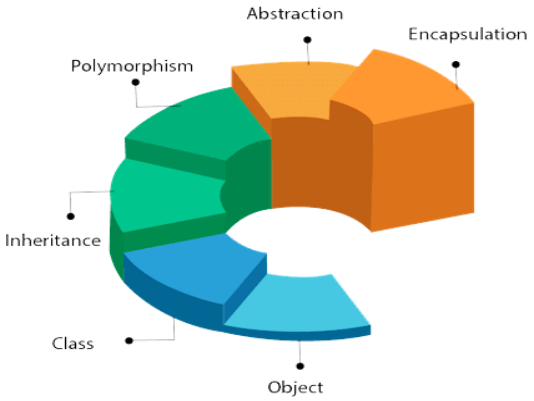
Lập Trình Hướng Đối Tượng – CT176



TS. Phan Thượng Cang
ptcang@cit.ctu.edu.vn
Khoa CNTT&TT - Đại học Cần Thơ

Phần 2

Lý thuyết Lập Trình Hướng Đối Tượn

TS. Phan Thượng Cang

Khoa CNTT&TT-Đại học Cần Thơ

Nội dung

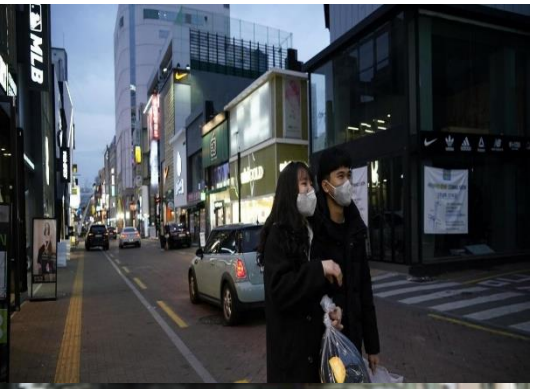
- Phương pháp Lập trình hướng đối tượng (OOP) ?
- Các khái niệm OOP
- Các tính chất quan trọng của OOP

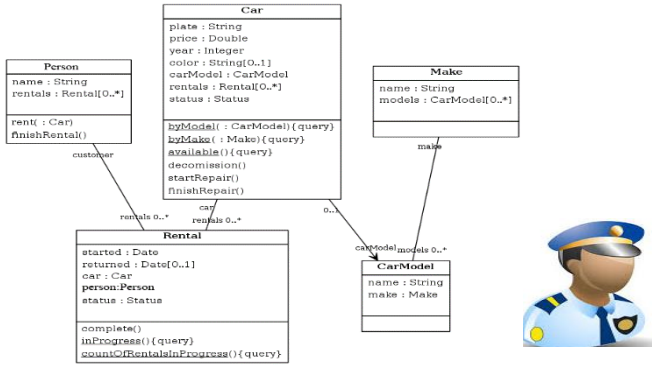
Nội dung

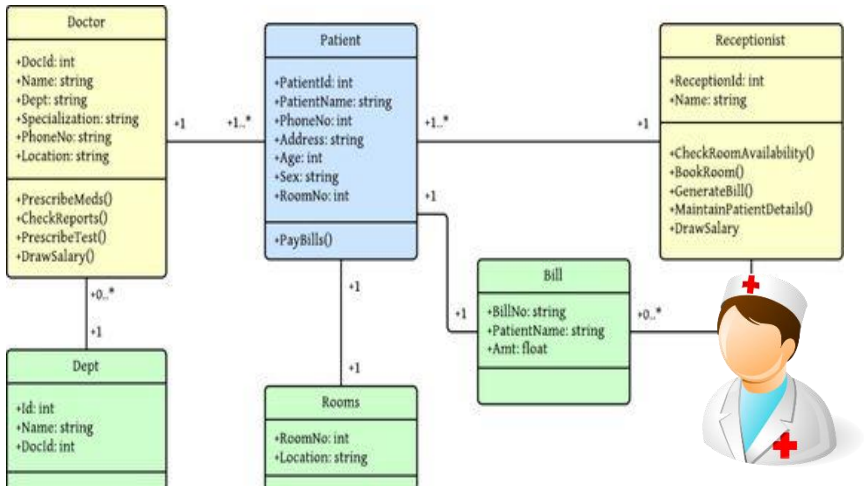
- **Phương pháp Lập trình hướng đối tượng (OOP) ?**
  - Ý tưởng – định nghĩa
  - Sự khác biệt OOP với Lập trình thủ tục - Tại sao nên dùng OOP ?
- **Các khái niệm OOP**
  - Đối tượng vs Lớp
  - Phạm vi truy cập
  - Thông điệp – Truyền thông điệp
- **Các tính chất quan trọng của OOP**
  - Trừu tượng hoá
  - Bao gói
  - Đa hình
  - Thừa kế

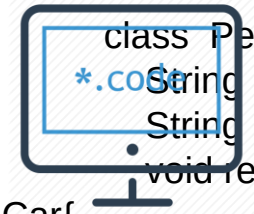
Phương pháp Lập trình hướng đối tượng

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










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

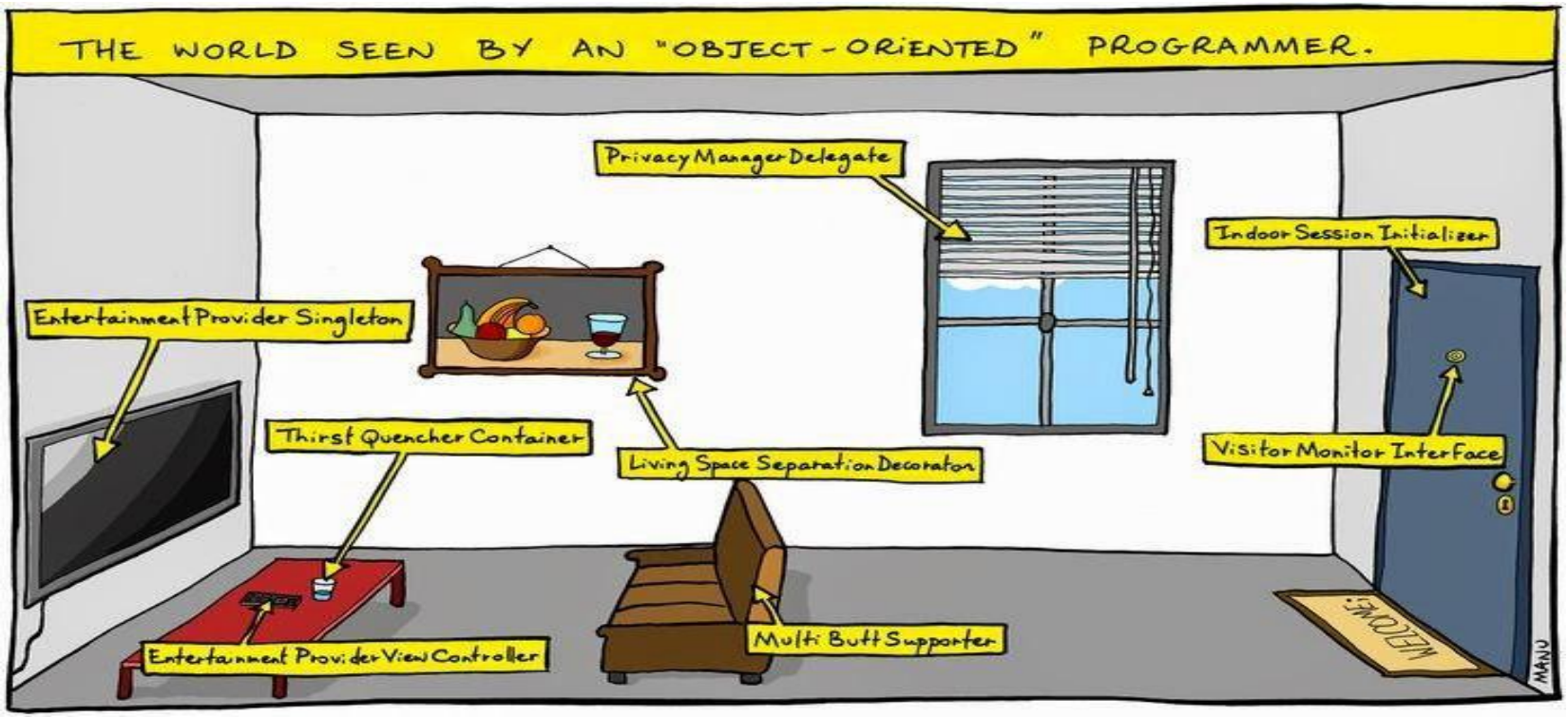
```
class  Car{
   String id;  }
   Person owner;
   void input()
   …….
}
```

