

Học lập trình Java cơ bản



Bài 7: Lập trình hướng đối tượng

Hanoi 2022

NỘI DUNG CHÍNH



**A. 3 hiểu biết về lập trình
hướng đối tượng**



B. 4 Tính Chất

Phần 1

3 hiểu biết về lập trình hướng đối tượng

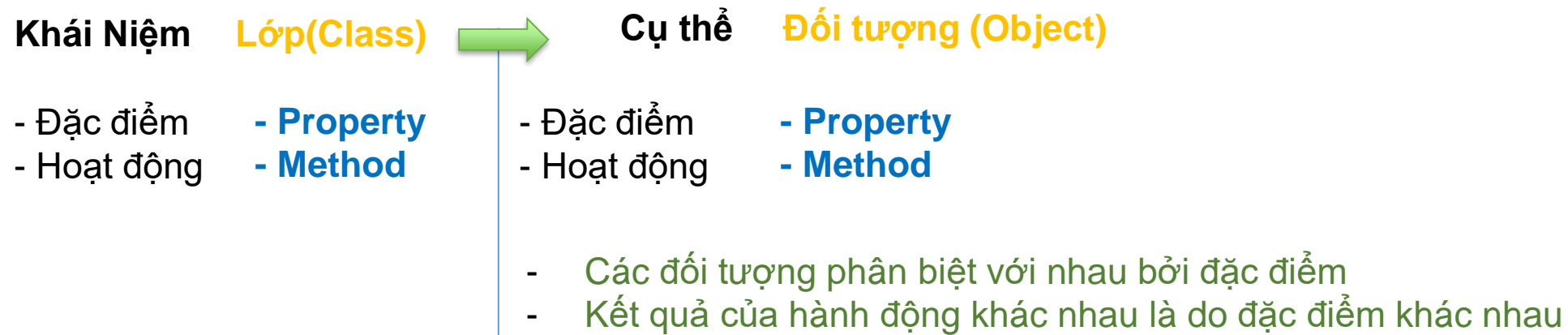
1. Lớp – Đối Tượng
2. Phạm vi truy cập
3. Phương Thức Thuộc Tính



1. Lớp – đối tượng

Lập trình hướng đối tượng (OOP)

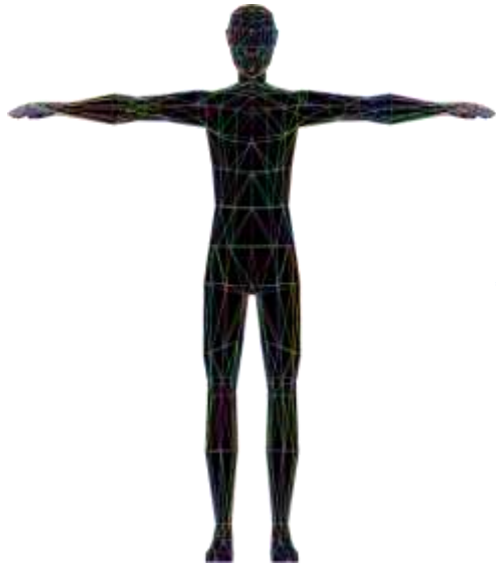
- Chỉ là một trong hơn 50 hình mẫu lập trình khác nhau (các hình mẫu khác như là Lập trình hàm, LT meta, LT tác tử, LT hướng mảng, LT hướng sự kiện, LT nguyên mẫu ... vv)
- OOP khá phổ dụng trong giới outsourcing và phát triển sản phẩm hiện thời



1. Lớp – đối tượng

Khái niệm: Con Người

- Đặc điểm: Khuôn Mặt, Họ Tên, Giới Tính, Năm Sinh ...
- Hoạt Động: Ăn, Chơi, Chạy, Nhảy...



Cụ Thể: Anh Nguyễn Văn A,
Nam, 1980
Hoạt Động: Ăn, Chơi, Chạy, Nhảy...



