**Write once, run anywhere**

Là khi viết mã bằng java, mã nguồn được biên dịch thành bytecode (ngôn ngữ trung gian dễ dàng được chuyển đổi), sẽ được chuyển đổi thành mã máy bởi JVM.

**JDK**

Là viết tắt của Java Development Kit. Đây là một công cụ phát triển java, bao gồm JRE và các công cụ phát triển khác (như trình biên dịch Java, các công cụ để tạo và quản lý file, và các công cụ khác để phát triển và thử nghiệm ứng dụng Java).

**JRE**

Là viết tắt của Java Runtime Environment. Đây là một môi trường thực thi cho các ứng dụng Java, bao gồm JVM (Java Virtual Machine) và các thư viện cần thiết để chạy các ứng dụng Java.

**Compile**

Là quá trình chuyển đổi mã nguồn Java có đuôi .java sang mã bytecode có đuôi .class bởi trình biên dịch Java

**Runtime**

Là một lớp trong Java cung cấp một số phương thức để tương tác với môi trường thực thi của hệ thống. Được sử dụng để lấy thông tin về hệ thống và thực hiện các hoạt động cơ bản như tạo tiến trình, tắt hệ thống, lấy thông tin tài nguyên hệ thống,…

Một số phương thức quan trọng trong lớp Runtime là:

* **availableProcessors()**: trả về số lượng bộ xử lý có sẵn trên hệ thống.
* **freeMemory()**: trả về số lượng bộ nhớ không sử dụng.
* **totalMemory()**: trả về tổng số lượng bộ nhớ có sẵn trên hệ thống.
* **maxMemory()**: trả về số lượng bộ nhớ tối đa có thể được sử dụng bởi JVM.
* **exec(String command)**: thực thi một lệnh được chỉ định trên hệ thống.
* **halt(int status)**: tắt hệ thống với mã trạng thái được chỉ định.

**Giá trị mặc đinh của kiểu dữ liệu nguyên thủy.**Boolean: false

Byte: 0

Short: 0

Int: 0

Long: 0L

Float: 0.0f

Double: 0.0d

Char: ‘\u0000’

**Các vòng lặp(for, while, do while)(đặc điểm)**

Vòng lặp for sẽ biết trước số lần lặp.

Vòng lặp while sẽ không biết trước số lần lặp.

Vòng lặp do-while sẽ lặp ít nhất 1 lần, rồi mới kiểm tra điều kiện.

**Array (reference || value ? ) chứng minh?**

Mảng là một kiểu dữ liệu tham chiếu, không phải kiểu dữ liệu giá trị

Khi khai báo một mảng trong Java, một biến tham chiếu được tạo ra để trỏ đến vùng nhớ lưu trữ các phần tử của mảng. Biến tham chiếu này chứa địa chỉ bộ nhớ của vùng nhớ chứa các phần tử của mảng. Khi truy cập đến phần tử của mảng, là đang sử dụng biến tham chiếu đấy để truy cập đến phần tử của mảng tại địa chỉ bộ nhớ tương ứng.

**passByReference & passByValue**

Trong tham trị, giá trị của biến được truyền vào phương thức được sao chép và truyền vào phương thức. Bất kì thay đổi nào trong phương thức thì không ảnh hưởng đến giá trị ban đầu.

Trong tham chiếu, tham chiếu đến vùng nhớ của biến được truyền vào phương thức được truyền vào thay vì giá trị thực tế của biến đó. Bất kì thay đổi nào được thực hiện trong phương thức đối với biến truyền vào đều ảnh hưởng đến giá trị của biến ban đầu.

**OOP là gì**

OOP là viết tắt của Object-Oriented Programming (Lập trình hướng đối tượng). là một phương pháp lập trình sử dụng đối tượng để mô hình hóa thế giới thực.

Trong OOP có 4 khái niệm:

*Tính kế thừa: cho phép một lớp con kế thừa các thuộc tính và phương thức của một lớp cha và mở rộng chúng hoặc ghi đè lên chúng.*

*Tính đa hình cho phép các đối tượng của các lớp khác nhau có thể được xử lý bằng cách sử dụng cùng 1 phương thức với cùng 1 tên, nhưng có thể thực hiện khác nhau tùy thuộc vào lớp của đối tượng đó.*

*Tính đóng gói cho phép ẩn giấu thông tin bên trong các đối tượng và chỉ cho phép truy cập thông qua các phương thức công khai.*

*Tính trừu tượng cho phép ta tập trung vào các khía cạnh quan trọng của đối tượng mà không cần quan tâm đến các chi tiết bên trong.*

**Class & quy tắc đặt tên**

Trong Java, một lớp là một bản thiết kế mô tả về các đối tượng sẽ được tạo ra

Quy tắc đặt tên cho lớp:

Viết hoa chữ cái đầu và không bắt đầu bằng số

Quy tắc đặt tên cho các thành phần của lớp:

Đặt tên theo quy tắc Camelcase