

CHƯƠNG 1

GIỚI THIỆU VỀ JAVA

NỘI DUNG

- Lịch sử phát triển
- Công nghệ Java
- Các dạng chương trình Java
- Đặc điểm của Java
- Máy ảo Java (Java Virtual Machine)
- Viết, dịch, thực thi chương trình HelloWorld
- Căn bản về ngôn ngữ Java

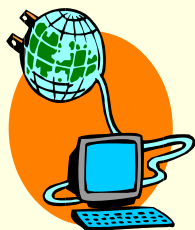
Lịch sử phát triển

- 1991, ra đời nhằm xây dựng các chương trình điều khiển các thiết bị điện tử như Tivi, máy giặt, lò nướng,...đáp ứng nhu cầu
 - Nhanh, gọn
 - Hiệu quả
 - Độc lập thiết bị (chạy trên nhiều loại CPU khác nhau)
- Java được xây dựng trên nền tảng của C và C++
- Do nhóm kỹ sư của Sun Microsystems phát triển (Green project- Patrick Naughton, Mike Sheridan, James Gosling)



Lịch sử phát triển

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



Lịch sử phát triển

- Cha đẻ là James Gosling



Java là gì?

- Là ngôn ngữ lập trình hướng đối tượng
- Là ngôn ngữ vừa biên dịch vừa thông dịch
 - Mã nguồn được biên dịch bằng công cụ JAVAC để chuyển thành dạng ByteCode
 - ByteCode được thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch
- Cho phép người lập trình viết chương trình một lần nhưng có thể chạy trên bất cứ phần cứng cụ thể
- Ngày nay, Java được sử dụng rộng rãi để viết chương trình chạy trên Internet, các thiết bị di động, desktop,...

Lịch sử phát triển Java Development Kit (JDK)

Môi trường phát triển và thực thi do Oracle cung cấp ([oracle.com](https://www.oracle.com))
Bao gồm phần mềm và công cụ giúp compile, debug và thực thi ứng dụng.

Các phiên bản:

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 21, 2017)
- Java SE 10 (March 20, 2018)
- Java SE 11 (September 25, 2018)
- Java SE 12 (March 19, 2019)

Công nghệ Java

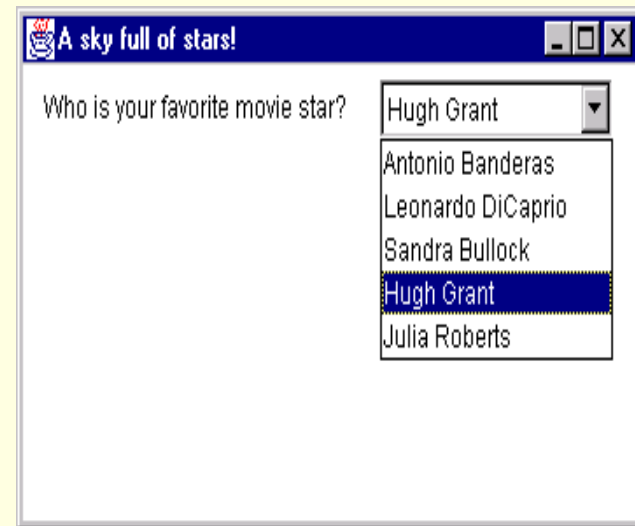
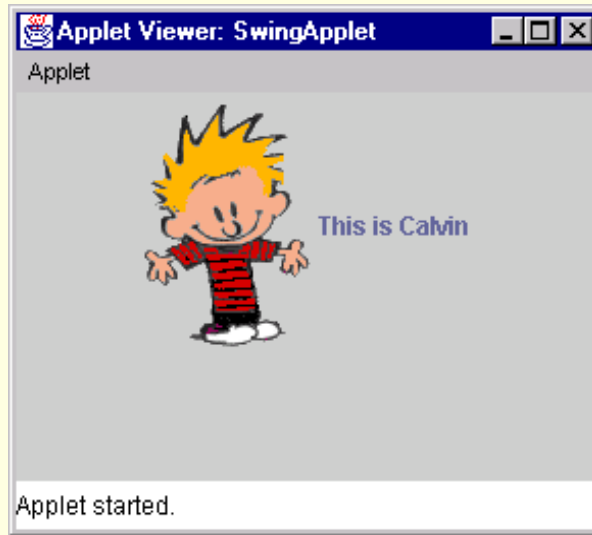
- Công nghệ:
 - ✓ Ngôn ngữ lập trình
 - ✓ Môi trường thực thi và triển khai
 - ✓ Môi trường phát triển
 - ✓ Xây dựng nhiều loại ứng dụng khác nhau
- Công nghệ J2SE (Java 2 Standard Edition)
- Công nghệ J2EE (Java 2 Enterprise Edition)
- Công nghệ J2ME (Java 2 Micro Edition)

