



Bắt đầu với Java



Nội dung

- Lịch sử của Java
- Các đặc trưng cơ bản
- Java applications và Java applets
- Tạo ứng dụng Java đơn giản



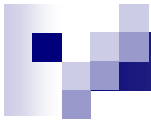
Tài liệu tham khảo

- Thinking in Java, chapter 1, 2, 3



Lịch sử hình thành

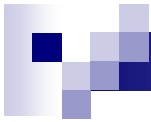
- 1991: được Sun Microsystems phát triển nhằm mục đích viết phần mềm điều khiển (phần mềm nhúng) cho các sản phẩm gia dụng
 - lúc đầu được đặt tên là Oak
- 1995: được phổ cập với sự phát triển mạnh mẽ của Internet
 - thị trường phần mềm nhúng không phát triển mạnh
 - WWW bùng nổ (1993~)
- Hiện nay, được chấp nhận rộng rãi với tư cách là một ngôn ngữ (công nghệ) đa dụng
 - khả chuyển, an toàn
 - hướng đối tượng, hướng thành phần



Java là một công nghệ

Java bao gồm

- Ngôn ngữ lập trình
- Môi trường phát triển
- Môi trường thực thi và triển khai



Mục tiêu của Java

- Ngôn ngữ dễ dùng
 - Khắc phục nhiều nhược điểm của các ngôn ngữ trước đó
 - Hướng đối tượng
 - Sáng sửa
- Môi trường thông dịch
 - Tăng tính khả chuyển
 - An toàn



Mục tiêu của Java

- Cho phép chạy nhiều tiến trình (threads)
- Nạp các lớp (classes) động vào thời điểm cần thiết từ nhiều nguồn khác nhau
 - Cho phép thay đổi động phần mềm trong khi hoạt động
- Tăng độ an toàn



Biên dịch và thông dịch

- Chương trình nguồn được biên dịch sang mã đích (bytecode)
- Mã đích (bytecode) được thực thi trong môi trường thông dịch (máy ảo)



Các dạng ứng dụng của Java

■ Desktop applications - J2SE

- Java Applications: ứng dụng Java thông thường trên desktop
- Java Applets: ứng dụng nhúng hoạt động trong trình duyệt web

■ Server applications - J2EE

- JSP và Servlets

■ Mobile (embedded) applications – J2ME



Đặc trưng của Java

- JVM – máy ảo Java
- Cơ chế giải phóng bộ nhớ tự động
- Bảo mật chương trình



JVM - Máy ảo Java

- Máy ảo phụ thuộc vào platform (phần cứng, OS)
- Cung cấp môi trường thực thi cho chương trình Java (độc lập với platform)
- Máy ảo đảm bảo an toàn cho hệ thống
- Máy ảo thông thường được cung cấp dưới dạng phần mềm
 - JRE - Java Runtime Environment
- Java platform: JVM + APIs

