

## Phần 2

# Lý thuyết Lập Trình Hướng Đối Tượng



TS. Phan Thượng Cang  
Khoa CNTT&TT-Đại học Cần Thơ

# Nội dung



- ⌘ Phương pháp Lập trình hướng đối tượng (OOP) ?
- ⌘ Các khái niệm OOP
- ⌘ Các tính chất quan trọng của OOP

# Nội dung

## ⌘ Phương pháp Lập trình hướng đối tượng (OOP) ?

- Ý tưởng – định nghĩa
- Sự khác biệt OOP với Lập trình thủ tục - Tại sao nên dùng OOP ?

## ⌘ Các khái niệm OOP

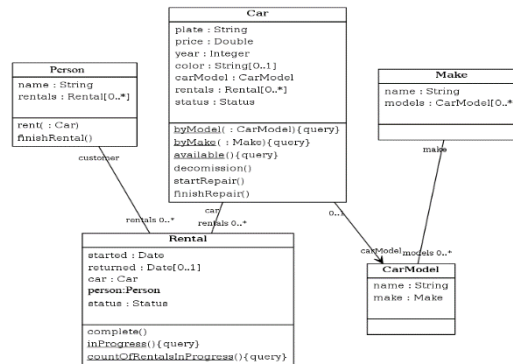
- Đối tượng vs Lớp
- Phạm vi truy cập
- Thông điệp – Truyền thông điệp

## ⌘ Các tính chất quan trọng của OOP

- Trừu tượng hoá
- Bao gói
- Đa hình
- Thừa kế

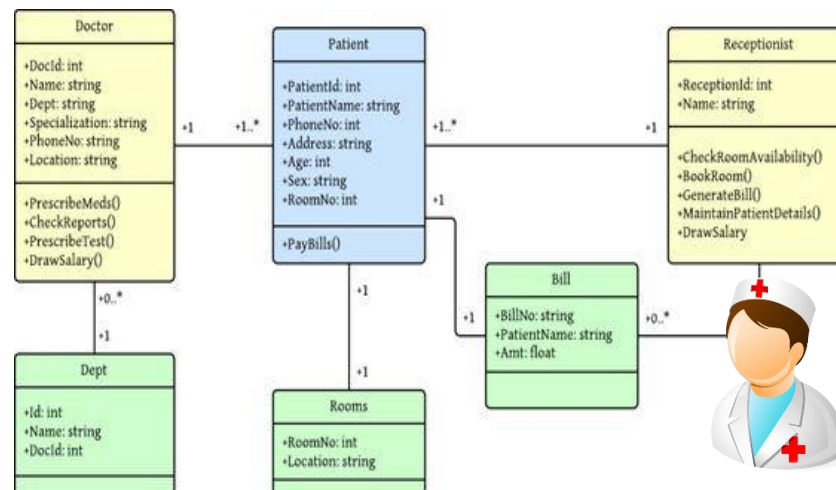
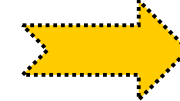
# Phương pháp Lập trình hướng đối tượng

