

LẬP TRÌNH JAVA

Bài 1: Làm quen với ngôn ngữ Java

Thông tin giảng viên

- ▣ Họ tên: **NGHIÊM MINH HOÀNG**
- ▣ Công tác:
Co-founder,
Giám đốc trung nghiên cứu và phát triển
CÔNG TY CỔ PHẦN OWS VIỆT NAM
- ▣ Email: **hoangnm@ows.vn**

NỘI DUNG

- ▣ Giới thiệu chung về ngôn ngữ Java
- ▣ Môi trường để lập trình Java
- ▣ Chương trình Java đầu tiên
- ▣ Dịch và chạy chương trình Java

Giới thiệu chung về ngôn ngữ Java và ứng dụng

Java là gì?

- ▣ Java là một ngôn ngữ lập trình Hướng Đối Tượng được phát triển bởi Sun Microsystems (nay là Oracle)
- ▣ Là một ngôn ngữ lập trình khá trẻ
 - ▣ Ban đầu được sử dụng để xây dựng ứng dụng điều khiển các bộ xử lý bên trong các thiết bị điện tử dân dụng.
 - ▣ Bắt đầu được sử dụng từ năm 1995



Green Team and James Gosling
(the leader)

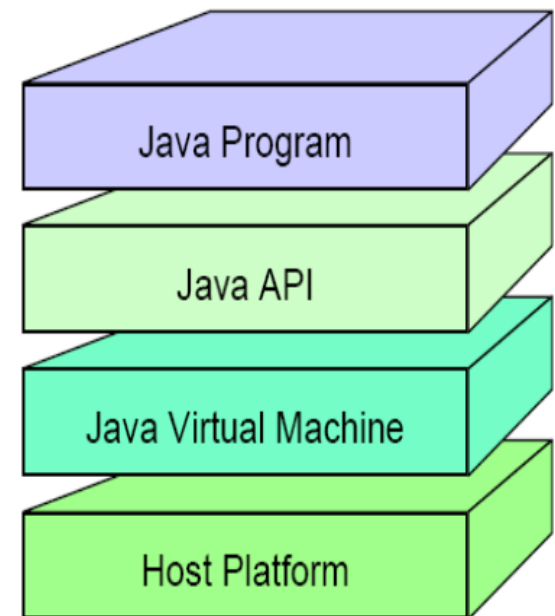
Java là gì?

- ▣ Java không chỉ là một ngôn ngữ mà còn là một công nghệ, một nền tảng phát triển.
- ▣ Java có một cộng đồng phát triển mạnh mẽ
 - ▣ Một tập hợp các thư viện với số lượng lớn (từ Sun và các nguồn khác)
- ▣ Tiêu chí phát triển:
“Write Once, Run Anywhere”



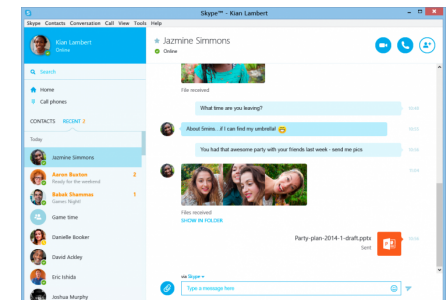
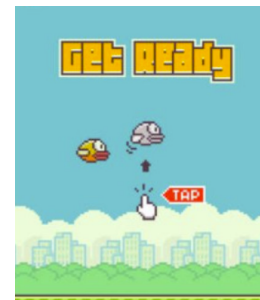
Java Platform

- ▣ Platform là môi trường phát triển hoặc triển khai.
- ▣ Java platform có thể chạy trên mọi hệ điều hành
 - ▣ Các platform khác phụ thuộc vào phần cứng
 - ▣ Java platform cung cấp
 - ▣ Máy ảo Java - Java Virtual Machine
 - ▣ Giao diện lập trình ứng dụng - Application Programming Interface (API)



Ứng dụng của Java

- ❑ Thiết bị điện tử dân dụng
- ❑ Thiết bị di động
- ❑ Ứng dụng Desktop
- ❑ Ứng dụng Web
- ❑ Hệ thống Doanh nghiệp
- ❑ ...



