

Chương 2

CÁC KHÁI NIỆM CƠ BẢN VỀ HƯỚNG ĐỐI TƯỢNG

Đối tượng (object)

Đối tượng là thành phần trọng tâm của cách tiếp cận hướng đối tượng. Một đối tượng là một đại diện của bất kỳ sự vật nào cần được mô hình trong hệ thống và đóng một vai trò xác định trong lãnh vực ứng dụng.

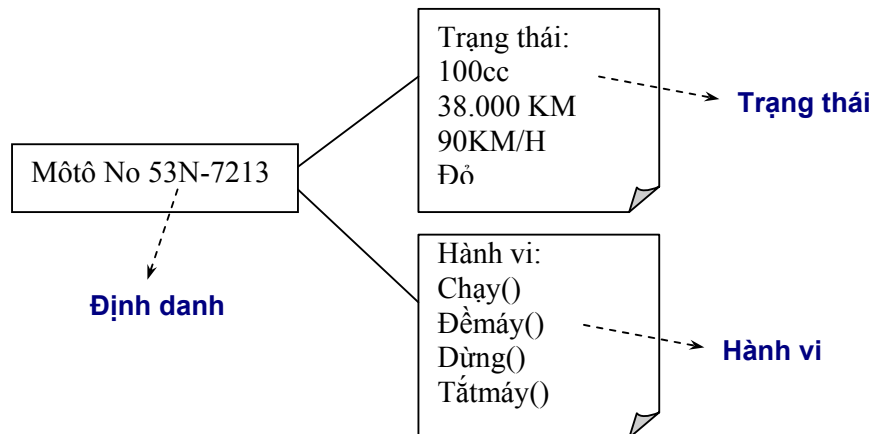
- Là một biểu diễn từ thế giới thực sang thể hiện của tin học (ví dụ : một chiếc xe ô tô trong thế giới thực được biểu diễn trong tin học dùng một khái niệm đối tượng xe ô tô).
- Là một sự trừu tượng hoá, một khái niệm có ý nghĩa trong lãnh vực ứng dụng.
- Diễn đạt một thực thể vật lý, hoặc một thực thể quan niệm, hoặc một thực thể phần mềm.
- Đối tượng có thể là một thực thể hữu hình trực quan (ví dụ : một con người, một vị trí, một sự vật,...) hoặc một khái niệm, một sự kiện (ví dụ : phòng ban, bộ phận, kết hôn, đăng ký, ...).

Một thực thể phải thoả ba nguyên lý :

- Phân biệt (distinction): đơn vị duy nhất (định danh)
- Thường xuyên (permanence) : quá trình sống (trạng thái)
- Hoạt động (activity) : vai trò, hành vi

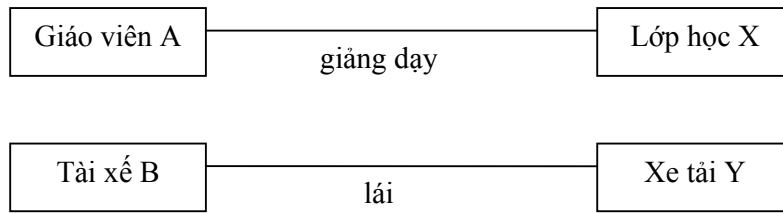
Đối tượng = định danh + trạng thái + hành vi

Ví dụ : một đối tượng xe mô tô

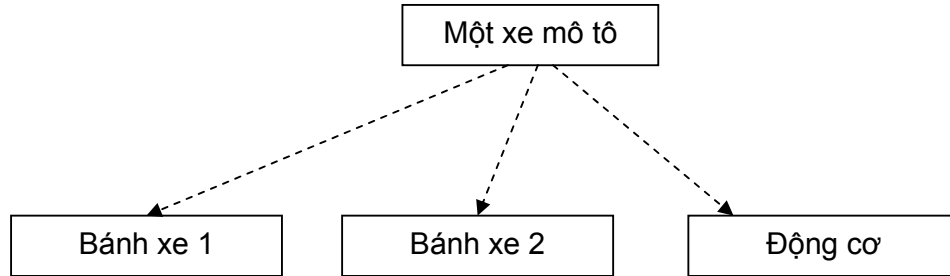


Liên kết giữa các đối tượng

- Mỗi kết hợp (association) – liên kết ngữ nghĩa :