⌘ Ý tưởng: mô hình hoá bài toán vào ngôn ngữ lập trình



```
class Person{
    String id;
    String name;
    void res()
}

class Car{
    .....
    String id; }
    Person owner;
    void input()
    .....
}
```



```
class Patient{
    String id;
    String name;
    void exam();
    .....
}

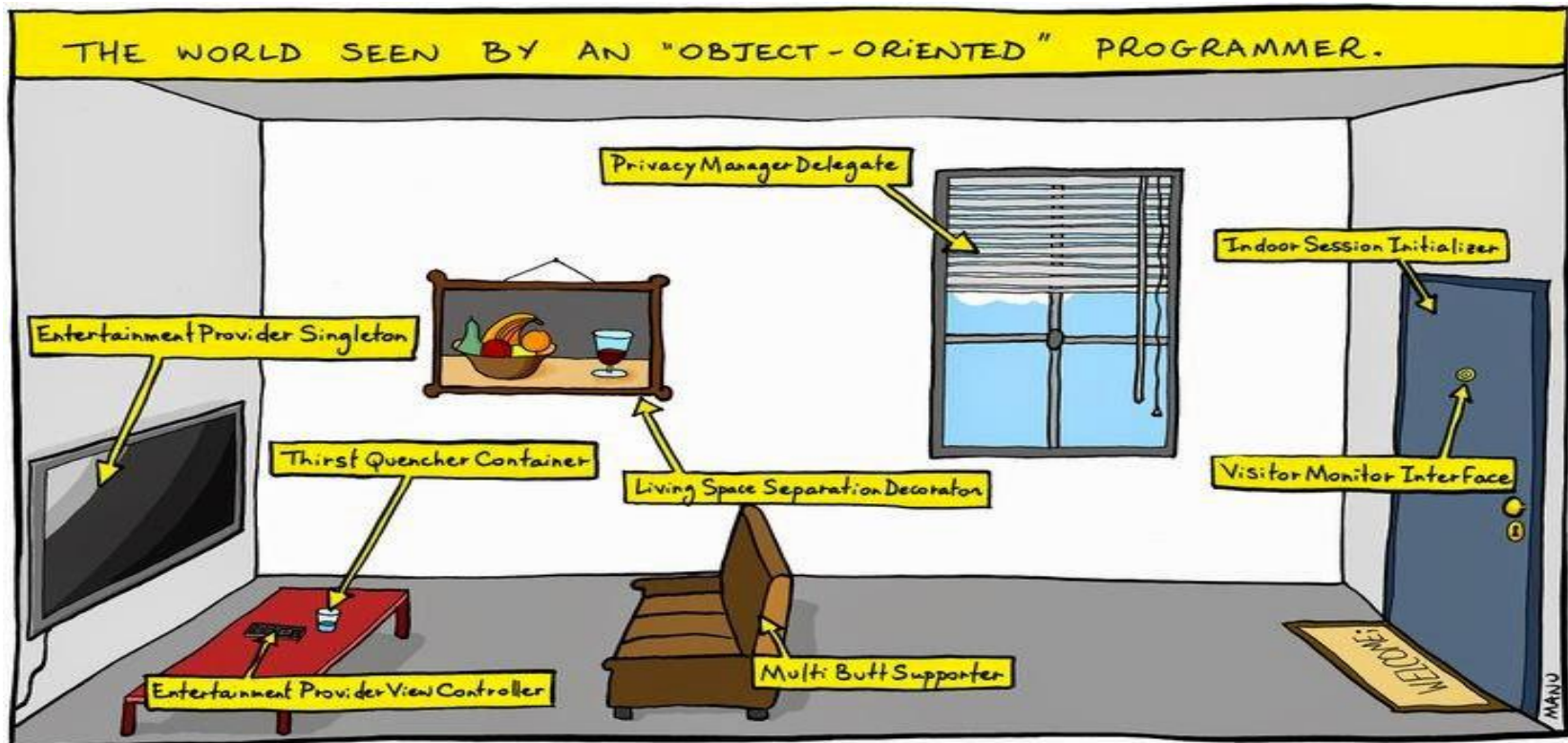
class Doctor{
    String name;
    Patient p[];
    void medicine();
    .....
}
```



# Phương pháp Lập trình hướng đối tượng

⌘ **Định nghĩa:** Object-oriented programming (OOP)