Các dạng chương trình java

1. Applets
2. Console
3. GUI
4. Servlet
5. CSDL
6. Mobile
7. Nhúng
8. Enterprise Applications

Các dạng chương trình java

- Applets:



Các dạng chương trình java

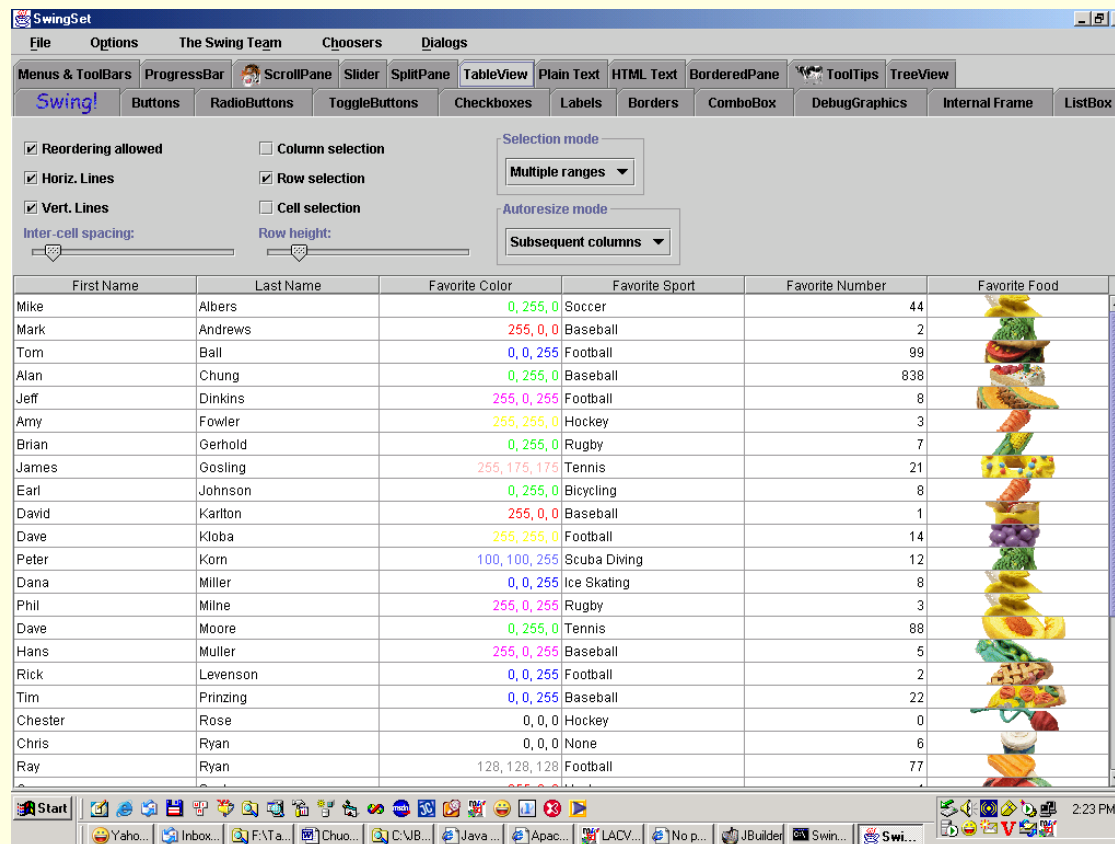
- Console Applications



```
C:\NT\System32\cmd.exe
C:\>java Arraytest
1
2
3
4
5
C:\>
```