- Phân cấp (hierarchy) – liên kết cấu trúc :

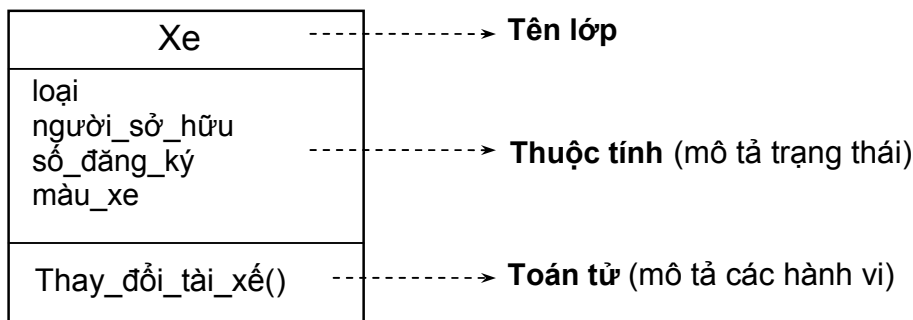


Đối tượng persistent/ transitoire

- Đối tượng transitoire : là đối tượng có quá trình sống tối đa tương ứng với quá trình chạy ứng dụng (đối tượng không được lưu trữ trạng thái)
- Đối tượng persistent : đối tượng có trạng thái được lưu trữ trong máy tính và có thể được thực thi bởi một ứng dụng khác ứng dụng tạo ra nó (quá trình sống của nó kéo dài và có thể từ ứng dụng này qua ứng dụng khác do trạng thái của nó được lưu trữ trong máy tính). Thông thường, trạng thái của đối tượng này sẽ được lưu trữ vào cơ sở dữ liệu trong quá trình sử dụng, và việc lưu trữ này sẽ duy trì được tình trạng của đối tượng và cung cấp tình trạng này cho những lần thực thi khác của ứng dụng hoặc cung cấp trạng thái của đối tượng cho những ứng dụng khác.

Lớp (class)

- Một lớp là một mô tả của một tập hợp/ một loại các đối tượng có :
 - o Cùng cấu trúc (định danh, đặc trưng)
 - o Cùng hành vi (trạng thái, vai trò)
- Trình bày của lớp : là một hình chữ nhật bao gồm ba phần (không bắt buộc)



- Trong giai đoạn cài đặt, định danh của lớp được cài đặt từ một khoá. Khoá này cho phép phân biệt rõ các đối tượng của lớp một cách duy nhất. Khái niệm khoá có thể cho phép truy cập bởi người dùng một cách tường minh hoặc ngầm định. Một khoá tường minh có thể được khai báo chung với trạng thái của lớp trong khi đó khái niệm định danh là một khái niệm độc lập, và có các ý nghĩa sau :
 - o Xác định tính duy nhất của đối tượng
 - o Có ý nghĩa sử dụng đối với người dùng

Ví dụ : trong lớp Xe có thể khai báo số_đăng_ký là một khoá.

Thể hiện của lớp (instance)

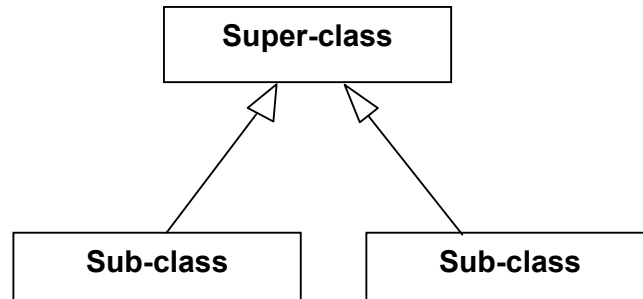
Thể hiện của lớp là một đối tượng cụ thể được tạo ra trên mô hình lớp :