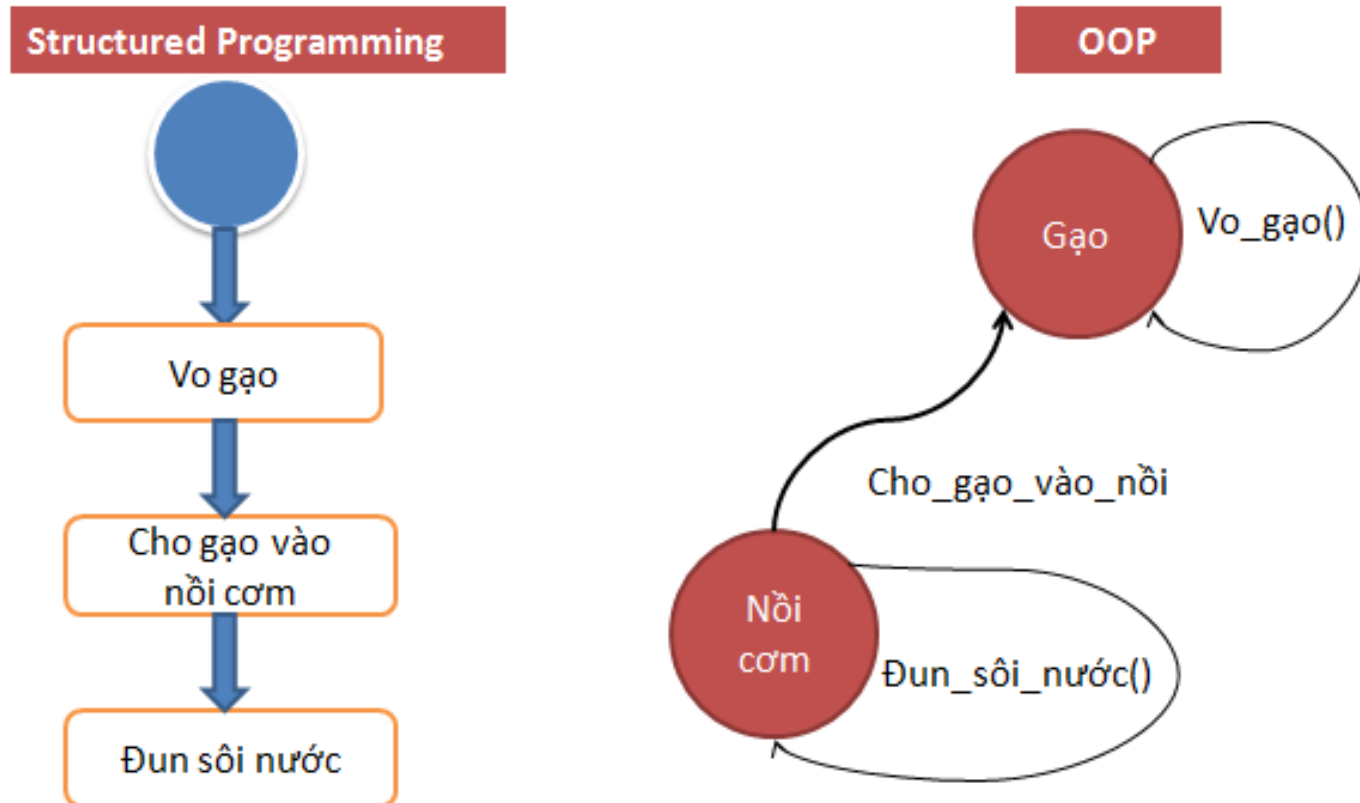
- Là cách viết chương trình máy tính bao gồm tập các đối tượng (chứa dữ liệu và phương thức) có thể tương tác với nhau.
- Là phương pháp lập trình mô hình hóa các vấn đề cần giải quyết bằng phần mềm dựa vào khái niệm đối tượng.