Các dạng chương trình java

- GUI: AWT, Swing, JavaFX



Các dạng chương trình java

- Servlet



Các dạng chương trình java

- CSDL



The screenshot shows a window titled "Qt Jambi application". Inside the window is a form with the following fields and values:

Field	Value
ID	986230914
First Name	Peter
Middle Name	J
Last Name	Thompson
Email	pjt11@gmail.com
Phone	7439874366

At the bottom of the form are seven buttons: "New...", "Update", "Delete", "First", "Previous", "Next", and "Last".

Các dạng chương trình java

- **Mobile:**
 - Java ME
 - Android



Các dạng chương trình java

- **Nhúng:**

- Các loại máy tính chuyên dụng
- SIM cards
- Blue-ray disk players
- Dụng cụ đo lường
- Tivi



Các dạng chương trình java

- **Enterprise Applications:** Java EE
 - Network applications
 - Web-services

Mức độ phổ biến

<https://www.tiobe.com/tiobe-index/>

Feb 2019	Feb 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.876%	+0.89%
2	2		C	12.424%	+0.57%
3	4	▲	Python	7.574%	+2.41%
4	3	▼	C++	7.444%	+1.72%
5	6	▲	Visual Basic .NET	7.095%	+3.02%
6	8	▲	JavaScript	2.848%	-0.32%
7	5	▼	C#	2.846%	-1.61%
8	7	▼	PHP	2.271%	-1.15%
9	11	▲	SQL	1.900%	-0.46%
10	20	▲▲	Objective-C	1.447%	+0.32%
11	15	▲▲	Assembly language	1.377%	-0.46%
12	19	▲▲	MATLAB	1.196%	-0.03%
13	17	▲▲	Perl	1.102%	-0.66%
14	9	▼▼	Delphi/Object Pascal	1.066%	-1.52%
15	13	▼	R	1.043%	-1.04%
16	10	▼▼	Ruby	1.037%	-1.50%
17	12	▼▼	Visual Basic	0.991%	-1.19%
18	18		Go	0.960%	-0.46%
19	49	▲▲	Groovy	0.936%	+0.75%
20	16	▼▼	Swift	0.918%	-0.88%

Đặc điểm java

1. Đơn giản
2. Tựa C++, hướng đối tượng hoàn toàn
3. Khả chuyển, độc lập nền.
4. Mạnh
5. An toàn, bảo mật.
6. Phân tán
7. Đa luồng

Đặc điểm java

1. Đơn giản: Loại bỏ các đặc trưng phức tạp của C và C++:

- Con trỏ
- Thao tác định nghĩa chồng toán tử
- Goto
- Struct
- Union

Đặc điểm java

2. Hướng đối tượng: Thuần hướng đối tượng

- Chương trình có ít nhất một lớp
- Hỗ trợ các tính chất hướng đối tượng
- Các khái niệm mới: lớp nội, interface

Đặc điểm java

3. Độc lập nền: Chạy được trên nhiều nền tảng (CPU) khác nhau (Write Once Run Anywhere):

- Trình biên dịch và trình thông dịch
- Java chuyển mã nguồn thành dạng bytecode độc lập với phần cứng mà có thể chạy trên bất kỳ CPU nào.
- Để thực thi chương trình dưới dạng bytecode, cần có trình thông dịch của Java hay còn gọi là máy ảo Java (JVM)
- Máy ảo Java chuyển bytecode thành mã lệnh mà CPU thực thi được.

```
public class MyClass  
{  
    ...  
}
```

