

# Tổng quan về Java

# Nội dung

---

- Giới thiệu Java
- Các đặc trưng của Java
- Định nghĩa về máy ảo Java (Java Virtual Machine)
- JDK (Java Development Kit) và packet của Java code API

# Giới thiệu Java

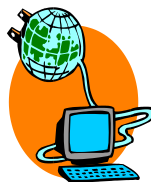
---

## ➤ Lịch sử phát triển

- 1991: Sun Microsystems phát triển OAK nhằm mục đích viết phần mềm điều khiển (phần mềm nhúng) cho các sản phẩm gia dụng.



- 1995: Internet bùng nổ, phát triển mạnh. Sun phát triển OAK và giới thiệu ngôn ngữ lập trình mới tên Java



# Giới thiệu Java (tt)

---

- Xây dựng trên nền tảng C và C++
- Sử dụng các cú pháp của C và các đặc trưng đối tượng của C++
- Ngôn ngữ độc lập với thiết bị
- Không phụ thuộc vào trình biên dịch của CPU
- Java là ngôn ngữ lập trình hướng đối tượng

# Giới thiệu Java (tt)

---

- Là ngôn ngữ lập trình vừa **biên dịch**, vừa **thông dịch**.
  - Đầu tiên mã nguồn được biên dịch bằng công cụ **JAVAC** để chuyển thành dạng **bytecode**
  - Sau đó được thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch
- Chạy trên bất kỳ phần cứng, không phụ thuộc vào hệ điều hành
- Trở thành ngôn ngữ lập trình Internet
- Xây dựng các chương trình điều khiển thiết bị cho điện thoại di động, PDA

# Các thư viện (API) của Java

---

## ➤ Gồm 3 thư viện:

- **Java 2 Standard Edition (J2SE):** Dùng để phát triển các ứng dụng đơn, hoặc ứng dụng Client-site
- **Java 2 Enterprise Edition (J2EE):** Phát triển các ứng dụng đơn, ứng server-site và JavaServerPage
- **Java 2 Micro Edition (J2ME):** Phát triển các ứng dụng di động (Mobile), không dây (PDA)

# Các đặc trưng của Java

---

- Đơn giản
- Hướng đối tượng
- Độc lập phần cứng và hệ điều hành
- Mạnh mẽ
- Bảo mật
- Phân tán
- Đa luồng
- Động

# Đơn giản

---

- Loại bỏ các đặc trưng phức tạp của C và C++ như :
  - Con trỏ
  - Định nghĩa chồng toán tử (operator overloading)
  - Không sử dụng lệnh goto
  - Không định nghĩa file header.h
  - Không dùng struct
  - Không dùng union



# Hướng đối tượng

---

- Thiết kế code xoay quanh mô hình hướng đối tượng
- Dữ liệu và các phương thức của một đối tượng được gói trong một lớp của java
  - Đảm bảo tính bảo mật
  - Thừa kế
  - Đa hình...vv
  - Không cho phép đa thừa kế (multi-inheritance) mà sử dụng các giao diện (interface).