Cụ Thể: Chị Phạm Thị B, Nữ, 1990
Hoạt Động: Ăn, Chơi, Chạy, Nhảy...

2. Phạm vi truy cập

Phạm vi truy cập (access modifier): để thể hiện khả năng gọi (truy cập) đến các thành phần bên trong lớp/đối tượng. Phụ thuộc vào ngôn ngữ mà có những phạm vi truy cập khác nhau.

Phạm vi truy cập	Ý nghĩa
Public	Không hạn chế phạm vi, thành phần mang thuộc tính này có thể được truy cập từ bất kì đâu .
Private	Thành phần mang thuộc tính này là thành phần riêng tư, chỉ nội bộ bên trong lớp chứa nó mới có quyền truy cập.
Protected	Tương tự như private , ngoài ra còn có thể truy cập từ lớp dẫn xuất lớp chứa nó.
Default	Truy cập được từ trong lớp khai báo và các lớp cùng gói với lớp khai báo.

2. Phạm vi truy cập

Java

```
class Tên_Lớp
{
    Phạm_vi_truy_cập Phương_thức
    Phạm_vi_truy_cập Thuộc_tính
}
```

Cùng xây dựng ví dụ sau

Xây dựng lớp Người gồm các thuộc tính :

- HoTen kiểu string → thể hiện họ tên đầy đủ của một người
- Tuổi kiểu int → thể hiện họ của một người

Phương thức:

- Hiện Thị không trả về → dùng để hiển thị lại toàn bộ các thuộc tính của một người.

3. Phương thức, thuộc tính

Thuộc tính
Thể hiện tính chất các đối tượng
Khai báo giống như biến
Có thể là một đối tượng thuộc về lớp khác
Trong một số ngôn ngữ như C#, VB, JS có hỗ trợ trực tiếp tính năng get-set để mô tả thuộc tính. Còn C,C++,Java không hỗ trợ, mà biến thể thành việc sử dụng các hàm có tiền tố get, set

Phương thức
Thể hiện hành vi của các đối tượng
Khai báo giống như hàm
Hàm tạo (constructor) là phương thức đặc biệt để khởi tạo ngay khi khai báo đối tượng, hàm tạo không có tên và có kiểu trả về trùng với tên lớp. Hàm tạo thường được dùng để khởi tạo các giá trị thuộc tính quan trọng cho lớp.
Tính Overloading : một lớp có thể có nhiều hàm trùng tên (bao gồm cả hàm tạo), miễn sao khác nhau tham số truyền
Ta cũng có thể định nghĩa lại cho các toán tử thông qua overloading operator

Phần 2

4 tính chất trong lập trình hướng đối tượng

1. Tính đóng gói
2. Tính kế thừa
3. Tính trừu tượng
4. Tính đa hình



Tổng quan 4 tính chất

Abstraction:

Modeling properties or method of entities as classes to represent the abstraction of some system or feature
Mô hình hóa các thuộc tính phương thức của các thực thể dưới dạng các lớp để thể hiện một cách trừu tượng của một hệ thống hoặc tính năng nào đó

Encapsulation:

Hidding internal state and functionality of an object and only allowing outsider access through public method
Che giấu đi trạng thái và tính năng bên trong đối tượng và chỉ cho phép bên ngoài truy cập thông qua các phương thức public

Inheritance:

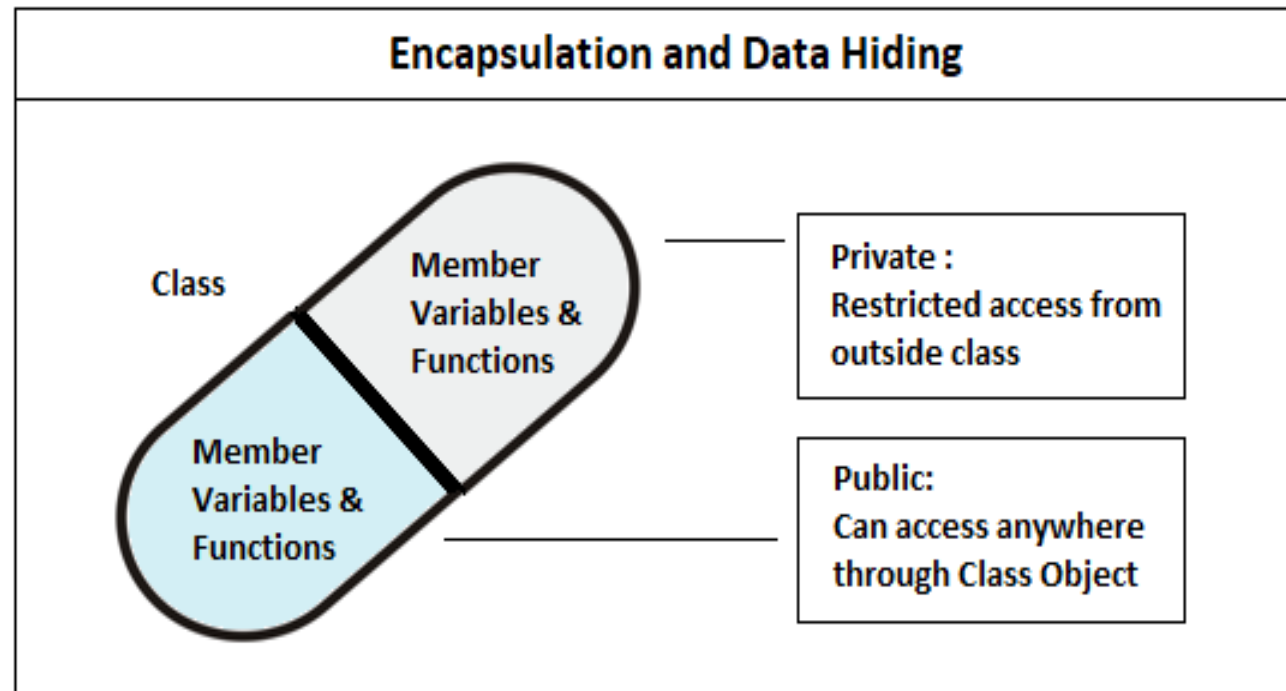
Ability to create new class based on existing class
Cung cấp khả năng tạo ra những lớp mới dựa trên những lớp đã có

Polymorphism:

Ability to implement inherited properties or method in different ways across multiple abstractions.
Khả năng thể hiện thuộc tính/phương thức kế thừa bằng nhiều cách khác nhau thông qua nhiều lớp trừu tượng

