

# 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

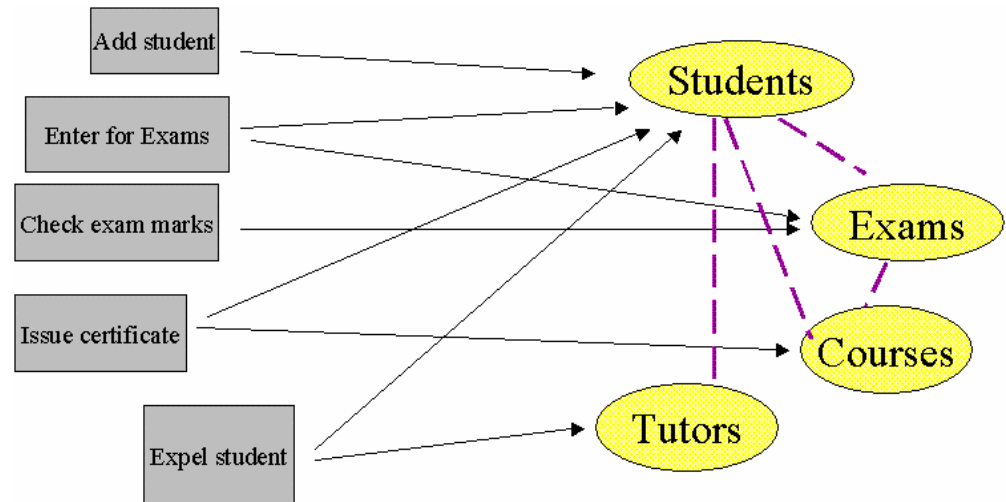
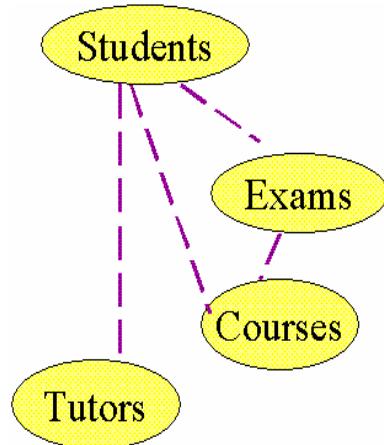
Email: [cuonggt@gmail.com](mailto:cuonggt@gmail.com)

# Lập trình cấu trúc (hướng chức năng)

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

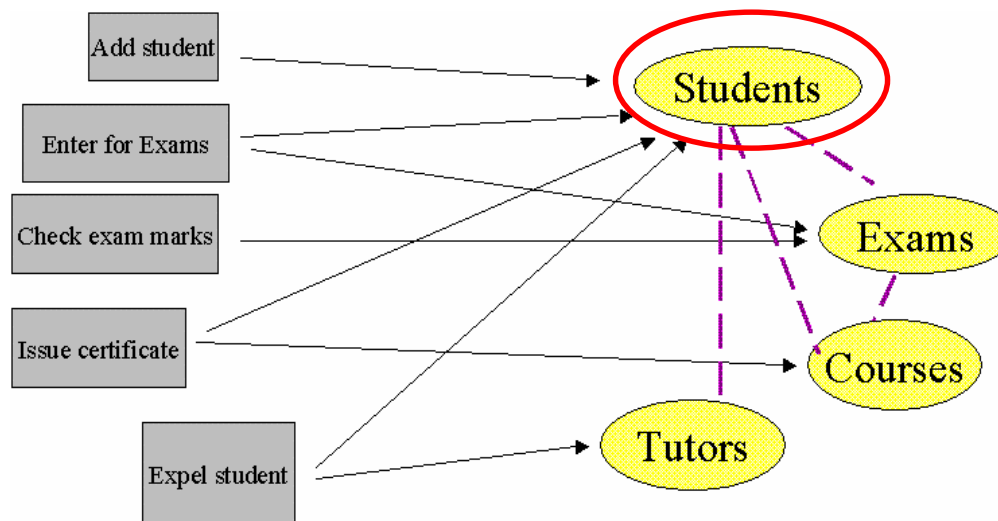
```
add_student()  
enter_for_exam()  
check_exam_marks()  
issue_certificate()  
expel_student()
```