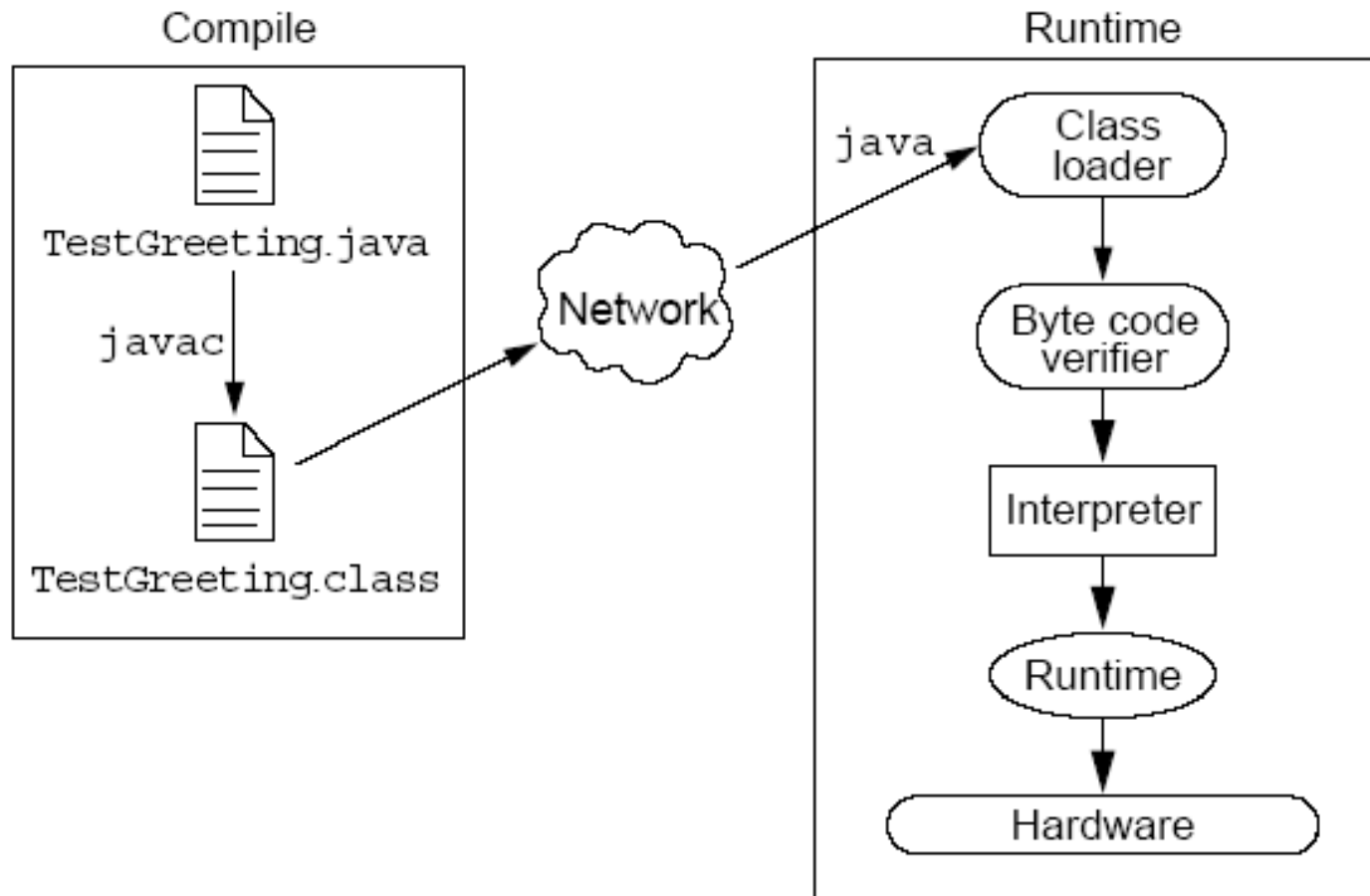


Giải phóng bộ nhớ

(Garbage Collection)

- Java cung cấp một tiến trình mức hệ thống để theo dõi việc cấp phát bộ nhớ
- Garbage Collection
 - Đánh dấu và giải phóng các vùng nhớ không còn được sử dụng
 - Được tiến hành tự động
 - Cơ chế hoạt động phụ thuộc vào các phiên bản máy ảo

