OOP

Oop có 4 tính chất cơ bản: đóng gói, đa hình, kế thừa, trừu tượng.

1. Đóng gói: là cách để che dấu những tính chất xử lý bên trong của đối tượng, các đối tượng khác chỉ có thể thay đổi trạng thái của lớp thông qua phương thức public của đối tượng của lớp đó.

* Các phạm vi: public, private, protected, default.
* Public: được truy cập từ bất cứ đâu.
* Private: chỉ được truy cập bên trong lớp và các lớp con nhưng không thể truy cập trực tiếp vào thành phần đó.
* Protected: có thể được truy cập bởi lớp hiện tại và lớp con nhưng không thể truy cập từ bên ngoài.
* Default: chỉ có thể truy cập trong các lớp cùng package.

1. Đa hình: Các lớp con kế thừa từ 1 lớp cha có thể thay đổi quá trình xử lý của lớp cha đó.
   * Cách thể hiện: overloading, overriding.
   * Overloading: trong cùng 1 class có thể định nghĩa phương thức cùng tên nhưng khác nhau về kiểu dữ liệu trả về và các tham số đầu vào.
   * Overriding: Class con kế thừa class cha có thể định nghĩa lại phương thức mà lớp cha đã tồn tại. Khi gọi phương thức nó sẽ xử lý theo lớp con đã định nghĩa.
     + **Virtual**: từ kháo này dùng để khai báo 1 phương thức cơ bản và có thể được ghi đè.
     + **Override**: từ khoá này dùng để đánh dấu phương thức đang ghi đè lên phương thức virtual của lớp cha.

Trong java khi các phương thức không phải final mặc định nó là virtual và có thể override lại.

1. Kế thừa: 1 lớp có thể kế thừa từ 1 lớp khác và nó thừa hưởng tất cả những thuộc tính, phương thức không phải là private. Thuộc tính này giúp giảm thiểu quá trình lặp code.
   * 1 class chỉ được kế thừa từ 1 lớp cha nhưng có thể kế thừa nhiều cấp.
2. Trừu tượng: là phương pháp trừu tượng hoá các phương thức, thuộc tính mà đối tượng đó cần phải có.
   * Abstract class:
     + Có 2 loại method: abstract method và method
       - abstract method: method này sẽ rỗng và không được thực hiện logic hay mang giá trị. Các class con phải xác định logic cho nó.
       - Method thường: method này vẫn mang giá trị và logic. Nó được sử dụng cho mục đích dùng chung. Các lớp kế thừa phải định nghĩa lại tất cả abstract method trừ lớp con cũng là abstract class.
       - Không có thuộc tính trừ tượng
   * Interface: trong interface không tồn tại method có logic. Các lớp kế thừa phải thực thi lại tất cả method đó. Nó giống như 1 khung mẫu để các lớp kế thừa dõi theo.
     + Trong interface sẽ không có constructor.
     + 1 lớp có thể implement nhiều interface.
     + Trong interface chỉ tồn tại phương thức trừu tượng và không tồn tại thuộc tính.

Sự khác nhau giữa class và instance: class là khuôn mẫu định nghĩa các đối tượng object.

Còn instance là 1 đối tượng được tạo ra từ class. Mỗi instance độc lập nhau.   
Instance được tạo khi khởi tạo đối tượng bằng từ khoá new

Nếu 1 class implement 2 interface có cùng phương thức nhưng khác kiểu dữ liệu trả về nó sẽ không có vấn đề xảy ra khi khác tham số đầu vào vì class đó bắt buộc phải override lại. còn nếu cùng tham số sẽ gây ra lỗi vì java không cho phép. Điểu này tương tự khi kế thừa 1 abstract và 1 interface.

2 từ khoá this và super có ý nghĩa hoàn toàn khác nhau.

This chỉ ra rằng hành động đang truy cập vào thuộc tính hay phương thức thuộc lớp đó.

Super sử dụng khi kế thừa 1 class có constructor và cần truyền tham số vào constructor đó.

1 phương thức là private hoặc final hoặc static sẽ không thể ghi đè.

Phương thức Static không thể ghi đè là vì nó thuộc class không phải instance nên không ghi đè.

\_\_

Static không thể kế thừa.

Abstract sử dụng khi: có những method dùng chung cho các lớp con. Nhưng mỗi lớp con xử lý 1 cách khác nhau. Cũng có những method dùng chung cho mọi class con mà xử lý giống nhau. Hạn chế lặp lại.

Interface: