***Giới thiệu ngôn ngữ lập trình Java***

## Giới thiệu

Là một trong những ngôn ngữ lập trình mạnh và được sử dụng đông đảo trong phát triển phần mềm, các trang web, game hay ứng dụng trên các thiết bị di động, ngôn ngữ lập trình Java đã và đang trở nên lớn mạnh hơn bao giờ hết. Bài viết này xin được trình bày một vài đặc điểm tiêu biểu của ngôn ngữ lập trình Java để các bạn có được một cái nhìn tổng quát về ngôn ngữ này.

## Đối tượng hướng đến

Bài viết này hướng đến những lập trình viên muốn tìm hiểu, nghiên cứu về ngôn ngữ lập trình Java. Bài viết không yêu cầu bạn đọc có kiến thức trước về ngôn ngữ lập trình Java.

## Lịch sử phát triển ngôn ngữ lập trình Java

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

## Đặc điểm của ngôn ngữ lập trình Java

### 1. Tựa C++, hướng đối tượng hoàn toàn

Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++.

Tuy nhiên, trong Java **thao tác với con trỏ bị lược bỏ** nhằm đảo bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được loại bỏ khỏi Java. Các vấn đề này xin được làm rõ ở những bài viết sau.

### 2. Độc lập phần cứng và hệ điều hành

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng "**cross-platform**”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ **mã nguồn**và cấp độ **nhị phân**.

**Ở cấp độ mã nguồn:** Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

**Ở cấp độ nhị phân:** Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

### 3. Ngôn ngữ thông dịch

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch. Bạn có thể xem thêm tại đây - [Phân Biệt Compile Và Interpret](https://www.stdio.vn/articles/read/23/phan-biet-compile-va-interpret).

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ **vừa biên dịch vừa thông dịch**. Cụ thể như sau

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã **byte code**. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành **machine code**  (hay native code) khi nhận được yêu cầu chạy chương trình.

**Ưu điểm** : Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.

**Nhược điểm** : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).

### 4. Cơ chế thu gom rác tự động

Khi tạo ra các đối tượng trong Java, JRE sẽ tự động cấp phát không gian bộ nhớ cho các đối tượng ở trên heap.

Với ngôn ngữ như C \ C++, bạn sẽ phải yêu cầu hủy vùng nhớ mà bạn đã  cấp phát, để tránh việc thất thoát vùng nhớ. Tuy nhiên vì một lý do nào đó, bạn không hủy một vài vùng nhớ, dẫn đến việc thất thoát và làm giảm hiệu năng chương trình.

Ngôn ngữ lập trình Java hỗ trợ cho bạn điều đó, nghĩa là bạn không phải  tự gọi hủy các vùng nhớ. Bộ thu dọn rác của Java sẽ theo vết các tài nguyên đã được cấp. Khi **không có tham chiếu** nào đến vùng nhớ, bộ thu dọn rác sẽ tiến hành thu hồi vùng nhớ đã được cấp phát.

### 5. Đa luồng

Java hỗ trợ lập trình đa tiến trình (**multithread**) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority...).

### 6. Tính an toàn và bảo mật

**Tính an toàn**

Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.

* Dữ liệu phải được khai báo tường minh.
* Không sử dụng con trỏ và các phép toán với con trỏ.
* Java kiểm soát chặt chẽ việc truy nhập đến mảng, chuỗi. Không cho phép sử dụng các kỹ thuật tràn. Do đó các truy nhập sẽ không vượt quá kích thước của mảng hoặc chuỗi.
* Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.
* Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

**Tính bảo mật**

Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.

* Mức 1 : Chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.
* Mức 2 : Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.
* Mức 3 : Trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.
* Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.