# Độc lập với phần cứng và HĐH

---

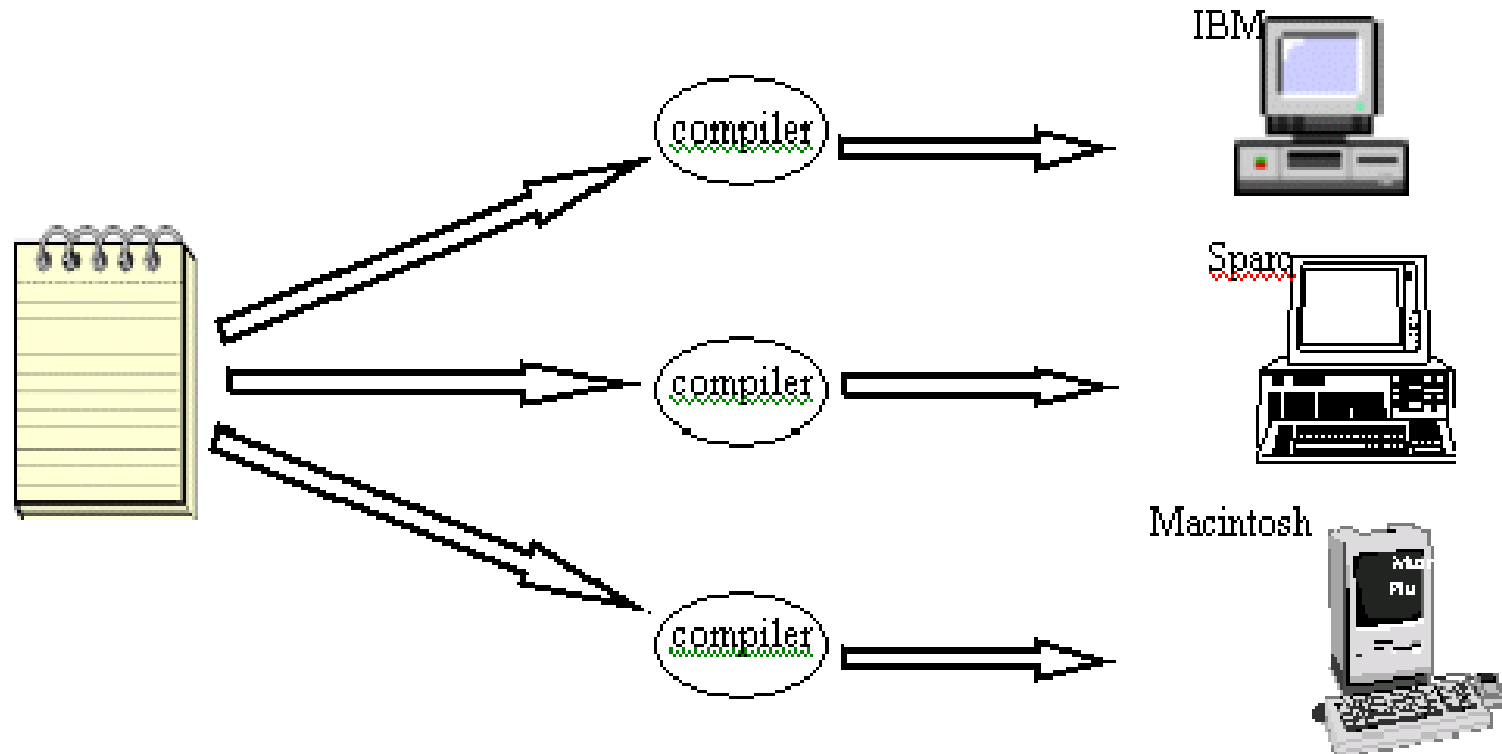
- Chương trình viết ở một máy nhưng có thể chạy bất kỳ ở đâu. Chúng được thể hiện 2 mức:
  - Mã nguồn
  - Mức nhị phân
- Ở mức mã nguồn
  - Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau
  - Java có riêng một thư viện các lớp cơ sở

# Độc lập với phần cứng và HĐH(tt)

---

- Tính độc lập ở mức nhị phân:
  - Một chương trình đã biên dịch có thể chạy trên nhiều nền (phần cứng, hệ điều hành) khác mà không cần dịch lại mã nguồn
  - Cần có phần mềm máy ảo java hoạt động như trình thông dịch tại máy thực thi

# Cách biên dịch truyền thống



# Trình dịch truyền thống (tt)

---

- Các chương trình viết bằng C, C++,...vv trình biên dịch sẽ chuyển tập lệnh thành mã máy (machine code), hay lệnh của CPU. Những lệnh này phục thuộc vào CPU trên trên máy thực thi.
- Với mỗi nền phần cứng khác nhau thì sẽ có một trình biên dịch khác nhau.
- Dịch lại khi muốn chạy trên nền phần cứng khác khác

# Chương trình dịch Java

Programs

Java code



compiler

Bytecode

(Platform Independent)

