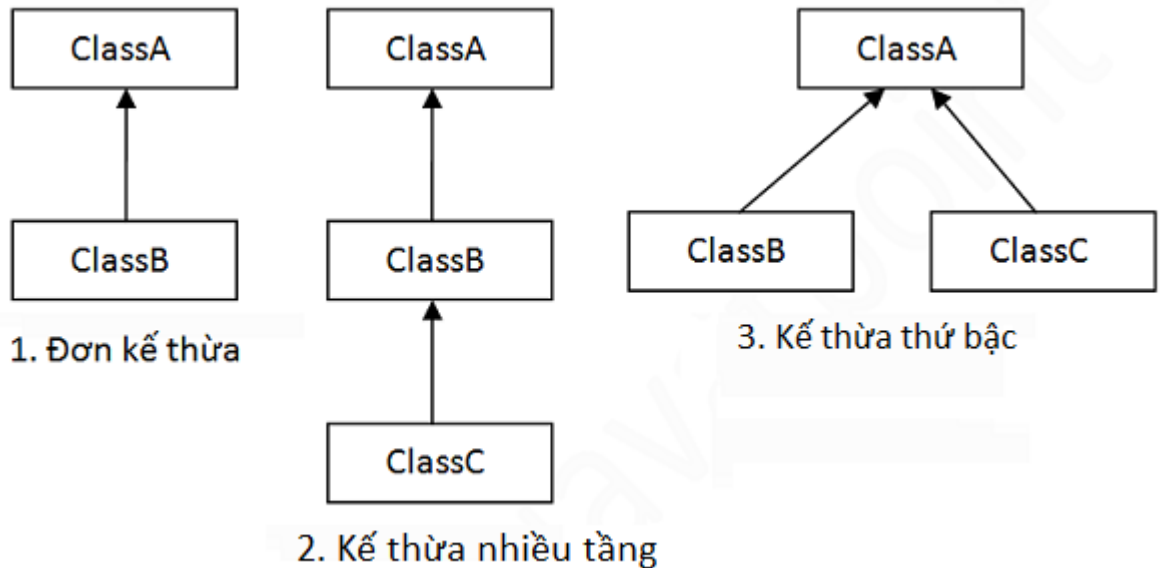


Object-Oriented Programming

Inheritance (Kế thừa)

- Định nghĩa lớp mới trên lớp có sẵn
- Lớp mới hấp thu lớp cũ và bổ sung thêm các tính năng
- Lớp con kế thừa lớp cha có thể truy cập vào các thành viên public và protected của class cha. Nó không được phép truy cập đến thành viên private của class cha.
- Có 3 kiểu kế thừa : đơn kế thừa, kế thừa nhiều tầng, kế thừa thứ bậc



Polymorphism (đa hình)

- Một đối tượng có nhiều hình thức khác nhau
- Liên quan đến kế thừa
- Cùng thông điệp hiểu theo cách khác nhau : ví dụ Hình tròn xoay khác mà hình vuông xoay khác
- Có 2 cách thực hiện : nạp chồng phương thức và ghi đè phương thức.

Abstraction (trừu tượng)

- Trừu tượng hóa là : biểu diễn bằng mô hình đơn giản, tập chung vào tính chất quan trọng. Đơn giản là tìm ra các thuộc tính và phương thức của đối tượng

Encapsulation (đóng gói)

- Sau khi trừu tượng hóa, chúng ta sẽ đóng gói nó lại thành một lớp. Lớp sẽ chứa các thuộc tính và hành vi của đối tượng ta vừa trừu tượng.
- Đóng gói còn dùng che dấu thông tin (information hiding), che dấu thông tin không cho thế giới bên ngoài biết

Object

- Một thực thể có trạng thái và hành vi thì gọi là đối tượng.
- Đối tượng là một thể hiện của một lớp

- Class (lớp) là mẫu thiết kế dùng để tạo ra đối tượng
- Một đối tượng sẽ có các attribute
- Đối tượng ô tô có nhiều loại ô tô còn lớp có một lớp là ô tô

Phạm vi truy cập

- Private: chỉ cho truy cập trong class
- Protected: có thể truy cập bên trong package và bên ngoài package. Bên ngoài package thì chỉ cho lớp kế thừa truy cập.
- Public: bất cứ đâu trong project đều có thể truy cập được
- Default: chỉ cho truy cập bên trong package

Attribute

- Lưu trữ các thông tin trạng thái của đối tượng
- Thuộc tính khai báo trong lớp có tên gọi là : biến thực thể (instance variable), thành viên dữ liệu (data member), trường (field).

