BKACAD®

HỌC VIỆN CÔNG NGHỆ BKACAD

Lập trình Java – Giới thiệu về Java

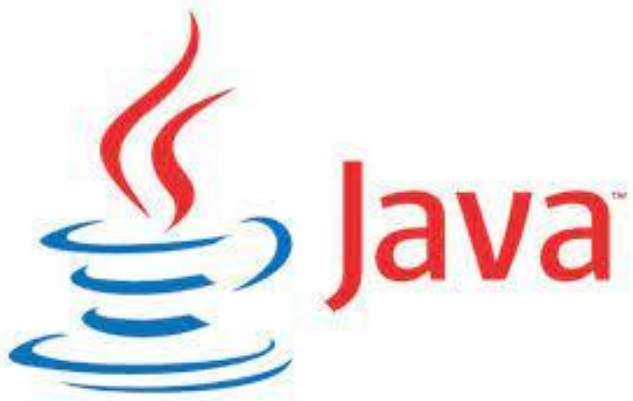
Ths. Vũ Duy Khương

Nội dung

1. Java là gì
2. Lịch sử hình thành và phát triển
3. Các đặc điểm của Java
4. Bắt đầu với lập trình Java
5. Q&A

Java là gì ?

- Java là một ngôn ngữ lập trình (*programming language):* một ngôn ngữ mà bạn có thể học cách viết và máy tính có thể hiểu được
- Java hiện đang là một ngôn ngữ rất phổ biến
- Java là một ngôn ngữ mạnh và có tầm bao quát rộng - nhưng nó ***không đơn giản***!
- Được so sánh với C++, Java rất "tao nhã" (elegant)



Lịch sử

- 1990, James Gosling và Sun Microsystems
- Tên ban đầu: Oak (cây sồi)
- Java, 20/05/1995, Sun World
- HotJava: Trình duyệt Web hỗ trợ Java đầu tiên
- JDK Evolutions
- J2SE, J2ME, and J2EE

Các đặc điểm của Java

• Java is simple	☞ đơn giản
• Java is object-oriented	☞ hướng đối tượng
• Java is distributed	☞ phân tán
• Java is interpreted	☞ thông dịch
• Java is robust	☞ mạnh mẽ
• Java is secure	☞ bảo mật
• Java is architecture-neutral	☞ kiến trúc trung tính
• Java is portable	☞ khả chuyển
• Java's performance	☞ hiệu quả cao
• Java is multithreaded	☞ đa tuyến
• Java is dynamic	☞ linh động

