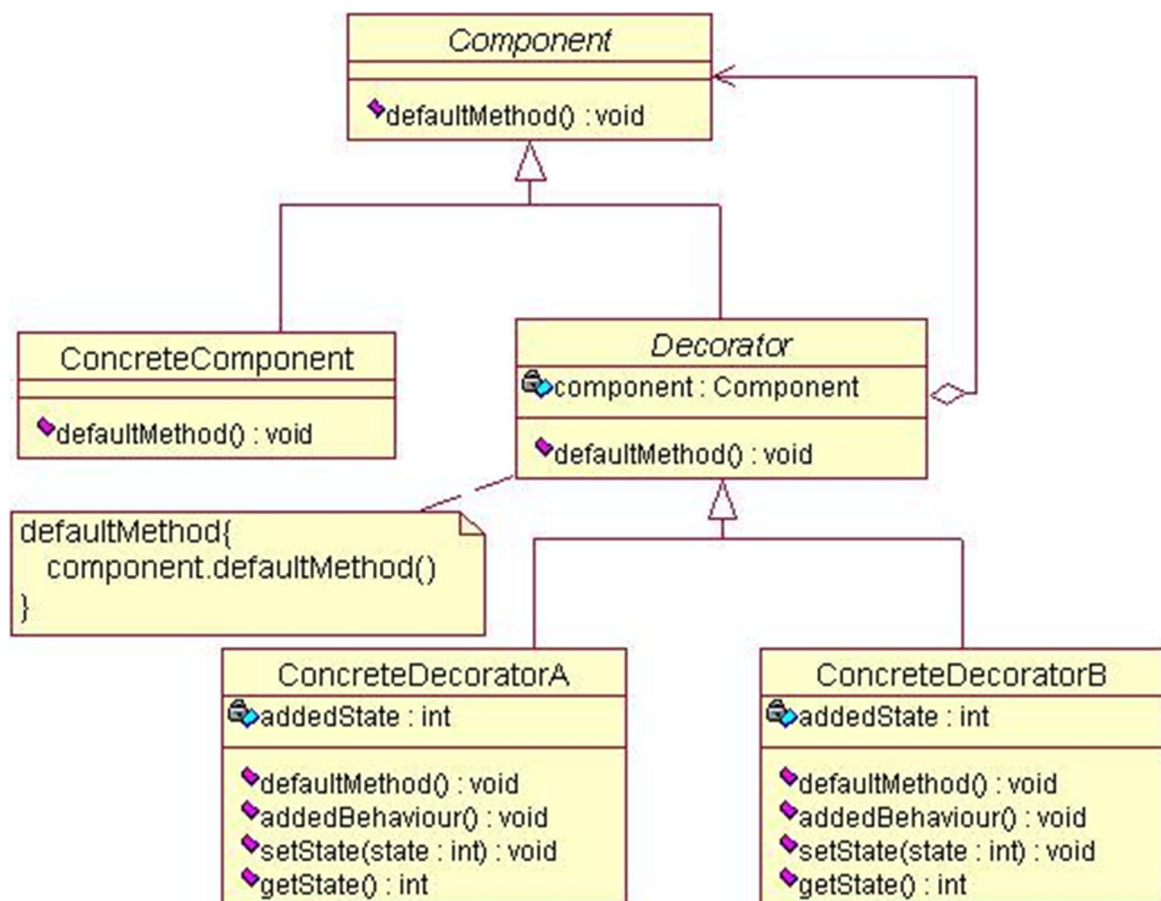
Hình 2.2: Mô hình mạng (Network Model)



Hình 2.3: Mô hình thực thể kết hợp (Entity Relationship Model)



Hình 2.4: Mô hình dữ liệu quan hệ (Relational Data Model)



Hình 2.5: Mô hình dữ liệu hướng đối tượng (Object Oriented Data Model)

2. Mô hình dữ liệu hướng đối tượng – ODMG

Mô hình dữ liệu hướng đối tượng - OODM (*Object Oriented Data Model*) ra đời từ cuối những năm 80 và đầu những năm 90.