Chống sao chép



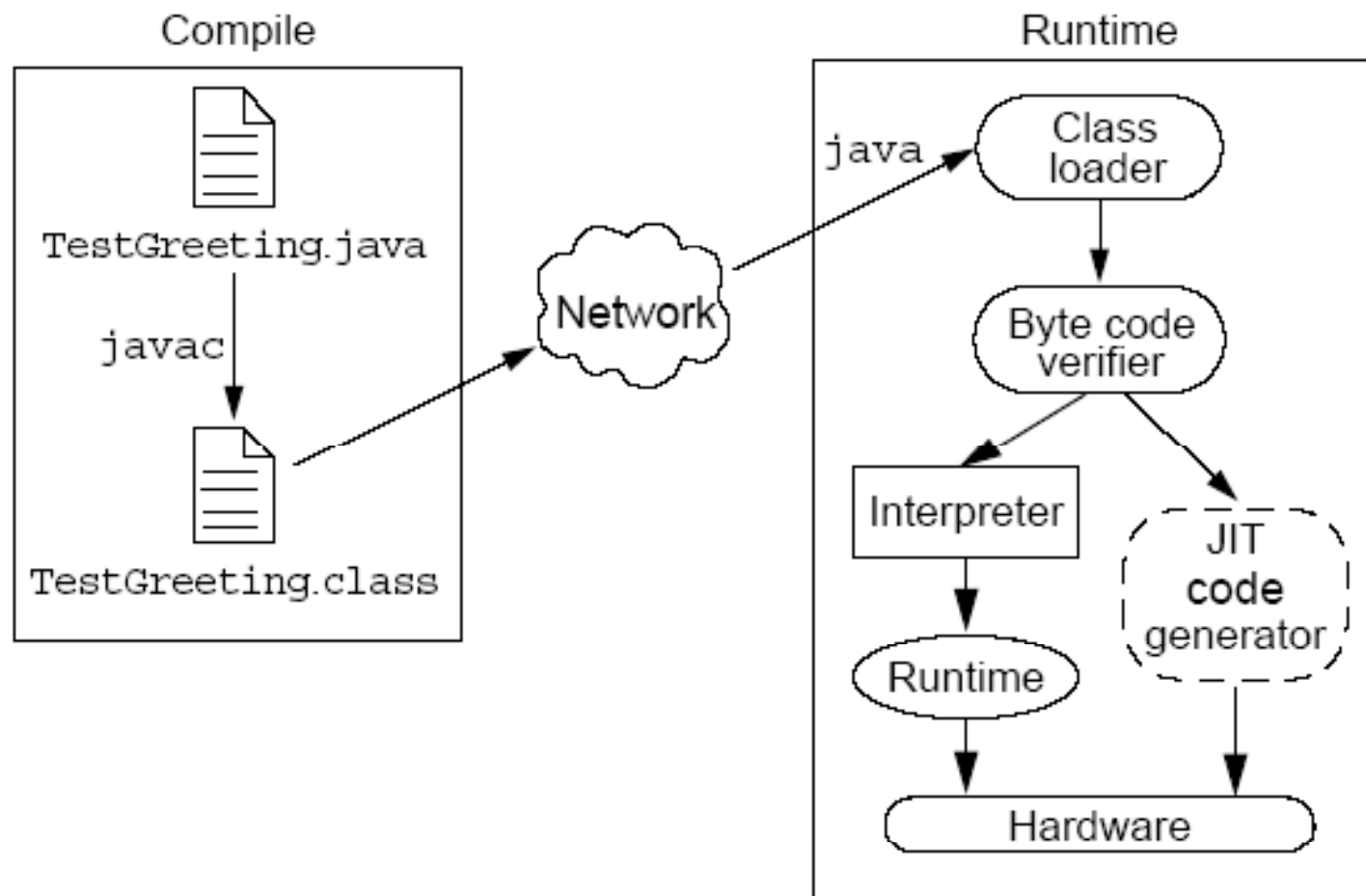


JDK

- Môi trường phát triển và thực thi do Sun Microsystems cung cấp (<http://java.sun.com>)
 - Phiên bản hiện tại J2SDK 6 (1.6)
- Bao gồm
 - *javac* Chương trình dịch chuyển mã nguồn sang *bytecode*
 - *java* Bộ thông dịch: Thực thi java application
 - *appletviewer* Bộ thông dịch: Thực thi java applet mà không cần sử dụng trình duyệt như *Netscape*, hay *IE*, v.v.
 - *javadoc* Bộ tạo tài liệu dạng HTML từ mã nguồn và chú thích
 - *jdb* Bộ gỡ lỗi (*java debugger*)
 - *javap* Trình dịch ngược *bytecode*

Công nghệ JIT

Just-In-Time Code Generator





Java Applications

- Chương trình ứng dụng hoàn chỉnh
- Giao diện dòng lệnh hoặc đồ họa
- Được bắt đầu bởi phương thức (hàm)
`main()` là phương thức `public static`

Chương trình Java đơn giản

TestGreeting.java:

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

Diagram annotations:

- public class**: points to `public class`
- public static method**: points to `public static void main`
- package**: points to `String[]`
- object**: points to `args`
- message**: points to `"Hello, world"`



Biên dịch và thực hiện

- Biên dịch `TestGreeting.java`

```
javac TestGreeting.java
```

- Thực hiện

```
java TestGreeting
```

- Kết quả

```
Hello, world
```



Một chút cải tiến

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("Hello, world");  
    }  
}
```



Biên dịch và thực hiện

- Biên dịch `TestGreeting.java`

```
javac TestGreeting.java
```

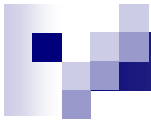
- `Greeting.java` được biên dịch tự động

- Thực hiện

```
java TestGreeting
```

- Kết quả

```
Hello, world
```



Java Applets

- Được nhúng trong một ứng dụng khác (web browser)
- Có giao diện hạn chế (đồ họa)
- Không truy cập được tài nguyên của client (không thực hiện được các hành vi *xấu*)



Applet đơn giản

Welcome.java:

```
// Java packages
import java.awt.Graphics;
import java.applet.Applet;

public class Welcome extends Applet {

    public void paint(Graphics g)
    {
        // call superclass version of method paint
        super.paint(g);

        // draw a String
        g.drawString("Welcome to Java programming!", 25, 25);
    }
}
```



Nhúng vào trang Web

Welcome.html:

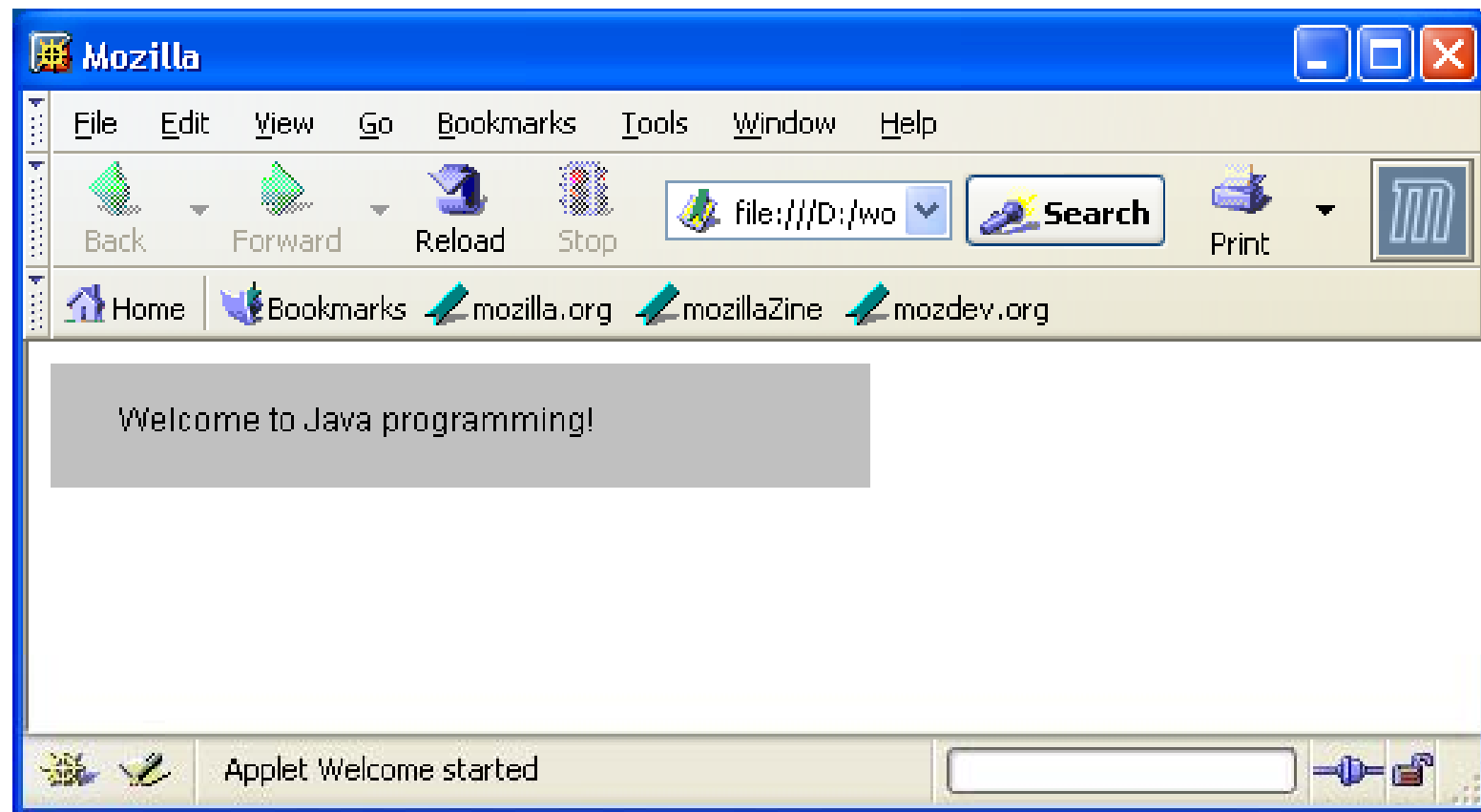
```
<html>
```