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

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

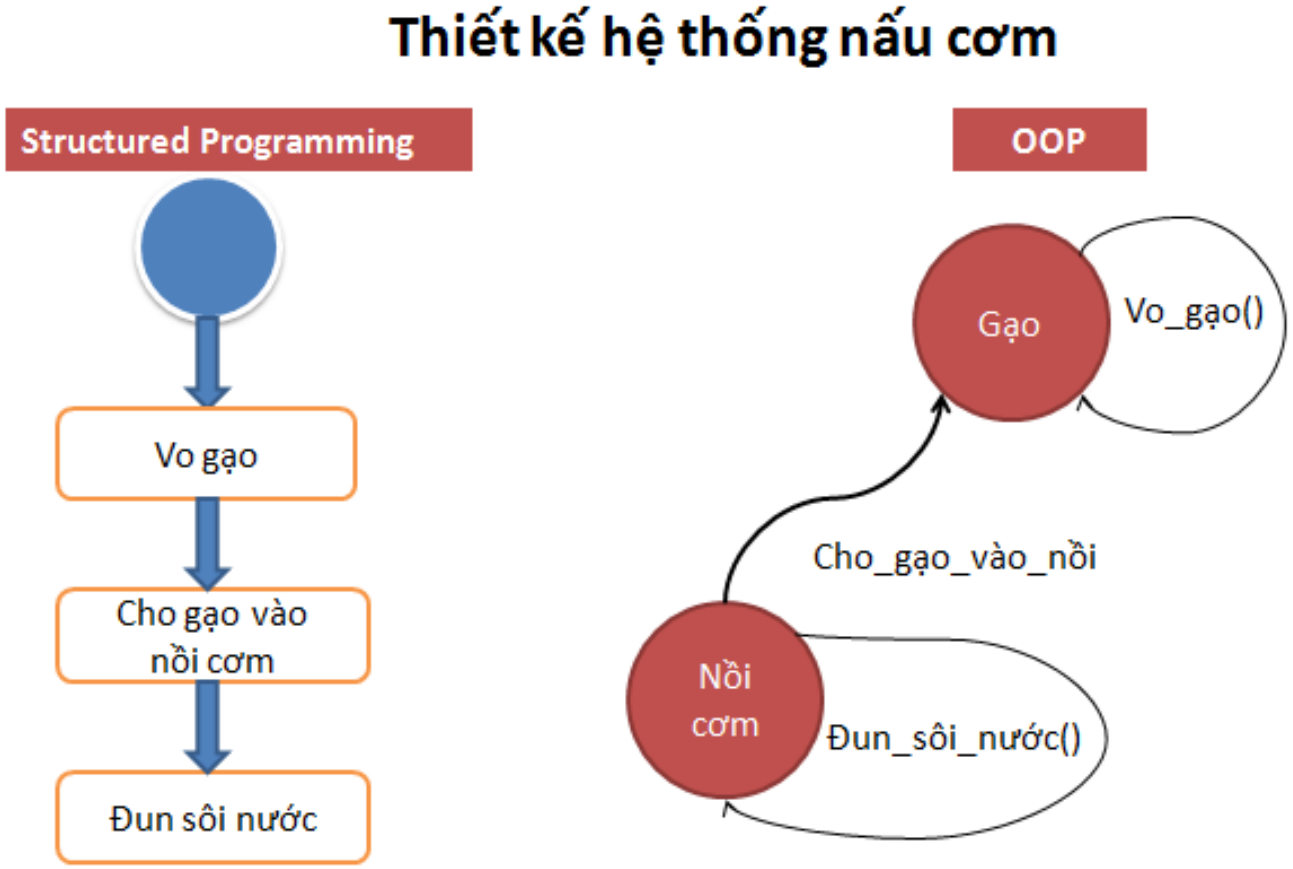
Phương pháp Lập trình hướng đối tượng

- 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 Lập trình thủ tục vs OOP

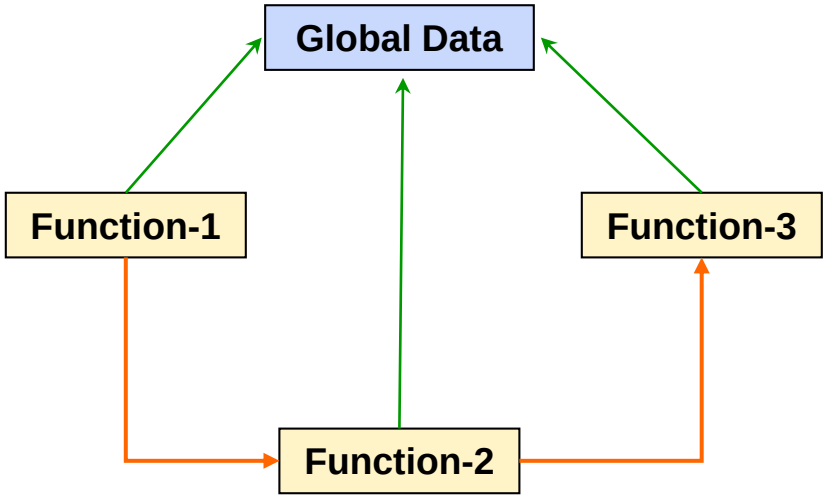


Chương trình = Hàm + dữ liệu

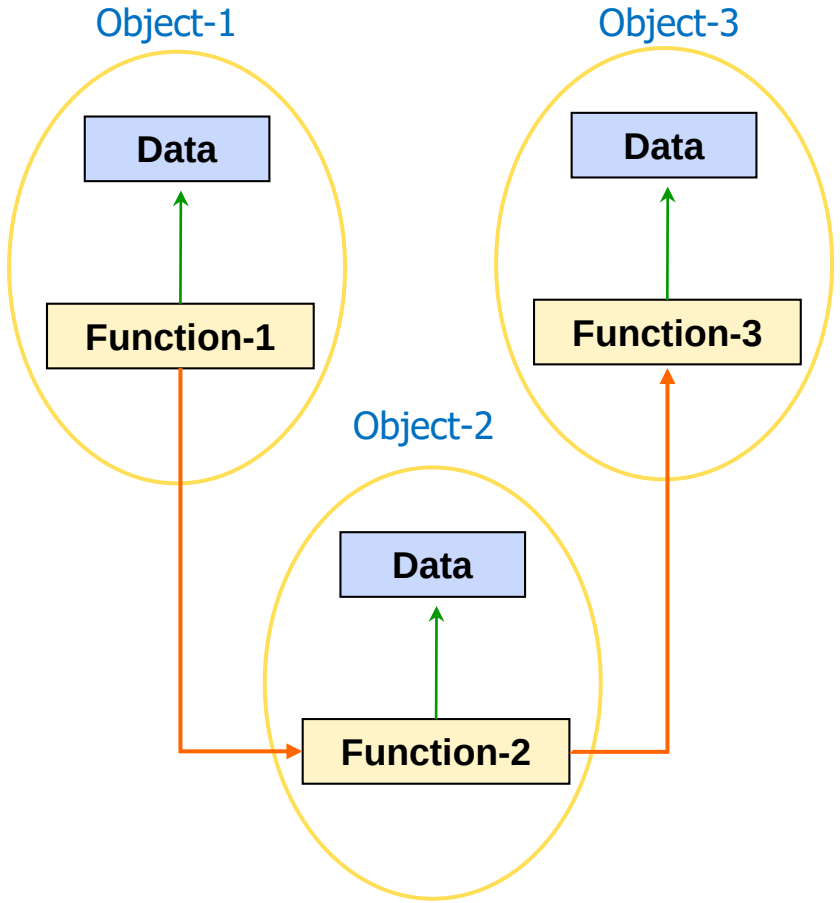
Chương trình = Tập các đối tượng tương tác nhau