# Phương pháp Lập trình hướng đối tượng

## ⌘ Sự khác biệt OOP với Lập trình thủ tục

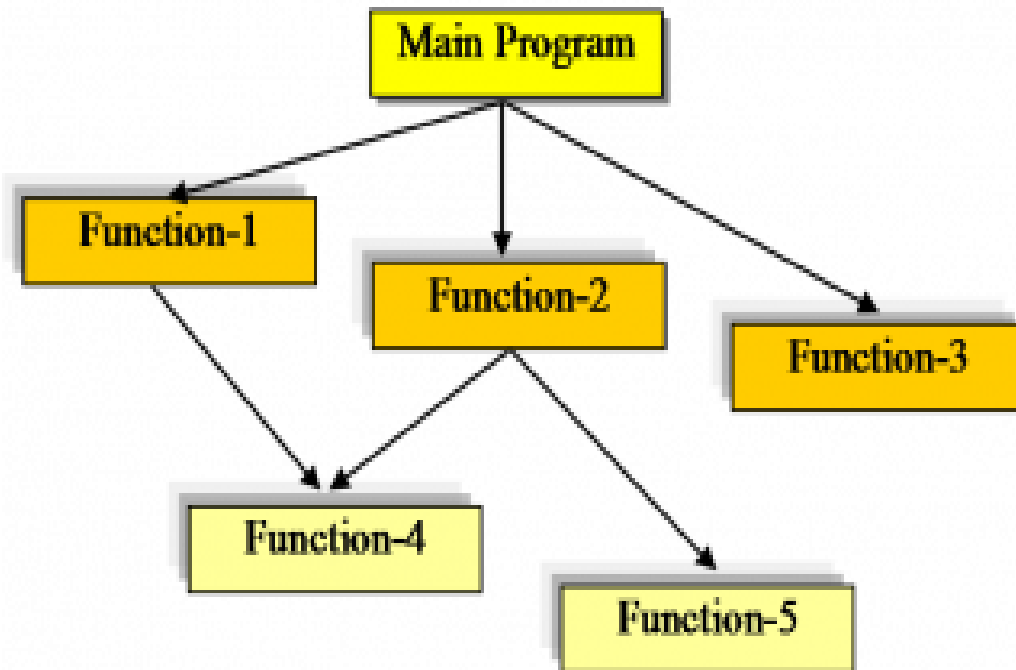
### Thiết kế hệ thống nấu cơm



# Phương pháp Lập trình hướng đối tượng

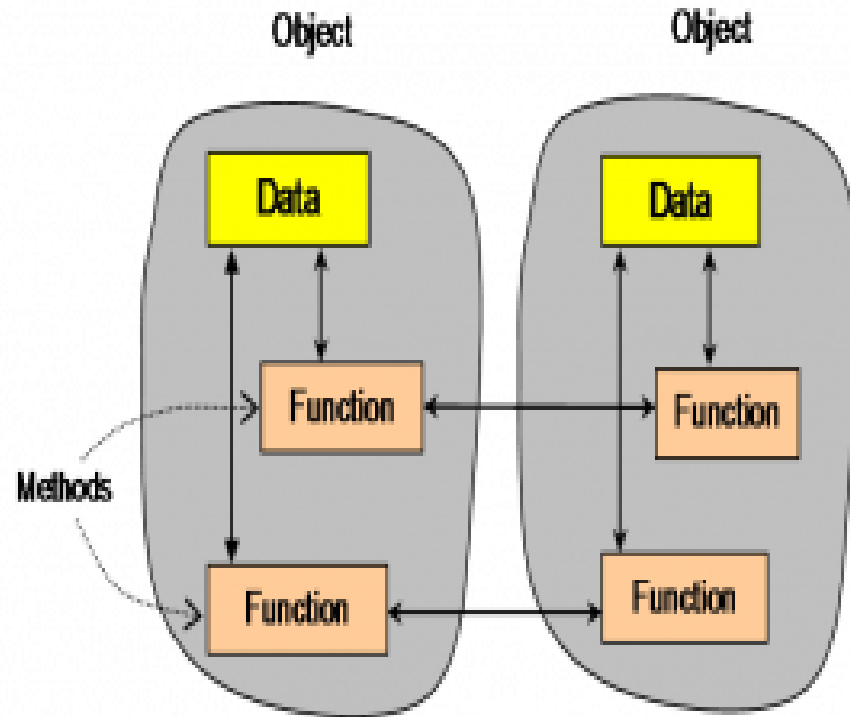
## ⌘ Sự khác biệt OOP với Lập trình thủ tục

Procedure-oriented Programming



Chương trình = Hàm + dữ liệu

Object-oriented Programming



Chương trình = Tập các đối tượng tương tác nhau



# Các khái niệm

## ⌘ Đối tượng

Là sự mô hình hóa cho một đối tượng thực như một con người, một cái bàn, một cái ghế, một con vật,... bao gồm tập các thuộc tính, hành vi và định danh

