

IT3120 - Phân tích và thiết kế hệ thống thông tin

Bài 2. Mô hình hóa hướng đối tượng

Nguyễn Hữu Đức

Viện CNTT&TT

17 - 01 - 2018

Công nghệ đối tượng là gì

Định nghĩa (Object technology: A manager's guide, Taylor, 1997)

Công nghệ đối tượng là một tập hợp nguyên lý định hướng việc phát triển phần mềm, kèm theo những ngôn ngữ, cơ sở dữ liệu, và các công cụ hỗ trợ các nguyên lý này.

Các nguyên lý cơ bản của tiếp cận hướng đối tượng

- Khái quát hóa/Cụ thể hóa (Generalization/Specialization)
- Tính đóng gói (Encapsulation)
- Tính mô đun hóa (Modularization)
- Tính phân cấp (Hierarchical)

Mô hình hóa hướng đối tượng

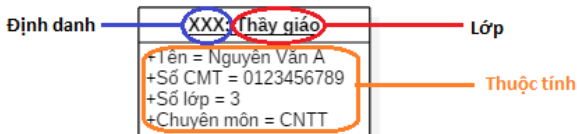
- **Mô hình** là một góc nhìn đơn giản của thực tế
- **Mô hình hóa** là cách thức biểu diễn hệ thống bằng các mô hình ở những góc nhìn khác nhau
 - Phương pháp, quy trình mô hình hóa (RUP,...)
 - Ngôn ngữ mô hình hóa (UML)
 - Công cụ mô hình hóa (Rational Rose, StarUML,...)
- **Mô hình hóa hướng đối tượng** (phân tích, thiết kế) là tiếp cận mô hình hóa hệ thống thành một tập đối tượng cùng với sự tương tác của chúng.
 - Đối tượng là những thực thể đóng vai trò quan trọng trong sự hình thành hệ thống
 - Sự tương tác giữa các đối tượng thường thể hiện thông qua việc chuyển giao thông điệp giữa chúng

Tại sao phải mô hình hóa?

- Mô hình được xây dựng với mục đích hiểu rõ hơn về hệ thống đang phát triển
- Mô hình hóa cần đạt bốn mục tiêu
 - Trực quan hóa hệ thống
 - Cho phép đặc tả cấu trúc và hành vi của hệ thống
 - Cung cấp khuôn mẫu hướng dẫn việc phát triển hệ thống
 - Tư liệu hóa các quyết định phát triển hệ thống
- Với các hệ thống phức tạp, mô hình hóa là cần thiết vì chúng ta khó nắm bắt được toàn bộ hệ thống

Đối tượng là gì?

- Đối tượng biểu diễn:
 - Thực thể vật lý: vd. Ngôi nhà, Ô tô,...
 - Thực thể ý niệm: vd. Quy trình xử lý,...
 - Thực thể phần mềm: vd. Danh sách liên kết, Cây nhị phân,...
- Đối tượng là thực thể có biên và định danh rõ ràng, đóng gói các trạng thái và hành vi
 - Trạng thái được biểu diễn bằng các thuộc tính và quan hệ
 - Hành vi được biểu diễn bằng các thao tác, phương thức, và máy trạng thái



Trạng thái của đối tượng

- Trạng thái của đối tượng là một trong những điều kiện tồn tại của đối tượng
- Trạng thái của đối tượng thường thay đổi theo thời gian

Hành vi của đối tượng

- Hành vi quyết định các hành động và phản ứng của đối tượng
- Hành vi quan sát được của đối tượng có thể mô hình hóa bằng một tập các thông điệp mà đối tượng có thể tiếp nhận
- Ví dụ: Nhận lớp học, Chấm điểm,...

Định danh của đối tượng

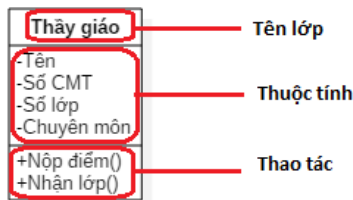
- Mỗi đối tượng có một định danh duy nhất kể cả khi trạng thái của nó giống hệt trạng thái của một đối tượng khác

XXX : Thầy giáo
+Tên = Nguyễn Văn A
+Số CMT = 0123456789
+Số lớp = 3
+Chuyên môn = CNTT

YYY : Thầy giáo
+Tên = Nguyễn Văn A
+Số CMT = 0123456789
+Số lớp = 3
+Chuyên môn = CNTT