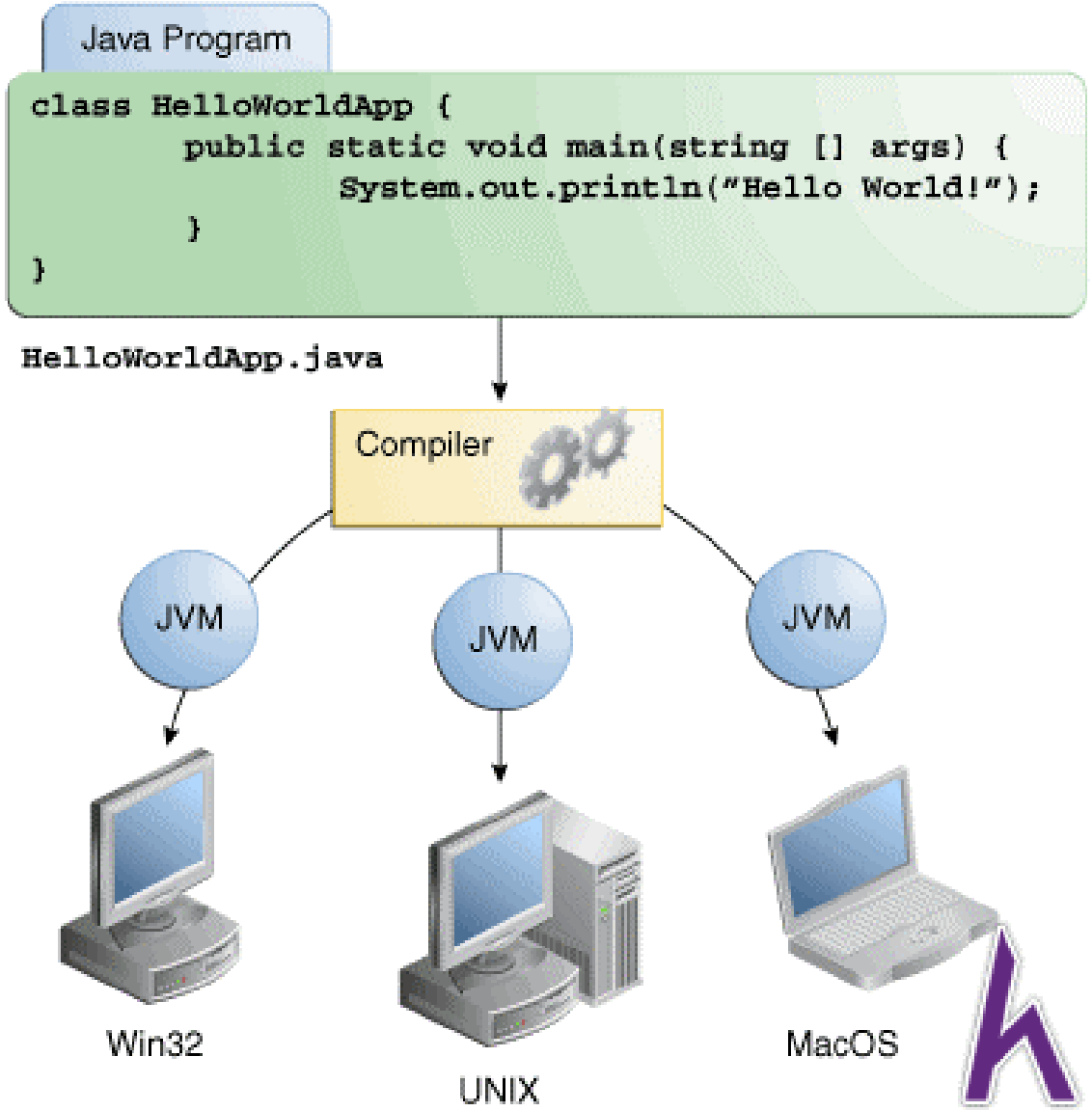
Các đặc trưng của Java

- Là ngôn ngữ **hướng đối tượng** (hướng đối tượng sẽ được giải thích rõ ràng trong bài sau)
- Chạy bằng máy ảo **Java**. Chương trình muốn thực thi phải biên dịch ra mã máy, mà mã máy mỗi hệ điều hành là khác nhau như Windows biên dịch dưới dạng file có đuôi **.EXE** còn Linux có dạng đuôi **.ELF**. Việc Java có thể chạy mọi hệ điều hành là do nhà phát triển **Sun Microsytems** phát triển máy ảo (JVM) chịu trách nhiệm việc này. Bài học sau ta sẽ cài đặt môi trường Java

Các đặc trưng của Java

- **Đa nhiệm – Đa luồng**: Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép chạy nhiều tiến trình chạy song song trong một thời điểm và tương tác lẫn nhau

- Java **bỏ đa kế thừa** trong C++ thay bằng **sử dụng Interface**



Các phiên bản JDK (Java Devenlopment Kit)

- Java 1
  - JDK 1.02 (1995)
  - JDK 1.1 (1996)
- Java 2
  - SDK v 1.2 (JDK 1.2, 1998)
  - SDK v 1.3 (JDK 1.3, 2000)
  - SDK v 1.4 (JDK 1.4, 2002)
  - SDK v 5.0 (JDK 5.0, 2004)
  - Java SE 6 (11th Dec 2006)
  - Java SE 7 (28th July 2011)
  - Java SE 8 (18th March 2014)
- Java SE 9 (21st Sep 2017)
- Java SE 10 (20th March 2018)
- Java SE 11 (Phiên bản hỗ trợ dài hạn)
- Java SE 12 (Vẫn đang được mở để sửa lỗi)

JDK Editions

- **Java Standard Edition (J2SE)**
  - J2SE có thể được dùng để phát triển các ứng dụng hoặc các applet độc lập phía client (client-side).
- **Java Enterprise Edition (J2EE)**
  - J2EE có thể được dùng để phát triển các ứng dụng phía server (server-side) như các Java servlet và Java ServerPages.
- **Java Micro Edition (J2ME).**
  - J2ME có thể được sử dụng để phát triển các ứng dụng cho các thiết bị di động như ĐTDĐ.

