PHẦN 1: **Java: Lịch sử và nền tảng**

**1. Lịch sử ra đời của Java**

-Năm 1990, **James Gosling** được công ty Sun Microsystems giao nhiệm vụ xây dựng phần mềm cho các thiết bị điện tử dân dụng như lò nướng, …Lúc đầu, Gosling và các cộng sự định chọn ngôn ngữ C++ nhưng thấy rằng C++ có những hạn chế. Chương trình viết bằng C++ khi chuyển sang chạy trên một hệ thống có bộ vi xử lý khác thì đòi hỏi **phải biên dịch lại**.

-Gosling quyết định xây dựng một ngôn ngữ mới dựa trên ngôn ngữ C/C++ và đặt tên là **Oak**(cây sồi).

-Năm 1995, **Oak**đổi tên thành **Java**

-Năm 2009, Oracle mua lại Sun Microsystems và quyền phát triển Java thuộc về Oracle

## ****2. Đặc điểm của ngôn ngữ lập trình Java****

Slogan nổi tiếng của Java là “**Write once, run everywhere**” – “**Viết một lần, chạy mọi nơi**”.

**Các đặc điểm của Java:**

**Ưu điểm:**

– **simple (đơn giản)**: cú pháp của nó dựa trên C++. Nhưng bỏ nhiều đặc điểm gây bối rối như con trỏ tường minh, nạp chồng toán tử,…

-**“Cleaner” trong cú pháp**

– **object-oriented (hướng đối tượng)**: Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Trong Java, mọi thứ đều là đối tượng.

– **distributed (phân tán)**: nhắm đến phân bố ứng dụng trên mạng, ứng dụng độc lập platform.

– **interpreted (thông dịch)**: chương trình nguồn Java có đuôi **\*.java**, đầu tiên được biên dịch thành tập tin có đuôi **\*.class** và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

– **robust (mạnh mẽ)**: Java đã loại bỏ một số loại cấu trúc lập trình dễ bị lỗi có trong các ngôn ngữ khác, có tính năng xử lý ngoại lệ,…

– **secure (bảo mật)**: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one-way hash) hoặc mã hóa công cộng (public key),…

– **architecture-neutral (kiến trúc trung lập)**: trình biên dịch Java sinh ra định dạng tập tin đối tượng có kiến trúc trung lập, có khả năng thực thi trên nhiều bộ xử lý, với sự hiện diện của hệ thống thực thi Java.

– **portable (khả chuyển)**: có thể mang Java Bytecode tới bất cứ nền tảng nào.

– **performance (hiệu suất cao)**: nhờ vào trình thu gom rác (Garbage Collection). Nó giúp giải phóng bộ nhớ đối với các đối tượng không được dùng đến.

– **multithreaded (đa luồng)**: Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc.

– **dynamic (linh động)**: Java được xem là linh động hơn C/C ++. Bởi vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

Nhược điểm

**- Sử dụng nhiều bộ nhớ hơn c ++ hoặc c #**

**- Less efficient(kém hiệu quả)**