Buổi 1: Giới thiệu về nghề lập trình android và cơ hội nghề nghiệp

* Cài đặt phần mềm Java SDK 1.8; Git SCM, Inteliji

Buổi 2: Giới thiệu về Git + GitHub, Agile Scrum

1. Giới thiệu về Agile Scrum
2. Giới thiệu về Git + Github

Quick Setup:

git init

git clone <https://github.com/trongtoanbk/Gitexample.git> // su dung neu clone thu muc tren github ve

git add Readme.md

git config user.email "trongtoanbk.hd@gmail.com"

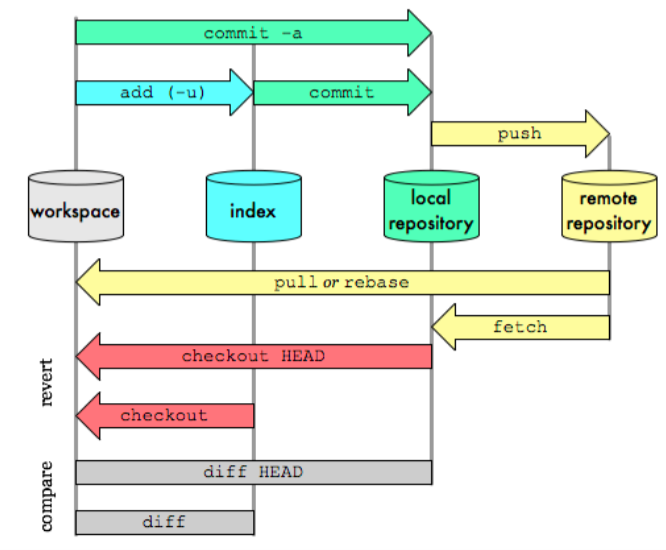
git commit –m “ first comit”

git branch –M Master

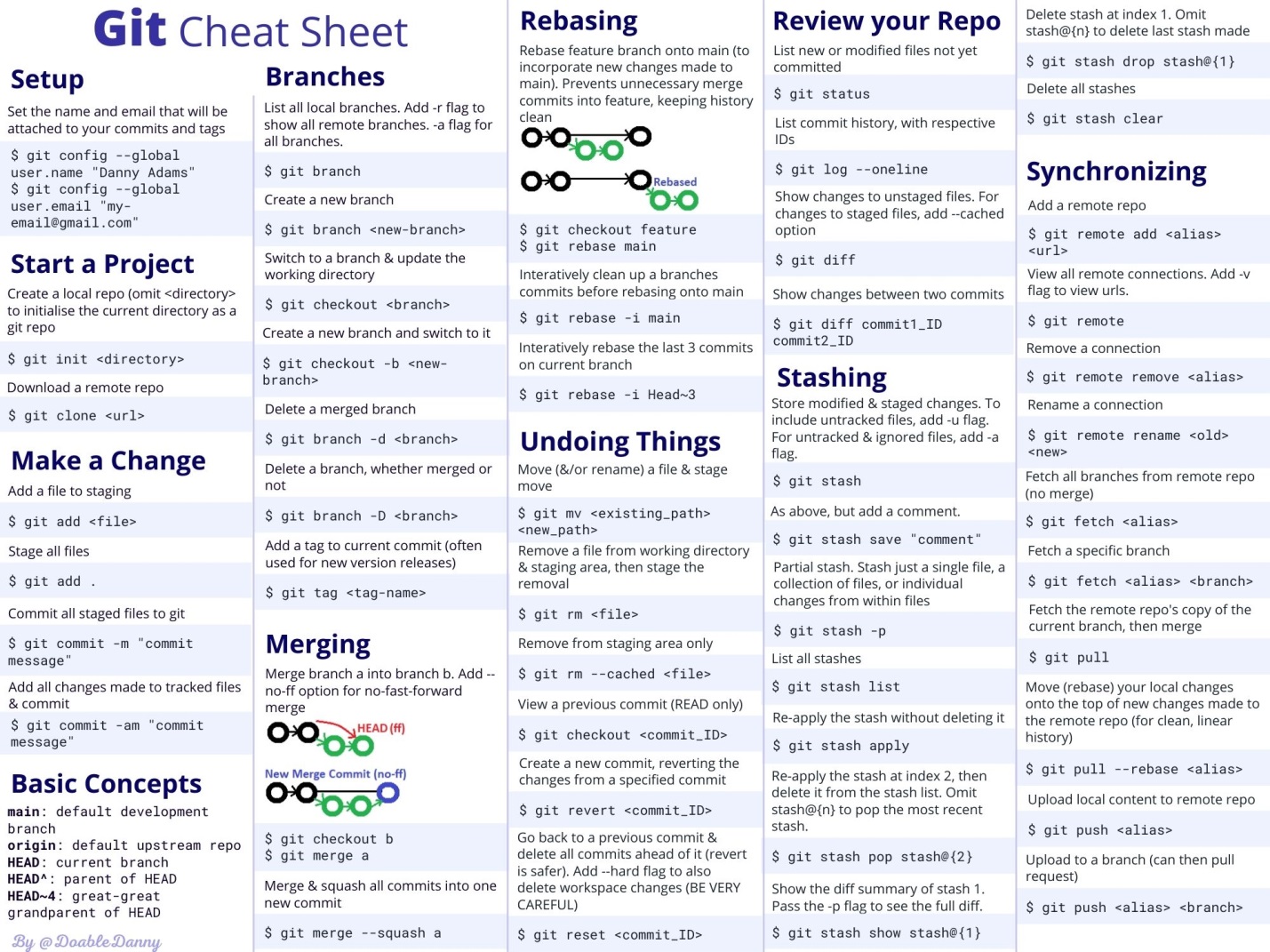
git remote add origin <https://github.com/trongtoanbk/Gitexample.git>

