# [BUỔI 1] WELCOME TO JAVA

## Java là gì?

- Java là một ngôn ngữ đa nền tảng, hướng đến đối tượng, lấy mạng làm trung tâm và có thể được sử dụng như một nền tảng. Là một ngôn ngữ lập trình nhanh, bảo mật, đáng tin cậy dùng để viết mã cho mọi thứ từ ứng dụng di động, phần mềm doanh nghiệp cho đến các ứng dụng dữ liệu lớn và công nghệ phía máy chủ.

- Khác biệt với các ngôn ngữ lập trình khác, Java mang một màu sắc riêng. Thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn mới có thể chạy được thì Java được thiết kế để biên dịch mã nguồn thành bytecode, sau đó bytecode sẽ được môi trường thực thi chạy.

- Ngôn ngữ Java được viết ra với tiêu chí “viết một lần, thực thi mọi nơi.” Hiện tại, chương trình phần mềm viết bằng Java có thể khởi tạo trên mọi nền tảng khác nhau chỉ qua một môi trường thực thi, điều kiện là môi trường đó phải phù hợp hỗ trợ nền tảng đó.

## Lí do ra đời:

- Ban đầu, Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng có tên là Oak.

- Java được phát hành năm 1994, và được Oracle mua lại từ Sun MicroSystem.

- Jaca được tạo ra với tiêu chí WORA.

## Cách Java hoạt động, điều gì xảy ra khi chạy một code java (.java)

- Để đảm bảo tính đa nền, Java sử dụng cơ chế máy ảo của Java (JVM – Java Virtual Machine).

- Bytecode là ngôn ngữ của JVM tương tự như các lệnh nhị phân của các máy tính thực.

- Biên dịch mã nguồn Java (.java) thành bytecode (.class): Một chương trình sau khi được viết bằng ngôn ngữ Java (có phần mở rộng là .java) phải được biên dịch thành tập tin thực thi được trên máy ảo Java (có phần mở rộng là .class). Tập tin thực thi này chứa các chỉ thị dưới dạng mã Bytecode mà máy ảo Java hiểu được phải làm gì.

- Thực thi trên JVM: Khi thực thi chương trình, máy ảo Java lần lượt thông dịch các chỉ thị dưới dạng Bytecode và chuyển đổi thành các chỉ thị dạng nhị phân tùy thuộc vào nền tảng của máy tính thực và thực thi thực sự chúng trên máy tính thực. Nó cũng cung cấp bộ nhớ và hỗ trợ quản lý ngoại lệ và đa luồng.

- Máy ảo thực tế đỏ là một chương trình thông dịch. Vì thế các hệ điều hành khác nhau sẽ có các máy ảo khác nhau. Để thực thi một ứng dụng Java trên một hệ điều hành cụ thể, cần phải cài đặt máy ảo tương tự cho hệ điều hành đó.

## Cấu trúc 1 chương trình Java

- Một project sẽ bao gồm các package (gói), trong mỗi các package sẽ có các Java class (file đuôi .java).

- Cấu trúc của 1 chương trình Java:

package <package\_name>;

import <other\_package>;

public class Cl*ass*Name {

  <Variables (also known as fields)>;

  <constructor method(s)>;

  <other methods>;

}

Trong đó:

- package (gói) mô tả không gian tên chứa các lớp Java, sử dụng ký tự thường và dấu chấm để định nghĩa tên, có thể xem package như một thư mục, còn class chính là các file thuộc thư mục đó.

- import: từ khóa được sử dụng trong Java nhằm để xác định các class hoặc package được sử dụng trong lớp này.

- class: từ khóa nhằm để định nghĩa lớp của Java. Nó đứng trước khai báo tene lớp của Java. Ngoài ra còn có từ khóa public (xác định phạm vi truy cập của lớp).

- variables (hay còn được gọi là attribute) : thuộc tính của lớp, chứa thông tin cụ thể liên quan tới các đối tượng là thể hiện của lớp.

- methods (phương thức): là hàm chứa các hành động thực thi của đối tượng.

- constructors (phương thức khởi tạo).

## Các syntax (cú pháp) cơ bản của Java:

* 1. Khai báo lớp và phương thức:

public class ExampleClass {

    // Phương thức main, điểm bắt đầu của chương trình

    public static *void* main(String[] *args*) {

        // Mã lệnh thực thi

    }

}

* 1. Khai báo biến:

*int* number = 5;

*double* pi = 3.14;

*String* message = "Hello World";

* 1. Câu lệnh điều kiện if-else:

if (condition) {

    // Khối lệnh thực hiện khi điều kiện đúng

} else {

    // Khối lệnh thực hiện khi điều kiện sai

}

* 1. Vòng lặp for, while, do-while:

for(*int* i=0;i<5;i++) {

    // Khối lệnh lặp

}

while(condition) {

    // Khối lệnh lặp

}

do{

    // Khối lệnh lặp

} while(condition);

* 1. Mảng (Array):

*int*[] numbers = {1, 2, 3, 4, 5};

*int*[] numbers = **new** *int*[10];

* 1. Hàm / Phương thức (Methods):

public *void*/*int*/*double*.... myMethod() {

    // Khối lệnh thực thi phương thức

}