- Các toán tử của lớp mô tả các hành vi chung của các thể hiện
- Tất cả các thể hiện của một lớp có chung các thuộc tính

Phân cấp (hierarchy)

Là cơ chế hỗ trợ việc tổng quát hoá theo cách thức sau :

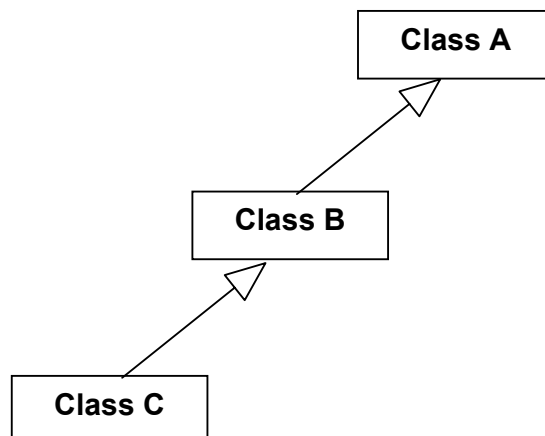
- Tổng quát hoá các đặc tính chung (định nghĩa các supper-class)
- Định nghĩa các đặc tính chuyên biệt nhất của các trường hợp cụ thể (định nghĩa các sub-class)



- Tổng quát hoá (generalisation) : xây dựng một lớp tổng quát từ các lớp khác cụ thể để đạt được một mức độ trừu tượng hoá có thể.
- Chuyên biệt hoá (specialisation) :
 - o Sự phân cấp của các lớp có phép mô tả các lớp chuyên biệt có thể từ các lớp trừu tượng
 - o Sự chuyên biệt hoá cũng có thể được tạo ra để :
 - Làm giàu thông tin : thêm mới thuộc tính hoặc toán tử vào lớp chuyên biệt so với các lớp trừu tượng
 - Có thể thay thế hoặc định nghĩa lại các thuộc tính, toán tử trong các lớp chuyên biệt từ thuộc tính, toán tử của các lớp trừu tượng

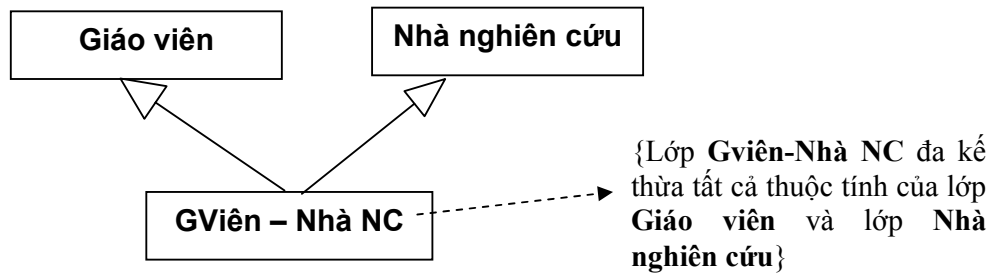
Trong quá trình phân tích hoặc thiết kế hệ thống hướng đối tượng, việc chuyên biệt hoá và tổng quát hoá cho phép định nghĩa các mối quan hệ tập con và làm sáng tỏ tính thừa kế. (Jacobson 1992). Nếu một lớp B thừa kế từ một lớp A, thì có nghĩa rằng tất cả các toán tử và các thuộc tính của lớp A trở thành toán tử và thuộc tính của lớp B.