1. Tính đóng gói

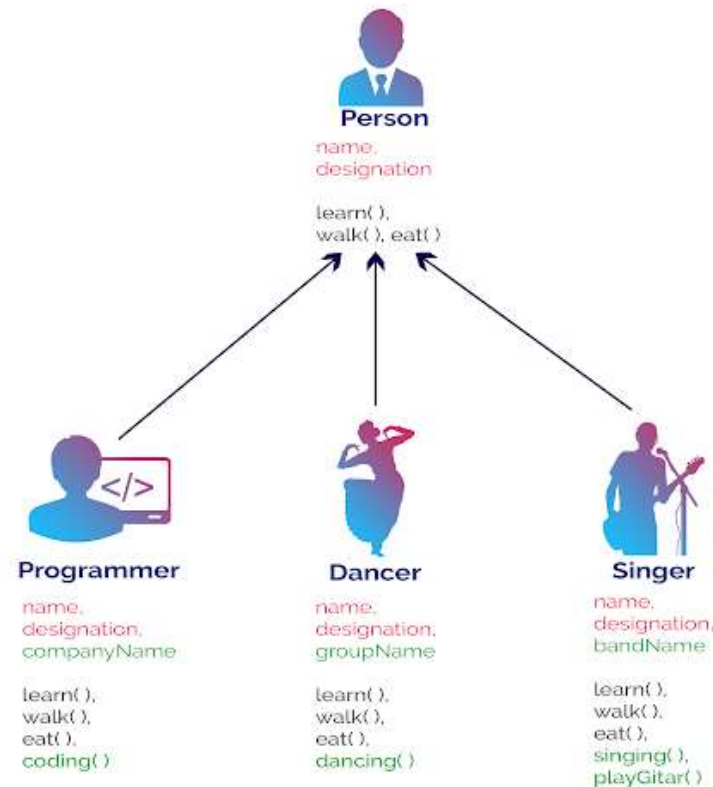
Tính đóng gói: là một trong 4 tính chất cơ bản của OOP nhằm chỉ ra cách thức lập trình mà ***lớp che giấu dữ liệu của mình***, không cho bên ngoài tác động vào. Hay nói cách khác, lớp sẽ quyết định xem bên ngoài được dùng đến dữ liệu nào và không được dùng dữ liệu nào.



Đối với các ngôn ngữ lập trình hướng đối tượng, việc thể hiện tính đóng gói có thể được thực hiện qua việc thực hiện các hàm get – set dữ liệu.

2. Tính kế thừa

Tính kế thừa: Nhằm tái sử dụng lại những đặc tính và hành vi đã có của lớp cha vào lớp con. Lớp con có cấu trúc tương đương nhưng mở rộng ra với những đặc tính và hành vi mới mà lớp cha không có.



Với các ngôn ngữ như C++, Lisp, Python... cho phép đa kế thừa, còn lại các ngôn ngữ OOP hiện đại bậc cao như Java, C#, Ruby, Swift... chỉ cho phép đơn kế thừa. Lớp cha được gọi là base class (lớp cơ sở), lớp con gọi là deliver class (lớp dẫn xuất)

Chú ý: Không nhầm lẫn giữa kế thừa đơn lớp và thực thi nhiều interface

Ghi đè phương thức – Overriding

Overriding : Khi lớp cha và lớp con có phương thức **trùng tên**, nếu ta khai báo một đối tượng của lớp con và sử dụng phương thức trùng tên đó, thì ngôn ngữ sẽ thực hiện ghi đè phương thức bằng cách thay thế nội dung thực thi của phương thức đó bằng nội dung phương thức của lớp con thay vì giữ lại phương thức kế thừa được từ lớp cha.