J  
a  
v  
a  
  
I  
n  
t  
e  
r  
p  
r  
e  
t  
e  
r

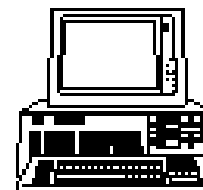
IBM



Macintosh



Sparc



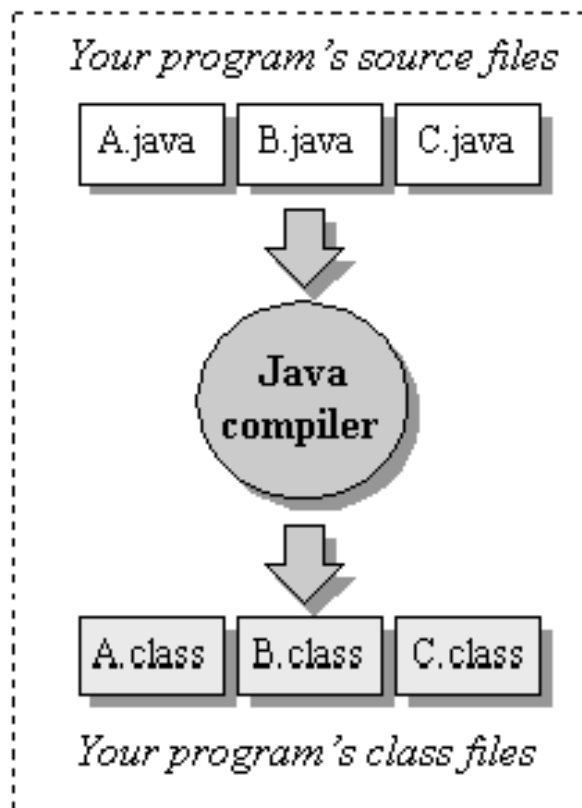
# Chương trình dịch Java

---

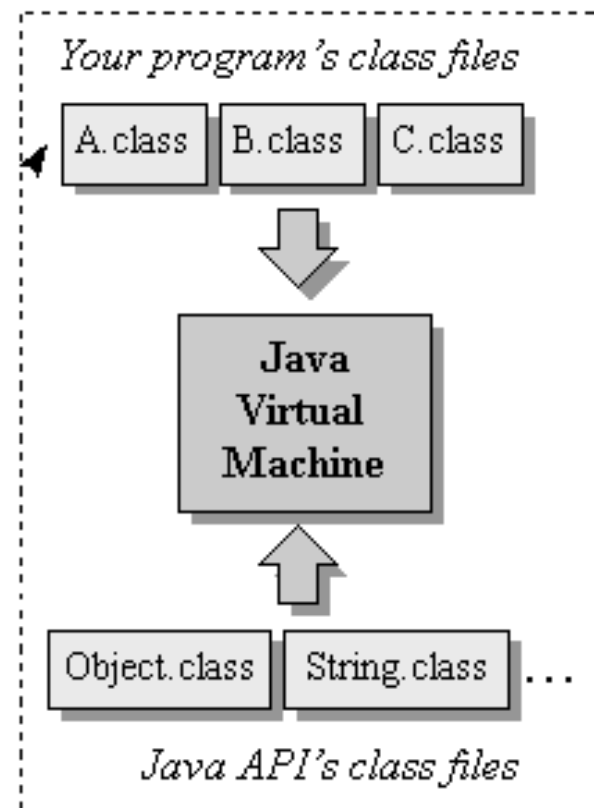
- Môi trường phát triển của java gồm 2 phần:
  - Trình biên dịch (**Javac**)
  - Trình thông dịch
- Trình biên dịch (**Javac**) chuyển mã nguồn thành dạng bytecode
- Trình thông dịch (máy ảo java)
  - Máy ảo Java chuyển bytecode thành mã CPU

# Môi trường lập trình JAVA

## compile-time environment



## run-time environment



*Your class files move locally or through a network*



# Máy ảo Java

## (JVM- Java virtual Machine)

---

- Là 1 phần mềm tập hợp các lệnh logic để xác định các hoạt động của máy tính (hệ điều hành thu nhỏ)
- Trình biên dịch chuyển mã nguồn thành các tập lệnh của máy ảo
- Trình thông dịch trên mỗi máy chuyển tập lệnh thành chương trình thực thi bằng cách
  - Nạp file .class
  - Quản lý bộ nhớ
  - Dọn rác