- Quan hệ kế thừa là quan hệ:
 - o Có tính bắc cầu



{Nếu lớp B là một chuyên biệt hoá của lớp A và lớp C là một chuyên biệt hoá của lớp B thì lớp C cũng là một chuyên biệt hoá của lớp A}

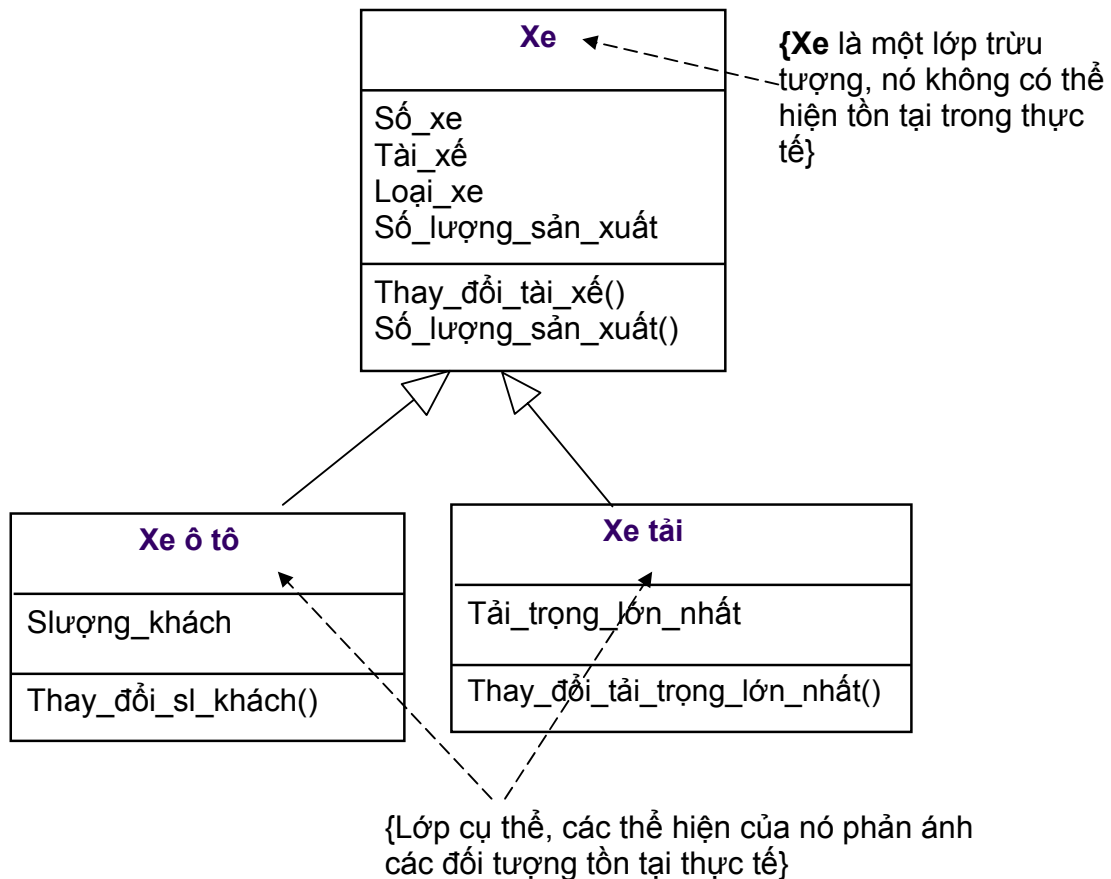
- Có thể đa kế thừa



- Và được cài đặt với mục tiêu tái sử dụng

- Khái niệm lớp trừu tượng (abstract) – lớp cụ thể (concrete) :

Tất cả các thể hiện đối tượng của một lớp trừu tượng đều xuất phát từ một trong những lớp cụ thể của nó. Một lớp trừu tượng không chứa đựng trực tiếp các đối tượng, các thể hiện của nó chỉ là sự xác định trừu tượng hơn của các thể hiện đối tượng trong các lớp cụ thể.