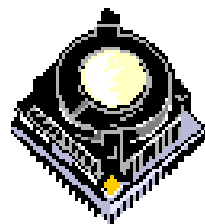
MyClass.java

javac
compile

```
0xCAFEFABE  
0x0001  
...
```

MyClass.class

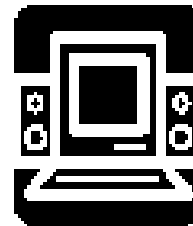
Bytecode
↓



Host
System



Machine Code



Java Virtual
Machine

Đặc điểm java

4. Mạnh mẽ

- Chặt chẽ kiểu dữ liệu: tường minh, phải cấp phát
- Kiểm tra lúc biên dịch và cả trong thời gian thông dịch vì vậy Java loại bỏ một một số loại lỗi lập trình nhất định
- Kiểm tra tất cả các truy nhập đến mảng, chuỗi khi thực thi
- Kiểm tra sự chuyển đổi kiểu dữ liệu từ dạng này sang dạng khác lúc thực thi
- Giải phóng vùng nhớ tự động (Garbage Collection)
- Quản lý ngoại lệ (Exception)

Đặc điểm java

5. An toàn

- Java cung cấp một môi trường quản lý thực thi chương trình: cung cấp nhiều mức để kiểm soát tính an toàn khi thực thi chương trình
- Dữ liệu và các phương thức được đóng gói bên trong lớp (**đóng gói**), không cho phép truy xuất bộ nhớ trực tiếp, không cho truy xuất thông tin bên ngoài kích thước của mảng, GC
- **Trình biên dịch:** kiểm soát để đảm bảo mã là an toàn, và tuân theo các nguyên tắc của Java
- **Trình thông dịch:** kiểm tra xem bytecode có đảm bảo các qui tắc an toàn trước khi thực thi
- Kiểm soát việc **nạp các lớp** vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống

Đặc điểm java

6. Phân tán

- Xây dựng các ứng dụng có thể làm việc trên nhiều phần cứng, hệ điều hành và giao diện đồ họa
- Hỗ trợ cho các ứng dụng chạy trên Internet

Đặc điểm java

7. Đa luồng

- Multithreading: để thực thi các công việc đồng thời.
- Cung cấp giải pháp đồng bộ giữa các luồng
- Cho phép xây dựng các ứng dụng trên mạng chạy hiệu quả