- **Thuộc tính (attribute)**

Là các giá trị hay trạng thái để mô tả cho đối tượng đó

- **Hành vi/Phương thức (behavior/method)**

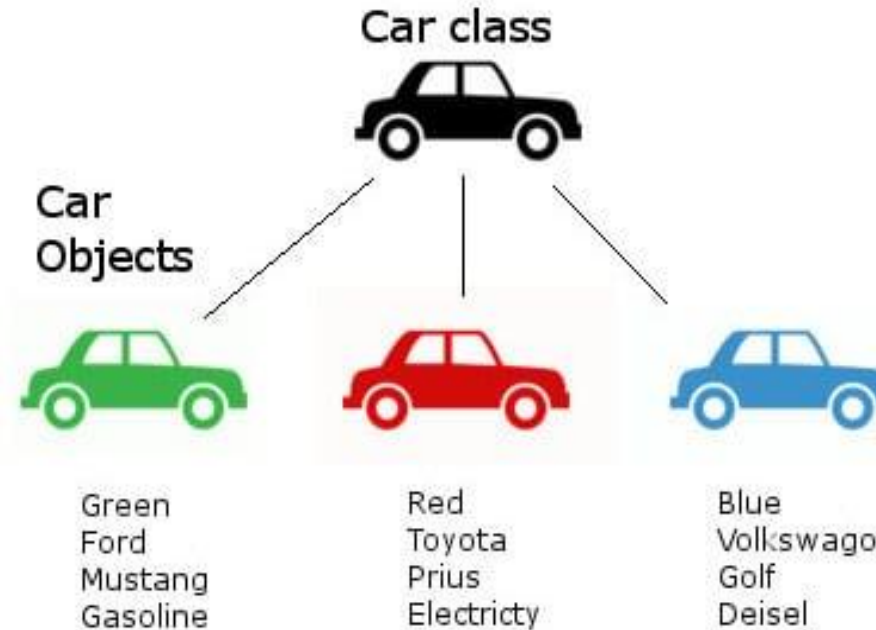
Cung cấp các thao tác/dịch vụ (service) trên các thuộc tính để các đối tượng khác tác động lên đối tượng đó.