- Dữ liệu lưu trong các tệp



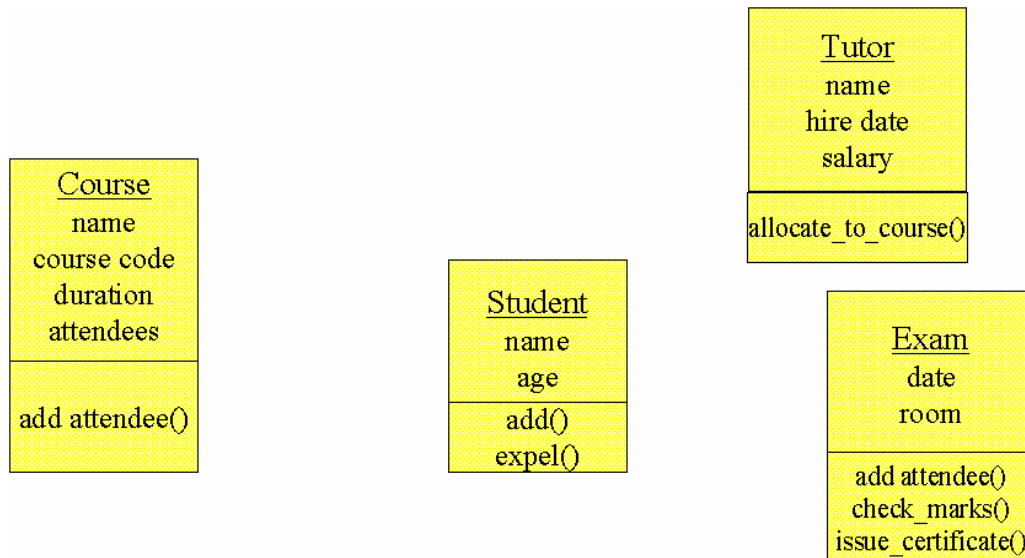
# Một vấn đề của lập trình cấu trúc

- 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
- Ví dụ: các hàm add() và expel() liên quan tới dữ liệu trong Student



# Sự đóng gói

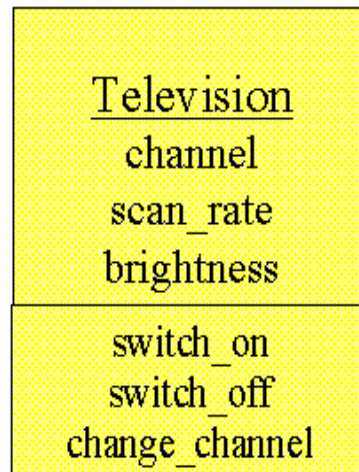
---

- Sự đóng gói: dữ liệu + thao tác
  - Chỉ các thực thể chứa dữ liệu có thể đọc hoặc chỉnh sửa dữ liệu đó
  - Trong ví dụ ở trên, thực thể Tutor không thể cập nhật và đọc dữ liệu age bên trong Student
- Lập trình viên của mô-đun Student có thể thực hiện một cách an toàn việc thay đổi dữ liệu trong mô-đun

# Đối tượng

---

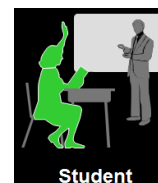
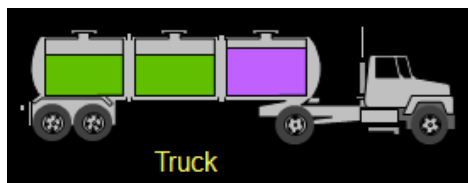
- Đối tượng (object) trong thế giới thực có thể là bất kỳ cái gì, có đặc trưng bởi dữ liệu (data) và các hành vi (behaviours)
- Khi cài đặt, dữ liệu là các thuộc tính (attributes), hành vi là các hàm hoặc còn gọi là các phương thức (methods)



# Đối tượng

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

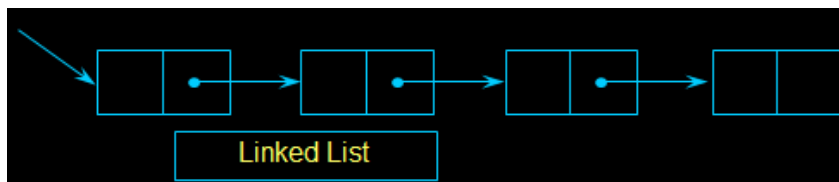
- Đối tượng vật lý



- Đối tượng khái niệm



- Đối tượng phần mềm



# Thông điệp và truyền thông điệp

---

- Thông điệp (message)

- Kích hoạt một hành động của của đối tượng

```
Television aTV;
```

```
aTV.switch_on();
```

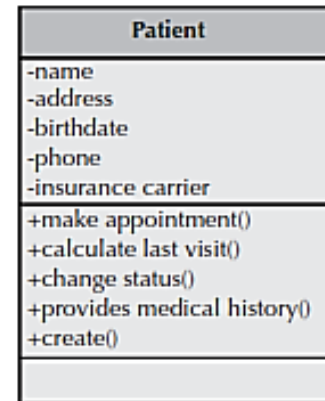
← Thông điệp `switch_on()`

- Phương thức  $\neq$  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?
  - Trừu tượng hoá dữ liệu
  - Đóng gói (dữ liệu + thao tác)
  - Che giấu thông tin
  - ...



