



Giới thiệu về ngôn ngữ Java

Trường ĐH Công nghệ, ĐHQGHN

Nội dung chính

- Lịch sử ngôn ngữ lập trình
- Lịch sử ngôn ngữ Java
- Môi trường lập trình Java, máy ảo, và JDK
- Giới thiệu về các loại chương trình Java
- Dịch và chạy chương trình

Phần mềm ngày càng phức tạp



- Kích thước ngày càng lớn*
 - LibreOffice: 9M dòng lệnh
 - Android: 13.3M dòng lệnh
 - NetBeans IDE: 97.2M dòng lệnh
- Người dùng ngày càng đòi hỏi nhiều chức năng, đặc biệt là chức năng *thông minh*
- Phần mềm luôn cần được sửa đổi

Nguồn: <https://www.openhub.net>

- Cần kiểm soát chi phí
 - Chi phí phát triển
 - Chi phí bảo trì
- Giải pháp chính là **sử dụng lại**
 - Giảm chi phí và thời gian phát triển
 - Nâng cao chất lượng (được sử dụng lại chứng tỏ có chất lượng tốt)

Để sử dụng lại (mã nguồn)



- Cần dễ hiểu
- Được coi là chính xác
- Có **giao diện** rõ ràng
- Không/ít yêu cầu thay đổi khi sử dụng trong phần mềm mới
- Có thể mở rộng nếu cần thiết

Các phương pháp lập trình

- Lập trình không có cấu trúc
- Lập trình có cấu trúc (lập trình thủ tục)
- Lập trình logic
- Lập trình hàm
- Lập trình hướng đối tượng

Java



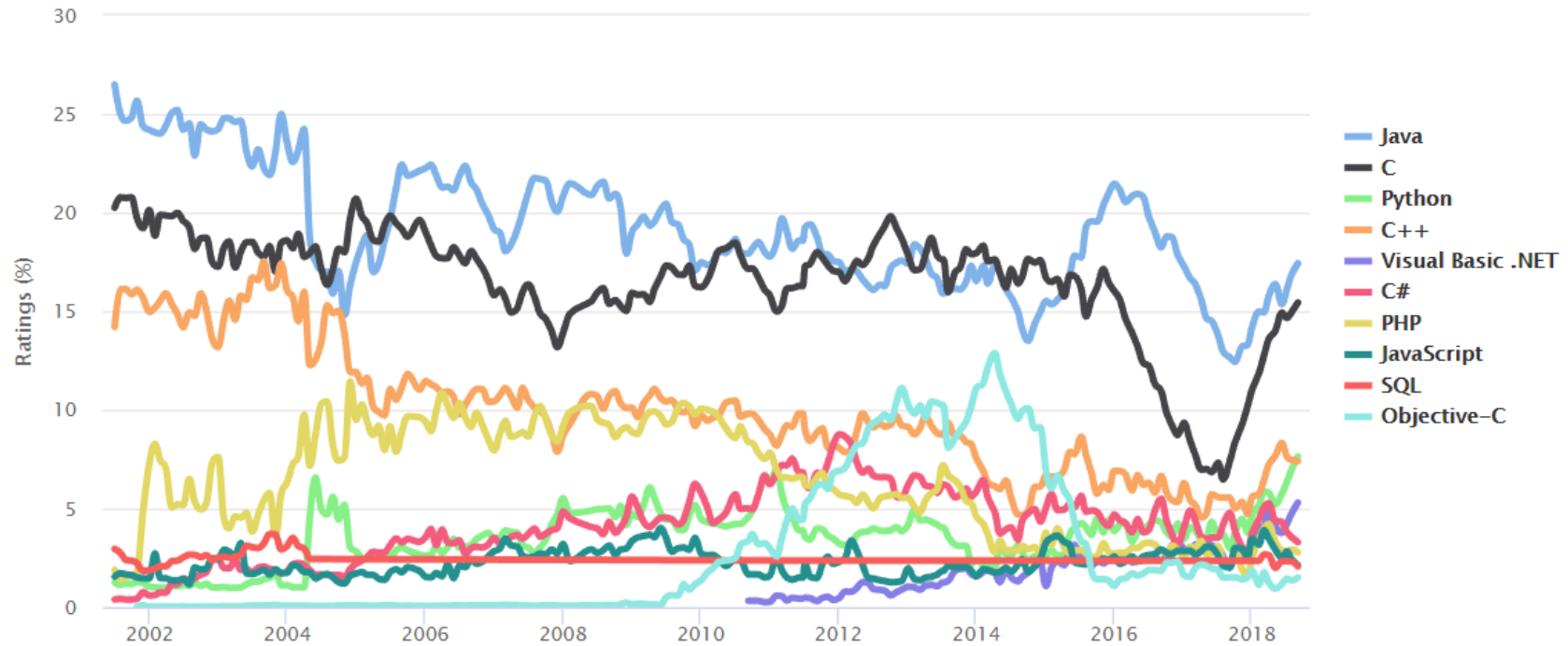
- 1991: phát triển bởi Sun Microsystems như là một ngôn ngữ lập trình cho môi trường nhúng
 - Tên ban đầu là Oak
- Java 1.0.2, 1.1
 - “Write Once, Run Anywhere”
 - Chậm
 - Được sử dụng cho các ứng dụng Web (applets)
- Java 2 (versions 1.2 - 1.4)
 - Nhanh và mạnh hơn,
 - 3 nền tảng: J2ME, J2SE, J2EE
- Java 5, 6, 7, 8 (versions 1.5-1.6...)
 - Được cải tiến nhiều hơn nữa