Tính năng của Java

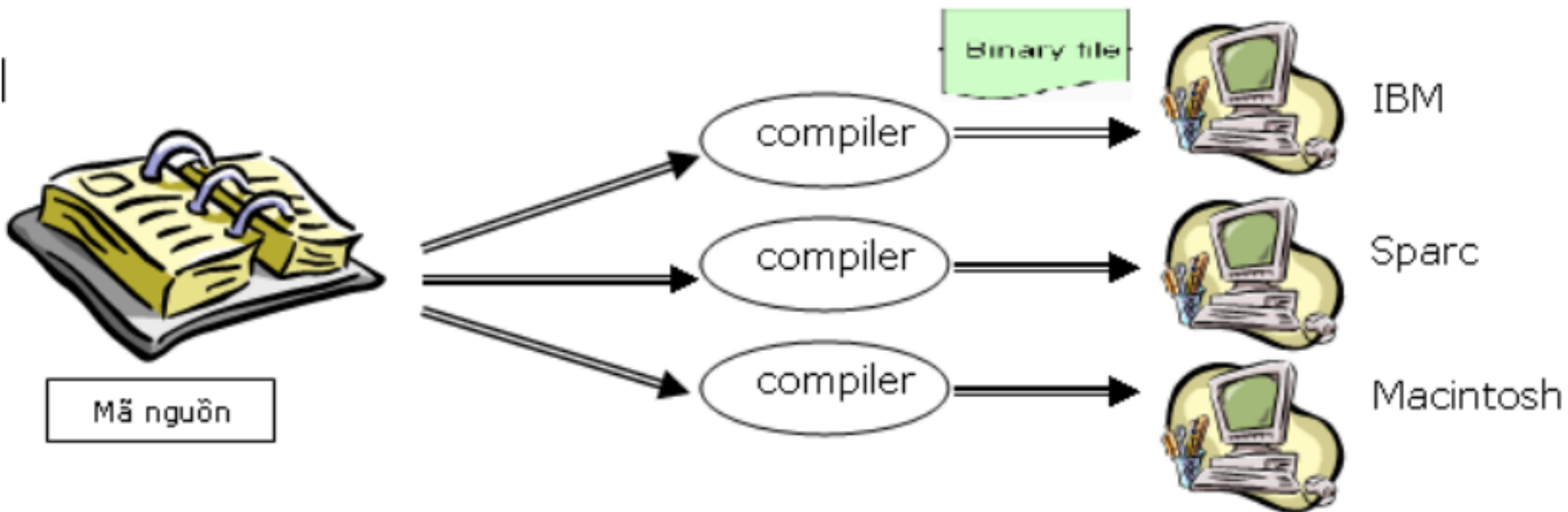
- ▣ Ngôn ngữ lập trình mạnh
- ▣ Hướng đối tượng
- ▣ Dễ học, dễ đọc
- ▣ Độc lập nền tảng
- ▣ Bảo mật cao
- ▣ Hỗ trợ lập trình đa luồng và lập trình phân tán

Môi trường để lập trình Java

Máy ảo Java - JVM

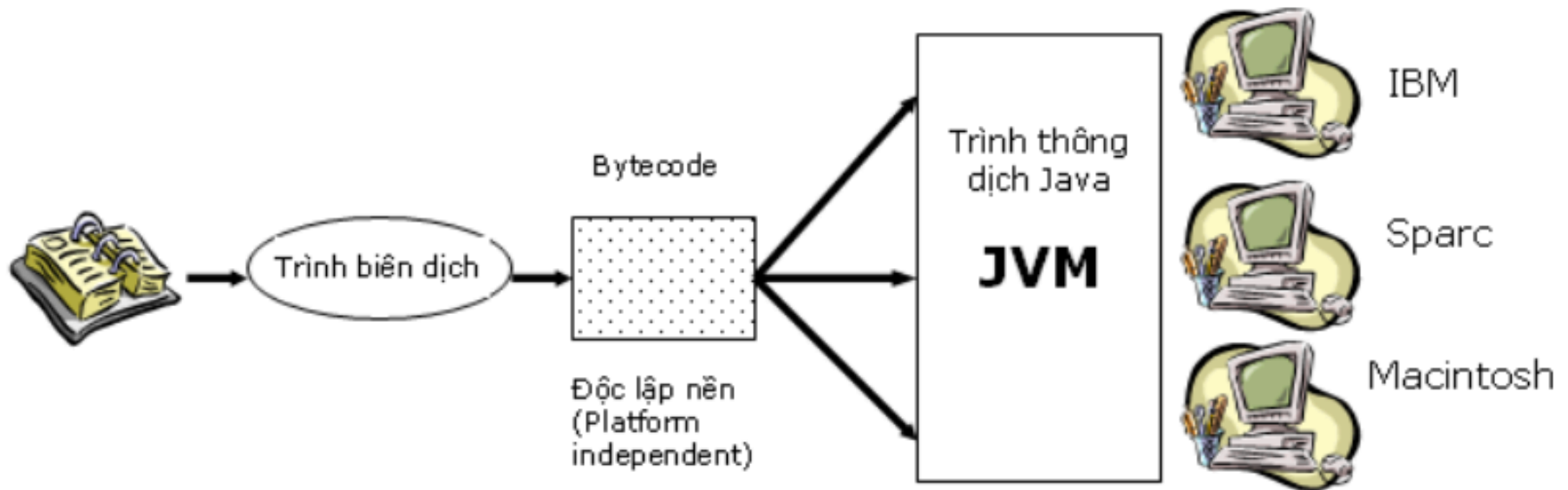
- ▣ Máy ảo Java là trái tim của ngôn ngữ Java
 - ▣ Đem đến cho các chương trình Java có khả năng viết một lần nhưng chạy được ở mọi nơi
- ▣ Tạo ra môi trường bên trong để thực thi lệnh:
 - ▣ Nạp các file .class
 - ▣ Quản lý bộ nhớ
 - ▣ Dọn “rác”
- ▣ Trình thông dịch “Just in time – JIT”
 - ▣ Chuyển tập lệnh bytecode thành mã máy cụ thể cho từng loại CPU