Bài giảng sử dụng J2SE để giới thiệu lập trình Java.

Java IDE Tools

- JCreator
- Forte by Sun MicroSystems
- Eclipse
- Netbean
- WebGain Café
- IBM Visual Age for Java

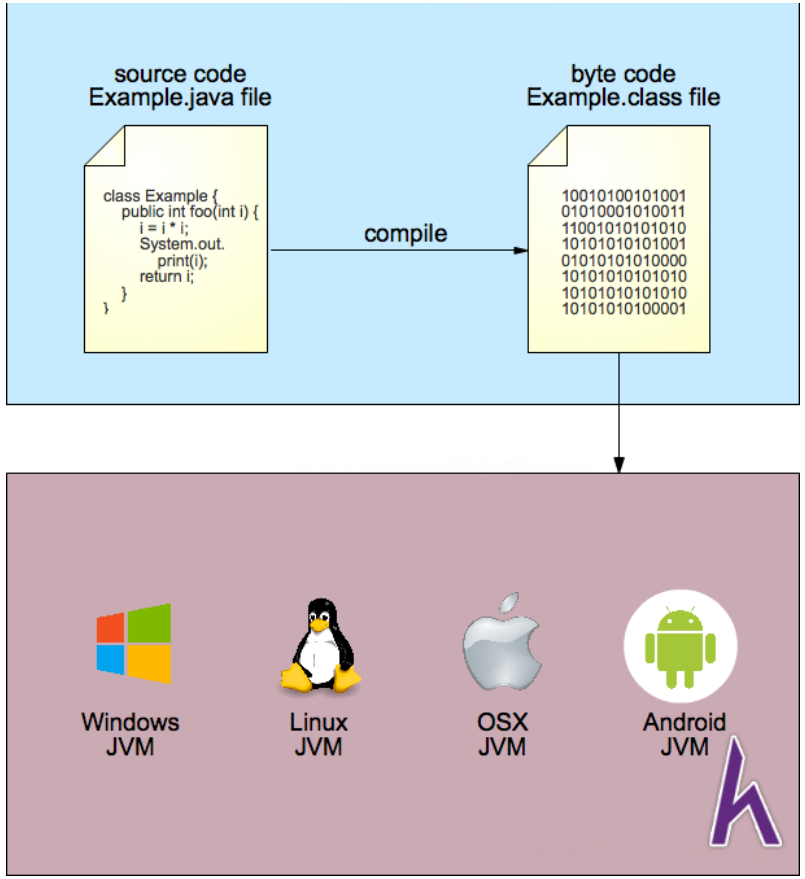
Bắt đầu với lập trình Java

- Một chương trình Java đơn giản
- Biên dịch chương trình
- Chạy chương trình

Một chương trình Java đơn giản

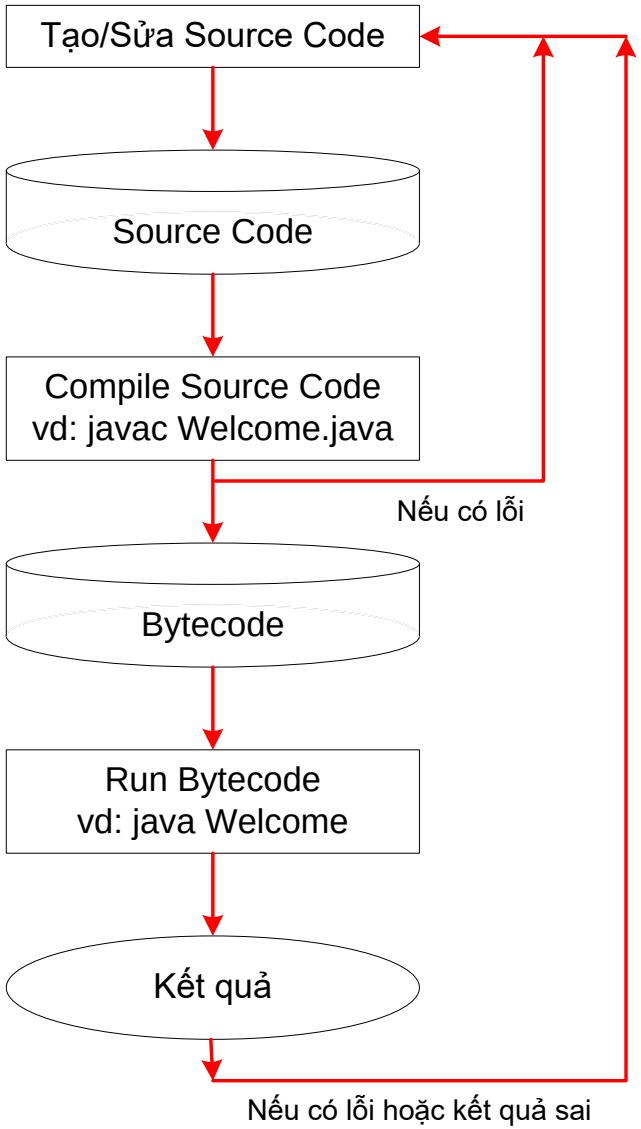
Ví dụ 1.1

```
// Chương trình in dòng: Welcome to Java!
package ch01;
public class Welcome {
  public static void main(String[] args) {
        System.out.println("Welcome to Java!");
  }
}
```