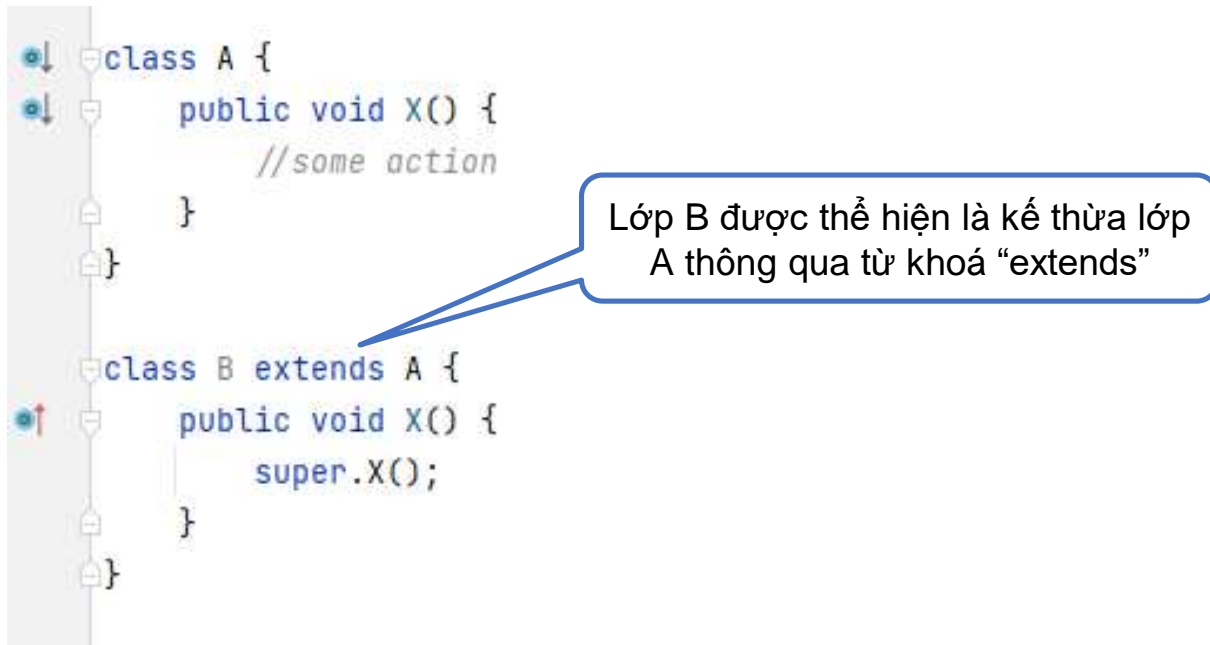
Overriding cũng được xem là đa hình về mặt phương thức – runtime polymorphism

Khi nào thì Overriding:

- Khi lớp cha có phương thức quan trọng và được sử dụng ở nhiều lớp con khác trước đó rồi, và ta muốn sử dụng đặc thù cho một lớp con mới thì để tránh ảnh hưởng đến các phần khác đã được dùng, thay vì ta sửa phương thức ở lớp cha, ta sẽ thực hiện overriding ở lớp con
- Khi ta muốn ghi đè lại các phương thức của lớp gốc như ToString(), GetHashCode(), Equals()... Theo cách mà ta mong muốn.

Overriding trong Java:

- Mặc định với Java khi ở 2 lớp cha con ta thiết lập 2 phương thức trùng tên thì các phương thức ở lớp cha sẽ tự nhận là virtual, lớp con mặc định là override.
- Gọi lại phương thức của lớp cha ta thông qua từ khóa **super**



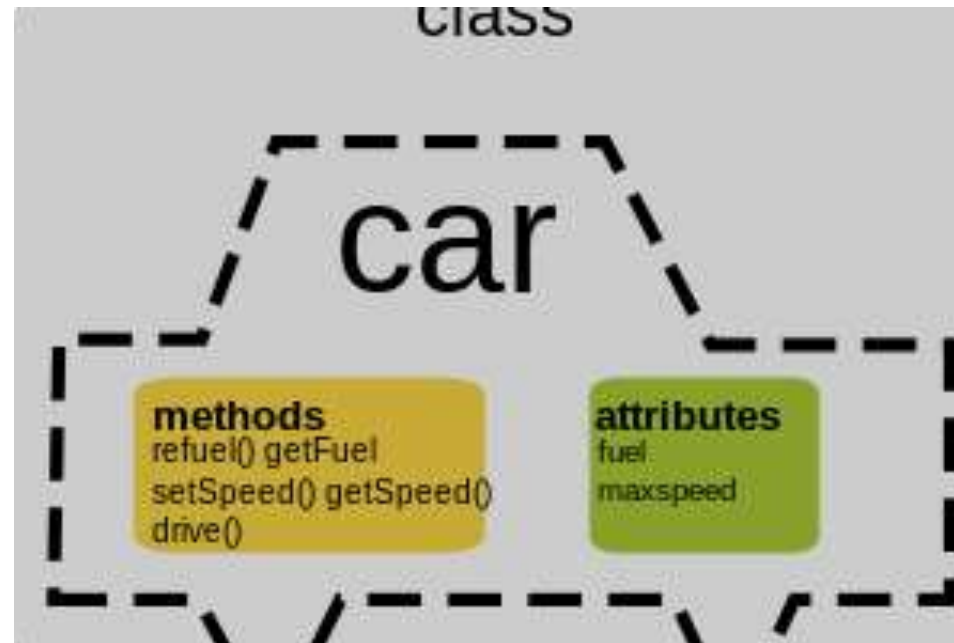
```
class A {  
    public void X() {  
        //some action  
    }  
}  
  
class B extends A {  
    public void X() {  
        super.X();  
    }  
}
```

