

# Overview

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun Microsystems năm 1991. Ban đầu ngôn ngữ này được gọi là Oak (có nghĩa là cây sồi; do bên ngoài cơ quan của ông Gosling có trồng nhiều loại cây này), họ dự định ngôn ngữ đó thay cho C++, nhưng các tính năng giống Objective C. Không nên lẫn lộn Java với JavaScript, hai ngôn ngữ đó chỉ giống tên và loại cú pháp như C. Công ty Sun Microsystems đang giữ bản quyền và phát triển Java thương mại. Tháng 04/2011, công ty Sun Microsystems tiếp tục cho ra bản JDK 1.6.24.

Java được tạo ra với tiêu chí "Viết (code) một lần, thực thi khắp nơi" ("Write Once, Run Anywhere" (WORA)). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó. Môi trường thực thi của Sun Microsystems hiện hỗ trợ Sun Solaris, Linux, Mac OS, FreeBSD & Windows. Ngoài ra, một số công ty, tổ chức cũng như cá nhân khác cũng phát triển môi trường thực thi Java cho những hệ điều hành khác như BEA, IBM, HP.... Trong đó đáng nói đến nhất là IBM Java Platform hỗ trợ Windows, Linux, AIX & z/OS.

## Điểm mạnh của Java:

- Object Oriented: Là một ngôn ngữ lập trình hướng đối tượng
- Platform Independent: Không giống như nhiều ngôn ngữ lập trình khác bao gồm C và C++, khi Java được biên dịch, nó không được biên dịch thành mã máy cụ thể của nền tảng, thay vào đó là mã byte độc lập với nền tảng. Mã byte này được Máy ảo (JVM) biên dịch trên bất kỳ nền tảng nào nó đang được chạy.
- Simple: Java được thiết kế để dễ học. Nếu bạn hiểu khái niệm cơ bản về OOP Java, nó sẽ dễ dàng thành thạo.
- Secure: Với tính năng bảo mật của Java, nó cho phép phát triển các hệ thống chống virus, không có virus. Kỹ thuật xác thực dựa trên mã hóa khóa công khai.
- Architecture-neutral: Trình biên dịch Java tạo ra một định dạng đối tượng trung lập kiến trúc, làm cho mã được biên dịch có thể thực thi được trên nhiều bộ xử lý
- Portable: Trung lập về kiến trúc và không có các khía cạnh phụ thuộc triển khai của đặc tả làm cho Java có thể di động. Trình biên dịch trong Java được viết bằng ANSI C với ranh giới khả năng di chuyển sạch, là tập hợp con POSIX.
- Robust: Java nỗ lực để loại bỏ các tình huống dễ xảy ra lỗi bằng cách tập trung vào kiểm tra lỗi khi biên dịch và kiểm tra runtime.
- Multithreaded: Với tính năng đa luồng của Java, có thể viết các chương trình thực hiện đồng thời nhiều tác vụ. Tính năng này cho phép các nhà phát triển xây dựng các ứng dụng

tương tác có thể chạy trơn tru.

- Interpreted: Mã byte Java được dịch nhanh chóng theo hướng dẫn của máy gốc và không được lưu trữ ở bất cứ đâu
- High Performance: Java hỗ trợ và xử lý tốt các bài toán yêu cầu hiệu suất cao
- Distributed: Java được thiết kế cho môi trường phân tán của internet.
- Dynamic: Java được coi là năng động hơn C hoặc C++ vì nó được thiết kế để thích ứng với môi trường phát triển. Các chương trình Java có thể mang lượng thông tin lớn về thời gian chạy có thể được sử dụng để xác minh và giải quyết các truy cập vào các đối tượng trong thời gian chạy

## 4 tính chất của lập trình hướng đối tượng trong Java

### 1. Tính đóng gói (encapsulation) và che giấu thông tin (information hiding)

- Trạng thái của đối tượng được bảo vệ không cho các truy cập từ code bên ngoài như thay đổi trong thái hay nhìn trực tiếp. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã.
- Đây là tính chất đảm bảo sự toàn vẹn, bảo mật của đối tượng Trong Java, tính đóng gói được thể hiện thông qua phạm vi truy cập (access modifier). Ngoài ra, các lớp liên quan đến nhau có thể được gom chung lại thành package.

### 2. Tính kế thừa (Inheritance)

- Tính kế thừa là khả năng cho phép ta xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có. Lớp đã có gọi là lớp Cha, lớp mới phát sinh gọi là lớp Con và đương nhiên kế thừa tất cả các thành phần của lớp Cha, có thể chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

### 3. Tính đa hình (polymorphism)

- Khi một tác vụ được thực hiện theo nhiều cách khác nhau được gọi là tính đa hình. Đối với tính chất này, nó được thể hiện rõ nhất qua việc gọi phương thức của đối tượng. Các phương thức hoàn toàn có thể giống nhau, nhưng việc xử lý luồng có thể khác nhau. Nói cách khác: Tính đa hình cung cấp khả năng cho phép người lập trình gọi trước một phương thức của đối tượng, tuy chưa xác định đối tượng có phương thức muốn gọi hay không. Đến khi thực hiện (run-time), chương trình mới xác định được đối tượng và gọi phương thức tương ứng của đối tượng đó. Kết nối trễ giúp chương trình được uyển chuyển hơn, chỉ yêu cầu đối tượng cung cấp đúng phương thức cần thiết là đủ.
- Trong Java, chúng ta sử dụng nạp chồng phương thức (method overloading) và ghi đè phương thức (method overriding) để có tính đa hình.
- Nạp chồng (Overloading): Đây là khả năng cho phép một lớp có nhiều thuộc tính, phương thức cùng tên nhưng với các tham số khác nhau về loại cũng như về số lượng. Khi được gọi, dựa vào tham số truyền vào, phương thức tương ứng sẽ được thực hiện.
- Ghi đè (Overriding): là hai phương thức cùng tên, cùng tham số, cùng kiểu trả về nhưng thằng con viết lại và dùng theo cách của nó, và xuất hiện ở lớp cha và tiếp tục xuất hiện ở lớp con. Khi dùng override, lúc thực thi, nếu lớp Con không có phương thức riêng, phương

thức của lớp Cha sẽ được gọi, ngược lại nếu có, phương thức của lớp Con được gọi.

#### 4. Tính trừu tượng (abstraction)

- Tính trừu tượng là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép bạn loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.
- Tính trừu tượng giúp bạn tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.
- Trong Java, chúng ta sử dụng abstract class và abstract interface để có tính trừu tượng.

---

Revision #4

Created 22 September 2019 15:55:34 by Admin

Updated 19 October 2019 16:55:59 by Admin