

Lập trình hướng đối tượng

(OOP : Objects-Oriented Programming)

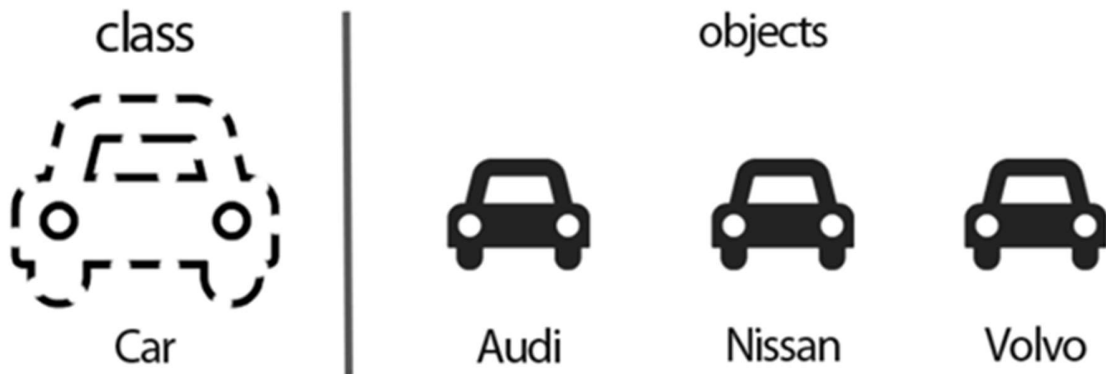
1. Giới thiệu về OOP

- Lập trình hướng đối tượng (OOP) là phương pháp lập trình cho phép lập trình viên tạo ra các đối tượng (objects) thông qua các lớp (class) giúp code dễ phát triển và bảo trì trong tương lai, giúp họ tập trung vào các thực thể giống như trong thực tế hơn. VD: nhà cửa, xe cộ, ...
- Tại sao lại gọi là “hướng đối tượng”: vì nó tập trung vào các đối tượng và lớp, các đối tượng này có thể tương tác với nhau để thực hiện chức năng chương trình

2. Từ hướng thủ tục đến hướng đối tượng

- Nói cho dễ hiểu thì trong lập trình hướng thủ tục là chương trình sẽ được chia thành nhiều hàm con và được gọi trong hàm chính.
- Nhược điểm: Khó bảo trì, bảo mật kém, sử dụng không linh hoạt khi chương trình quá lớn.
- Sự xuất hiện của 2 khái niệm mới là objects và class chính là đặc trưng của phương pháp lập trình hướng đối tượng. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng thủ tục để lại. Đồng thời giúp mô hình hóa thế giới thực vào máy tính dễ dàng hơn.

- Đối tượng (Object): bao gồm 2 thông tin: thuộc tính và phương thức.
 - Thuộc tính (attribute): là các biến được sử dụng để lưu trữ những thông tin, đặc điểm của đối tượng. Họ tên, tuổi, địa chỉ, ...
 - Phương thức (method): chính là các hàm được sử dụng để thực hiện các hành động của đối tượng đó. Ăn, ngủ, đi chơi, ...
- Lớp (Class): là nơi định nghĩa các thông tin trên củ đối tượng, có thể hiểu như là kiểu dữ liệu do chúng ta tự định nghĩa, gần tương tự Struct



3. Định nghĩa lớp và đối tượng

- Cách định nghĩa class gần tương tự như việc định nghĩa struct

```
class ten_lop
{
    Thuoc_tinh_1;
    Thuoc_tinh_2;
    ...
    Thuoc_tinh_n;

    Phuong_thuc_1;
    Phuong_thuc_2;
    ...
    Phuong_thuc_n;
};
```

VD:

```
class SinhVien
{
    // thuoc tinh
    int msv;
    string hoten, gioitinh;

    // phuong thuc
    void lambaitap()
    {
        cout << "Dang lam bai tap";
    }
    void hocbai()
    {
        cout << "Dang hoc bai";
    }
};
```

4. Bốn tính chất quan trọng của OOP

- Tính đóng gói (Encapsulation)
- Tính kế thừa (Inheritance)
- Tính đa hình (Polymorphism)
- Tính trừu tượng (Abstraction)

5. Các thuộc tính truy cập (Access modifier)

- private
- public
- protected

⇒ Là khả năng chỉ định phạm vi truy cập của các thuộc tính và phương thức trong lớp của đối tượng

6. Các phương thức setter, getter

- Sử dụng để thao tác với các thuộc tính của đối tượng thay vì truy xuất trực tiếp bằng dấu “.” ở struct

7. Quan hệ kết tập

- Là khi thuộc tính của lớp này là đối tượng của lớp khác

VD:

```
class DiaChi
{
    private:
        string huyen, thanhpho;
};

class SinhVien
{
    private:
        int msv;
        string hoten;
        DiaChi diachinha;
};
```

8. Sơ đồ lớp

Class Name
Attributes
Methods

Student
-name: String -age: int
+Student() +getInformation(): void +display(): void