# Trình thông dịch “Just In Time-JIT”

---

- JIT: dùng để chuyển mã bytecode thành mã máy cụ thể cho từng loại CPU
- Các trình duyệt web thông dụng : IE, Netscape đều có JIT để tăng tốc độ xử lý

# Ví dụ 1: chương trình Hello World

- Dùng Notepad soạn thảo đoạn lệnh bên dưới và lưu lại với tên

HelloWorld.java

Khai báo thư viện java.io

```
import java.io.*;
```

Định nghĩa lớp tên “**HelloWorld**”

```
class HelloWorld
```

Bắt đầu đoạn lệnh

```
{
```

Phương thức **main**

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

```
{
```

```
    System.out.print("Hello Class");
```

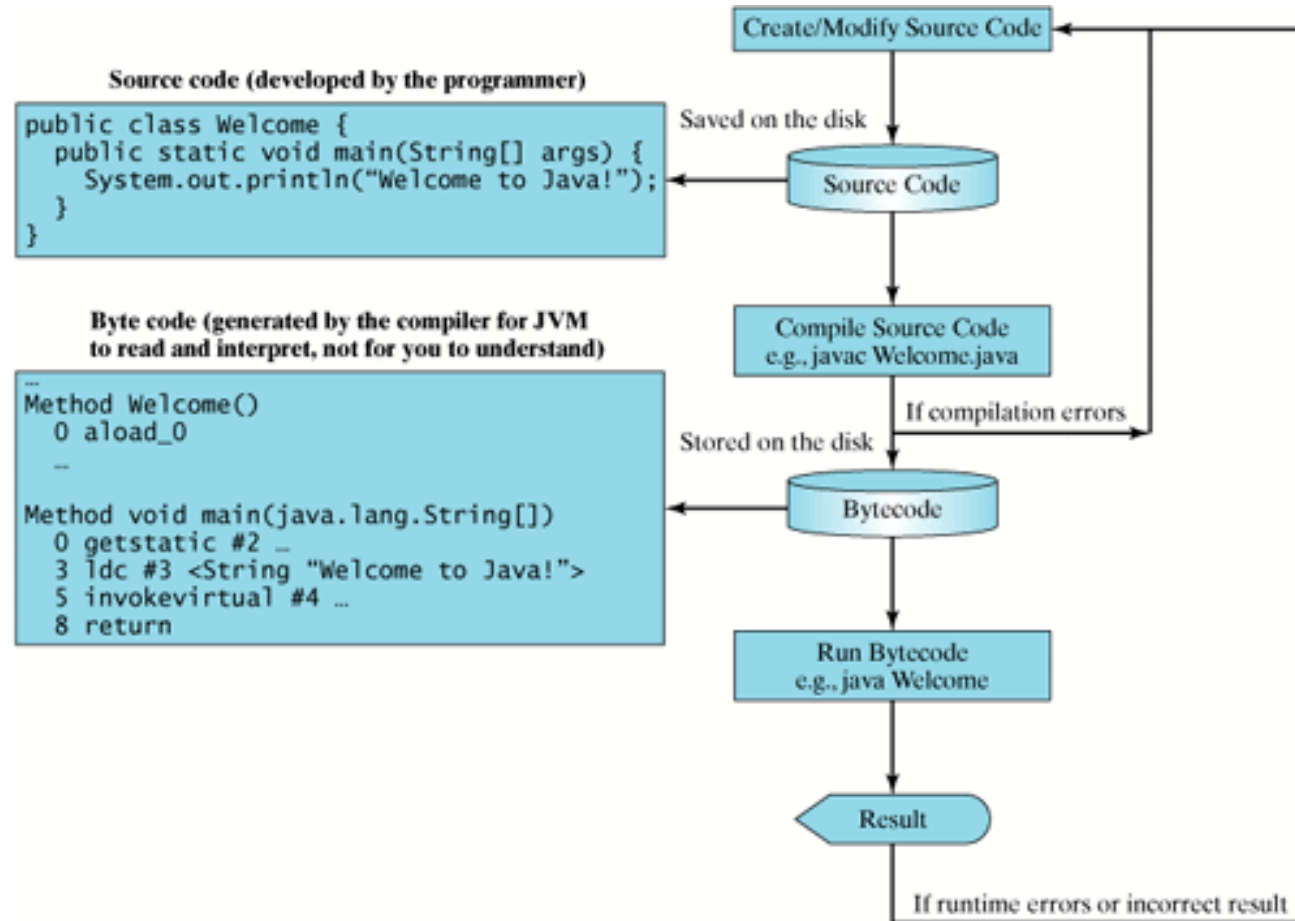
Xuất ra Console  
thông báo

```
}
```

```
}
```

Kết thúc đoạn lệnh

# Chương trình chạy Java



# Mạnh mẽ

---

- Phải khai báo kiểu dữ liệu tường minh trước
- Không sử dụng con trỏ và các phép toán trên con trỏ
- Không phải bận tâm về việc cấp phát và giải phóng vùng nhớ
- Cơ chế bắt lỗi của java giúp đơn giản quá trình xử lý lỗi và phục hồi sau lỗi

# Bảo mật

---

- Java cung cấp nhiều mức để kiểm soát tính an toàn khi thực thi chương trình
  - **Mức 1:** Dữ liệu và phương thức được gói trong lớp
  - **Mức 2:** Trình biên dịch kiểm soát để đảm bảo mã an toàn và tuân theo các nguyên tắc của Java
  - **Mức 3:** Được đảm bảo bởi trình thông dịch, chúng kiểm tra xem bytecode có đảm bảo các quy tắc an toàn trước khi thực thi
  - **Mức 4:** Kiểm tra việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống

# Phân tán, đa luồng, động

---

- Java hỗ trợ xây dựng các ứng dụng được làm việc trên nhiều phần cứng, HĐH và giao diện đồ họa khác nhau
- Chương trình Java đa luồng(Multithreading) để thực thi các công việc song song
- Java là mã nguồn mở. Kiểm soát truy cập lúc chạy giúp liên kết mã động

# Các ứng dụng trong Java

---

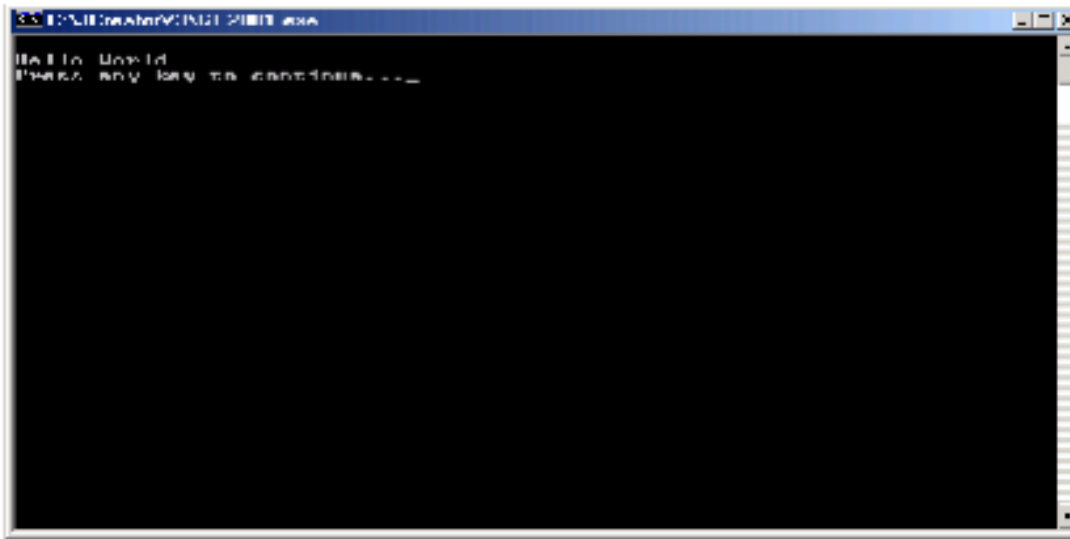
- Ứng dụng Console
- Ứng dụng Applet
- Ứng dụng Desktop dùng AWT
- Ứng dụng Web
- Ứng dụng nhúng