Đây là loại mô hình tiên tiến nhất hiện nay dựa trên cách tiếp cận hướng đối tượng đã quen thuộc trong các phương pháp lập trình hướng đối tượng, nó sử dụng các khái niệm như lớp (*class*), sự kế thừa (*inheritance*), kế thừa bội (*multi-inheritance*).

Đặc trưng cơ bản của cách tiếp cận này là tính đóng gói (*encapsulation*), tính đa hình (*polymorphism*) và tính tái sử dụng (*Reusability*).

Lược đồ ODMG (*Object Database Management Group*): ODMG đề xuất một cơ sở dữ liệu tiêu chuẩn với mục tiêu thống nhất mô hình đối tượng hạt nhân của nhiều hệ quản trị cơ sở dữ liệu đối tượng khác nhau.

ODMG đưa ra một chuẩn mới cho OODM:

- Một mô hình đối tượng (OM)
- Một ngôn ngữ định nghĩa đối tượng (ODL)
- Một ngôn ngữ hỏi đối tượng với cú pháp tựa SQL
- Ràng buộc ngôn ngữ C++(Java/Smalltalk)

Những kết cấu chính được đặc tả bởi mô hình dữ liệu của ODMG:

- Đối tượng và literal
- Kiểu
- Các kiểu con và tính kế thừa
- Ngoại diện
- Khoá
- Kiểu sưu tập và kiểu có cấu trúc

Đối tượng bền vững (đối tượng cơ sở dữ liệu): là các đối tượng tiếp tục tồn tại khi thủ tục hay quá trình tạo ra chúng đã kết thúc. Chúng được cấp phát bộ nhớ và được lưu trữ bởi hệ quản trị cơ sở dữ liệu hướng đối tượng.

Đối tượng không bền (chuyển tiếp): chỉ tồn tại bên trong thủ tục hay quá trình tạo ra chúng. Chúng được cấp phát bộ nhớ bởi hệ thống thời gian chạy của ngôn ngữ lập trình.

2.1. Mô hình hóa các đối tượng

- Đối tượng (Object): Bộ sưu tập các yếu tố DL có cấu trúc, được đồng nhất bởi một dẫn trỏ (tham chiếu) duy nhất.
 - Mọi đối tượng đều được đặc trưng bằng một tên duy nhất, gọi là OID (Object Identifier)
 - Hai đối tượng là đồng nhất ($O_1 == O_2$) nếu chúng có cùng OID
 - Hai đối tượng là bằng nhau ($O_1 = O_2$) nếu chúng có cùng giá trị
 - Các đối tượng đặc trưng bởi các tính chất
- Tính chất (Property): đặc trưng của một đối tượng được chỉ định bằng một tên có thể ứng với một thuộc tính, một hàm hay một đối tượng con thành phần

Ví dụ:

- Thuộc tính đơn: tên của một người,...
- Hàm: Hàm tuổi (của một người),...
- Thuộc tính kép: các con của một người,...
- Lớp: nhóm các đối tượng có cùng tính chất, được đặc trưng bởi một cấu trúc và tập các phép toán tác dụng lên các đối tượng của lớp bằng cách che dấu cấu trúc
 - Việc đặc tả tiến triển của các lớp đối tượng làm thành một CSDL hướng đối tượng, cho phép mô hình hoá hành vi chung của các đối tượng một cách đơn thể và mở rộng được.

Ví dụ: các con người, các hình tròn,...

2.2. Mô hình hóa tính động

- Phương pháp: thao tác liên kết với một lớp, xử lý hay đưa trả lại trạng thái của một đối tượng hay một phần của đối tượng thuộc lớp
 - o Một đối tượng được thao tác bởi phương pháp của lớp và được thấy qua các phương pháp: nguyên lý bọc kín
 - o Phương pháp có thể áp dụng được cho nhiều đối tượng thuộc các lớp khác nhau: đa lớp → dùng để mô hình hoá các mối liên kết giữa các lớp

Ví dụ:

class cửa

public: {các thuộc tính thấy được từ bên ngoài lớp}

trạng thái: mở, đóng

chiều cao: real {kiểu thực}

chiều rộng: real

chiều dày: real

private:

trục: vector

góc: real

public operation: {các phương pháp}

mở(lực: real) ... end;

đóng ... end;

- Thông báo: các đối tượng trao đổi (giao lưu thông tin) với nhau bằng thông báo.
 - o Thông báo gồm tên của một phương pháp và các tham số của nó
 - o Khởi tham số cho phép bằng việc gửi đi dẫn gọi một phương pháp công cộng của một đối tượng
 - o Đối tượng phản ứng lại một thông báo bằng cách thực hiện phương pháp liên kết và đưa trả về các tham số kết quả của phương pháp

Ví dụ: Có thể gửi thông báo tới một đối tượng p của lớp cửa:

p: mở(30)

p: đóng

p: chiều rộng.read

p: chiều rộng.write

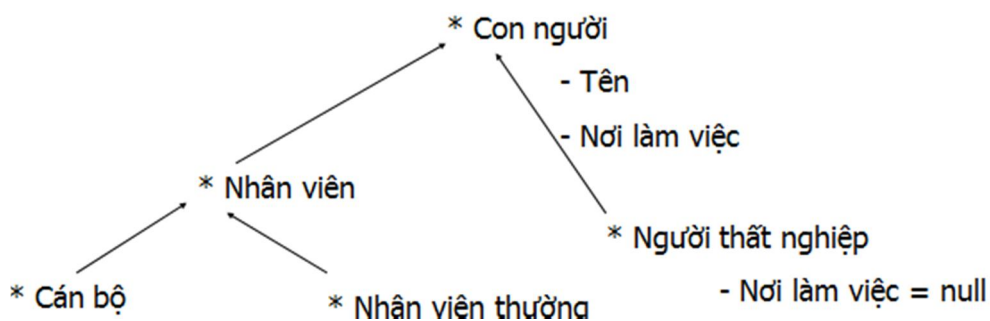
- Demon: Là thao tác trên các đối tượng được khởi phát bởi hệ thống khi có xuất hiện một điều kiện đặc biệt

Ví dụ: Demon có thể được thêm vào lớp cửa nhằm duy trì tự động trạng thái của nó: **if** góc > 10° **then** trạng thái = mở

2.3. Các liên kết ngữ nghĩa giữa các lớp

- Sự tổng quát hoá: liên kết phân cấp giữa hai lớp xác định rằng các đối tượng của lớp trên tổng quát hơn các đối tượng của lớp dưới, các đối tượng của lớp dưới có các tính chất đầy đủ và tinh tế hơn

Ví dụ:

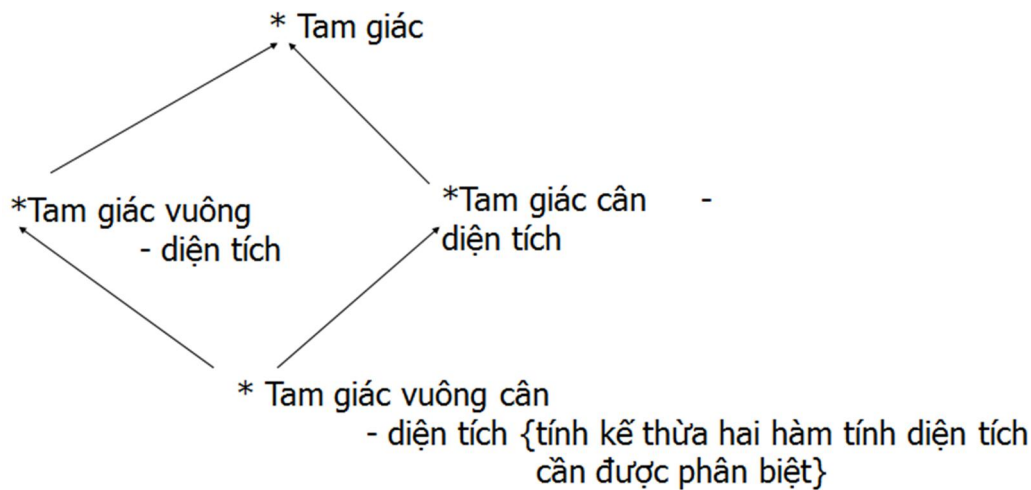


- Tính kế thừa: sự truyền tính chất của một lớp tới lớp con của nó
 - o Mọi phần tử của lớp con kế thừa các tính chất của lớp trên
 - o Một số tính chất của lớp con có thể được làm tinh tế hơn → định nghĩa lại

