Phân tích thiết kế hướng đối tượng

Bài 2: Cơ sở lập trình hướng đối tượng

TS. Nguyễn Hiếu Cường
Bộ môn CNPM, Khoa CNTT, Trường ĐH GTVT
Email: cuonggt@gmail.com

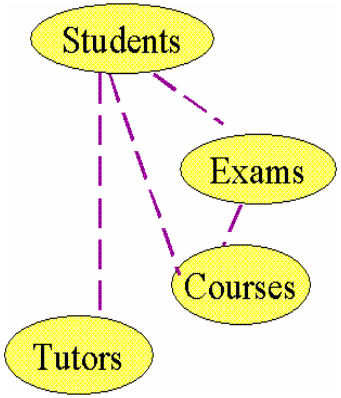
Hà Nội, 2025

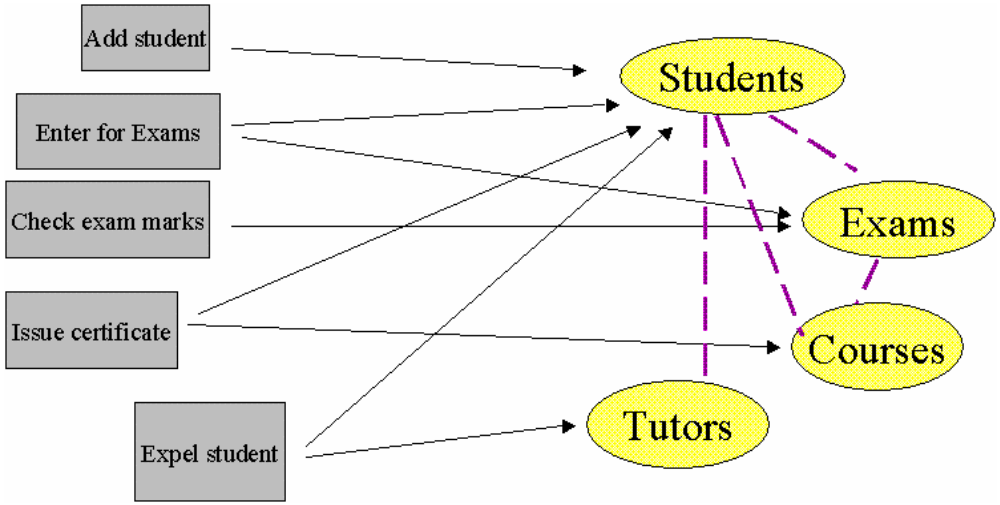
Lập trình cấu trúc (hướng chức năng)

- Phân rã bài toán thành các chức năng, cài đặt bằng các hàm

```
add_student()
enter_for_exam()
check_exam_marks()
issue_certificate()
expel_student()
```