Tính bao bọc(encapsulation)

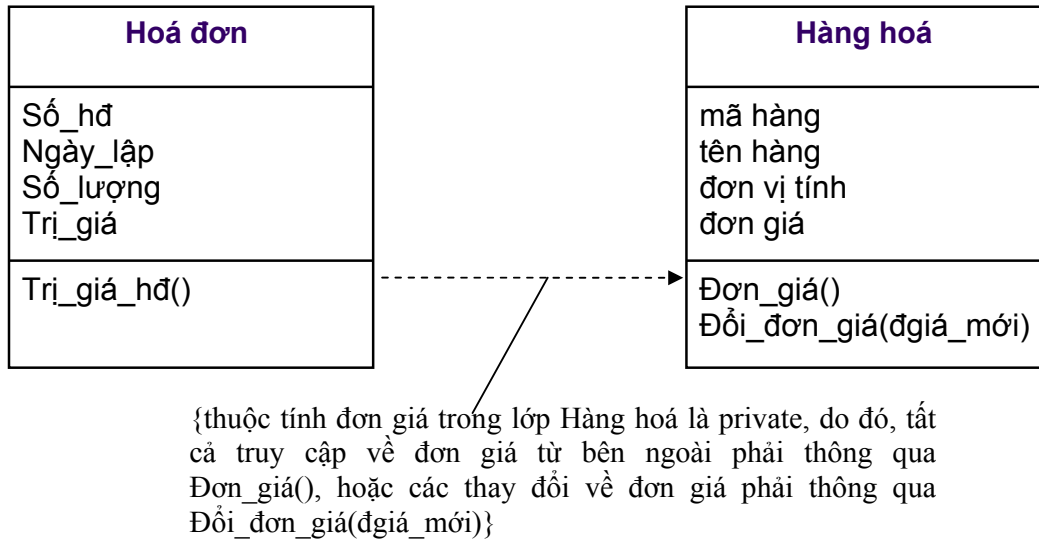
Che dấu thông tin là nguyên lý để che dấu những dữ liệu và thủ tục bên trong của một đối tượng và cung cấp một giao diện tới mỗi đối tượng như là một cách để tiết lộ ít nhất có thể được về nội dung bên trong của đối tượng.

Các cơ chế bao bọc đối tượng tổng quan bao gồm : public, private, và protected

- public : thuộc tính và hành vi của đối tượng có thể được truy cập từ mọi nơi

- private : thuộc tính và hành vi của đối tượng chỉ được bên trong lớp
- protected : thuộc tính và hành vi của đối tượng chỉ được truy cập từ các lớp con

Tính bao bọc là một mục tiêu trong thiết kế hướng đối tượng. Thay vì cho phép một đối tượng truy cập trực tiếp đến dữ liệu của một đối tượng khác, thì đối tượng này sẽ yêu cầu dữ liệu đó thông qua việc gọi thi hành một hành vi đã được thiết kế cho việc cung cấp dữ liệu và một thông điệp sẽ được gửi tới đối tượng đích thông tin được yêu cầu. Điều này không chỉ đảm bảo rằng các lệnh đang hoạt động trong dữ liệu đúng mà còn không cho phép các đối tượng có thể thao tác trực tiếp lên dữ liệu của đối tượng khác.