Ví dụ: thuộc tính “Nơi làm việc” của lớp “Con người” có thể được định nghĩa lại với giá trị null ở mức của lớp “Người thất nghiệp”

- Tính kế thừa bội: cho phép một lớp có nhiều lớp trên trực tiếp

- Lớp con kế thừa các tính chất và phương pháp của các lớp trên
- Có thể xảy ra và cần được giải quyết những xung đột về tên các tính chất hay phương pháp



Các mô hình đối tượng thường phân biệt các tính chất được phân chia bởi nhiều lớp và nhóm hợp chúng trong những lớp đặc biệt gọi là các mối liên kết

Mối liên kết là liên hệ cấu trúc cho phép liên kết các lớp đối tượng với nhau bằng các tính chất phân chia

Ví dụ: “Người” và “Sách” là hai lớp gộp một số tính chất (Tên, ..., Tên sách,...), thì có thể định nghĩa mối liên kết Tác giả của như sau:

Người → Tác giả của → Sách

2.4. Tổ chức các nhóm đối tượng

- Tác tử xây (constructor): lớp cấu trúc, cho phép áp đặt một cấu trúc lên một tập đối tượng và định nghĩa các tính chất cấu trúc đa trị

Các tác tử xây:

- bộ (tuple): cho phép nhóm gộp các thuộc tính (tích Đề các)
- tập (set): cho phép định nghĩa các nhóm không sắp thứ tự, không chứa các phần tử giống nhau
- túi (bag): các tập không sắp thứ tự, có các phần tử giống nhau

- danh sách (list): cho phép định nghĩa các nhóm có thứ tự, được phép có các phần tử giống nhau
- bảng (table): các nhóm có thứ tự và có chỉ số

Một nhóm đối tượng kế thừa các tính chất của tác tử xây nếu có tác tử xây đứng trước.

Ví dụ: có thể quản lý dễ dàng các danh sách các bảng sau:

class câu

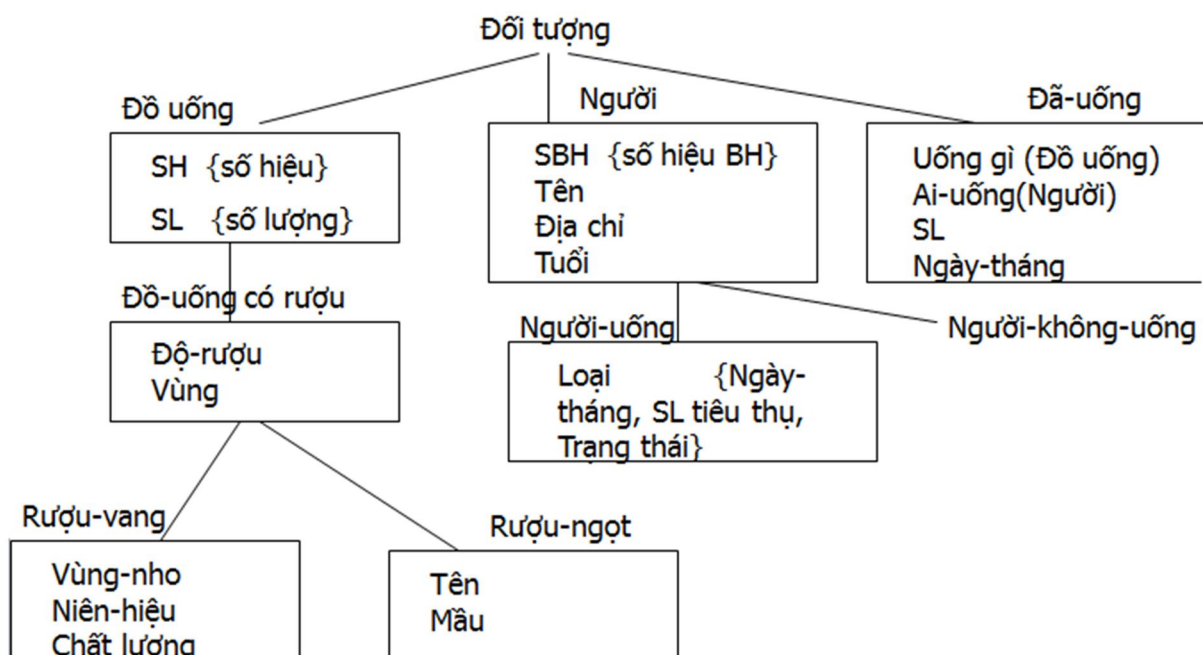
nội dung: list array char;