git push origin Master

----------------------------------------------------------------------------------------------------------------







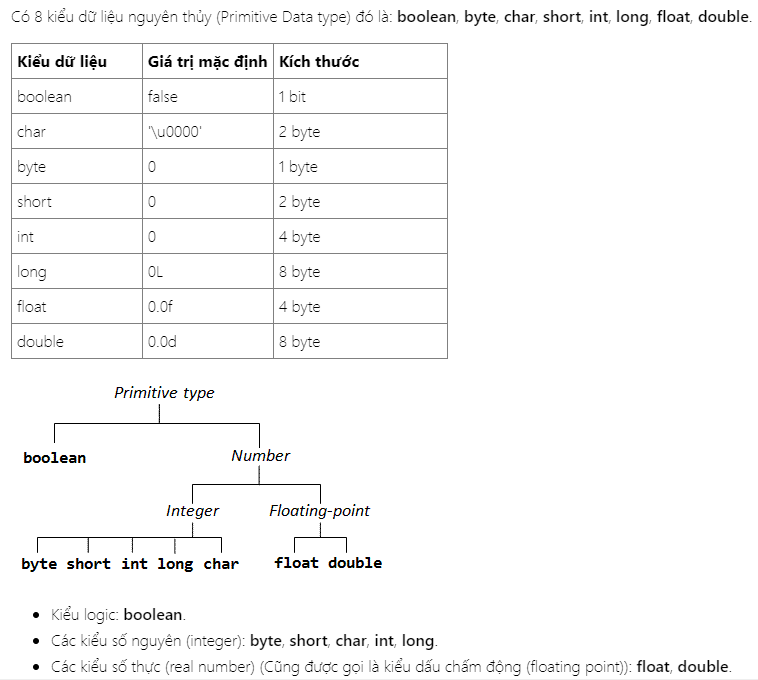
**Buổi 3: Mở đầu về Java**

1. Kiểu dữ liệu

02 loại kiểu dữ liệu:

Kiểu dữ liệu nguyên thủy (primitive)