Phương pháp Lập trình hướng đối tượng

Lập trình hướng thủ tục (POP)

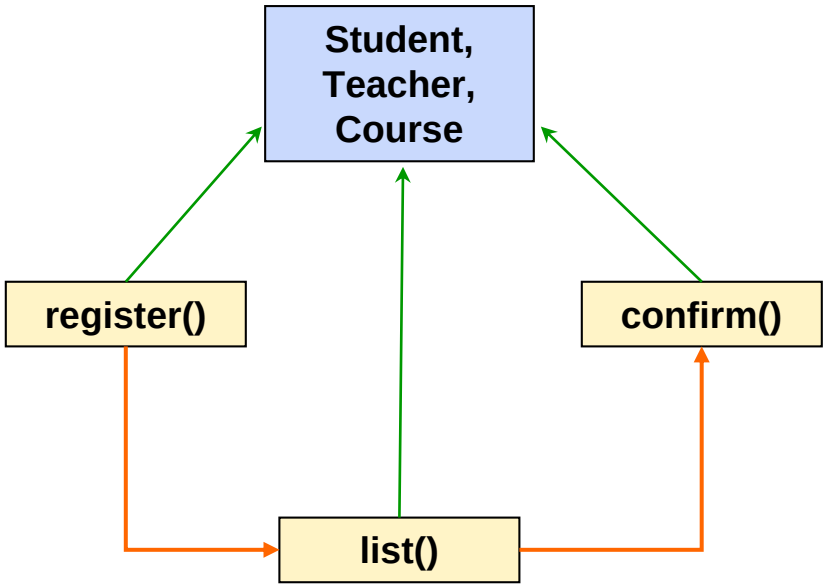


Lập trình hướng đối tượng (OOP)

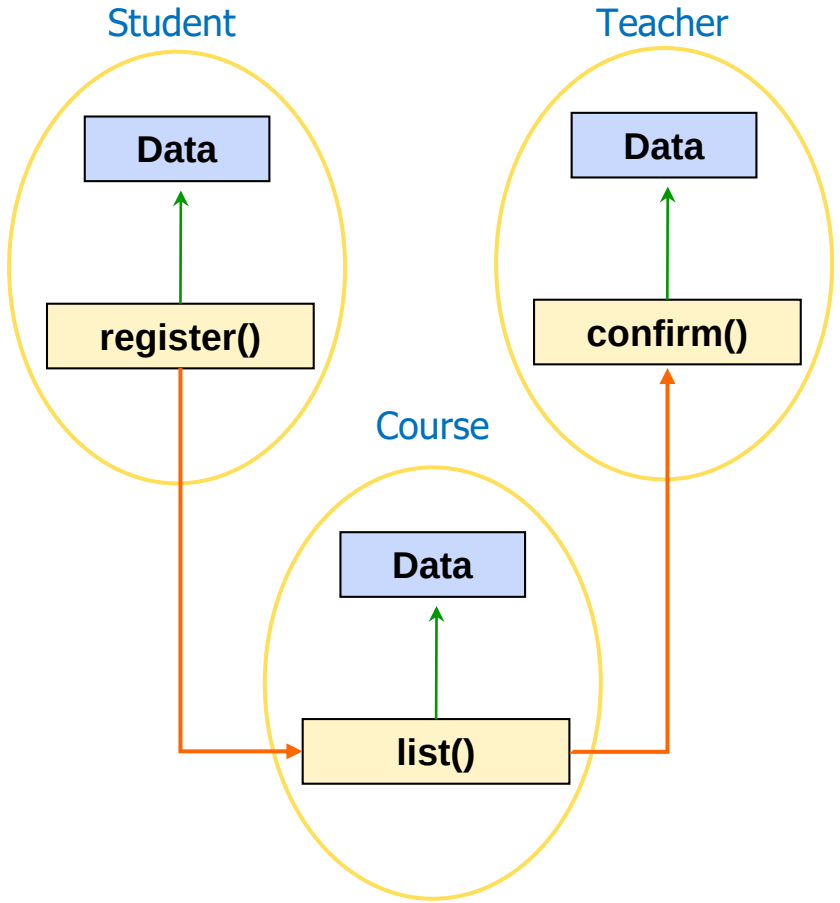


Phương pháp Lập trình hướng đối tượng

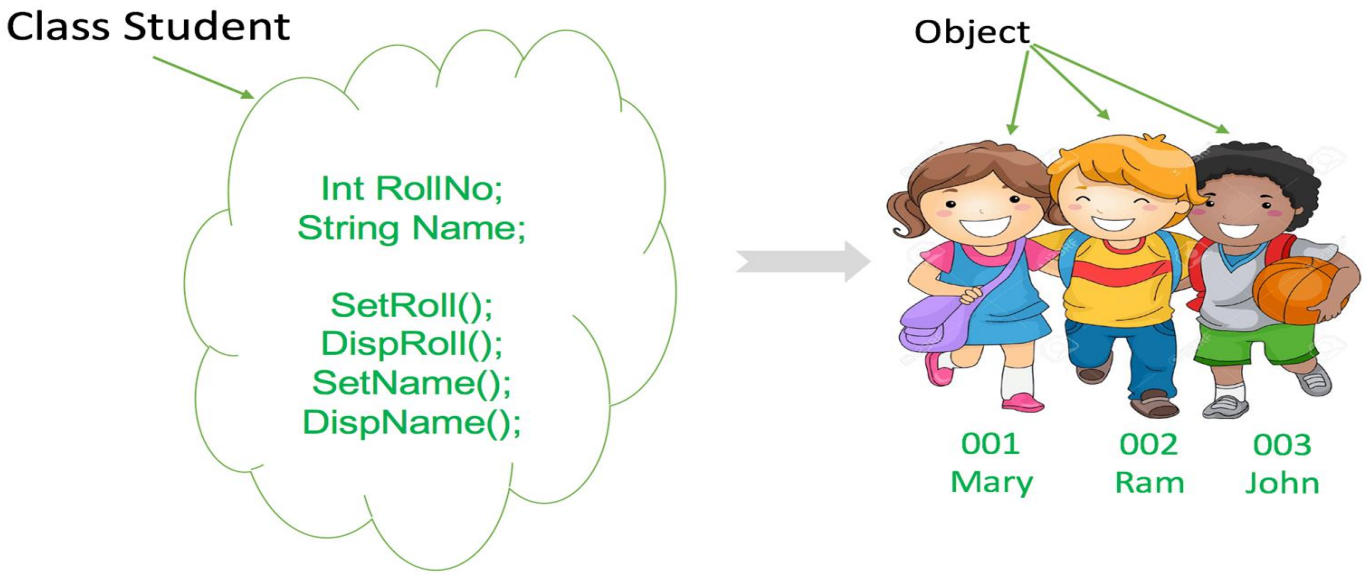
Lập trình hướng thủ tục (POP)



