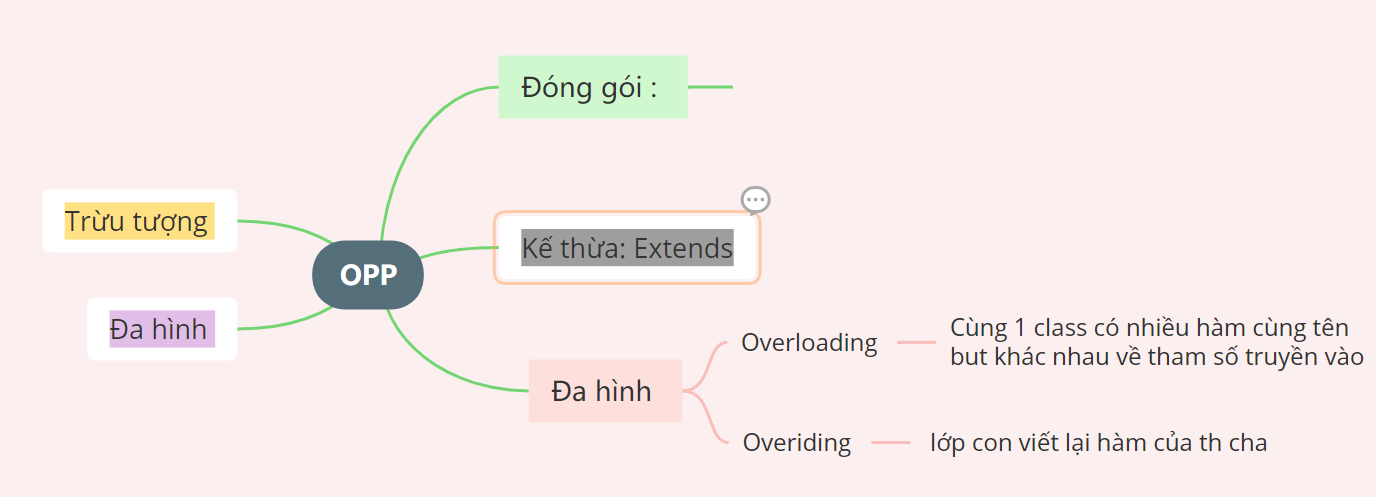
****

* **Đóng gói**: **(Encapsulation)** bảo mật thông tin
* Properties:
* Modifier:

+Private: trong Class

+Default: trong cùng 1 Class + same package

+Protected: các lớp con, cùng package, lớp con, cùng class dc phép truy cập

+Public:

* Method:
* **Kế thừa: ( Inheritance)** tái sử dụng; từ khóa extend .
* **Đa hình: (Polymorphism)**

- Overiding : Con viết lại của cha: 2 lớp cha con

+ Sub class change modifier: có nhưng mức độ rộng rãi hơn

+ Overring methods: xài lại hàm của th cha

+ Hidding method: th con viết lại hàm static của th cha

- Overloading: 1 lớp nhiều hàm cùng tên, khác nhau kiểu dữ liệu và tham số truyền vào

* **Trừu tượng: (Abtraction)**
* **Final** Class ko cho kế thừa

|| Method cho kế thừa but ko cho overiding

|| Properties hằng số (final) public int MAXN = 5) (hằng số ko thay đổi) viết hoa

Empty propertites: phải dc khởi tạo ở cóntructor

* **Static**
* Static khai báo trc Method dùng để truy cập thông qua class mà ko cần khởi tạo đối tượng
* Static **block** để thực thi đầu tiên khi hàm đó dc gọi tới

( khi vừa gọi là chạy vào )

* **This**

Từ khóa this dùng để tham chiếu đến thuộc tính hay phương thức

Để phân biệt giữa Attribute (thuộc tính) và Parameter (tham số).