Ex: char, int, long, float, double, boolean : viết thường



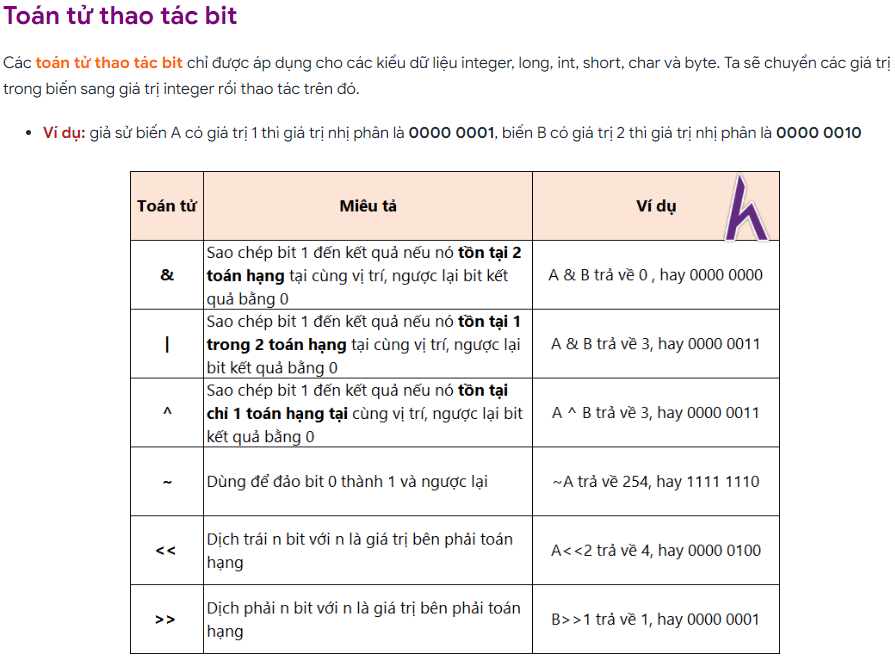
Kiểu dữ liệu đối tượng (Object)

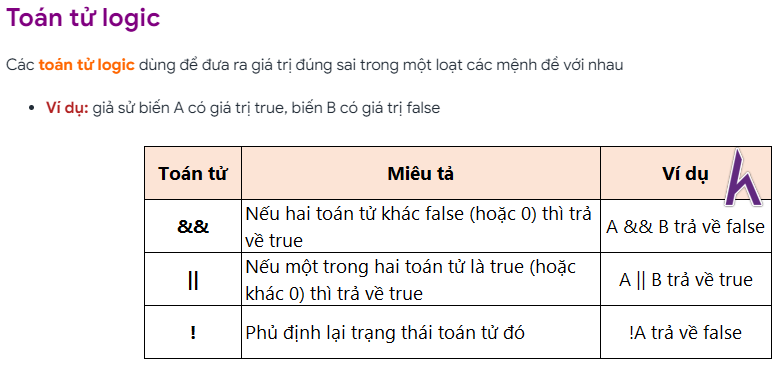
Ex: String, Array, Interger, Long, Float, Double, Boolean,.. : Viết hoa

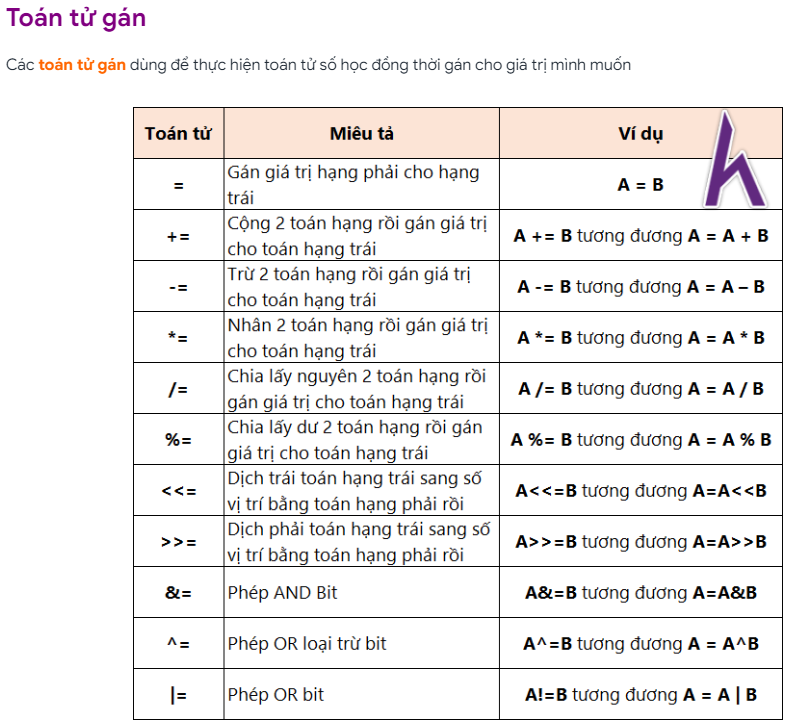
**Sau đây là một số quy chuẩn đặt tên thường dùng trong dự án:**