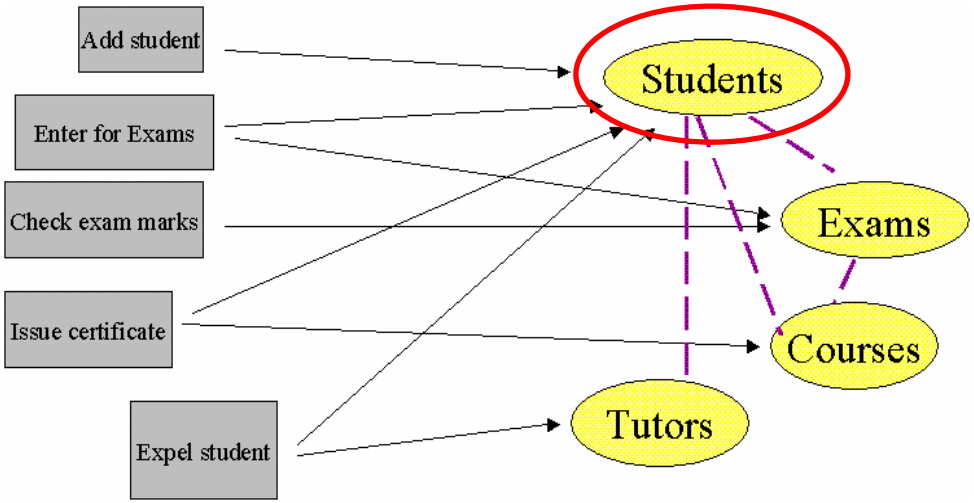
- Dữ liệu lưu trong các tệp





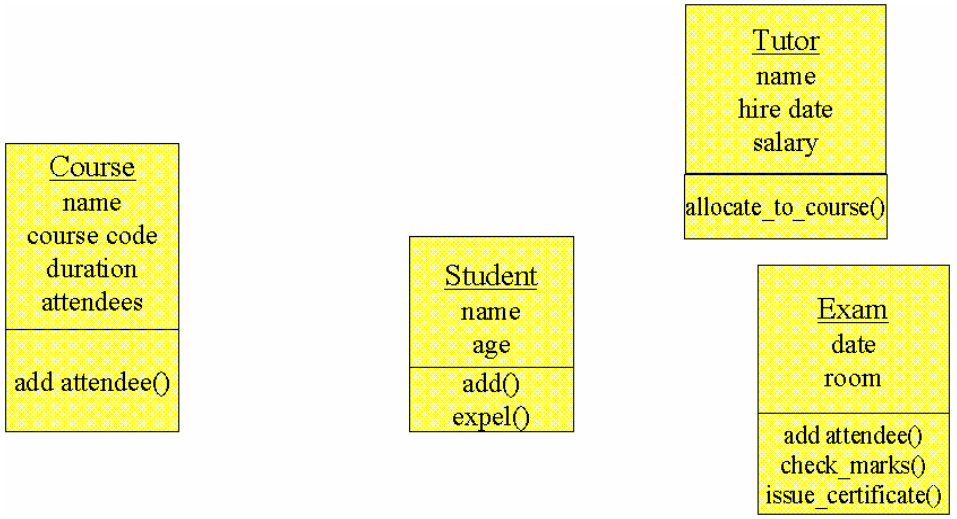
Một vấn đề của lập trình cấu trúc

- Điều gì xảy ra khi có sự thay đổi về dữ liệu trong Students?



Tiếp cận hướng đối tượng

- Cách tiếp cận hướng đối tượng khắc phục vấn đề như trên
  - Dữ liệu và các hàm được liên kết với nhau trong cùng một mô-đun
  - Dữ liệu trong Student chỉ ảnh hưởng đến các hàm add() và expel()

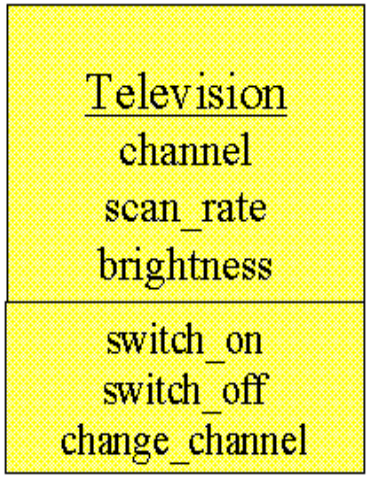


Sự đóng gói

- Sự đóng gói: dữ liệu + thao tác
  - Chỉ các thực thể chứa dữ liệu có thể đọc hoặc chỉnh sửa dữ liệu đó
  - Mỗi lớp thực hiện một cách an toàn việc thay đổi dữ liệu bên trong
- Trong ví dụ ở trên
  - Chỉ các hàm bên trong lớp Student có thể cập nhật và đọc dữ liệu *age*
  - Nếu định dạng của *age* thay đổi thì chỉ các hàm bên trong Student có sử dụng thông tin này cần điều chỉnh
  - Các hàm ở lớp khác không thể truy cập trực tiếp vào *age* mà cần phải thông qua các hàm của Student nếu muốn sử dụng dữ liệu này

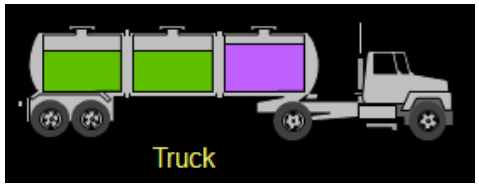
Đối tượng

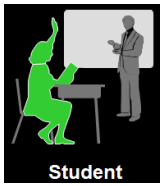
- Đối tượng (object) trong thế giới thực có thể là bất kỳ cái gì, có đặc trưng bởi dữ liệu (data) và các hành vi (behaviours)

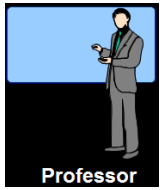


Đối tượng

- Thế giới thực bao gồm các *đối tượng* (object)!
  - Đối tượng vật lý

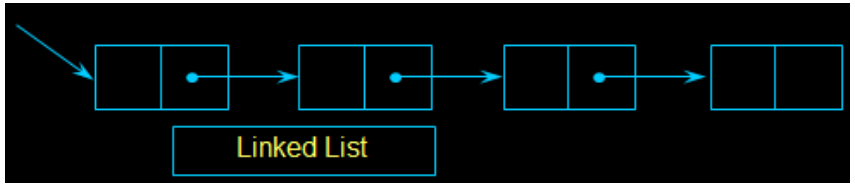






  - Đối tượng khái niệm

  - Đối tượng phần mềm



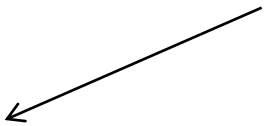
- Mỗi đối tượng có các thuộc tính riêng và các hành vi

Thông điệp và truyền thông điệp

- Thông điệp (message)
  - Kích hoạt một hành vi của của đối tượng

```
Television aTV;

aTV.switch_on();
```



Thông điệp `switch_on()` được truyền tới đối tượng `aTV`

- Phương thức ≠ thông điệp?