Lập trình hướng đối tượng (OOP)



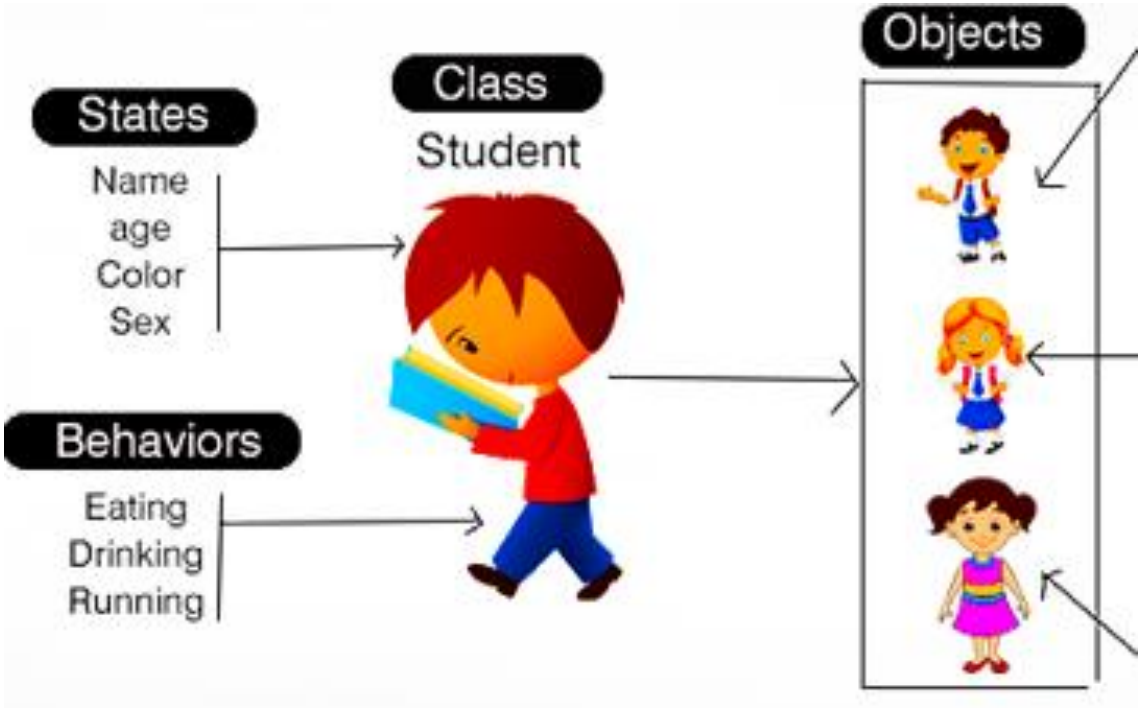
Các khái niệm



- **Lớp (class)**
  - Sự tổng quát hóa các đối tượng có cùng đặc trưng (thuộc tính, phương thức).
  - Một kiểu dữ liệu (Student)
- **Đối tượng (object)**
  - Sự cụ thể hóa (thể hiện) của một lớp bởi việc gán các giá trị cụ thể cho các đặc trưng của một lớp.
  - Một biến (svien1, svien2)

Các khái niệm

- **Thuộc tính** (attribute/data/state)
  Là thành phần dữ liệu để mô tả về một lớp/đối tượng.
  Nó được dùng để lưu trữ trạng thái của một đối tượng tại một thời điểm.
- **Phương thức** (method/function/behavior)
  Là thành phần xử lý trên các thuộc tính của lớp/đối tượng.
  Nó được dùng để các đối tượng khác tác động lên đối tượng đó.

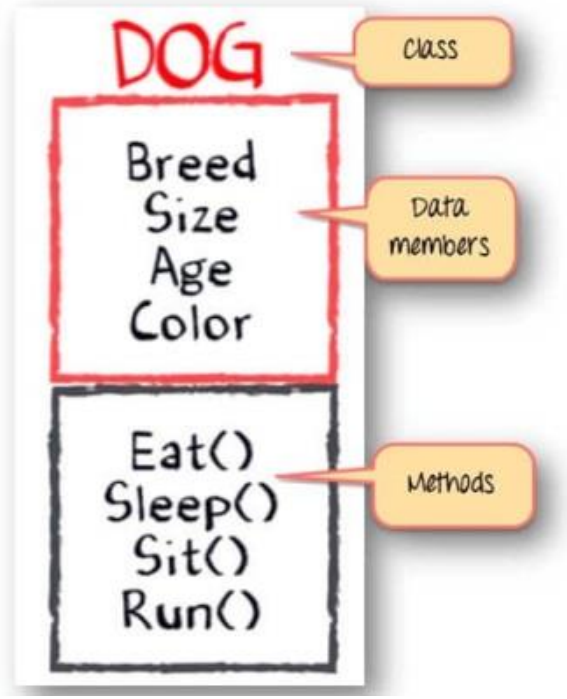


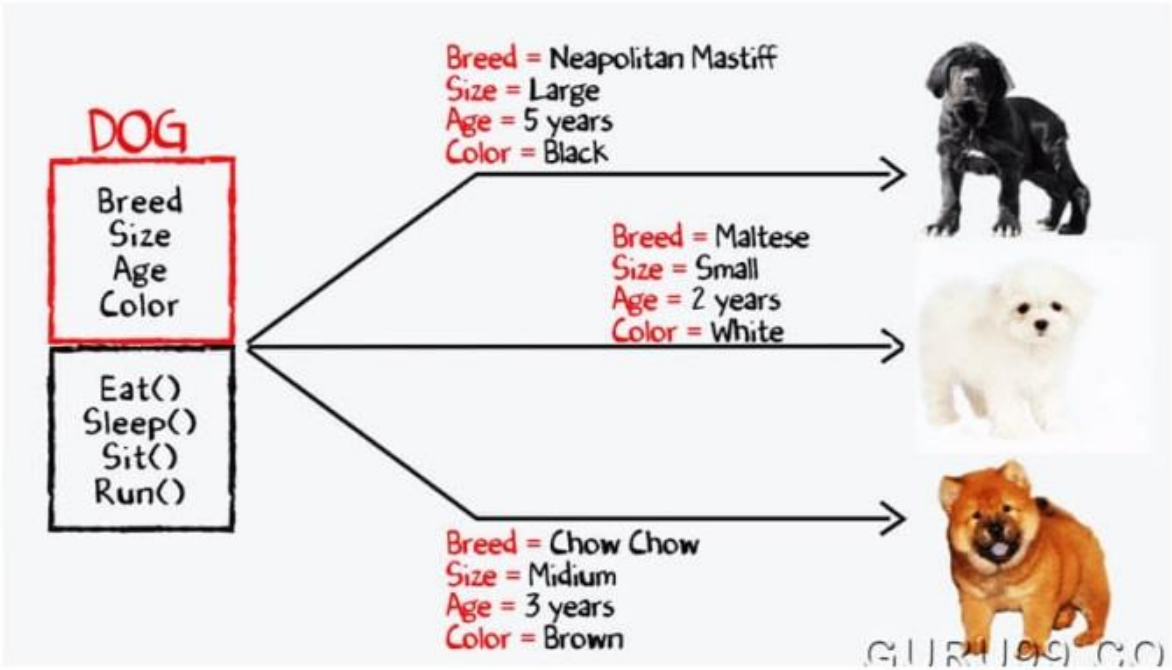
Name: Nguyễn Văn Hải
Age: 20
Color: đen
Sex: Nam
- eat(): Hải ăn nhiều
- drink(): Hải uống nhiều
- run(): Hải chạy chậm