1. **Tên lớp** đặt theo PascalCase, ví dụ: UserClass, CategoryClass…
2. **Tên hàm** và phương thức sử dụng camelCase, ví dụ getUser, getCategory…
3. **Tên biến** cũng sử dụng camelCase loginUser,loginUser,categoryList…
4. **Tên hằng số** thì đặc biệt, viết hoa hết và cách nhau bởi dấu gạch dưới DISCOUNT\_PERCENT, LIMIT\_RATE…
5. **Tên bảng**, **tên cột** trong Database sử dụng underscore và sử dụng danh từ số nhiều, ví dụ bảng oauth\_clients, oauth\_refresh\_tokens.
6. **Tên phần tử trong HTML**, ví dụ khi bạn sử dụng Vue.js, React… tạo ra thì nó sẽ có dạng KebabCase, ví dụ <my-component>.
7. **Toán tử**









**Buổi 4+5: Các câu điều kiện và vòng lặp cơ bản**

Điều kiện if … else

Điều kiện switch … case

Vòng lặp for

Vòng lặp while

Vòng lặp do …. While

**Buổi 6: Java, tìm hiểu về hướng đối tượng**

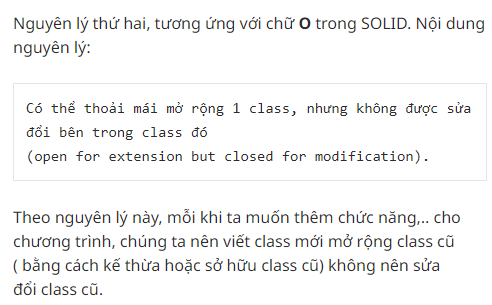
1. **Solid**

[**https://toidicodedao.com/2015/03/24/solid-la-gi-ap-dung-cac-nguyen-ly-solid-de-tro-thanh-lap-trinh-vien-code-cung/**](https://toidicodedao.com/2015/03/24/solid-la-gi-ap-dung-cac-nguyen-ly-solid-de-tro-thanh-lap-trinh-vien-code-cung/)

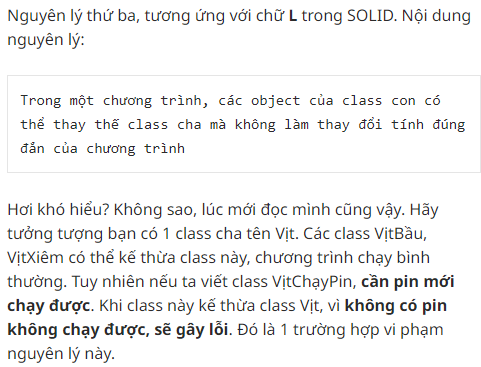
**Single responsibility principle**

****

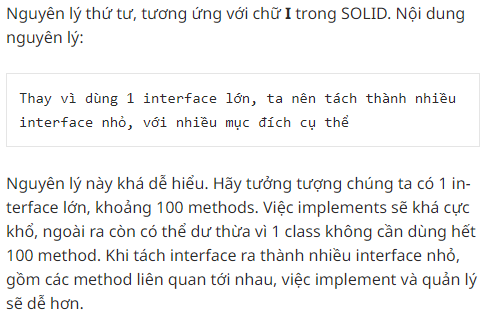
### ****Open/closed principle****

****

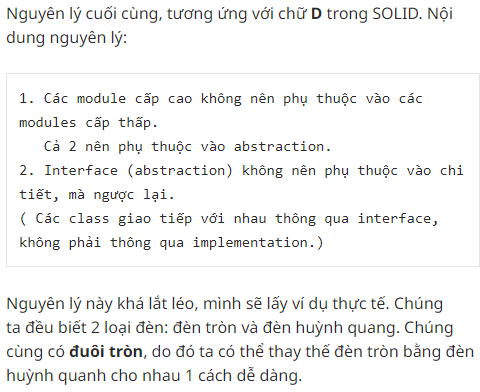
### ****Liskov Substitution Principle****