Super truy cập đến lớp cha

* **Classes relationship: các khái niệm về**

**Is a/ a kind of** mối qua hệ cha con

**Has a** : để biểu diễn thuộc tính

**Association**: mối quan hệ 1vs1; 1vs nhiều; nhieu vs nhieu

**Data type (~class)**

→là cái khuôn để tạo ra object

**1/ Primitive:** int, char float,

**2/Reference** : class, dog, animal, student

**instance** đã dc cấp phát vùng nhớ >< object

**Class** : Method (là cái để thay đổi các thuộc tính của property)

thuộc tính là cái thuộc về riêng tư của class ko dc truy cập từ bên ngoài )

**Class** ( thongtin chung chuaw rõ) → object (chi tiết)

**+ Encapsulation** ( ĐÓNG GÓI ) che đậy thông tin

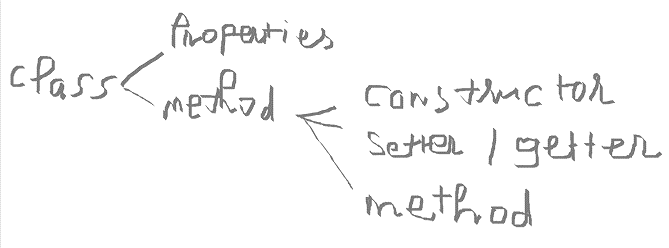
Modifies: đjnh nghĩa mức độ che đậy tthong tin tạo ra 4level

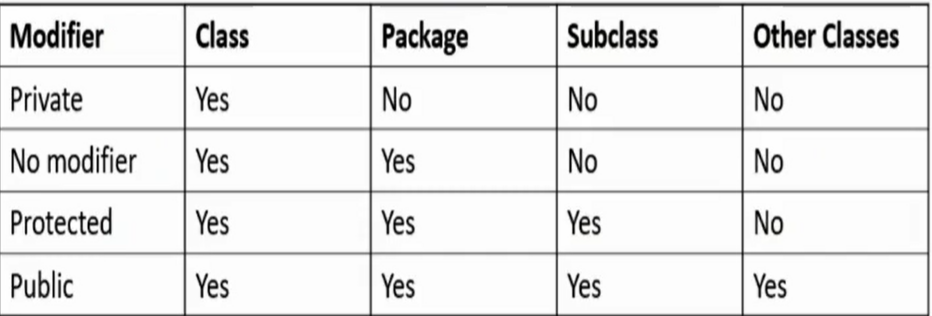
Private: mạc định chon cho thuộc tính

No modifier

Protected

Public ko dc xài cho thuộc tính





**Constructor (hàm khởi tạo )**

**1/ cũng vs tên lớp**

**2/dc thực thi khi tạo ra đối tượgn = từ khóa new**

**3/ cú pháp public \_\_\_tênclass nếu ko có hàm khởi tạo thì trình biên dịch sẽ tạo ra 1 hàm mạc định ko tham số**

**Getters/Setters**

**Get** để **lấy** giá trị thuộc tính

**Set** để **gán** giá trị thuộc tính

**3. Review Class Object+ Encapsulation+ Inheritance+ Polymorphism (sc:youtube)**

* Java datatype : \_\_Primitive : nguyên thủy

\_\_Reference: tham chiếu (class) class & dataType &object

**Class**: ( Name theo noun kí tự đầu viết hoa) \_ nhận biết danh từ chính

* **Properties**: biểu diễn đặc điểm của class khai báo private \_ danh từ bổ trợ

( name theo noun related to class kí tự đầu thường )

* Method : biểu diễn hành động của class

( name theo verb related to class kí tự đầu thường )

\_ nhận biết động từ

Contructor: tên giống tên Class dc thực thi khi tạo mới đối tượng qua từ khóa New

Khi đã có hàm khởi tạo r thì khi tạo mới đối tượng phải dùng hàm do người dùng đã định nghĩa

+ Contructor có tham số

+ Contructor ko có tham số

**SLOT 4**

Co 2 cach khoi tao new khoi tao gia tri vs static

This phan biet thuoc tinh va tham so param

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Abstract method

bắt buộc phải override

**SLOT 7: Objectives**

Tthis truy cập chính nó

Super truy cập th cha