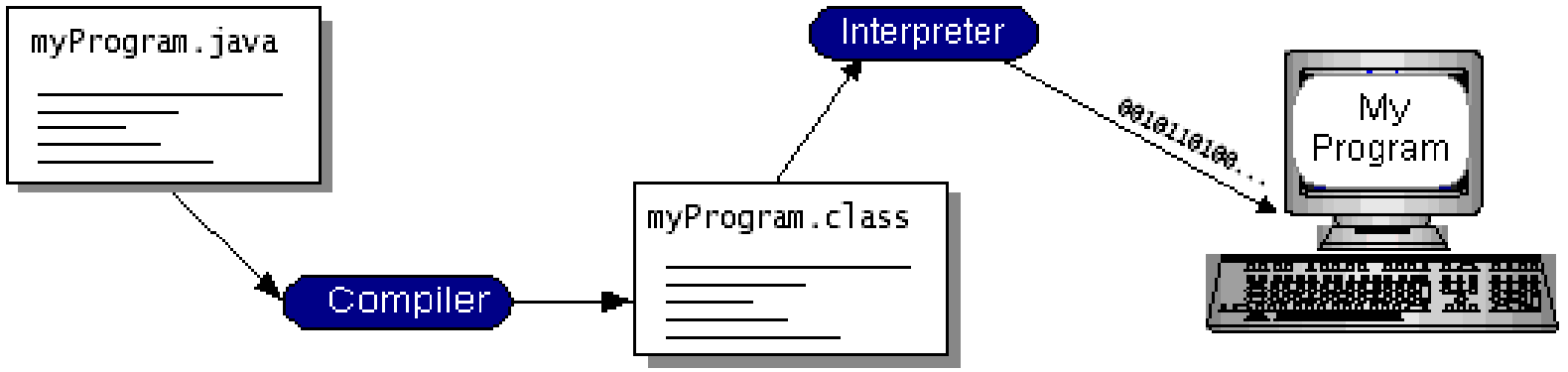


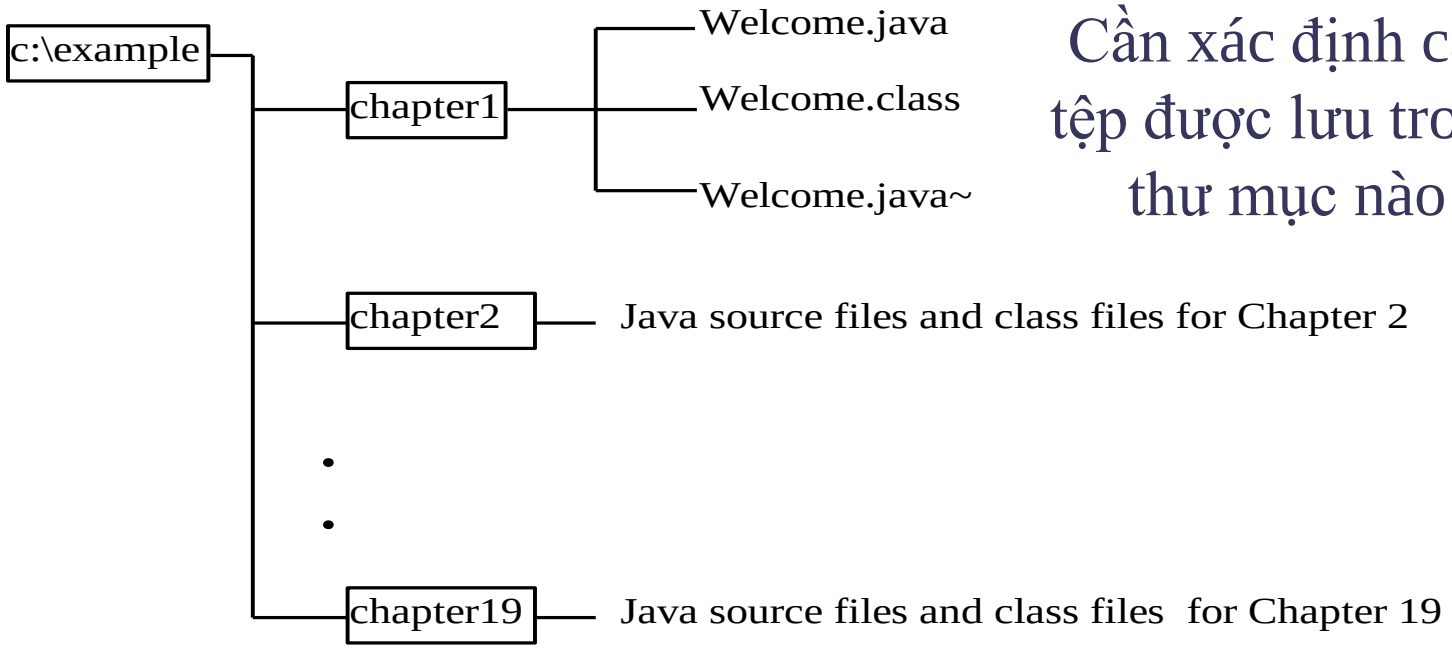
Tạo, biên dịch, chạy chương trình

- Tạo:
  - Soạn thảo chương trình (Notepade, Wordpad…)
  - Ghi tệp tên Welcome.java vào thư mục C:\javapro
- Biên dịch:
  - Trên cửa sổ lệnh (cmd.exe)
  - cd\ ↵
  - cd javapro ↵
  - javac Welcome.java ↵
- Chạy:
  - java Welcome ↵



Biên dịch và chạy một chương trình





Cần xác định các tệp được lưu trong thư mục nào

Các thành phần của một chương trình Java

- Chú giải - Comments
- Đóng gói - Package
- Từ khóa - Reserved words
- Từ bổ nghĩa - Modifiers
- Câu lệnh - Statements
- Khối - Blocks
- Lớp - Classes
- Phương thức - Methods
- Phương thức chính - The main method

Comments

- Trong Java, các chú giải có thể được đặt :
  - sau 2 dấu gạch chéo // trên 1 dòng
  - giữa dấu mở /* và đóng */ trên 1 hoặc nhiều dòng
- Khi trình biên dịch gặp:
  - //, nó bỏ qua tất cả các ký tự sau // trên dòng đó
  - /*, nó quét tìm đến */ tiếp sau và bỏ qua mọi ký tự nằm giữa /* và */.