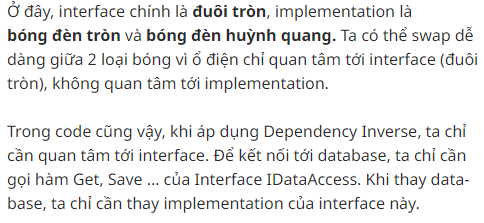
****

### ****Interface Segregation Principle****

****

### ****Dependency inversion principle****

****

****

**Uml Diagram**

**Các kiểu nguyên thủy không có giá trị null, chỉ có kiểu Object mới có giá trị Null.**

**Null là đối tượng không có vùng nhớ, còn chuỗi rỗng là có vùng nhớ**

**Buổi 7: Java hướng đối tượng – Buổi 2**

Tham chiếu: => truyền vùng nhớ (kiểu đối tượng)

Tham trị: => Truyền phần giá trị. Kiểu nguyên thủy

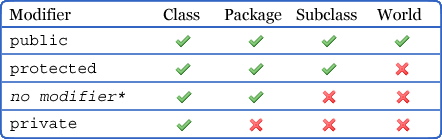
Int a = 5;

Int b=10;

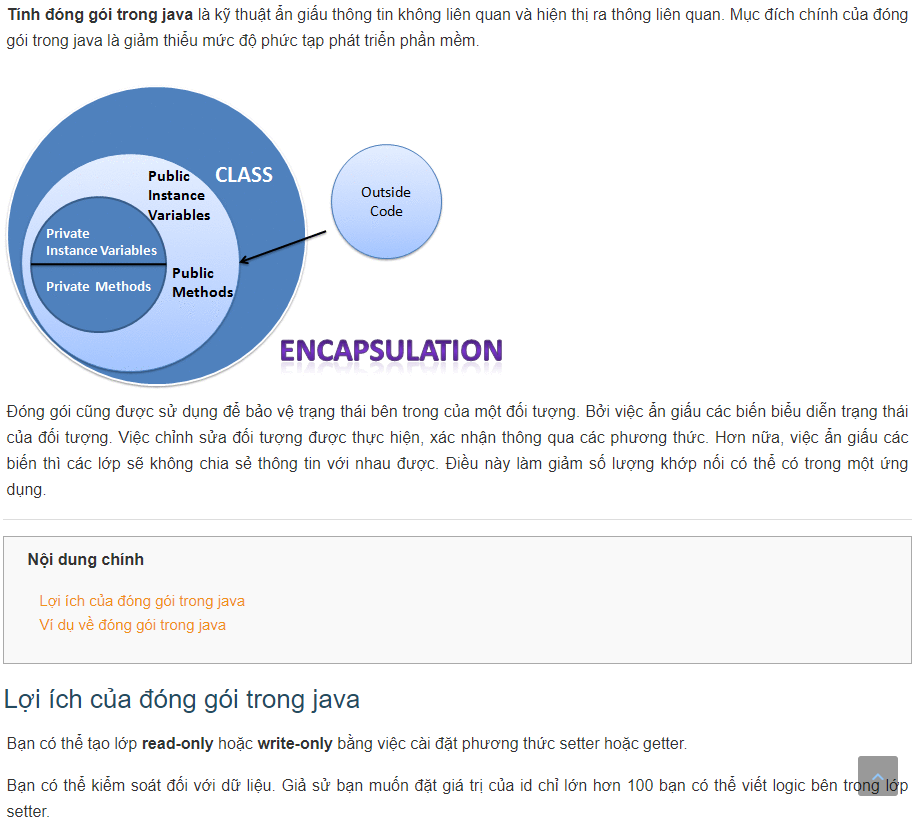
If (a >b){} => Đúng

// Access modifier  
// tính kế thừa  
// tính đóng gói  
// tính đa hình  
// tình trừu tượng