Name: Phan Thị Cẩm Tiên
Age: 19
Color: vàng
Sex: Nữ
- eat(): Tiên ăn ít
- drink(): Tiên uống ít
- run(): Tiên chạy nhanh

Các khái niệm

- **Lớp vs Đối tượng**

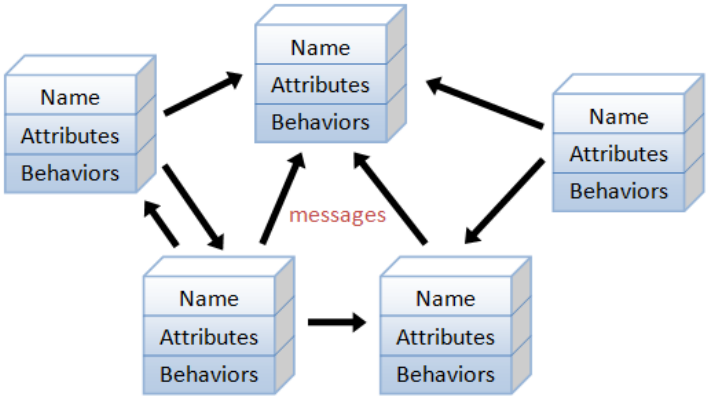




Các khái niệm

- **Phạm vi truy cập** (access modifier)
  - Thiết lập giới hạn truy cập đến các hàm thành viên hoặc các dữ liệu thành vên 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** (message)
  - 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
  - VD: Sudent svien1

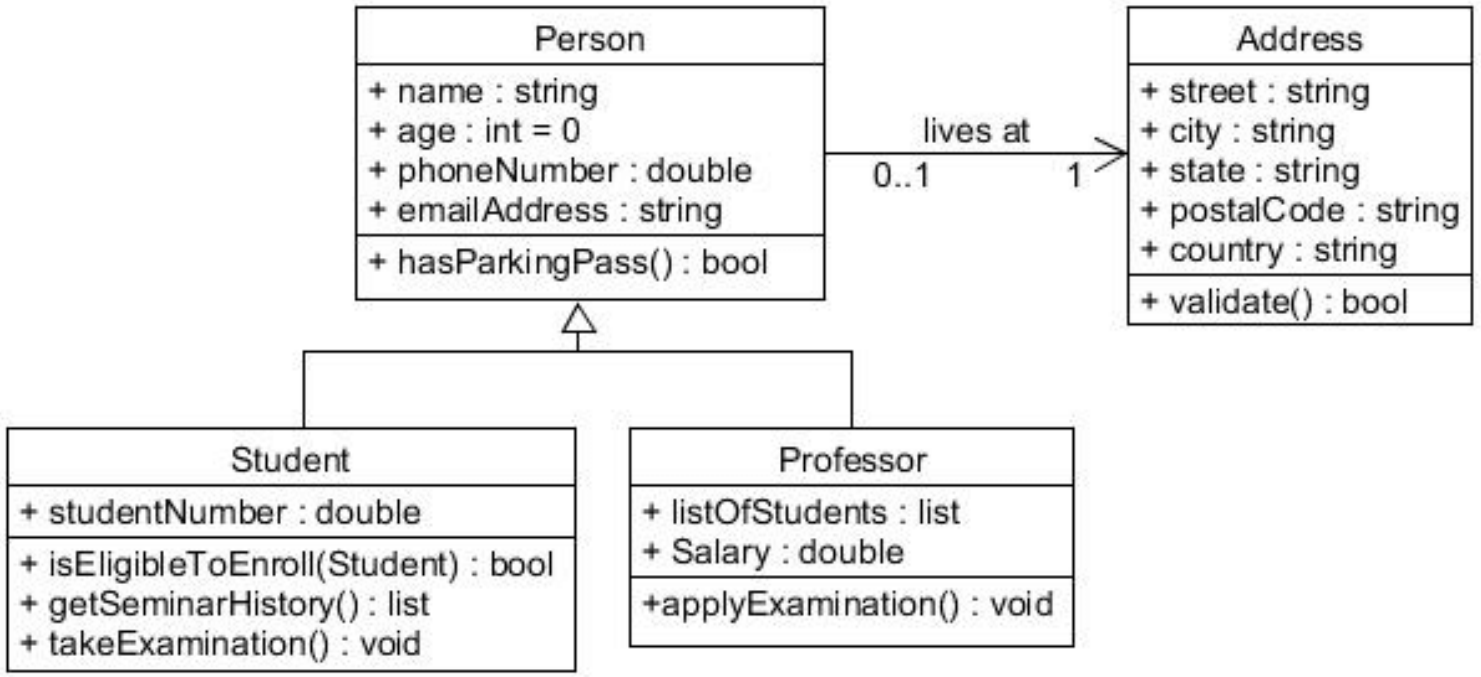
    svien1 . run()



Các tính chất OOP

- **Tính trừu tượng** (Abstraction)

Là sự mô hình hóa của các đối tượng. Quá trình xác định các thuộc tính, phương thức, quan hệ 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

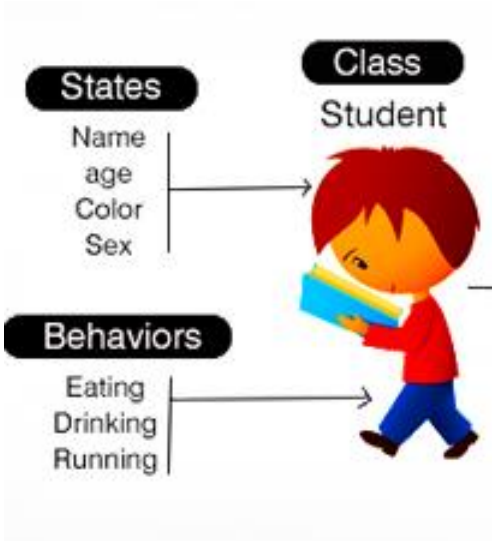


Các tính chất OOP

- **Tính bao gói** (Encapsulation)

  Thể hiện ở 2 khía cạnh :

  - Việc gom nhóm các thuộc tính và phương thức vào một lớp/đối tượng
  - Phạm vi truy cập

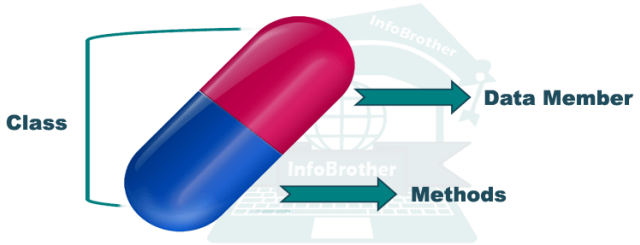


Thành phần bên ngoài

Student

- private:
Name: Nguyễn Văn Hải
Age: 20
Color: đen
Sex: Nam

+ public:
- eat(): Hải ăn nhiều
- drink(): Hải uống nhiều



Các tính chất OOP

- **Tính bao gói**
  - Phạm vi truy cập:

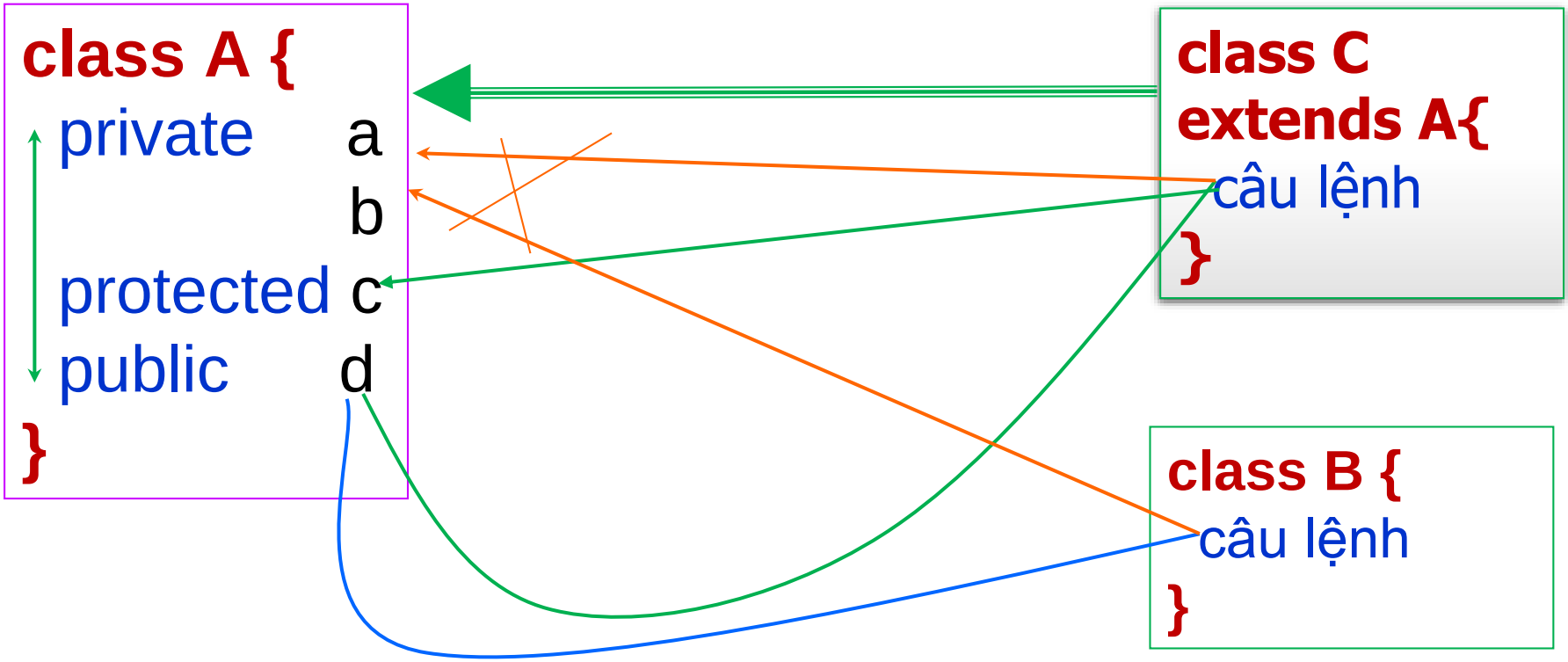
-private: chỉ được truy cập trong cùng lớp

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

+public: không giới hạn

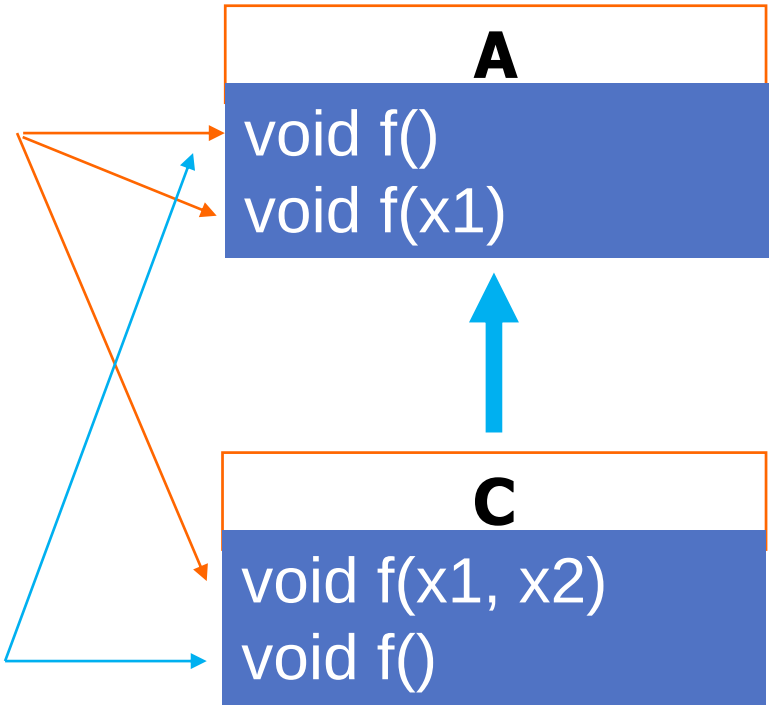
Các tính chất OOP

- **Tính bao gói -** Phạm vi truy cập:



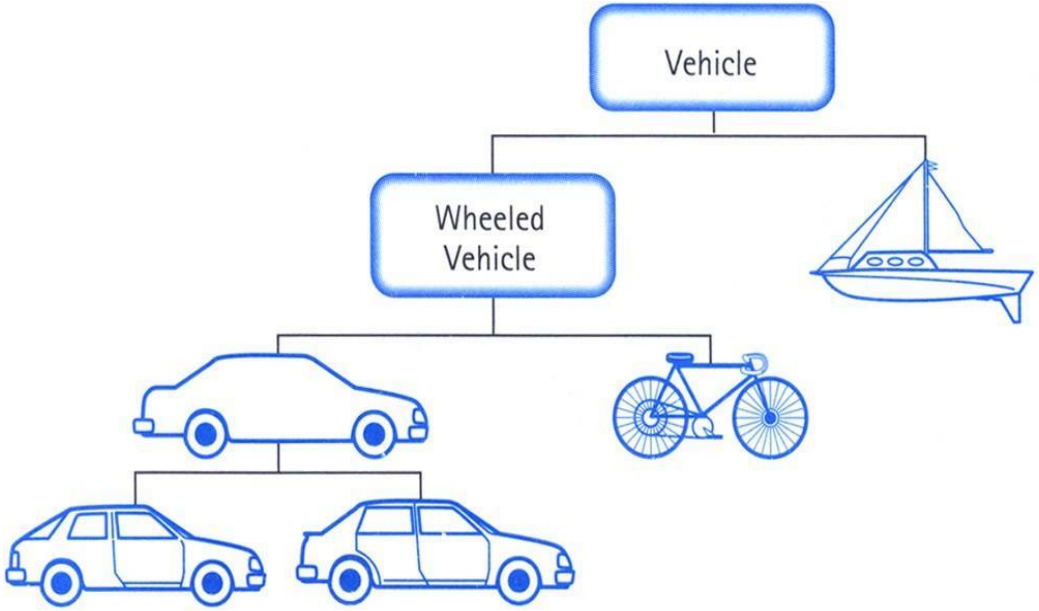
Các tính chất OOP

- **Tính đa hình** (Polymorphism)
  - Một thành phần có ý nghĩa/xử lý khác nhau trong môi trường khác nhau.
  - Có hai cách để thực hiện đa hình một
    - Nạp chồng/tái định nghĩa (Overloading)
      - Cho phép định nghĩa nhiều phương thức trùng tên nhưng khác nhau về tham số (kiểu, thứ tự, số lượng) trong cùng lớp hay lớp con cháu.
    - Nạp đè (Overriding)
      - Cho phép định nghĩa nhiều phương thức trùng tên và trùng đối số nhưng trong lớp con cháu.