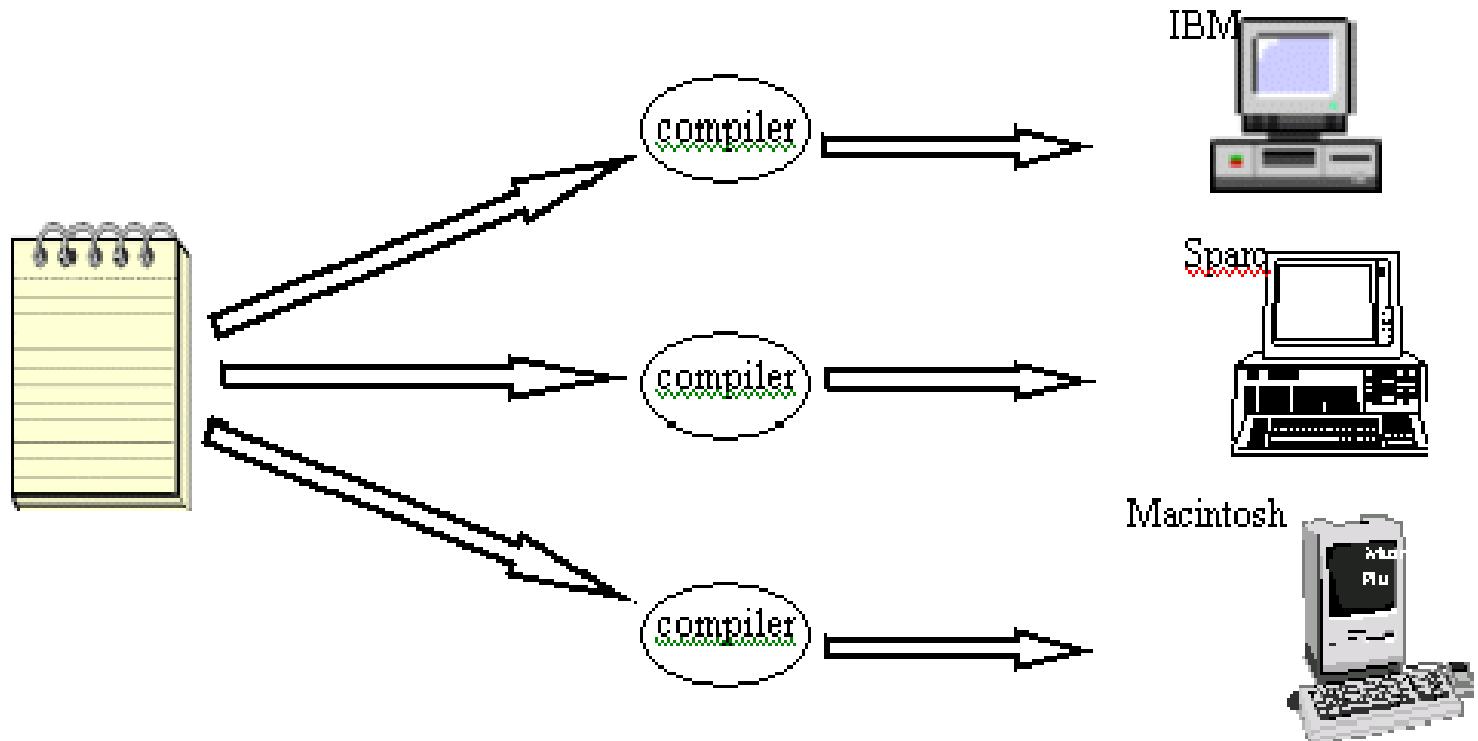
Java Virtual Machine

- Là trái tim/linh hồn của ngôn ngữ Java
- Là phần mềm dựa trên cơ sở máy tính ảo.
- Có thể xem như 1 hệ điều hành thu nhỏ, có tập hợp các lệnh logic để xác định các hoạt động của máy tính
- Thiết lập các lớp trừu tượng cho: Phần cứng bên dưới, hệ điều hành, mã đã biên dịch
- Chương trình java chỉ chạy khi có JVM
- JVM đọc và thực thi từng câu lệnh java
- ...

Java Virtual Machine

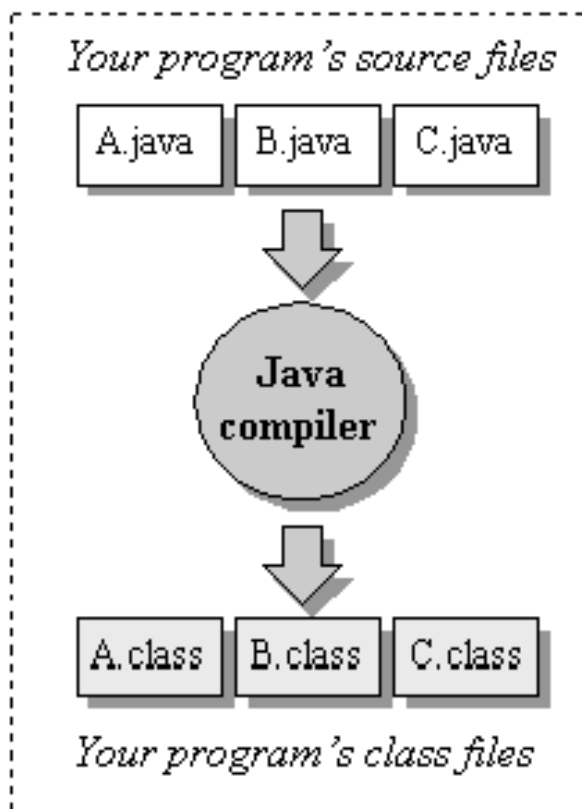
- Máy ảo tạo ra một môi trường bên trong để thực thi các lệnh bằng cách:
 - Nạp các file .class
 - Quản lý bộ nhớ
 - Dọn “rác”