* 1. Accesss modifier (phạm vi truy cập): Sử dụng cho biến, cho phương thức



1. Tính đóng gói

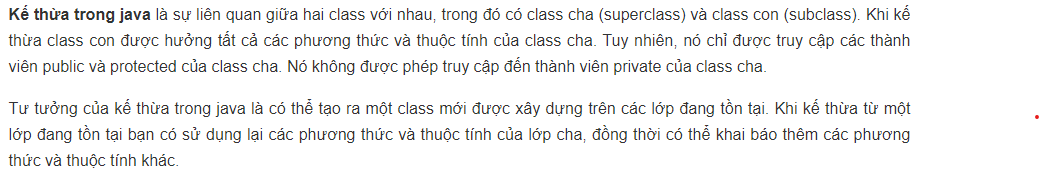
 

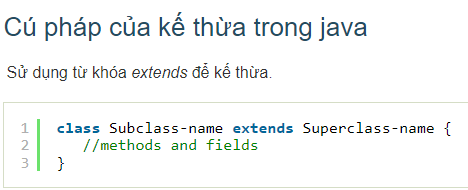
1. Tính kế thừa :

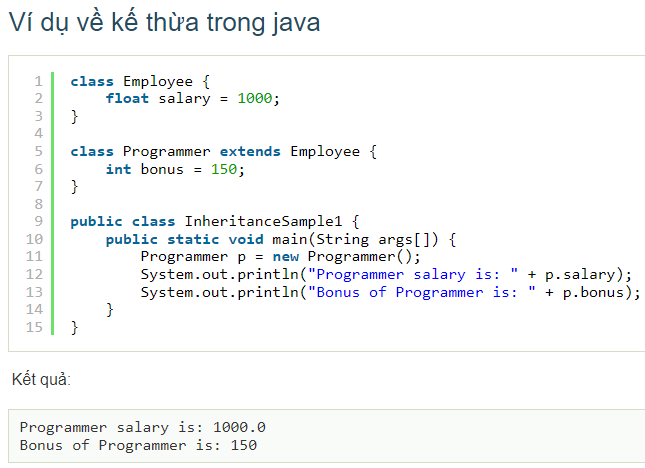
Base => clone base 1 => sử dụng từ khóa extends => lớp kế thừa lớp