class văn-bản

đoạn : list câu

2.5. Lược đồ

- Lược đồ cơ sở dữ liệu hướng đối tượng mô tả các thành phần sau:
 - Mô tả các lớp. Mỗi lớp bao gồm các tính chất (tuỳ theo tình hình được tổ chức thành các nhóm bởi các toán tử xây) và các phương pháp.
 - Mô tả các mối liên kết giữa các lớp.

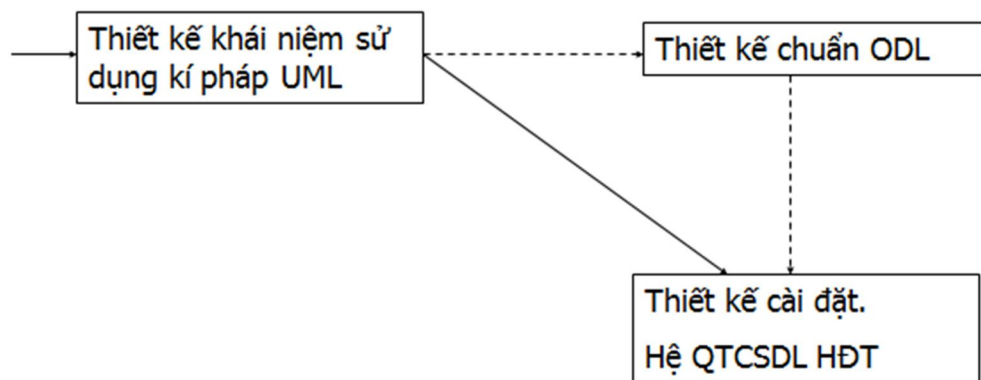


Hình 2.6: Lược đồ đối tượng của cơ sở dữ liệu rượu vang

3. Phương pháp xây dựng mô hình dữ liệu hướng đối tượng

3.1. Phương pháp chuyển đổi

- **Định nghĩa lược đồ khái niệm** trong một ngôn ngữ (mô hình) sao cho gần với người dùng và độc lập với cài đặt cuối cùng. Mô hình được dùng trong bước này phải có khả năng biểu diễn mọi yêu cầu của người dùng (UML – Unified Modeling Language).
- **Dịch chuyển trực tiếp sang cài đặt** cuối cùng trong một hệ QTCSDL hướng đối tượng xác định.
- **Có thể qua một bước trung gian** để có một lược đồ được mô tả trong ODL (Object Definition Language), biểu diễn các chi tiết thiết kế độc lập với sản phẩm cuối cùng