Lớp

- Lớp là một đặc tả cho một tập đối tượng có cùng *thuộc tính*, *thao tác*, *quan hệ*, và ngữ nghĩa.
 - Đóng vai trò khuôn mẫu để tạo ra đối tượng
 - Đối tượng là một thể hiện của lớp
- Lớp là một sự trừu tượng hóa, không phải tập đối tượng



Thuộc tính của lớp

<Phạm vi> Tên : Kiểu [Cơ số] = Giá trị mặc định

- Thuộc tính (attribute) là một tính chất được đặt tên cho một lớp, mô tả phạm vi giá trị mà thuộc tính đó có thể nhận.
- Phạm vi
 - "+" : Public, các đối tượng khác có thể "nhìn thấy"
 - "-" : Private, các đối tượng khác không thể "nhìn thấy"
 - "#" : Protected, chỉ các đối tượng thuộc lớp đó hoặc lớp "hậu duệ" có thể "nhìn thấy"
 - "~" : Package, chỉ các đối tượng thuộc lớp cùng gói có thể "nhìn thấy"

<Phạm vi> Tên : Kiểu [Cơ số] = Giá trị mặc định

- Kiểu: Phạm vi giá trị mà thuộc tính có thể nhận được
- Giá trị mặc định: Giá trị được gán mặc định khi đối tượng được tạo mới
- Cơ số: Số lượng giá trị mà thuộc tính có thể nhận
 - "*" : Nhiều
 - "+" : Nhiều hơn hoặc bằng 1
 - "m-n" : Từ m đến n

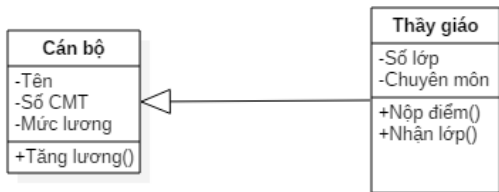
Thao tác của lớp

<Phạm vi> Tên(Tham số 1 : Kiểu [Cơ số] = Giá trị mặc định, Tham số 2, ...) : Kiểu trả về

- Thao tác là một "dịch vụ" của các đối tượng của lớp..
- Phạm vi
 - "+" : Public, các đối tượng khác có thể "nhìn thấy"
 - "-" : Private, các đối tượng khác không thể "nhìn thấy"
 - "#" : Protected, chỉ các đối tượng thuộc lớp đó hoặc lớp "hậu duệ" có thể "nhìn thấy"
 - "~" : Package, chỉ các đối tượng thuộc lớp cùng gói có thể "nhìn thấy"

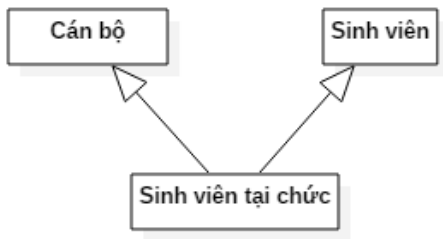
Kế thừa

- Một lớp (lớp con) kế thừa các thuộc tính, thao tác, liên kết từ một lớp khác (lớp cha).
- Lớp con có thể bổ sung thêm các thuộc tính, thao tác, hay liên kết mới
- Lớp con có thể viết đè lên các thao tác của lớp cha, qua đó thay đổi hành vi của thao tác tương ứng trên các đối tượng của lớp cha



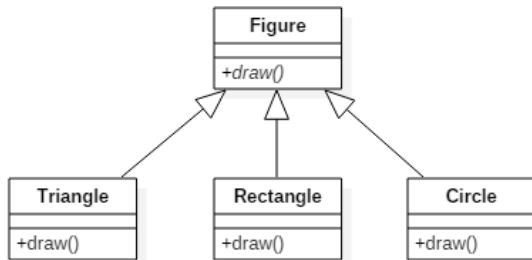
Đa kế thừa

- Một lớp (lớp con) có thể kế thừa từ nhiều lớp cha.
- Tình huống nhập nhằng có thể xảy ra khi các lớp cha có các thuộc tính, thao tác hoặc liên kết cùng tên



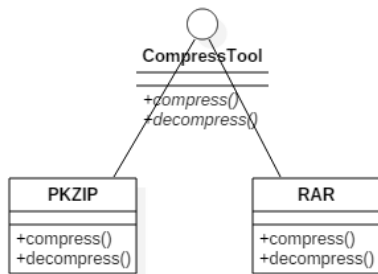
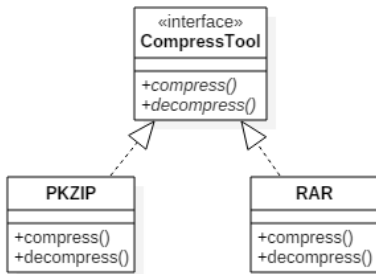
Tính đa hình

- Đa hình (polymorphism) là sự tồn tại nhiều hình thái khác nhau của cùng một biểu diễn.
- Trong tiếp cận hướng đối tượng, đa hình thể hiện khi các lớp con viết đè cùng một phương thức của lớp cha, dẫn đến có nhiều thể hiện khác nhau của cùng phương thức đó.