Lớp B được thể hiện là kế thừa lớp A thông qua từ khoá "extends"

Chú ý: Không nhầm lẫn, cần phân biệt sự khác nhau giữa nạp chồng phương thức Overloading method và Ghi đè phương thức Overriding method

3. Tính trừu tượng

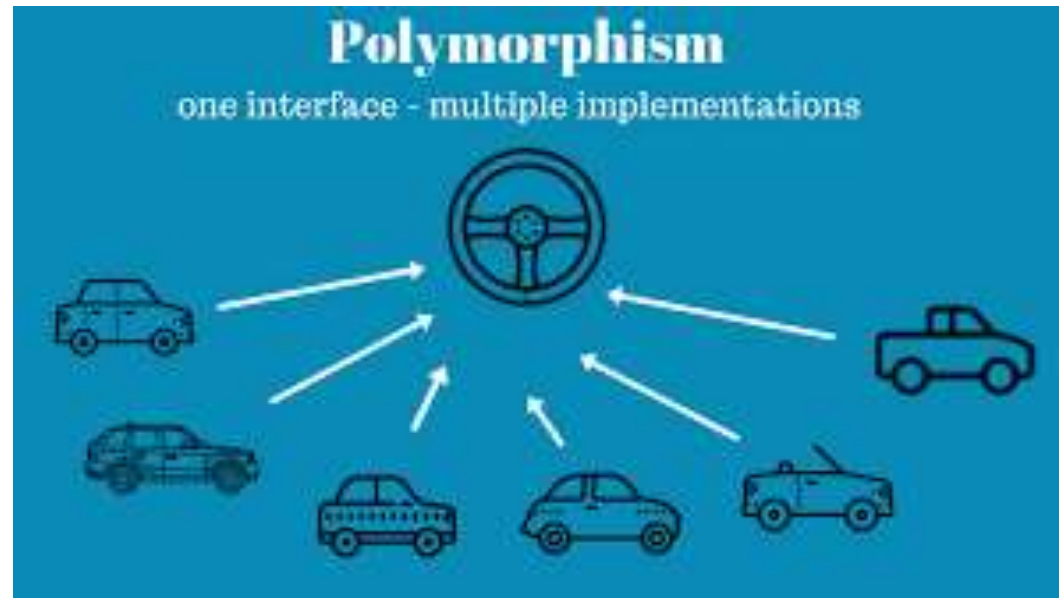
Tính trừu tượng: Tính trừu tượng nhằm che dấu đi chi tiết cụ thể của mã nguồn, chỉ đưa ra bức tranh tổng quan về hành vi của lớp hoặc của phương thức.



Với các ngôn ngữ có hỗ trợ OOP, việc thể hiện tính trừu tượng có thể sử dụng mã nguồn thuần túy hoặc sử dụng một công cụ được hỗ trợ là lớp trừu tượng (abstract class).

4. Tính đa hình

Tính đa hình: Là tính chất thể hiện sự biến đổi của hành vi trong một lớp hoặc của những lớp có những dạng đặc điểm tương tự nhau.



Có hai loại đa hình chính :

- Static Binding (Compile Time Polymorphism)
- Dynamic Binding (Runtime Polymorphism)



Xin trân trọng cảm ơn



Lotus Academy



lotusacademy.edu.vn