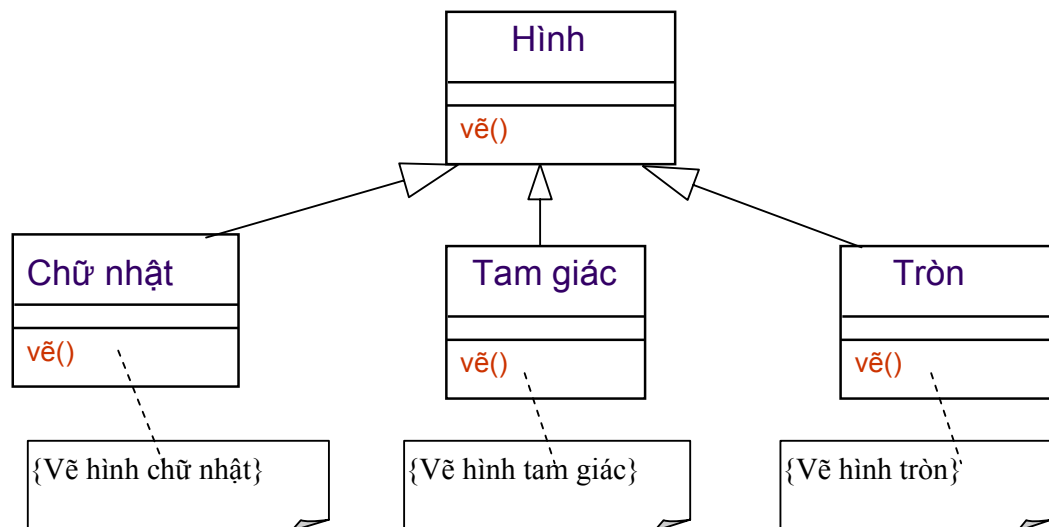
Một yếu tố quan trọng của tính bao bọc là việc thiết kế khác nhau của các đối tượng có thể sử dụng một phương thức (protocol) chung hoặc giao diện chung cho người dùng đối tượng. Điều này có nghĩa rằng nhiều đối tượng sẽ trả lời tới cùng thông điệp nhưng mỗi đối tượng sẽ thi hành thông điệp sử dụng các toán tử đã được biến đổi thích ứng tới lớp của nó. Bằng cách này, một chương trình có thể gọi một thông điệp tổng quát và để lại việc cài đặt cho đối tượng nhận. Điều này làm giảm sự phụ thuộc lẫn nhau và gia tăng số lượng trao đổi và tái sử dụng của đối tượng.

Ví dụ : các động cơ xe ô tô có thể khác nhau về cách cài đặt và vận hành cụ thể, giao diện giữa tài xế và xe ô tô là thông qua một phương thức chung : ví dụ, đạp cần gas để tăng lực và nhả cần gas để giảm lực của xe. Tất cả tài xế đều biết phương thức này và tất cả tài xế đều sử dụng phương thức này trong tất cả xe ô tô mà không qua tâm đến động cơ của xe ô tô được thực hiện như thế nào (tất nhiên, các động cơ khác nhau thì có cách vận hành khác nhau).

Tính đa hình (polymorphism)

Thuật ngữ đa hình dùng để mô tả một yếu tố có nhiều dạng thức. Trong tin học, thuật ngữ đa hình được hiểu như là quan hệ đối tượng của nhiều lớp khác nhau có chung một (vài) lớp cha (superclass), đó là, bất kỳ đối tượng nào được chỉ định bởi tên này thì có thể trả lời tới một vài tập toán tử chung trong một cách khác nhau :

- Cùng toán tử có thể thi hành khác nhau trong các lớp khác nhau.
- Các phương thức khác nhau cùng cài đặt cho toán tử này trong các lớp khác nhau phải có cùng ký hiệu (tên, tham số và giá trị trả về)
- Cài đặt của toán tử được xác định bởi lớp đối tượng mà được sử dụng trực tiếp



Cấu trúc phân cấp trên cho thấy, lớp Hình là lớp tổng quát chung cho các lớp : Chữ nhật, Tam giác, Tròn. Vì cả ba lớp này đều có thể vẽ, do đó, có thể xác định một phương thức vẽ() chung trong lớp Hình. Tuy nhiên, các đối tượng trong các lớp chuyên biệt có thể được thực hiện trong một cách thức có thể khác so với phương thức vẽ() chung. Do đó, mỗi lớp chuyên biệt có thể cài đặt lại phương thức vẽ() chồng lên phương thức vẽ() của lớp tổng quát.

Nếu hệ thống xử lý một danh sách các đối tượng của lớp Hình, thì hệ thống sẽ dò tìm và thực hiện phương thức vẽ() phù hợp cho mỗi đối tượng. Nếu đối tượng đó là của lớp con Chữ nhật thì nội dung cài đặt của phương thức vẽ() trong lớp Chữ nhật sẽ được thực thi. Tương tự cho đối tượng của lớp con Tam giác và Tròn.

Đa hình ở đây là cho phép có nhiều hình thức cài đặt của cùng một hành vi (phương thức). Hệ thống sẽ tự động thực hiện phương thức thích hợp cho mỗi đối tượng.

Câu hỏi và bài tập