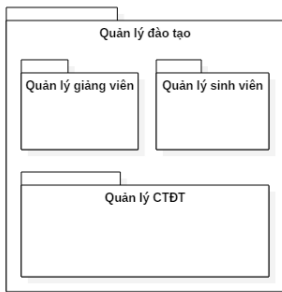


Giao diện

- Giao diện là một lớp đặc biệt chỉ chứa các thao tác *trừu tượng*.
 - Thao tác trừu tượng là thao tác không có cài đặt cụ thể cho phương thức thực hiện
- Giao diện có thể được hiện thực hóa bởi một hoặc nhiều lớp
- Giao diện thường được dùng như một dạng "hợp đồng" giữa bên sử dụng dịch vụ và bên cung cấp dịch vụ



- Gói (package) là kỹ thuật tổ chức các phần tử thành các nhóm.
- Là thành phần mô hình chứa các thành phần mô hình khác.
- Gói có thể sử dụng để:
 - Tổ chức các mô hình trong quá trình phát triển.
 - Làm một đơn vị trong quản lý cấu hình.



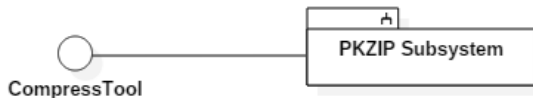
Hệ thống con

- Hệ thống con (subsystem) Là sự kết hợp giữa gói (chứa các thành phần mô hình) và lớp (có hành vi).
- Hiện thực hóa các giao diện



Thành phần

- Thành phần (component) là một phần không tầm thường của hệ thống, tương đối độc lập và có thể thay thế, thỏa mãn những chức năng rõ ràng trong một kiến trúc.
- Thành phần có thể là:
 - Thành phần mã nguồn.
 - Thành phần thực thi.
 - Các tệp dữ liệu



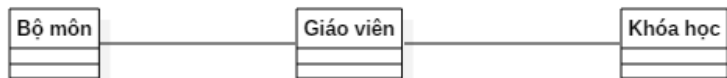
Thiết kế



Thực hiện

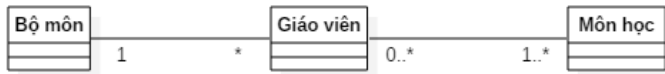
Liên kết (Association)

- Liên kết là một quan hệ có tính cấu trúc, chỉ ra rằng đối tượng của một lớp có thể kết nối với đối tượng của một lớp khác.
- Ví dụ:
 - Giáo viên thuộc bộ môn.
 - Giáo viên dạy khóa học.



Cơ số (multiplicity)

- Chỉ số lượng đối tượng của một lớp có kết nối tới một đối tượng lớp khác.
 - "1", "*", "+", "0-1", "m-n"
- Mỗi liên kết có cơ số ở hai đầu liên kết



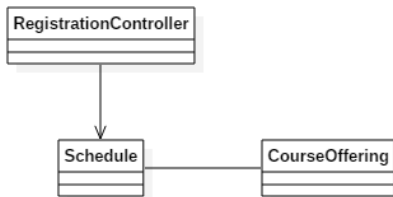
Hợp thành (composition), và kết nhập (aggregation)

- Chỉ các liên kết đại diện cho quan hệ tổng thể-bộ phận.
 - Hợp thành: ràng buộc chặt giữa bộ phận với tổng thể, tổng thể bị hủy thì bộ phận cũng bị hủy theo
 - Kết nhập: ràng buộc lỏng giữa bộ phận và tổng thể, tổng thể bị hủy, bộ phận vẫn tồn tại



Lưu hành (navigability)

- Chỉ ra khả năng có thể tìm đến một đối tượng từ một đối tượng khác
 - Một chiều: có mũi tên
 - Hai chiều: Không có mũi tên



Phụ thuộc

- Là quan hệ giữa hai yếu tố thiết kế chỉ ra rằng sự thay đổi của một yếu tố này dẫn đến sự thay đổi của yếu tố còn lại
 - Phụ thuộc giữa các lớp
 - Phụ thuộc giữa các thành phần
 - Phụ thuộc giữa các gói