Thống kê về ngôn ngữ lập trình

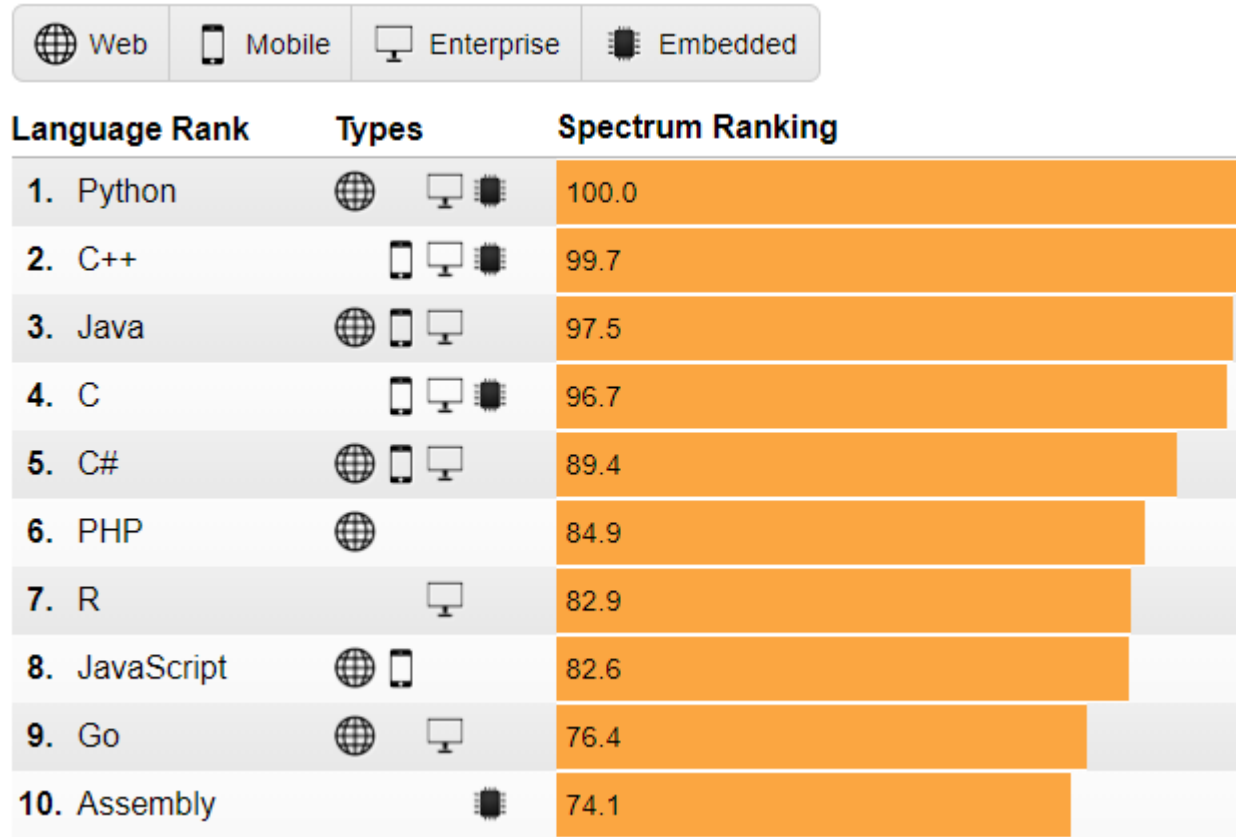


TIOBE Programming Community Index

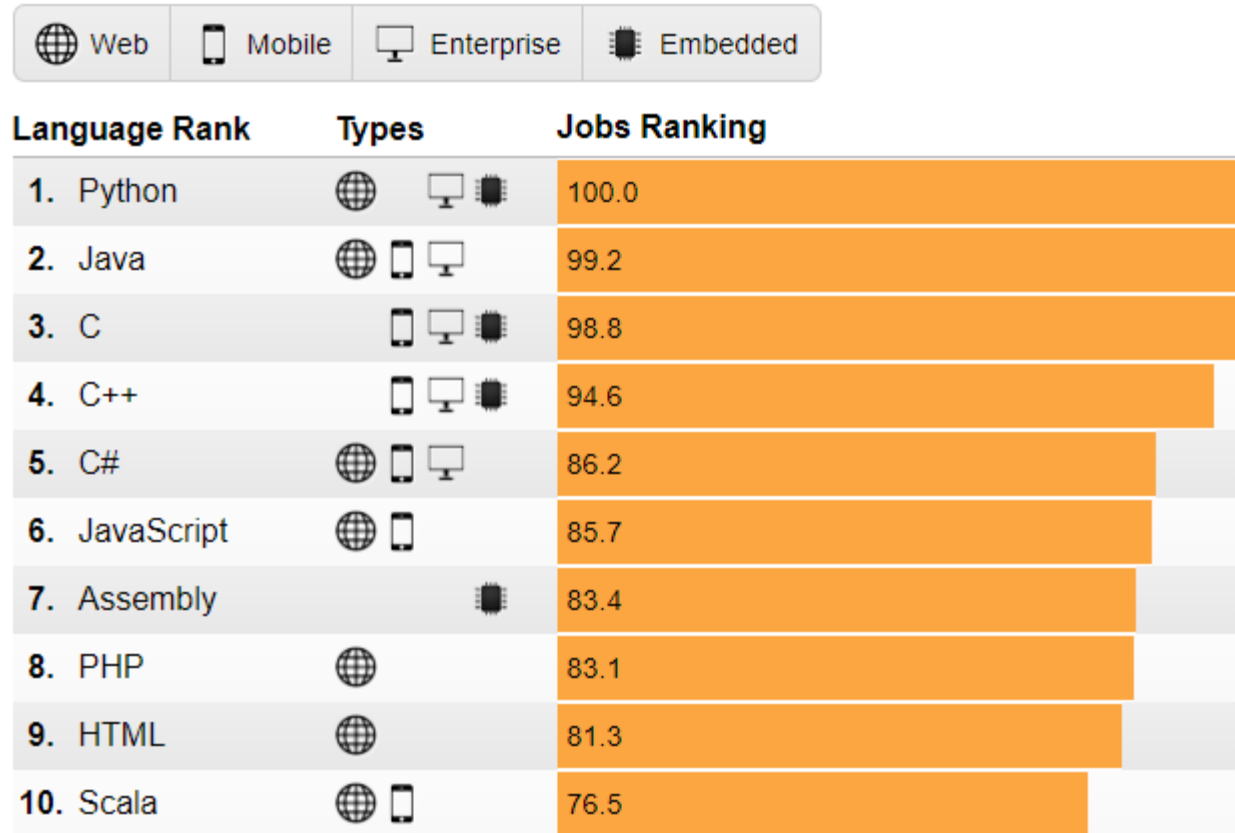
Source: www.tiobe.com



Thống kê của IEEE



Thống kê của IEEE (theo nhu cầu tuyển dụng)



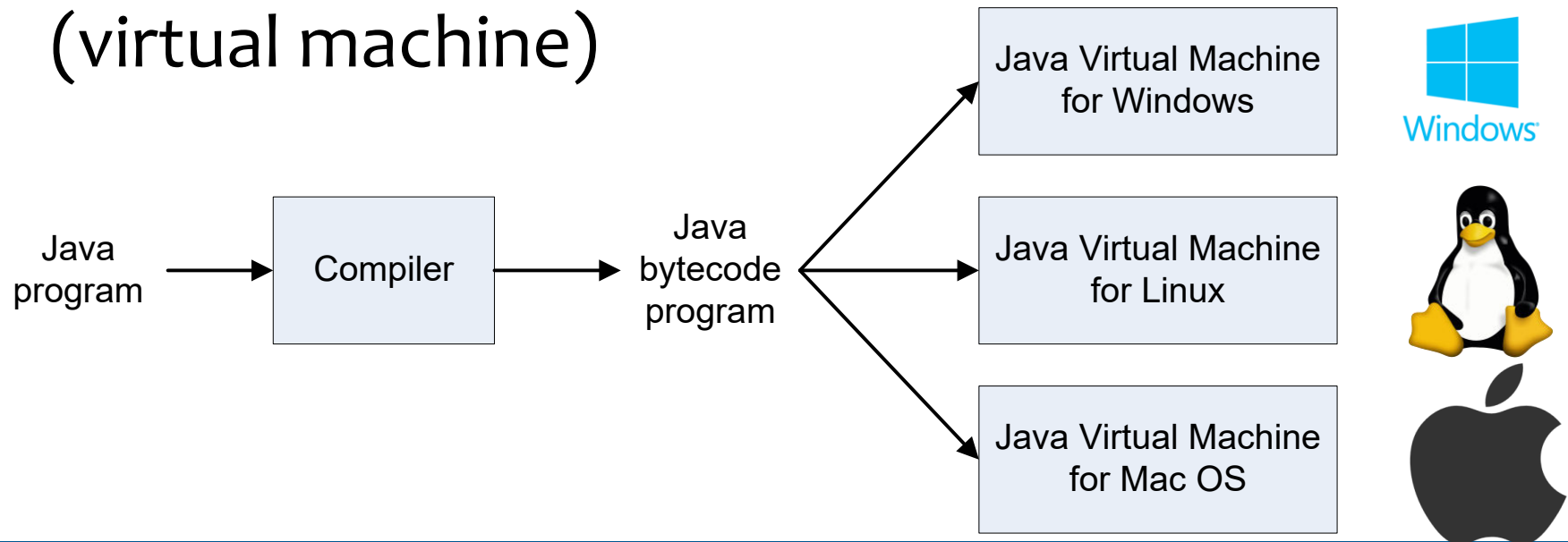
Cấu trúc của chương trình Java



- Tập hợp của các lớp đối tượng
- Thông thường mỗi lớp được mô tả trong 1 tệp mã nguồn (trùng tên với tên lớp)
 - Tăng tính độc lập
 - Dễ sửa đổi, tiết kiệm thời gian biên dịch

Biên dịch

- Mã nguồn Java được biên dịch thành mã bytecode
- Bytecode độc lập với nền tảng
- Bytecode được thực thi qua máy ảo Java (virtual machine)



Java Virtual Machine (JVM)



- JVM được hiện thực phụ thuộc vào các nền tảng (hardware, OS)
- Đảm bảo các chương trình Java (bytecode) có thể thực thi trên các nền tảng khác nhau (platform-independent)
- Đảm bảo an ninh
- Thường được hiện thực như là một phần mềm
 - JRE - Java Runtime Environment
- Java platform: JVM + APIs

Các loại ứng dụng Java



- Ứng dụng để bàn – *Java Standard Edition (Java SE)*
- Ứng dụng phân tán, ứng dụng chủ – *Java Enterprise Edition (Java EE)*
- Ứng dụng di động – *Java Micro Edition (Java ME)*
- Ứng dụng trên thẻ - Java Card

Chương trình HelloWorld



trùng tên lớp

HelloWorld.java:

Lớp

Tên lớp

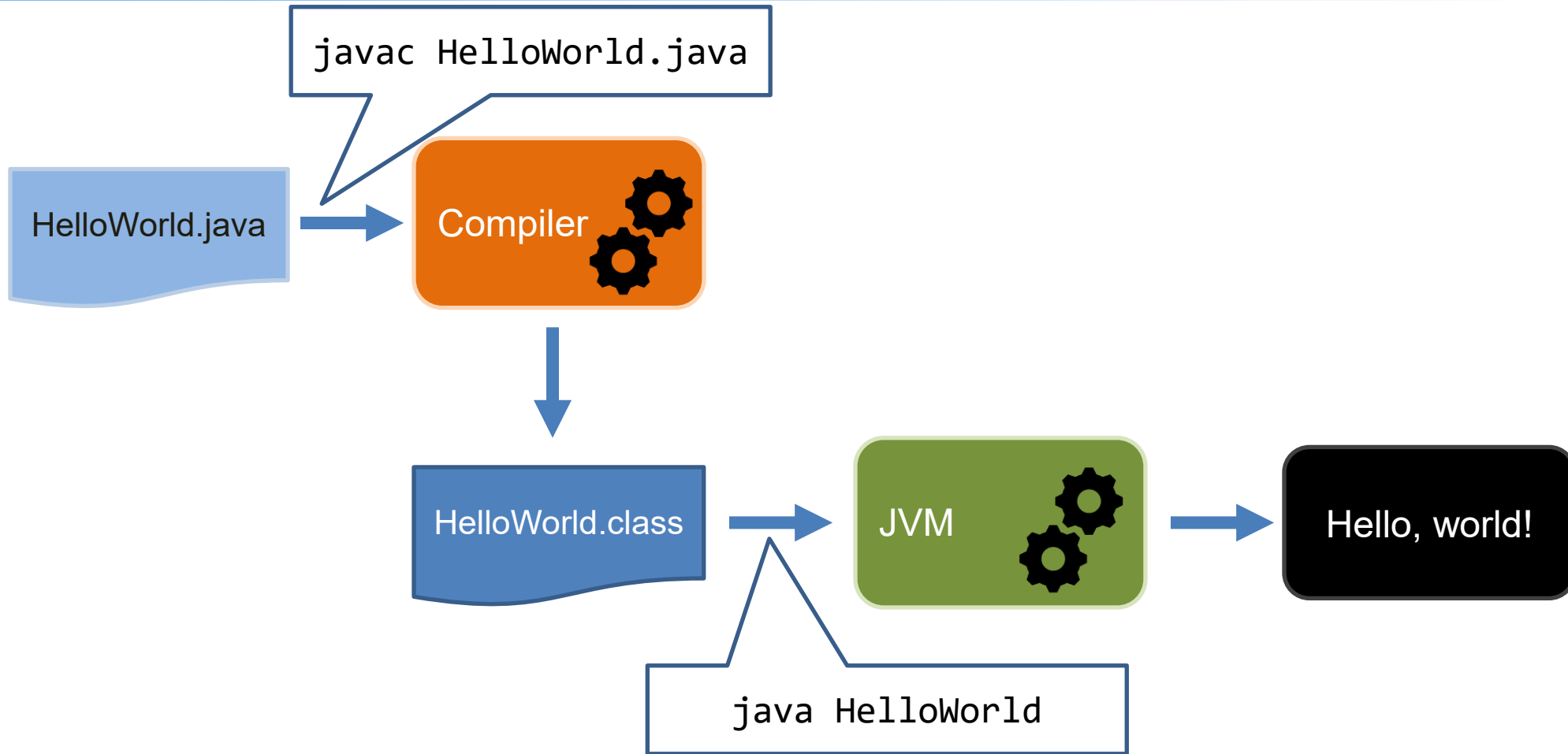
```
public class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Hello, world");  
    }  
}
```

Tên phương thức

Phát biểu trong phương thức

public, từ khóa quy định mức truy xuất

Chương trình HelloWorld



Ứng dụng với nhiều hơn một lớp



2 lớp được hiện thực bởi 2 file

TestGreeting.java:

```
public class TestGreeting {  
    public static void main(String[] args) {  
        Greeting gr = new Greeting();  
        gr.greet();  
    }  
}
```

Greeting.java:

```
public class Greeting {  
    public void greet() {  
        System.out.print("Hi there!");  
    }  
}
```

Biên dịch và thực thi

- Compile

```
javac TestGreeting.java
```

(*Greeting.java* được dịch tự động)

- Run

```
java TestGreeting
```

```
%> javac TestGreeting.java
%> java TestGreeting
Hi there!
```

JDK – Java Development Kit



- Môi trường phát triển ứng dụng Java
- Các thành phần chính
 - *javac*: trình biên dịch, chuyển mã nguồn Java thành bytecode
 - *java*: trình thông dịch
 - *javadoc*: công cụ sinh tài liệu từ các chú thích trong mã nguồn
 - *jdb*: trình gỡ lỗi

Tổng kết



- Lịch sử ngôn ngữ lập trình
- Lịch sử ngôn ngữ Java
- Môi trường lập trình Java, máy ảo, và JDK
- Giới thiệu về các loại chương trình Java
- Dịch và chạy chương trình