

COMPUTER SCIENCE PROGRAMMING C++

Lesson 19: Tổng quan về OOP trong C++

Nguyễn Văn Hiếu



NỘI DUNG CHÍNH

- Tổng quan về OOP, tầm quan trọng của OOP
- Các khái niệm cơ bản tồn tại trong OOP: class, object, attribute, method, constructor, destructor
- Chương trình demo về việc triển khai OOP để giải quyết bài toán
- Làm bài tập ví dụ định nghĩa class



Nội dung chính

01

MAIN

CONTENTS

Công cụ và môi trường lập trình hướng đối tượng

02

REQUIREMENTS

Nắm được các khái niệm cơ bản của lập trình hướng đối tượng

03

NOTES

Một số chương trình cơ bản demo mẫu hình OOP

04

LEARNING

SKILLS
Sẽ được một chương trình cơ bản ứng dụng OOP



01

INTRODUCTION

OOP – Lập trình hướng đối tượng
là gì ?



OOP in C++

Lập trình hướng đối tượng là một mẫu hình lập trình dựa trên khái niệm "đối tượng".

Đối tượng chứa các dữ liệu:

- Giá trị được gọi là các thuộc tính.
- Hàm nội tại được gọi là các phương thức.



Tác dụng của OOP

Một số ứng dụng thường thấy:

- OOP mô hình hóa những thứ phức tạp dưới dạng cấu trúc đơn giản.
- Code OOP có thể sử dụng lại, giúp tiết kiệm tài nguyên.
- Giúp sửa lỗi dễ dàng hơn. So với việc tìm lỗi ở nhiều vị trí trong code thì tìm lỗi trong các đơn giản và ít mất thời gian hơn.
- Có tính bảo mật cao thông qua đóng gói.
- Dễ mở rộng dự án.



Tại sao phải học OOP



OOP được ứng dụng trong việc triển khai chương trình lớn



Các framework lớn đều cần sử dụng OOP để định nghĩa



Ứng dụng trong nghề nghiệp và phân tích chức năng chương trình rất nhiều





Trăm hay không bằng tay quen



02

Các khái niệm trong OOP

Các khái niệm cơ bản trong OOP:

- Class, object
- Attributes, method
- Constructor, destructor

Chương trình demo

```
#include <iostream> // input - output - stream
using namespace std;
class Person
{
private:
    // Attributes
    string name;
    int age;

public:
    Person() {} // Default constructor
    Person(string name, int age) // Custom constructor
    {
        this->name = name;
        this->age = age;
    }
    ~Person() {} // Destructor
    // Methods
    void NhapThongTin(){}
    void InThongTin(){}
};

int main()
{
    Person p = Person("nam", 20);
    p.
    re InThongTin inline void Person::InThongTin()
    NhapThongTin
    ~Person
4 UTF-8 C
```

House

Chương trình demo

Class : lớp

```
#include <iostream> // input - output - stream
using namespace std;
class Person
{
private:
    // Attributes
    string name;
    int age;
```

Attributes : thuộc tính

Constructor : hàm tạo

```
public:
    Person() {} // Default constructor
    Person(string name, int age) // Custom constructor
    {
        this->name = name;
        this->age = age;
    }
    ~Person() {} // Destructor
    // Methods
    void NhapThongTin(){}
    void InThongTin(){}
};
```

Methods : phương thức

Destructor : hàm hủy

Object : đối tượng

```
int main()
{
    Person p = Person("nam", 20);
    p.
    re InThongTin inline void Person::InThongTin()
    NhapThongTin
    ~Person
4 UTF-8 C
```

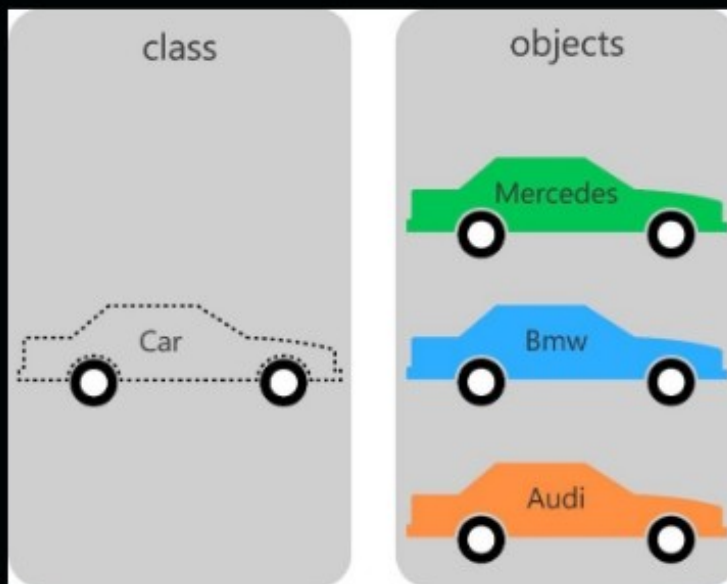


FullHouse

Class

Lớp là sự trừu tượng hóa của đối tượng
Lớp được coi là bản thiết kế để tạo ra một đối tượng