# Ứng dụng Console

---

- Là loại ứng dụng nhập xuất theo kiểu văn bản thông qua màn hình console tương tự như màn hình Console của hệ điều hành MS-DOS

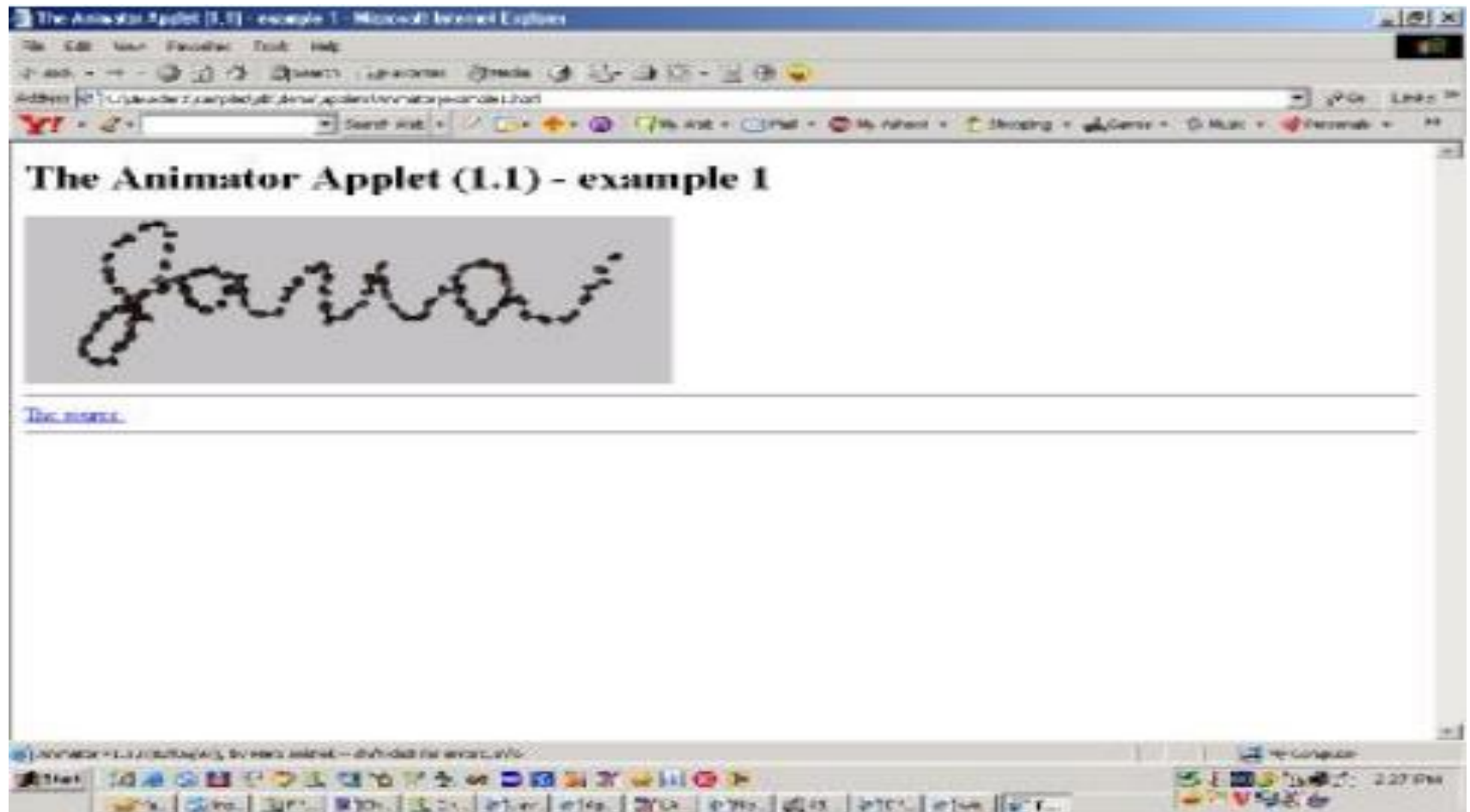


# Ứng dụng Applets

---

- JavaApplet dùng để nhúng trong trang Web, applet được tải về và thực thi khi duyệt web
- Chương trình được tạo ra để sử dụng trên Internet thông qua các trình duyệt web hỗ trợ Java: Chrome, IE, Firefox...

# Ứng dụng Applets



# Ứng dụng Desktop

---

- Dùng các gói thư viện AWT và Swing để tạo ra giao diện cho chương
- JFC (**Java Foundation Classess**) là thư viện phong phú và hỗ trợ mạnh mẽ cho việc thiết kế hơn nhiều so với AWT và Swing
- JFC giúp người dùng tạo ra giao diện trực quan cho bất kỳ ứng dụng nào

# Ví dụ tạo giao diện bằng JFC

[illegible]

# Ứng dụng Web

---

- Java hỗ trợ mạnh mẽ trong việc phát triển các ứng dụng Web thông qua công nghệ J2EE (Java 2 Enterprise Edition)
- Công nghệ J2EE hoàn toàn có thể tạo ra một ứng dụng Web hiệu quả không thua kém công nghệ .NET
- Hiện nay có rất nhiều Website sử dụng công nghệ Java

# Ứng dụng nhúng

---

- Java đưa ra công nghệ J2ME (Java 2 Platform, Micro Edition) hỗ trợ phát triển các phần mềm nhúng trên các thiết bị di động, PDA cũng như các thiết bị nhúng khác



# Bộ công cụ phát triển JDK (java development kit)

---

➤ Sun đưa ra ngôn ngữ lập trình Java qua sản phẩm có tên JDK. Các phiên bản:

JDK 1.0 - 1996

JDK 1.1 - 1997