Lớp

- Lớp (class)
  - Định nghĩa trừu tượng của các đối tượng có cùng những đặc tính chung
  - Đối tượng (object) là thể hiện cụ thể (instance) của một lớp

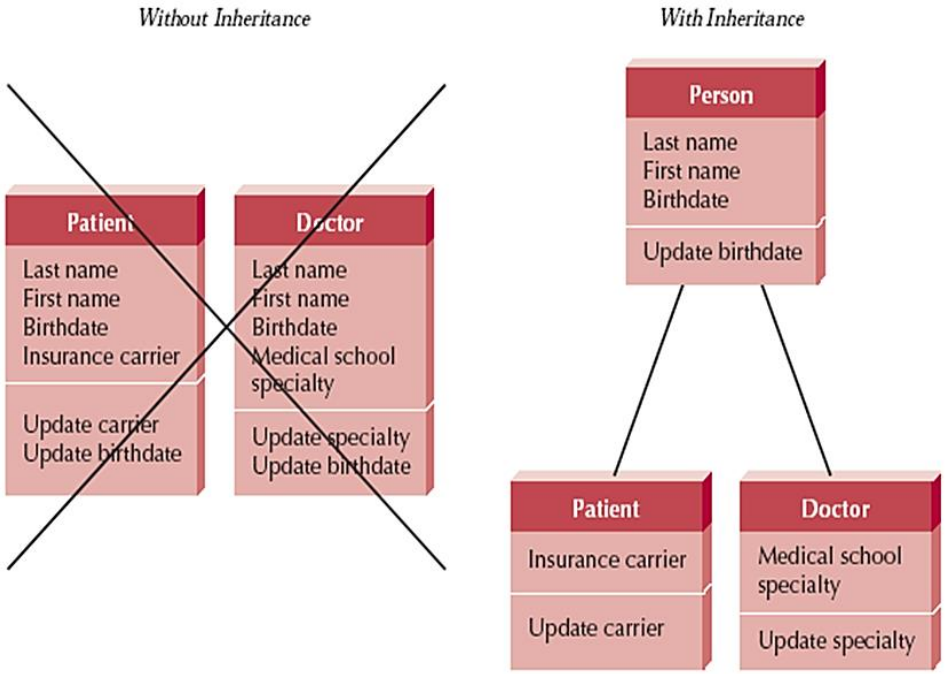
- Tác dụng của lớp?

Kế thừa

- Mục đích của kế thừa
  - Xây dựng lớp mới (lớp con) bằng cách sử dụng lại lớp đã có (lớp cha)
  - Có thể có những thiết kế hiệu quả hơn
- Một lớp có thể
  - Được thừa kế từ một hoặc nhiều lớp khác, đồng thời có thể là cơ sở của một hoặc nhiều lớp khác
  - Được kế thừa cả dữ liệu và các hành vi từ lớp cha

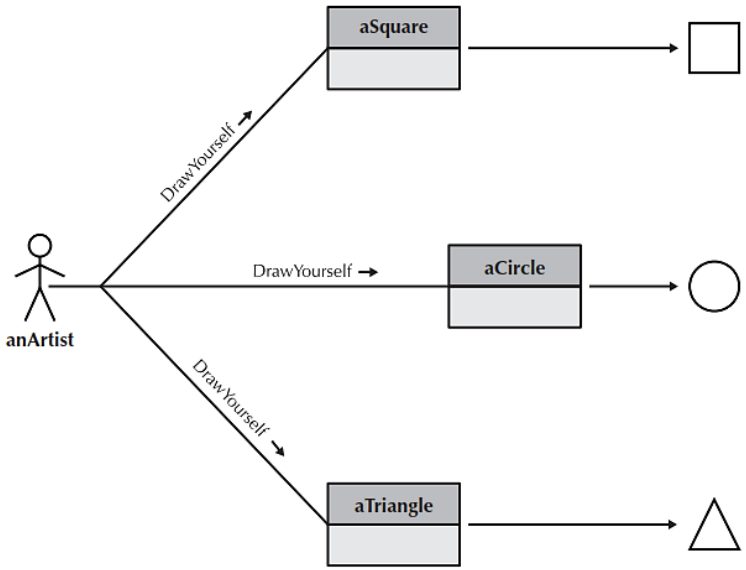
Tác dụng của kế thừa

- Ngoài tác dụng sử dụng lại, kế thừa còn là cơ chế cho phép tạo nên các thiết kế hiệu quả hơn



Đa hình

- Đa hình hoặc tương ứng bội (polymorphism)
  - Một thông điệp có thể được diễn giải theo các cách khác nhau
  - Có thể ẩn các chi tiết cài đặt dưới một "giao diện" chung



Câu hỏi và bài tập

1. Trong OOP, giải thích ngắn gọn: Encapsulation là gì? Cho ví dụ minh họa với lớp Student.
2. Hãy trình bày ngắn gọn về các khái niệm trong OOP: Abstraction, Inheritance, Polymorphism.
3. Viết lớp Account với các thuộc tính owner, balance, các phương thức: deposit(), withdraw(). Hãy giải thích tính kế thừa (inheritance) nếu muốn xây dựng lớp SavingsAccount kế thừa từ Account.
4. Trong hệ thống Library System, giả sử có 2 lớp Book và EBook.Hãy cho biết đa hình (polymorphism) được thể hiện thế nào qua 1 ví dụ C++ có phương thức read() mà hai lớp cài đặt khác nhau.