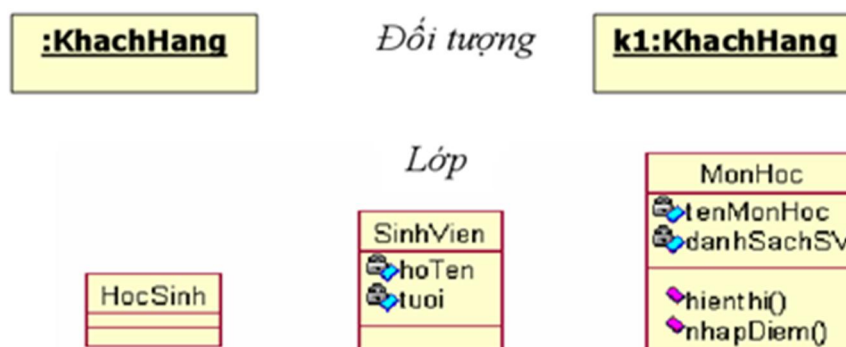


Hình 2.7: Quá trình thiết kế một lược đồ cơ sở dữ liệu hướng đối tượng

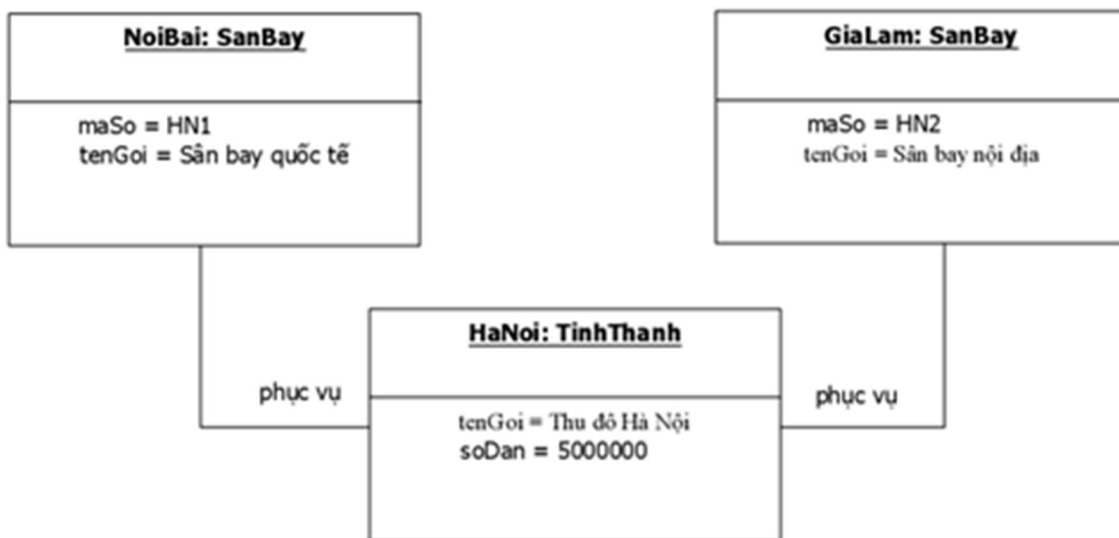
3.2. Phương pháp phân tích và xây dựng trực tiếp

a. Thiết kế khái niệm (UML)

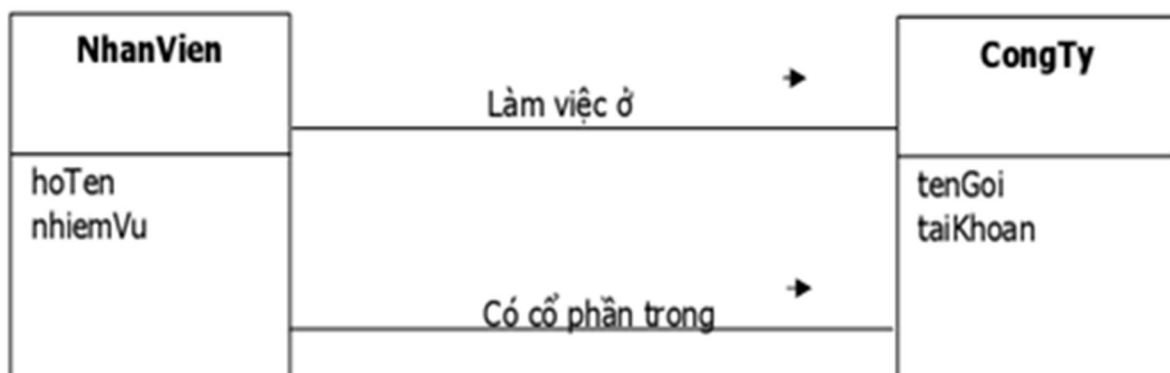
- Ký pháp UML:



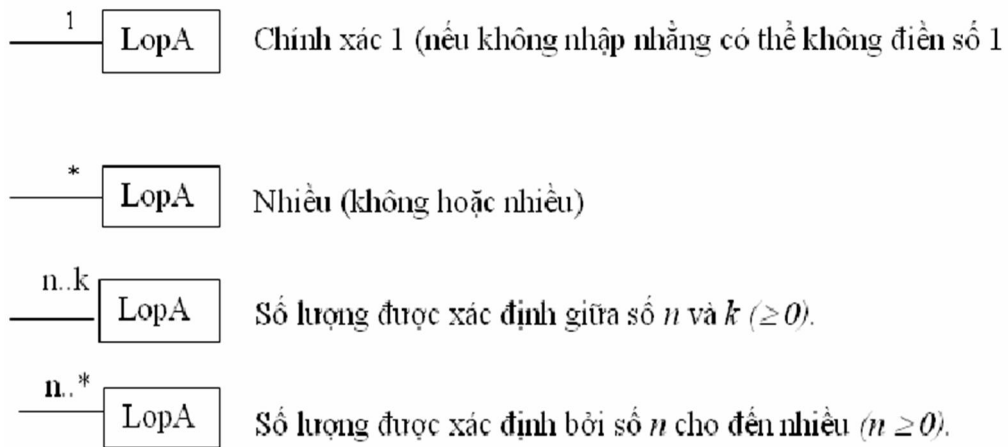
- ‘+’ đứng trước tên thuộc tính, hàm xác định tính công khai (public). Trong Rose kí hiệu là ổ khoá không bị khoá.
- ‘#’ đứng trước tên thuộc tính, hàm xác định tính được bảo vệ (protected). Trong Rose kí hiệu là ổ khoá bị khoá nhưng có chìa để bên cạnh.
- ‘-’ đứng trước tên thuộc tính, hàm xác định tính sở hữu riêng (private). Trong Rose kí hiệu là ổ khoá bị khoá nhưng không có chìa để bên cạnh.



Hình 2.8: Liên kết giữa các đối tượng



Hình 2.9: Quan hệ kết hợp giữa các lớp



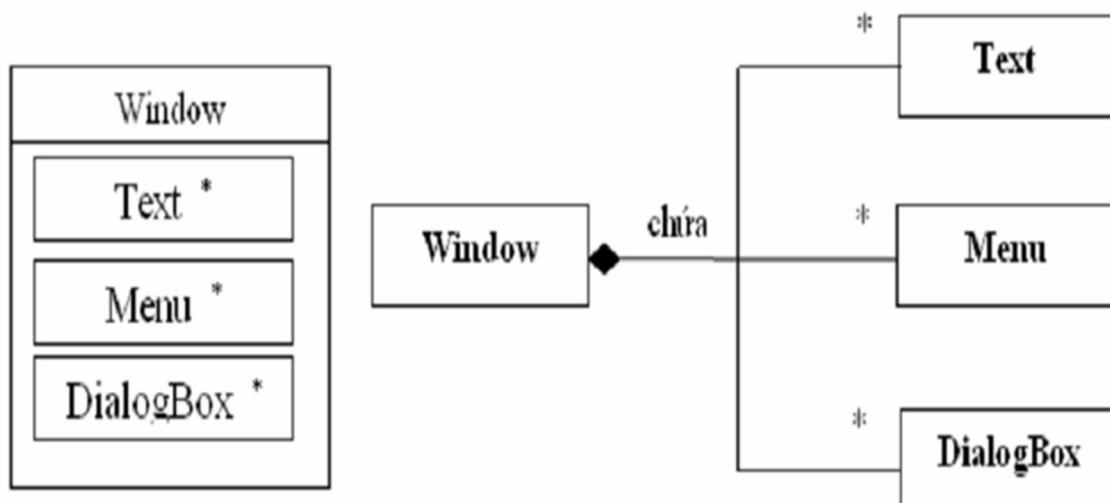
Hình 2.10: Biểu diễn các bội số



Hình 2.11: Quan hệ kết tập thông thường



Hình 2.12: Quan hệ kết tập chia sẻ



Hình 2.13: Quan hệ kết tập hợp thành