- **Định danh đối tượng**

Dùng để xác định một đối tượng trong tập các đối tượng



Red  
Toyota  
Prius  
Electricity



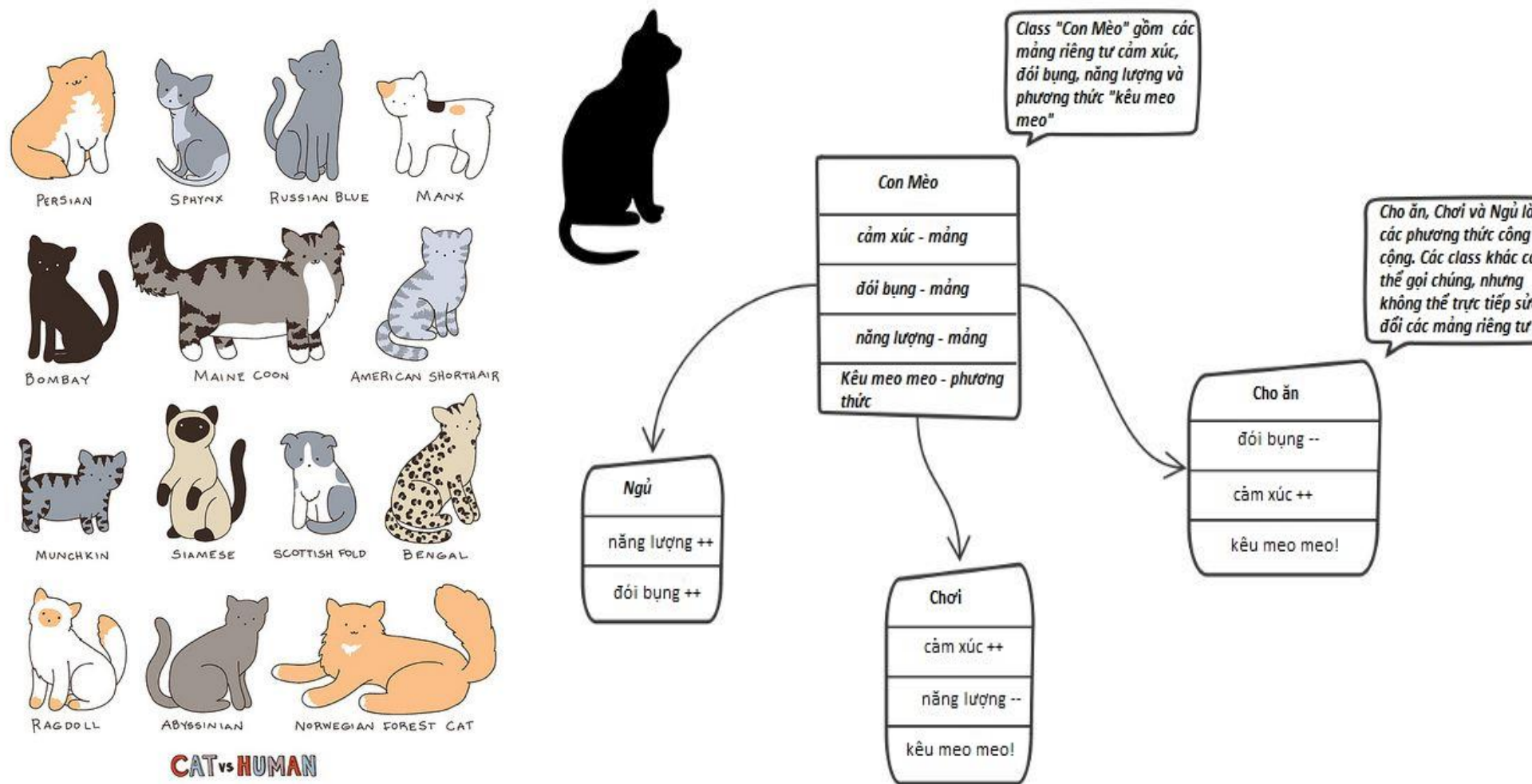
## ⌘ Lớp

Mô tả các thuộc tính và hành vi chung của các đối tượng cùng kiểu



# Các khái niệm

## ⌘ Đối tượng vs Lớp



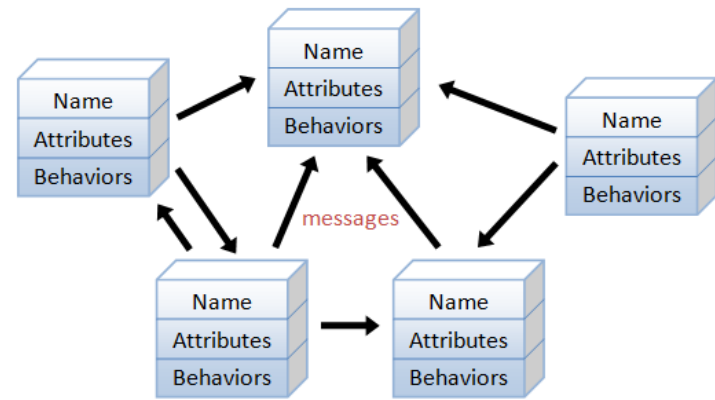
# Các khái niệm

## ⌘ Phạm vi truy cập

- Thiết lập giới hạn truy cập đến các hàm thành viên hoặc các dữ liệu thành viên của lớp đó thông qua các chỉ định truy cập. Có 3 chỉ định truy cập: private, protected và public.