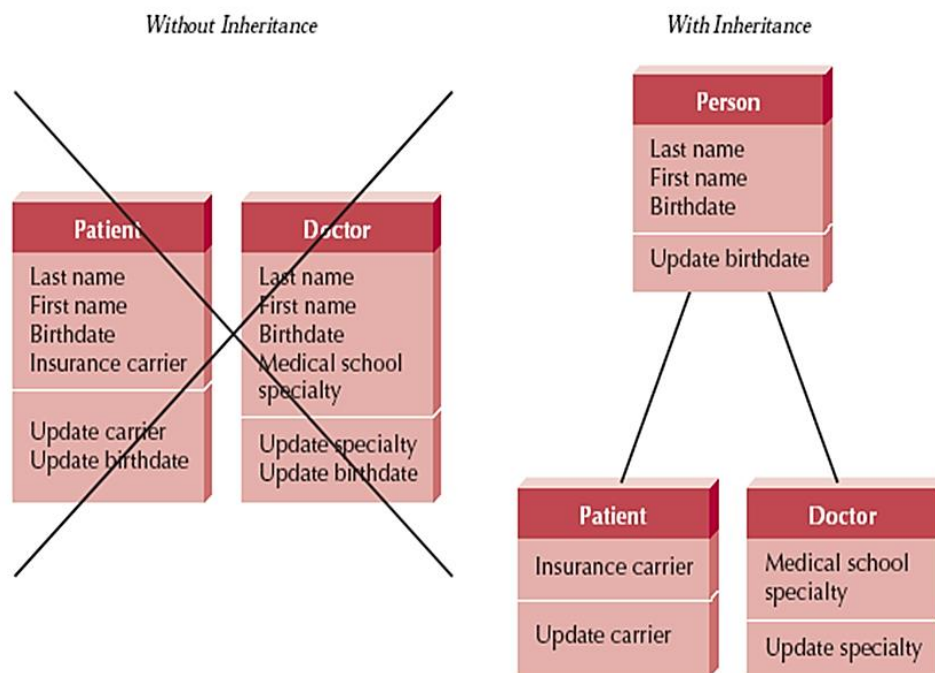
# Kế thừa

---

- Mục đích
  - Xây dựng một lớp mới bằng cách kế thừa các lớp đã có
- Một lớp có thể
  - Được thừa kế từ một hoặc nhiều lớp khác
  - Là cơ sở của một hoặc nhiều lớp khác
- Những gì được kế thừa?