Package

Dòng thứ hai trong chương trình (package ch01;) xác định một tên gói, ch01, cho class Welcome. Trình biên dịch source code trong tệp Welcome.java, tạo ra tệp Welcome.class, và lưu Welcome.class trong thư mục ch01.

Reserved Words

- ***Reserved words*** hay ***keywords*** là những từ có nghĩa xác định đối với trình biên dịch và không thể sử dụng cho các mục đích khác trong chương trình.
- VD: khi trình biên dịch gặp từ **class**, nó hiểu rằng từ ngay sau class là tên của class.
- Các từ khóa khác trong ví dụ 1.1 là **public**, **static**, và **void**. Chúng sẽ được giới thiệu ở phần sau.

Modifiers (Từ bổ nghĩa)

- Java sử dụng một số từ khóa gọi là ***modifiers*** để xác định các thuộc tính của dữ liệu, các phương thức, lớp, và chúng có thể được sử dụng như thế nào.
- Các ví dụ từ bổ nghĩa là **public, static, private, final, abstract**, và **protected.**
- Một dữ liệu, phương thức, hoặc lớp **public** thì có thể truy nhập được bởi chương trình khác. Một dữ liệu hay phương thức **private** thì không thể.
- Modifiers sẽ được thảo luận ở Chương sau, "Objects and Classes."

Statements

- Một câu lệnh (***statement***) đại diện cho một hành động hoặc một chuỗi các hành động.
- Câu lệnh *System.out.println("Welcome to Java!")* trong chương trình ví dụ 1.1 là một câu lệnh hiển thị lời chào "*Welcome to Java*!".
- Mọi câu lệnh trong Java kết thúc bởi một dấu chấm phẩy **(;)**.

Blocks

- Một cặp dấu ngoặc nhọn trong một chương trình hình thành một khối nhóm các thành phần của một chương trình.
- Vai trò tương tự cặp từ khóa Begin …end; trong Pascal

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Class block

Method block

Classes

- ***Class*** (*lớp*) là thiết yếu trong xây dựng cấu trúc Java. Một class là một khuôn mẫu hay bản thiết kế cho các đối tượng.
- Để lập trình trong Java, bạn phải hiểu các class và có thể viết, sử dụng chúng.
- Những bí ẩn của class sẽ tiếp tục được khám phá dần xuyên suốt khóa học.
- Bây giờ bạn chỉ cần hiểu một chương trình được xác định bằng cách sử dụng một hay nhiều class.

Methods

- ***System.out.println*** là gì? Đó là một ***method*** *(phương thức)*: một tập các câu lệnh thực hiện một chuỗi các thao tác để hiển thị một thông tin trên màn hình.
- Nó thậm chí có thể được sử dụng mà không cần hiểu đầy đủ chi tiết nó làm việc như thế nào.
- Nó được sử dụng bằng cách gọi một câu lệnh với tham số chuỗi ký tự (string) được bao bởi cặp dấu nháy kép. Trong trường hợp này, tham số là "Welcome to Java!"
- Bạn có thể gọi phương thức println với các tham số khác nhau để in ra những message khác nhau.

Main Method

- **main** method cung cấp sự kiểm soát luồng chương trình. Trình biên dịch Java thực hiện ứng dụng bằng cách gọi đến main method.
- Mọi chương trình Java phải có main method, nó là điểm khởi đầu khi thực hiện chương trình.
- Dạng thức của main method:

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

Hiển thị văn bản trong Message Dialog Box

- Ta có thể sử dụng phương thức **showMessageDialog** trong lớp JOptionPane. JOptionPane là một trong nhiều lớp được định nghĩa trước trong hệ thống Java để có thể tái sử dụng.

Phương thức showMessageDialog

```
JOptionPane.showMessageDialog(null,
  "Welcome to Java!",
  "Example 1.2 Output", JOptionPane.INFORMATION_MESSAGE));
```

Q&A

THANK YOU