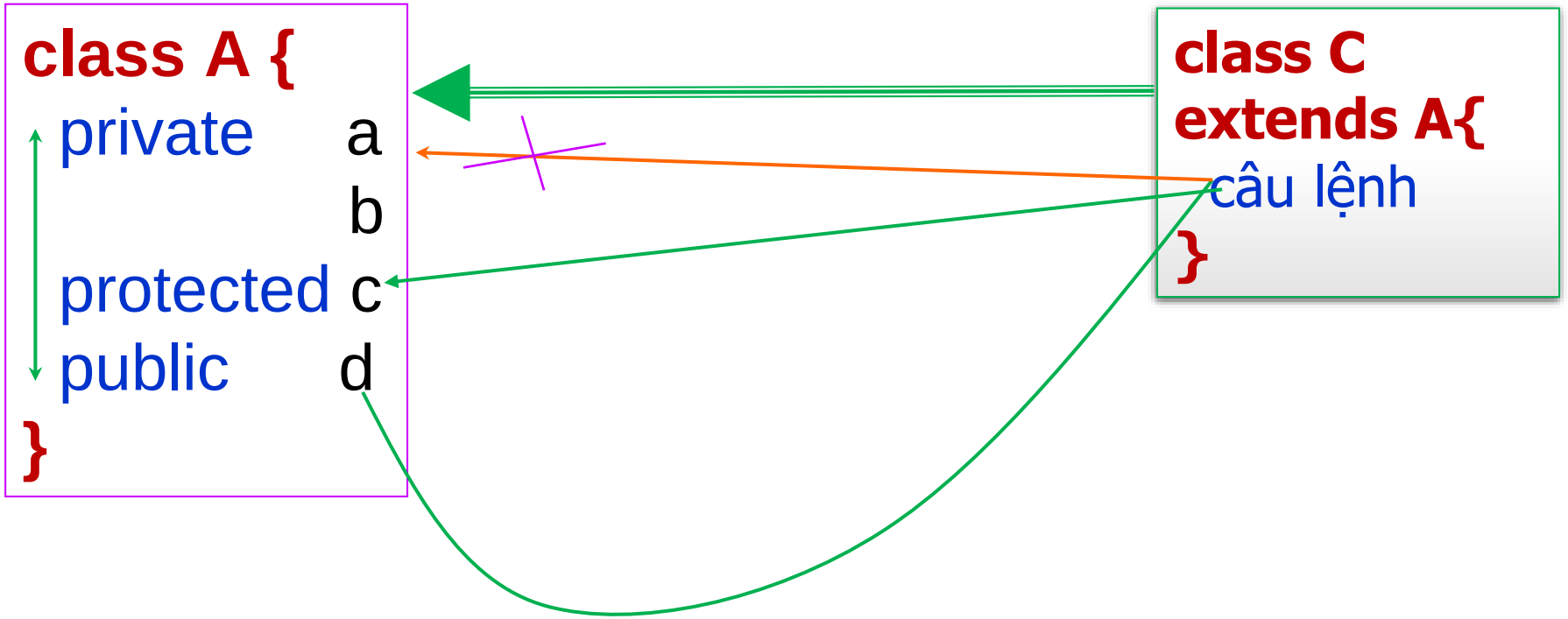
Các tính chất OOP

- **Tính thừa kế** (Inheritance)
  - 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.



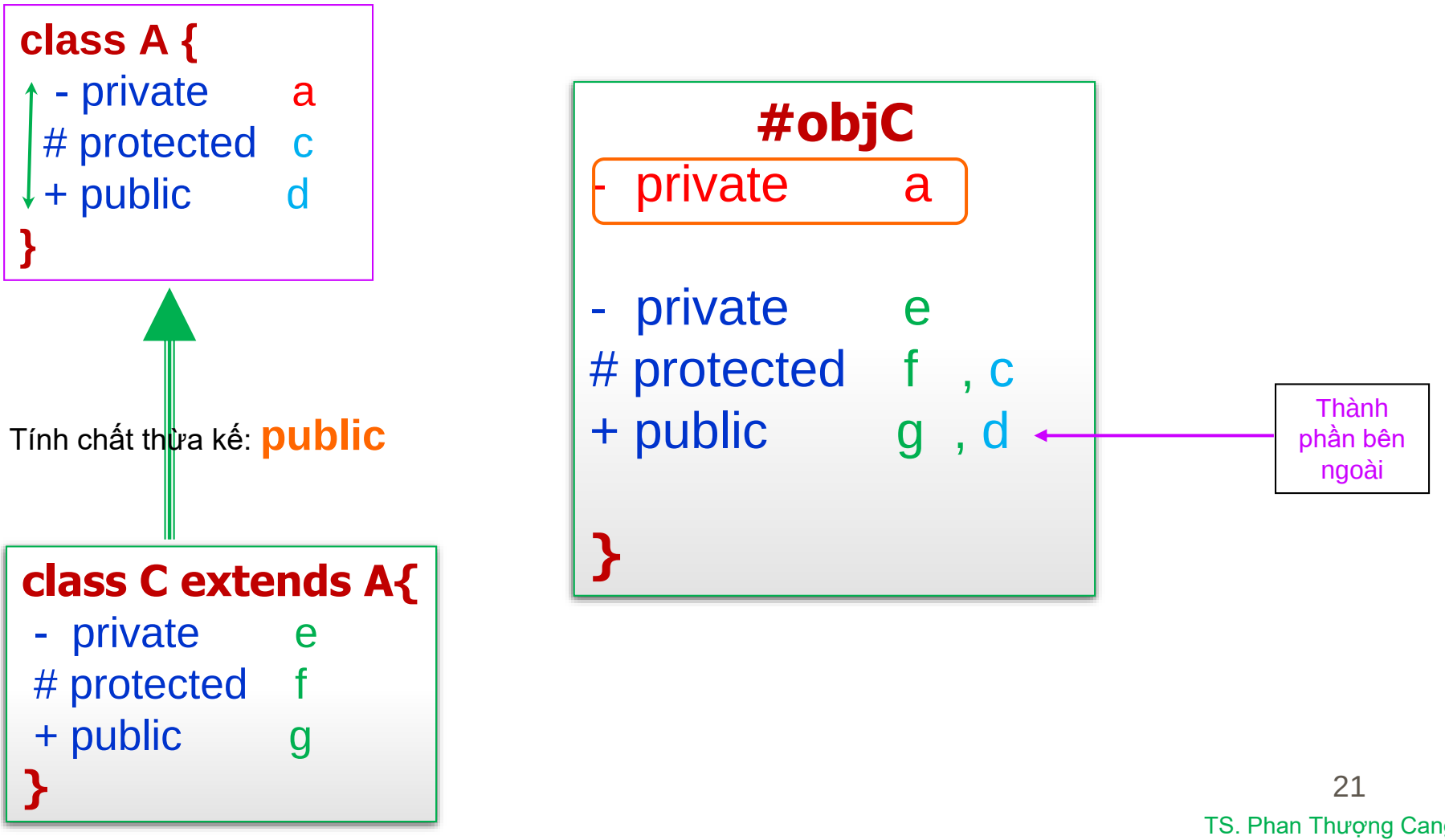
Các tính chất OOP

- **Tính thừa kế -** Phạm vi truy cập:



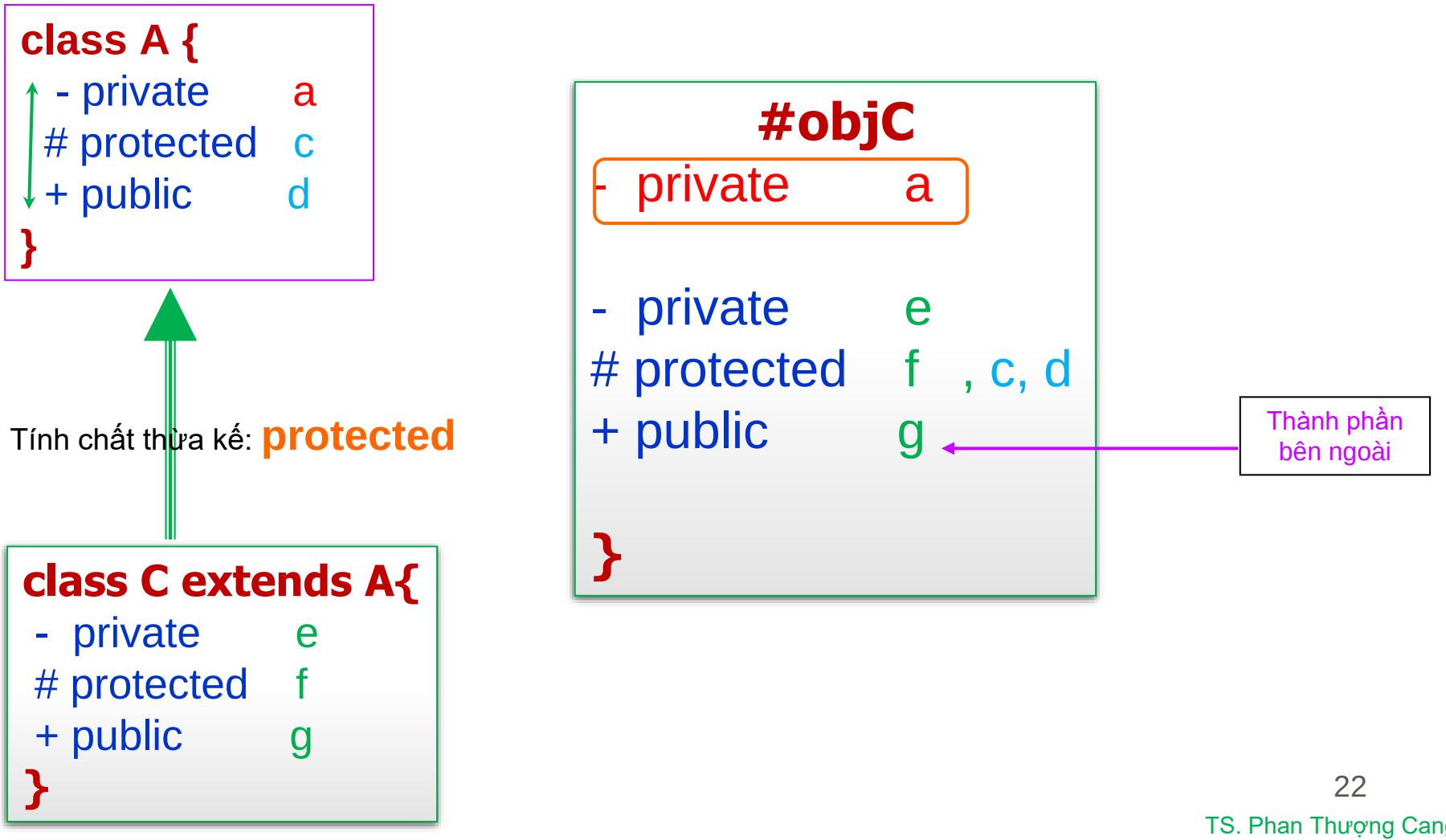
Các tính chất OOP

- **Tính thừa kế -** Phạm vi truy cập



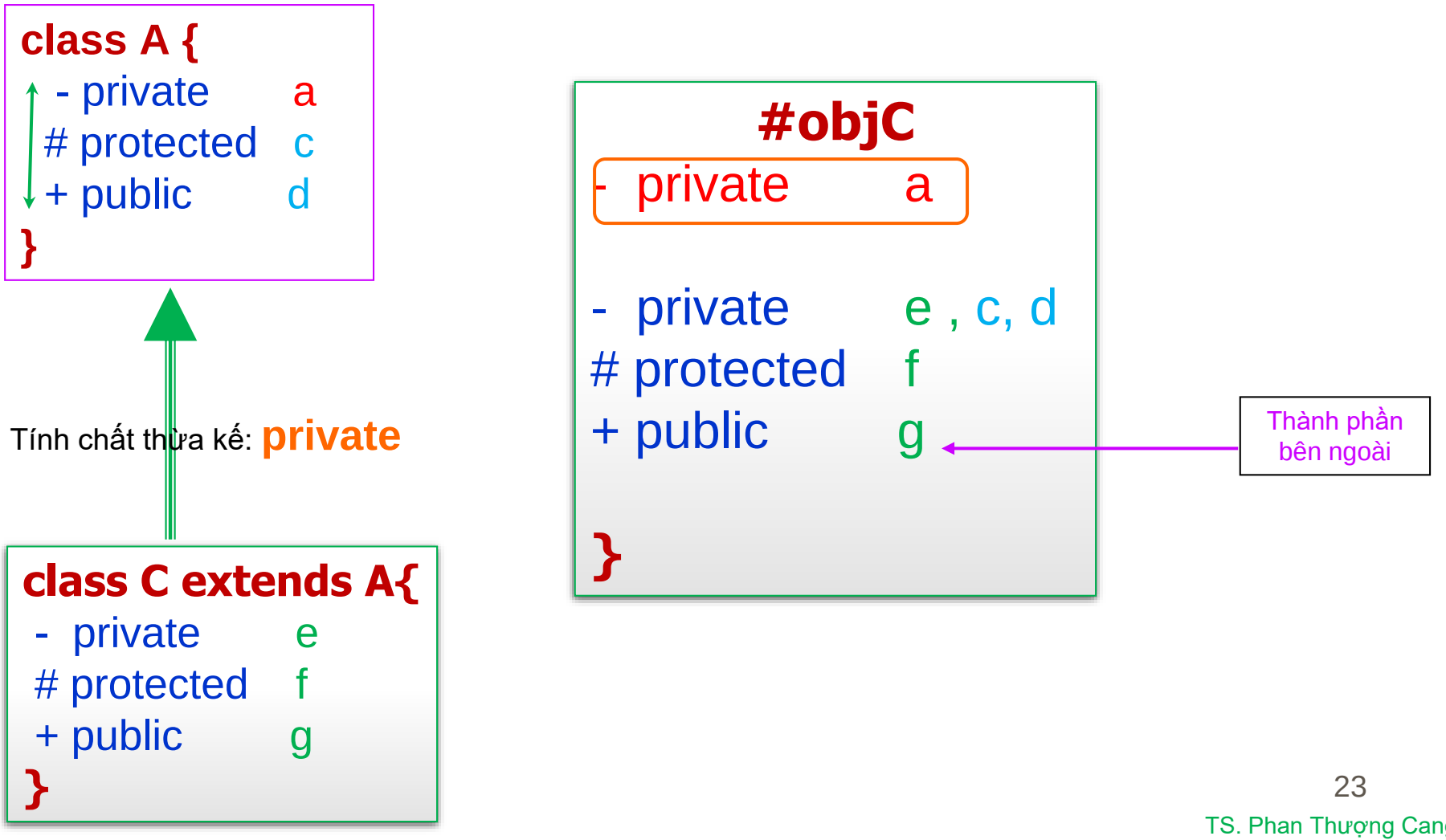
Các tính chất OOP

- **Tính thừa kế -** Phạm vi truy cập



Các tính chất OOP

- **Tính thừa kế -** Phạm vi truy cập



Các tính chất OOP

- **Tính thừa kế - Các dạng**
  - Đơn thừa kế
  - Đa thừa kế