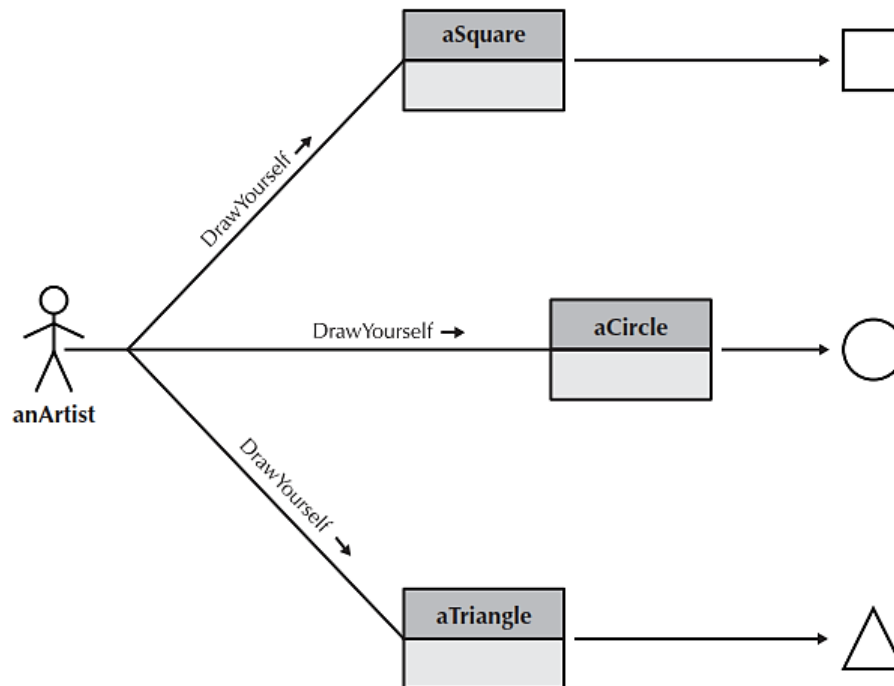
# Tác dụng của kế thừa

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

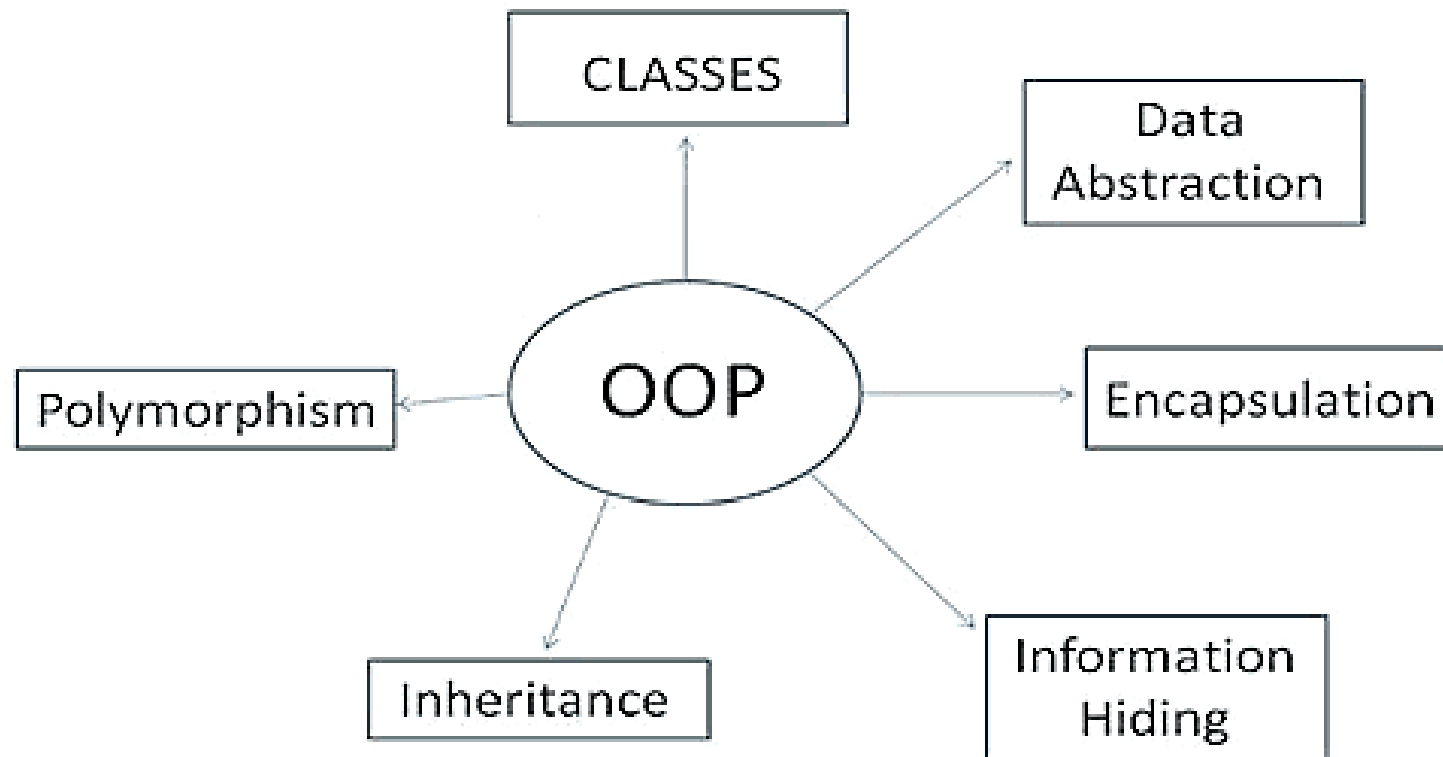


# Đa hình

- 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



# Tóm tắt





- Một bài toán khó hơn một chút so với mức độ cơ bản  
→ Nhu cầu cần phải phân tích và thiết kế
- Sơ lược về cách phân tích thiết kế theo hướng đối tượng
- Vận dụng các khái niệm hướng đối tượng thông qua việc cài đặt chương trình C++

# Bài toán cần giải quyết

---

- Một cô gái (Suzy) có:
    - Một *chuồng* gồm nhiều *ngăn* để nuôi các con vật làm cảnh, mỗi ngăn chỉ nuôi 1 con, mỗi con đều có tên
    - Có nhiều loại con vật khác nhau (mèo, chó)
  - Do số lượng các con vật ngày càng nhiều, Suzy cần:
    - Xây dựng một chương trình để quản lý các con vật trên
  - Chương trình có nhiệm vụ gì?
    - Xác định danh tính con vật trong từng ngăn
    - Thêm con vật vào chuồng, lấy một con vật ra khỏi chuồng
    - Có thể sửa chữa, mở rộng chương trình một cách dễ dàng
-