```
<applet code = "Welcome.class"  
  width = "300" height = "45">
```

```
</applet>
```

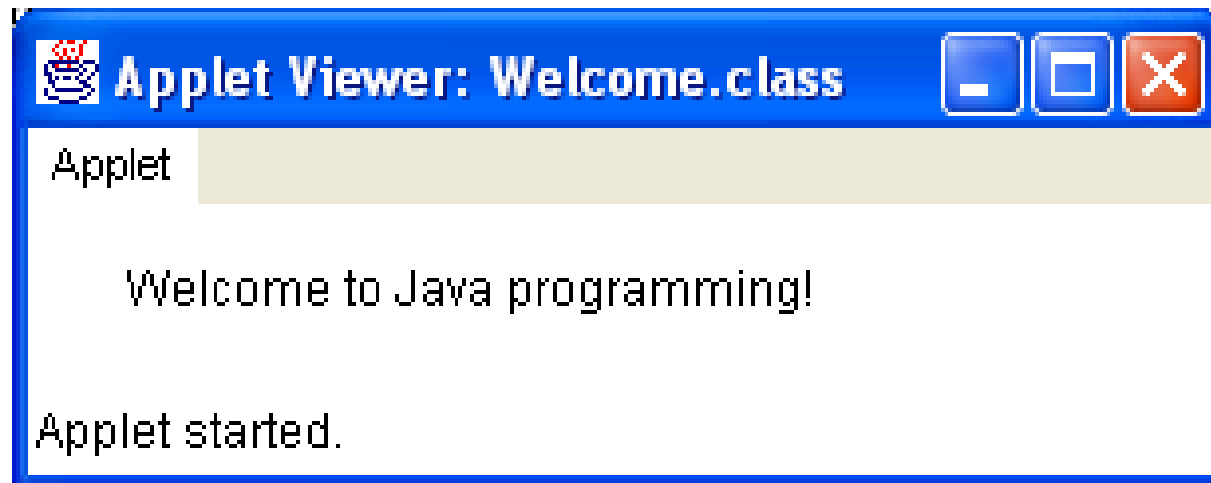
```
</html>
```

Thực hiện (trong web browser)



Thực hiện

`appletviewer Welcome.html`





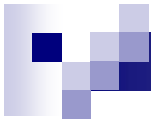
Các phương thức của Applet

- `init()`: khởi tạo applet
- `start()`: khởi động applet
 - mặc định sẽ gọi `paint()`
- `stop()`: dừng applet
- `destroy()`: giải phóng (hủy) applet



Tự thực hành

- Đăng nhập vào website môn học
- Làm quen với môi trường phát triển Java trên Linux và/hoặc Windows
- Tập viết các ứng dụng nhỏ
 - các ví dụ trong bài giảng (application/applet)
 - chuyển các bài thực hành cơ bản của môn C/C++ sang Java



Bài tập: Tìm hiểu về Java

- Các kiểu dữ liệu cơ bản
 - các kiểu số nguyên, kiểu ký tự, kiểu logic
- Từ khóa, cách đặt tên (lớp, phương thức, thuộc tính)
- Các cấu trúc điều khiển cơ bản
 - điều kiện
 - vòng lặp
 - switch