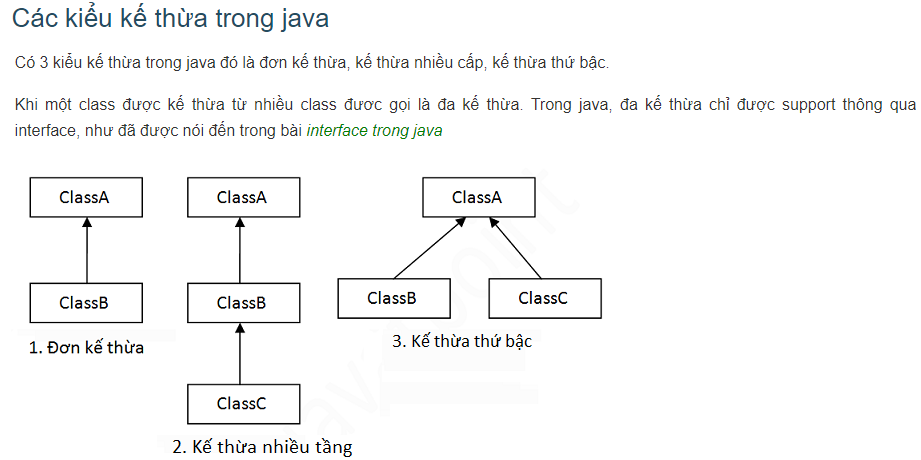
Super: phương thức của lớp cha

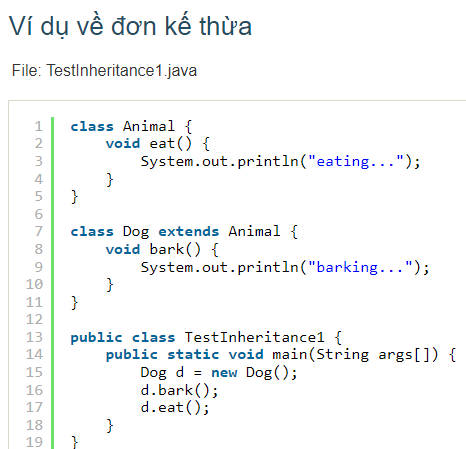
1 lớp được kế thừa tối đa 1 lớp

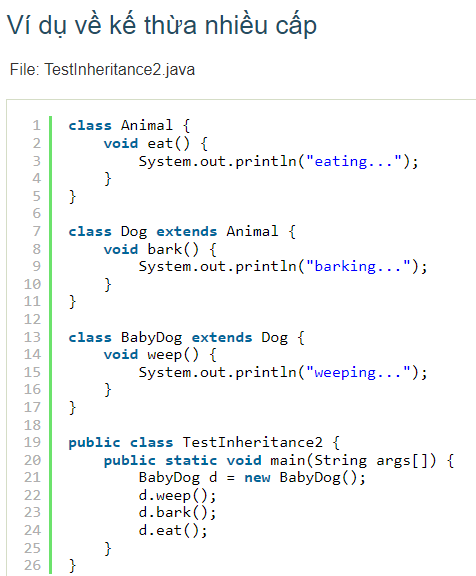












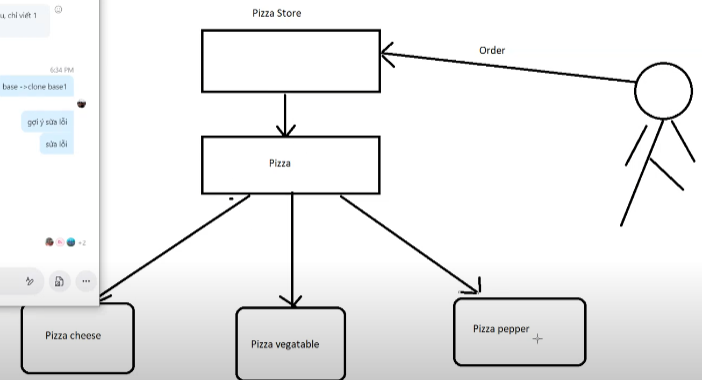
1. **Tính đa hình**

Overide => phương thích ghi đè thằng con có thể tùy chỉnh từ phương thức của lớp cha=> tính đa hình

* Đa hình là khi sử dụng một phương thức nhưng cách sử dụng khác nhau trong lớp cha và lớp con

1. **Tính trừu tượng**

Trừu tượng



**Buổi 7: Mở đầu về Android**

**API:** Version của device trong android studio => recommend đặt API >=6.

**API:** Thư viện để giao tiếp với các đối tượng bên ngoài

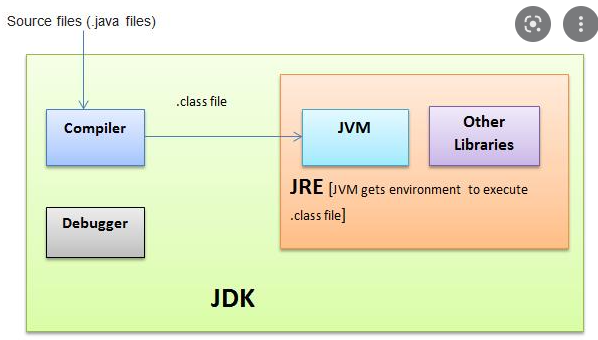
**Gradle => Compiler để biên dịch code**

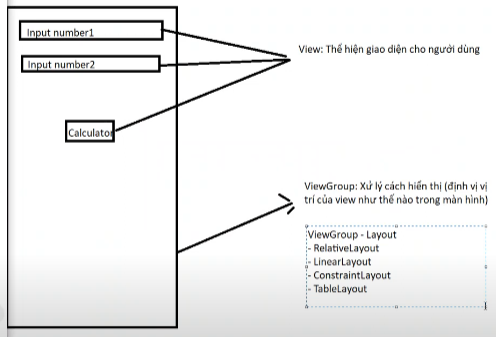
**3 package mặc định: Manifest, Java, Res**

**Manifest: chứa file cấu hình: tên, màn hình, icon, quyền**

**Java: sử dụng để viết code xử lý**

**Res: chứa tài nguyên của phần mềm: database, layout, string,…**

****

****