State

- Giá trị của thuộc tính theo thời gian

Behavior

- Các hành động của đối tượng

Method

- Cài đặt hành vi
- Thao tác khi nhận thông điệp
- Định nghĩa trong lớp

Constructor

- Constructor dùng để khởi tạo các đối tượng
- Constructor khởi tạo các giá trị để cung cấp dữ liệu cho các đối tượng
- Quy tắc tạo constructor: Tên constructor phải giống tên lớp chứa nó
- Có 2 kiểu : có tham số và không tham số (mặc định)

Constructor Overloading

- Tạo nhiều constructor trong cùng một lớp với danh sách tham số truyền vào khác nhau. Trình biên dịch phân biệt các constructor này thông qua số lượng và kiểu của các tham số truyền vào.

Method overloading

- Nạp chồng phương thức
- Một lớp có nhiều phương thức có tên giống nhau nhưng khác nhau về kiểu dữ liệu hoặc số lượng các tham số.

Method overriding

- Ghi đè phương thức
- Xảy ra nếu lớp con có phương thức giống lớp cha.

- Nếu lớp con cung cấp sự cài đặt cụ thể cho phương thức đã được cung cấp bởi một lớp cha của nó được gọi là ghi đè phương thức
- Nguyên tắc ghi đè: tên giống với lớp cha, tham số giống với lớp cha, mối quan hệ kế thừa.

Từ khóa this

- Một biến tham chiếu
- Dùng để tham chiếu đến đối tượng của lớp
- Dùng phân biệt biến toàn cục và biến cục bộ trong constructor
- Dùng `this()` để gọi constructor của lớp hiện tại. Tác dụng tái sử dụng lại code tạo constructor, chỉ việc thêm mới. Quy tắc: phải để ở dòng đầu tiên

Lớp abstract

- Một phương thức được khai báo là abstract và không có trình triển khai thì đó là phương thức trừu tượng.
- Một lớp chứa một phương thức cụ thể nhưng muốn triển khai phương thức đó được quyết định bởi các lớp con.
- `abstract void printStatus();` không có định nghĩa, được theo sau bởi dấu chấm phẩy, không có dấu ngoặc nhọn

Get và set

- Dùng để đóng gói. Tạo một lớp với phạm vi truy cập là private. Muốn lấy thông tin thì dùng get và set.

Instance

- Thực thể hay đối tượng của lớp

Upcasting

- Khi biến tham chiếu của lớp cha tham chiếu tới đối tượng của lớp con, thì đó là Upcasting
- Ví dụ:
- `class A{}`
- `class B extends A{}`
- `A a=new B();//upcasting`

Đa hình với thành viên dữ liệu

- Nếu lớp cha và lớp con có thành viên dữ liệu giống nhau thì khi tham chiếu lớp cha tham chiếu đến đối tượng lớp con. Thành viên dữ liệu của lớp cha sẽ không bị ghi đè bởi thành viên dữ liệu lớp con. Nghĩa là khi ta truy cập thành viên dữ liệu thì sẽ truy cập vào thành viên dữ liệu của lớp cha.

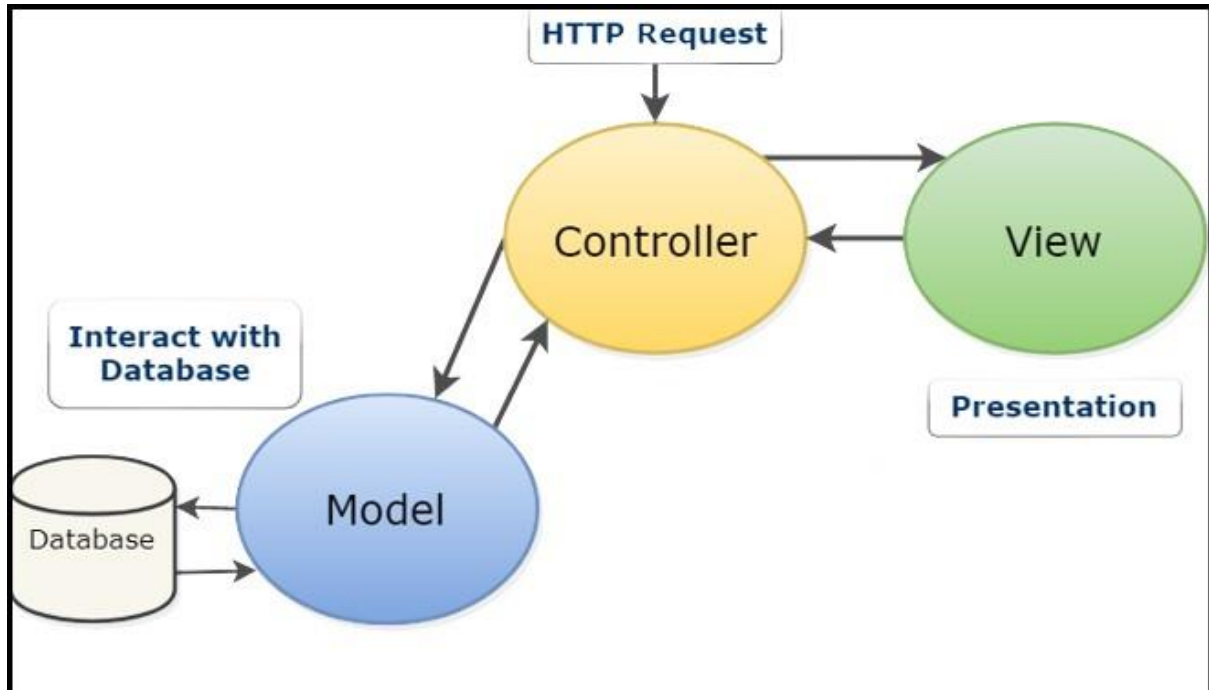
Đa hình nhiều tầng

- Dù nhiều tầng kế thừa như tham chiếu cha vẫn có thể tham chiếu đến các đối tượng con

Mô hình MVC

- Tên đầy đủ là : Model-View-Controller

- Kiến trúc phần mềm hay mô hình thiết kế
- Để tạo lập giao diện người dùng trên máy tính.
- Chia thành 3 thành phần riêng, hoạt động độc lập và có thể tương tác qua lại với nhau gồm model, view, controller



- **Model** là thành phần chịu trách nhiệm lưu trữ dữ liệu và logic nghiệp vụ của ứng dụng. Model tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu. View: là giao diện tương tác với người dùng
- **View** là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng. View tạo ra giao diện người dùng (UI) cho ứng dụng. View sử dụng dữ liệu từ Model để tạo ra các trang web, biểu đồ, báo cáo,...
- **Controller** là thành phần chịu trách nhiệm xử lý các yêu cầu từ người dùng. Nó nhận yêu cầu từ người dùng thông qua View, sau đó gửi yêu cầu đến Model để xử lý. Nó cũng nhận kết quả từ Model và gửi kết quả đến View để hiển thị cho người dùng.

Từ tiếng anh it

- System crash: sự cố hệ thống
- platform independent : độc lập nền tảng
- platform : hệ thống máy tính có nền tảng cụ thể như Windows, Macintosh.
- Write once, run anywhere : Viết một lần, chạy bất cứ đâu
- Interact with database : "tương tác với cơ sở dữ liệu", giao tiếp với cơ sở dữ liệu để lưu trữ, truy xuất và chỉnh sửa dữ liệu. sử dụng ngôn ngữ truy vấn dữ liệu (SQL). SQL là một ngôn ngữ chuẩn được sử dụng để truy vấn dữ liệu từ cơ sở dữ liệu.