Máy ảo Java - JVM



Biên dịch truyền thống

Máy ảo Java - JVM



Java Development Kit - JDK

- ▣ Bộ công cụ cho nhà phát triển
- ▣ Cung cấp phần mềm, thư viện, công cụ cho việc compiling, debugging và executing 1 chương trình Java
- ▣ JDK bao gồm 2 công cụ chính quan trọng:
 - ▣ javac (compiler)
 - ▣ java (interpreter)

Java Runtime Environment - JRE

- ▣ Cung cấp môi trường runtime
- ▣ Triển khai của JVM
- ▣ Bao gồm tập các thư viện và các file mà JVM sử dụng trong quá trình runtime

Cài đặt môi trường

- ▣ Cài đặt Java Development Kit – JDK

- ▣ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- ▣ Cài đặt IDE

- ▣ Notepad / Notepad++

- ▣ Eclipse

- ▣ NetBeans

- ▣ IntelliJ IDEA

Chương trình Java đầu tiên

B1 - Khai báo class

- ▣ Sử dụng từ khoá **class** để khai báo class
- ▣ Tên **class** trùng với tên file
- ▣ Lưu file đuôi **.java**

```
class <class_name> {  
  
}
```

B1 - Khai báo class

cú pháp

```
class <class_name> {  
  
}
```

ví dụ

```
class HelloWorld {  
  
}
```

B2 - Viết hàm main

```
public static void main(String[] args) {  
  
}
```

- ❑ **public** là keyword để chỉ phạm vi truy cập của 1 phương thức
- ❑ **static** là keyword cho phép thực hiện phương thức từ bên ngoài của class mà không cần tạo thể hiện của class
- ❑ **void** là keyword chỉ ra phương thức này không có kiểu dữ liệu trả về
- ❑ **args** là một mảng giá trị kiểu **String**

B3 - Viết nội dung hàm

```
System.out.println("Welcome to the world of Java");
```

- ▣ **System.out.println** là câu lệnh để in một đoạn text ra màn hình console

B4 - Lưu chương trình

- ▣ Chương trình được lưu dưới dạng file mở rộng .java
- ▣ Tên file phải trùng với tên class khai báo trong phần mã nguồn

Dịch và chạy chương trình Java

B1 – Biên dịch

- ▣ Sử dụng Java compiler, *javac.exe*
- ▣ Cú pháp

javac [option] tên_file.java

B1 – Biên dịch

- ▣ Sử dụng Java compiler, *javac.exe*
- ▣ Cú pháp

javac [option] tên_file.java

- ▣ Sau khi biên dịch sẽ thu được 1 file có phần mở rộng .class, đây chính là file chứa mã bytecode sau khi được biên dịch

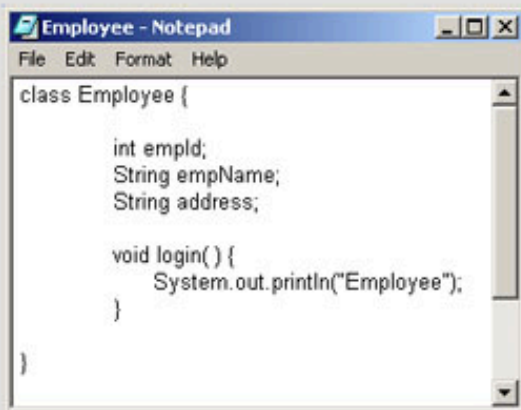
B2 - Thực thi

- ▣ Sử dụng Java interpreter, *java.exe*
- ▣ Cú pháp

java [option] tên_file.class

B2 - Thực thi

Source Code: Employee.java



```
class Employee {  
    int empId;  
    String empName;  
    String address;  
  
    void login() {  
        System.out.println("Employee");  
    }  
}
```