Chương trình truyền thông

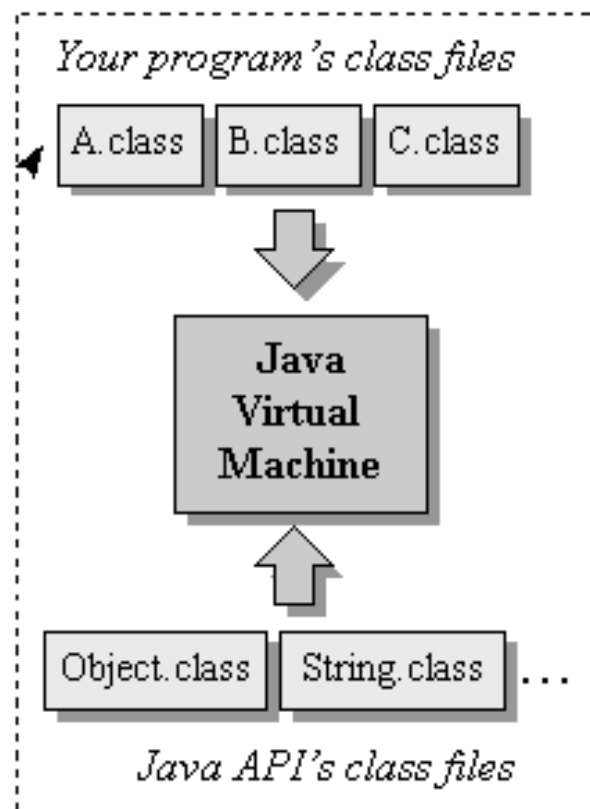


MÔI TRƯỜNG LẬP TRÌNH JAVA

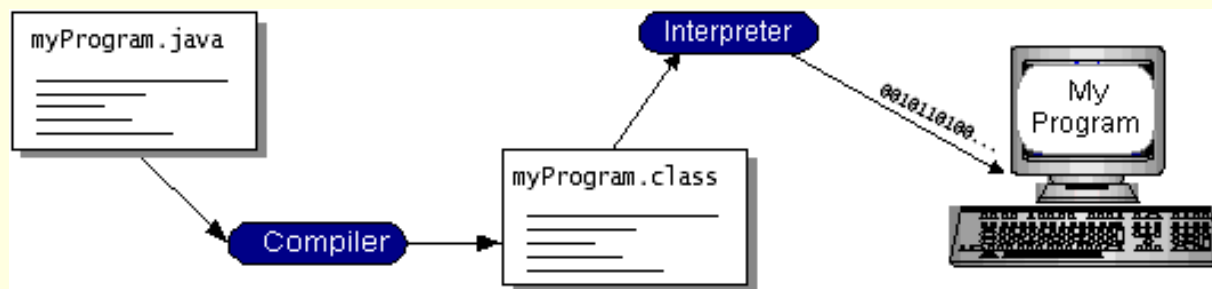
compile-time environment



run-time environment



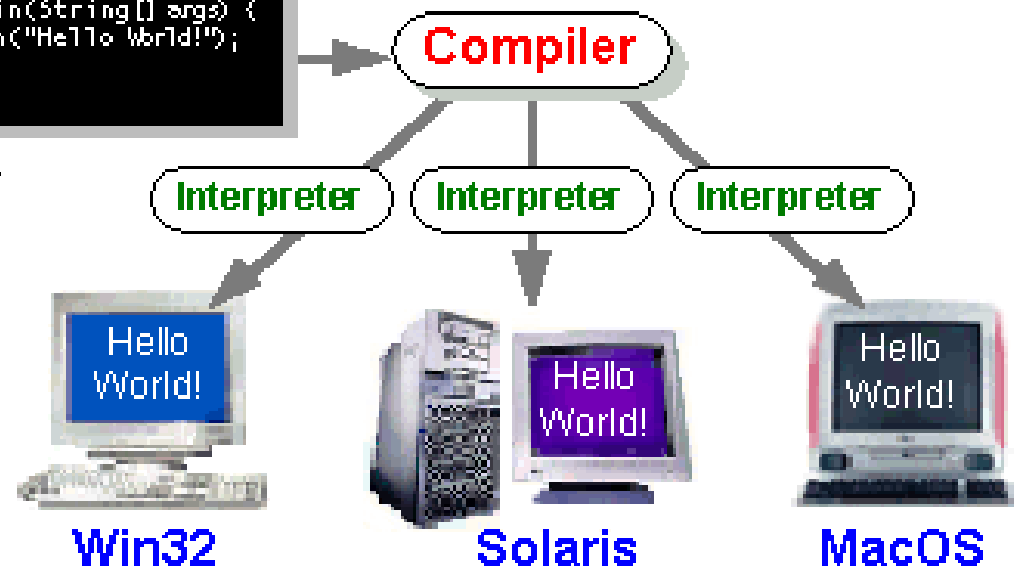
Dịch và thực thi chương trình java



Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



Viết và thực thi 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

}

Viết và thực thi chương trình Hello World (tt)

- **Biên dịch:** dùng chương trình **javac**

C:\> javac HelloWorld.java

Biên dịch thành công tạo ra tập tin có đuôi .class (HelloWorld.class)

- **Thông dịch (thực thi):** dùng chương trình **java**

C:\> java HelloWorld

Lưu ý: Phải khai báo đường dẫn chỉ đến thư mục cài đặt java, và thư mục chứa các class cần thực thi

Ví dụ: *C:\> set path=C:\jdk1.5\bin*

C:\> set classpath = D:\ThucHanhJava\BT1

Môi trường, công cụ

- Môi trường phát triển và thực thi – JDK

(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

- **IDE (Integrated Development Enviroment)**

- ✓ NetBeans

- ✓ Eclipse

- ✓ Jcreator

- ✓ Jbuilder

- ✓ ...

Thực hành

- Cài đặt JDK
- Cài đặt Eclipse, Netbean.
- Viết và thực thi chương trình HelloWorld dùng công cụ soạn thảo Notepad.
- Làm quen với môi trường phát triển NetBean, Eclipse, JCreator.
- Viết và thực thi chương trình HelloWorld dùng công cụ NetBean, Eclipse, JCreator.
- Tìm hiểu cách sử dụng jdk document để tra cứu.

CĂN BẢN VỀ NGÔN NGỮ' JAVA

NỘI DUNG

- Biến & Hằng
- Kiểu dữ liệu (kiểu cơ sở, kiểu tham chiếu)
- Toán tử, biểu thức
- Các cấu trúc điều khiển (chọn, rẽ nhánh, lặp)
- Lớp bao kiểu cơ sở

Biến

- Biến là một **vùng nhớ** lưu các giá trị của chương trình
- Mỗi biến gắn với 1 kiểu dữ liệu và 1 định danh duy nhất là tên biến
- Tên biến phân biệt chữ hoa và chữ thường. Tên biến bắt đầu bằng 1 dấu _, \$, hay 1 ký tự, không được bắt đầu bằng 1 ký số.

Khai báo

<kiểu dữ liệu> <tên biến>;

<kiểu dữ liệu> <tên biến> = <giá trị>;

Gán giá trị

<tên biến> = <giá trị>;

Phân loại biến

- Biến trong Java có 2 loại: **member** variable và **local** variable.
- **Member**
 - Không cần khởi tạo giá trị (được tự động gán giá trị mặc định)
 - Được khai báo là thành phần của lớp
- **Local**
 - Bắt buộc phải khởi tạo giá trị trước khi sử dụng (nếu không sẽ tạo ra lỗi khi biên dịch)
 - Được khai báo trong một phương thức
- Đặt tên: Viết hoa chữ cái đầu tiên các từ (SoLuongNhanVien, HoTen, NgaySinh,...)