Những đối tượng có những đặc tính tương tự nhau sẽ được tập hợp thành một lớp



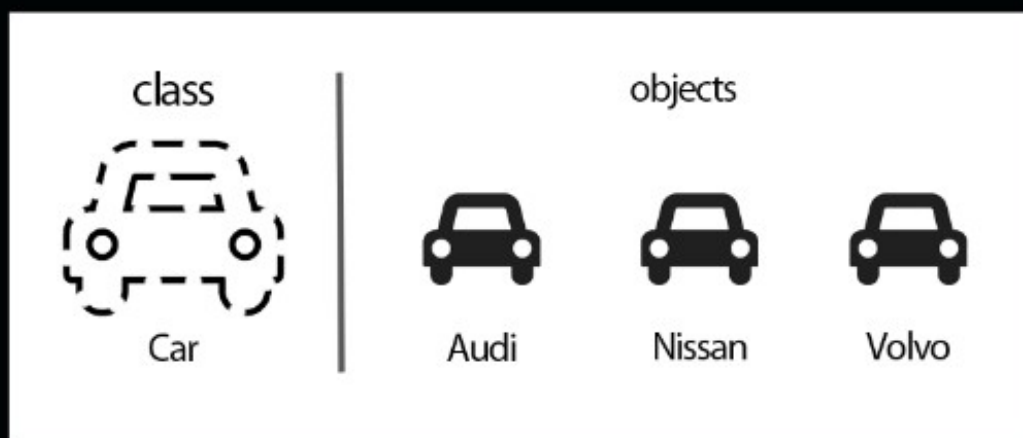
Lớp bao gồm 2 thành phần:

- Thuộc tính
- Phương thức



Object

Đối tượng là một thực thể được sinh ra từ một lớp có sẵn



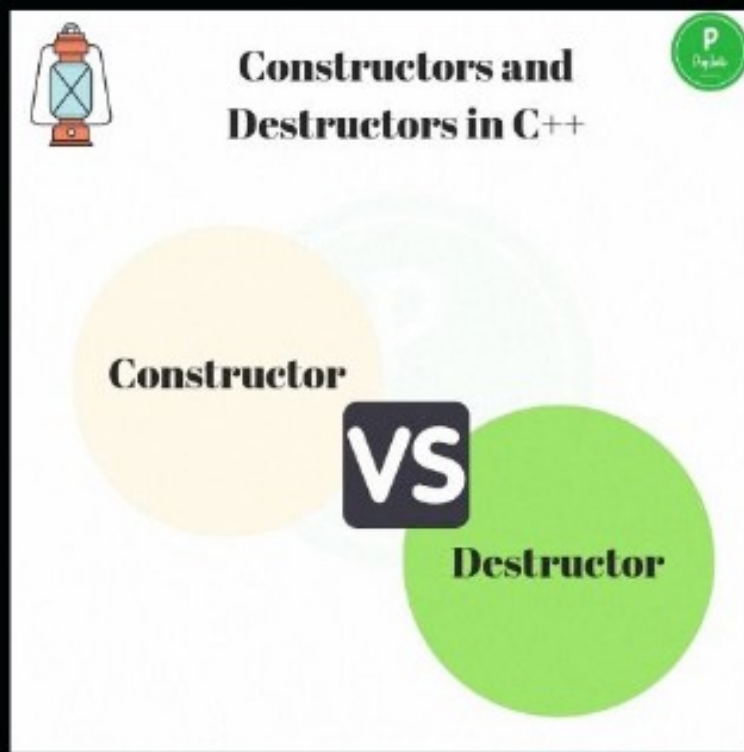
Đối tượng trong OOP bao gồm 2 thành phần chính:

- Thuộc tính (Attribute): là những thông tin, đặc điểm của đối tượng
- Phương thức (Method): là những hành vi mà đối tượng có thể thực hiện



Constructor

Constructor là hàm đặc biệt được sử dụng để khởi tạo giá trị cho object



Đặc điểm của constructor:

- Không có kiểu trả về, không tồn tại return
- Tên hàm trùng với tên của class



Một số bài tập ứng dụng





Thank

Cảm ơn tất cả các bạn đã theo dõi