## ⌘ Thông điệp – truyền thông điệp

- Là một yêu cầu thực hiện hành vi, bao gồm tên thông điệp và các thông tin kèm theo thông điệp
- Gửi thông điệp đến 1 đối tượng nào đó nhằm yêu cầu đối tượng thực hiện hành vi tương ứng. Một thao tác truyền thông điệp bao gồm định danh của các đối tượng được yêu cầu thực hiện thông điệp và thông điệp cần gửi



# Các tính chất OOP



## ⌘ Tính trừu tượng

Là sự mô hình hóa của các đối tượng. Quá trình xác định các thuộc tính, phương thức của đối tượng và bỏ qua các thuộc tính không liên quan đến vấn đề cần giải quyết là một quá trình trừu tượng hóa

# Các tính chất OOP

## ⌘ Tính bao gói

- Việc đóng gói dữ liệu và phương thức của một thực thể vào một đối tượng
- Việc giới hạn truy cập đến các thành phần của một đối tượng

## ⌘ Có 2 góc nhìn khác nhau đối với đối tượng:

- Góc nhìn từ bên trong: bao gồm các chi tiết về sự cài đặt của các phương thức và cấu trúc dữ liệu của các thuộc tính và được che dấu khỏi các đối tượng khác. Thường là thành phần private của đối tượng
- Góc nhìn từ bên ngoài: bao gồm các dịch vụ (hay thao tác) của đối tượng mà các đối tượng khác có thể yêu cầu. Thường là thành phần public của đối tượng

# Các tính chất OOP

## ⌘ Tính bao gói

☑ Phạm vi truy cập:

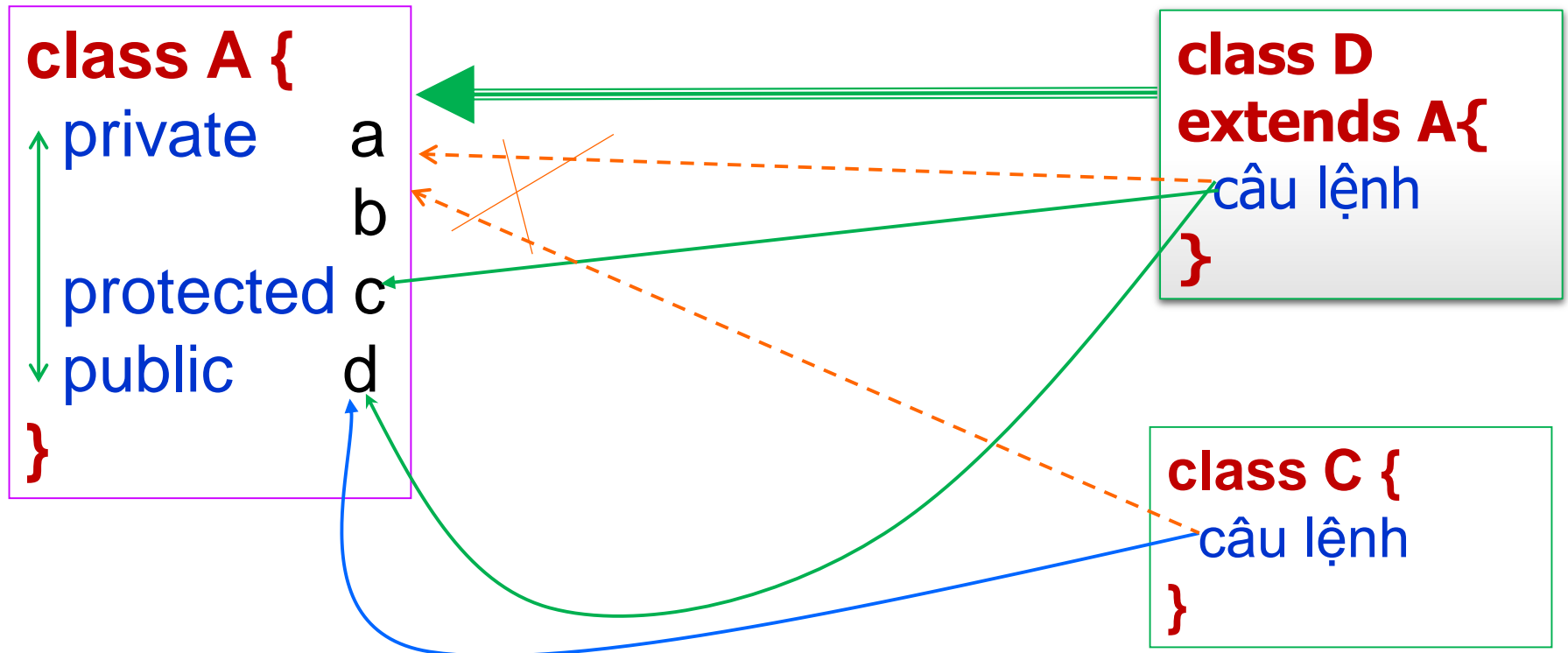
**-private**: chỉ được truy cập trong cùng lớp

