

 Cập nhật tháng 8 năm 2024

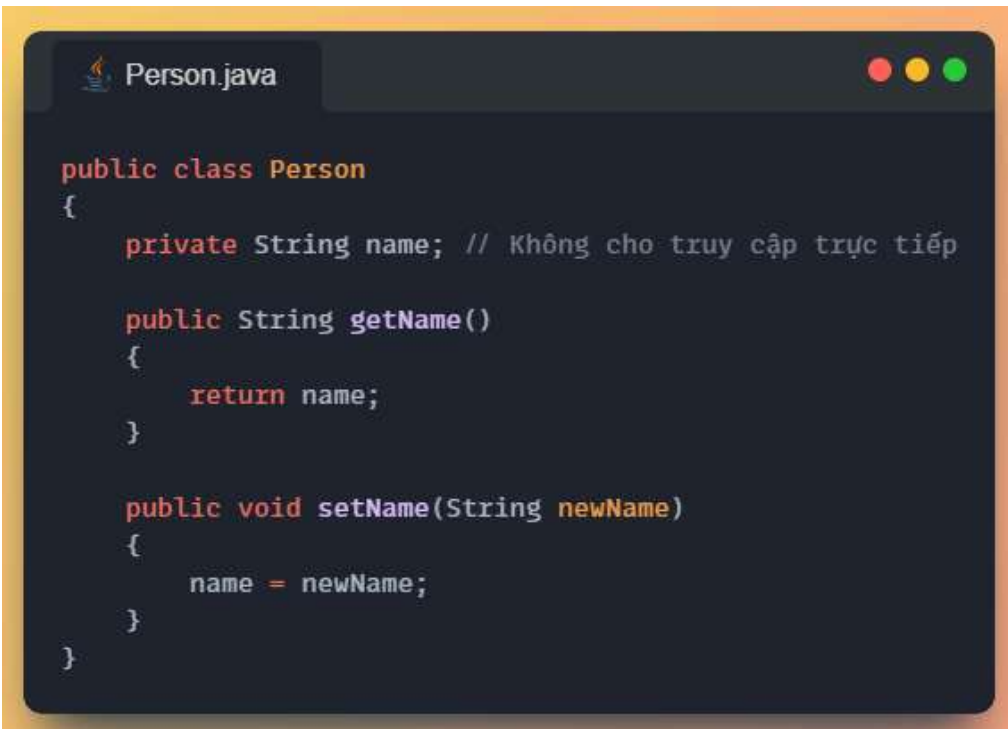
[Bài đọc] Bốn tính chất lập trình hướng đối tượng

Lập trình hướng đối tượng (OOP - Object-Oriented Programming) là một mô hình lập trình mô phỏng cách con người suy nghĩ về thế giới thực bằng cách chia chương trình thành các **đối tượng**.

Sau khi hiểu khái niệm về **class** và **object**, người học cần nắm 4 tính chất quan trọng của OOP:

1. Encapsulation (Đóng gói):

- **Khái niệm:**
 - Đóng gói là việc **che giấu thông tin nội bộ** của đối tượng và **chỉ cho phép truy cập thông qua các phương thức** (thường là **getter** và **setter**)
- **Mục đích:**
 - Bảo vệ dữ liệu khỏi bị thay đổi ngoài ý muốn
 - Giúp quản lý và kiểm soát dữ liệu dễ hơn
- **Ví dụ:**



```
Person.java

public class Person
{
    private String name; // Không cho truy cập trực tiếp

    public String getName()
    {
        return name;
    }

    public void setName(String newName)
    {
        name = newName;
    }
}
```

2. Inheritance (Kế thừa):

- **Khái niệm:**

- Kế thừa cho phép lớp con kế thừa các thuộc tính và phương thức của lớp cha. Giúp tái sử dụng mã và tổ chức chương trình dễ hơn
- Từ khóa: **extends**
- Ví dụ:



```
Animal.java

public class Animal {
    void speak() {
        System.out.println("Animal speaks");
    }
}
```



```
Dog.java

public class Dog extends Animal
{
    void bark()
    {
        System.out.println("Dog barks");
    }
}
```

3. Polymorphism (Đa hình):

- Khái niệm:
 - Đa hình là khả năng một hành động có thể được thực hiện theo nhiều cách khác nhau, tùy thuộc vào đối tượng cụ thể
- Có 2 loại đa hình:
 - **Compile-time Polymorphism** (nạp chồng – method overloading)
 - **Run-time Polymorphism** (ghi đè – method overriding)
- Ví dụ - Ghi đè phương thức:

```
Animal.java

class Animal
{
    void sound()
    {
        System.out.println("Some sound");
    }
}

class Cat extends Animal
{
    void sound()
    {
        System.out.println("Meow");
    }
}
```

4. Abstraction (Trừu tượng):

- Khái niệm:
 - Trừu tượng là che giấu chi tiết cài đặt bên trong và chỉ hiển thị phần quan trọng mà người dùng cần biết
- Cách thực hiện trong Java:
 - Dùng lớp trừu tượng (abstract class)
 - Hoặc giao diện (interface)
- Ví dụ:

```
Animal.java

abstract class Shape {
    abstract void draw(); // Không có phần thân – chỉ khai báo
}

class Circle extends Shape {
    void draw() {
        System.out.println("Draw a circle");
    }
}
```

- Người dùng chỉ quan tâm đến việc "vẽ hình", không cần biết bên trong **draw()** làm gì

5. Tóm tắt bảng bốn tính chất

Tính chất	Mô tả ngắn	Từ khóa liên quan
Đóng gói	Ẩn thông tin, chỉ cho truy cập qua method	private, getter/setter
Kế thừa	Lớp con dùng lại mã từ lớp cha	extends
Đa hình	Một hành động - nhiều cách thực hiện	@Override, overload
Trừu tượng	Che giấu chi tiết, chỉ lộ phần quan trọng	abstract, interface

Tài nguyên đọc thêm: <https://www.geeksforgeeks.org/four-main-object-oriented-programming-concepts-of-java>

Danh sách các bài học