J2SE 1.2 - 1998

J2SE 1.3 - 2000

J2SE 1.4 - 2002

J2SE 5 (1.5) - 2004

Java SE 6 - 2006

Java SE 7 - 2011

Java 8 - 2014



# Các packet của Java code API

---

## ➤ **Java.lang**

- Lớp quan trọng nhất của java
- Gồm các kiểu dữ liệu cơ bản, ký tự, số nguyên
- Chứa tập lệnh nhập xuất chuẩn
- Chứa 1 số lớp quan trọng như: String, StringBuffer

## ➤ **Java.applet**

- Package nhỏ nhất chứa 1 mình lớp Applet
- Các Applet nhúng trong trang web, hay chạy trong Appletviewer đều thừa kế lớp này

# Các packet của java code API (tt)

---

- **Java.awt (abstract Window Toolkit)**
  - Chứa các lớp tạo giao diện đồ họa
  - Một số lớp bên trong: Button, GridBagLayout, Graphics
- **Java.io**
  - Cung cấp thư viện vào ra chuẩn
- **Java.util**
  - Cung cấp một số công cụ hữu ích như: Date, hashtable, vector, StringTokenizer

# Các packet của java code API (tt)

---

## ➤ **Java.net**

- Cung cấp khả năng giao tiếp từ xa
- Cho phép tạo và kết nối tới Socket, URL

## ➤ **Java.awt.event**

- Chứa các lớp, giao diện dùng để xử lý các sự kiện như bàn phím, chuột

## ➤ **Java.rmi**

- Công cụ gọi hàm từ xa

# Các packet của java code API (tt)

---

## ➤ **Java.security**

- Cung cấp các công cụ cần thiết để mã hóa dữ liệu
- Đảm bảo tính an toàn khi dữ liệu truyền đi giữa client và Server

## ➤ **Java.sql**

- Package này chứa Java DataBase Connectivity (JDBC)
- Dùng để truy xuất cơ sở dữ liệu quan hệ: SQL server, Oracle, MySQL