**#protected**: không giới hạn trong cùng  
lớp, và lớp con

**+public**: không giới hạn

# Các tính chất OOP

## ⌘ Tính bao gói - Phạm vi truy cập:



# Các tính chất OOP



## ⌘ Tính đa hình

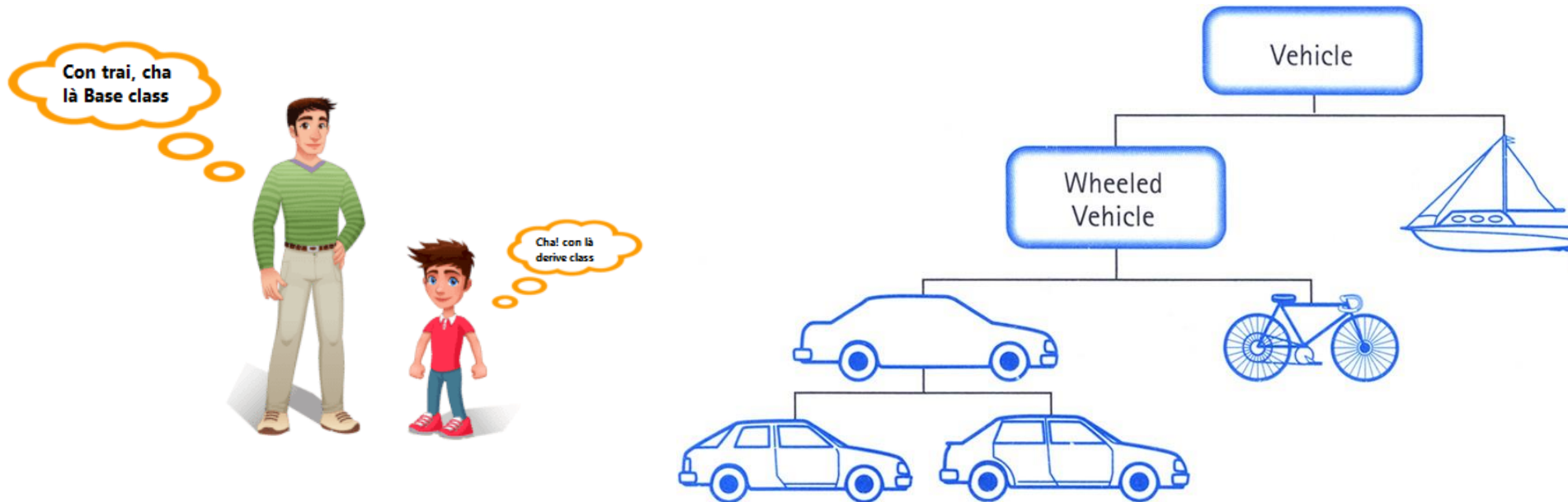
- Cung cấp khả năng gửi một thông điệp đến một nhóm các đối tượng có kiểu khác nhau và thông điệp này sẽ được thực hiện một cách khác nhau tùy vào đối tượng nhận được thông điệp thuộc loại nào



# Các tính chất OOP

## ⌘ Tính thừa kế

- Một lớp con sẽ thừa kế tất cả các thành phần của lớp cha. Tuy nhiên lớp con chỉ được truy xuất các thành phần public và protected của lớp cha.





October 1999