Hằng

- Là một giá trị **bất biến** trong chương trình
- Tên đặt: Viết hoa tất cả các kí tự (NUMER_OF_MEMBER, PI,...)
- Được khai báo dùng từ khóa **final**, và thường dùng tiếp vĩ ngữ đối với các hằng số (l, L, d, D, f, F) nhằm chỉ rõ kiểu dữ liệu
- Ví dụ:

final int x = 10; // khai báo hằng số nguyên x = 10

final long y = 20L; // khai báo hằng số long y = 20

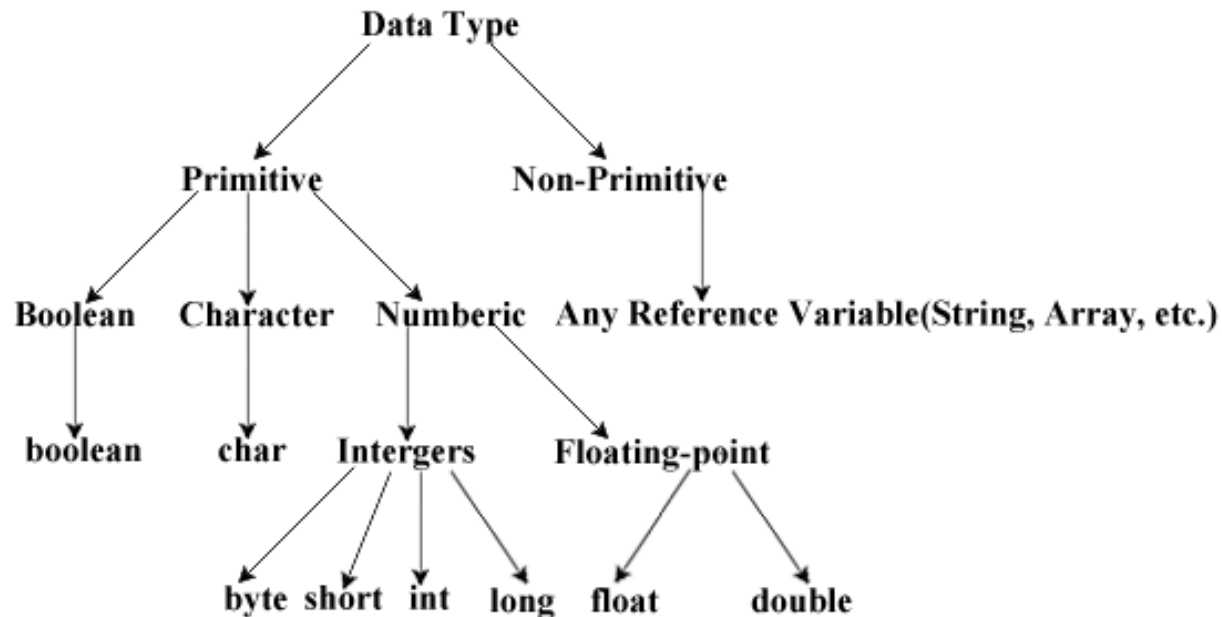
- Hằng ký tự: đặt giữa cặp nháy đơn ``
- Hằng chuỗi: là một dãy ký tự đặt giữa cặp nháy đôi ""

Hàng ký tự đặc biệt

Ký tự	Ý nghĩa
\b	Xóa lùi (BackSpace)
\t	Tab
\n	Xuống hàng
\r	Dấu enter
\"	Nháy kép
\'	Nháy đơn
\\	\
\f	Đẩy trang
\uxxxx	Ký tự unicode

Kiểu dữ liệu

- Kiểu dữ liệu cơ sở (primitive data type)
- Kiểu dữ liệu tham chiếu (reference data type)



Kiểu dữ liệu cơ sở (tt)

Kiểu	Kích thước (bits)	Giá trị	Giá trị mặc định
boolean	[<i>Note:</i> The representation of a boolean is specific to the Java Virtual Machine on each computer platform.]	true và false	false
char	16	'\u0000' to '\uFFFF' (0 to 65535)	null
byte	8	-128 to +127 (-2^7 to $2^7 - 1$)	0
short	16	-32,768 to +32,767 (-2^{15} to $2^{15} - 1$)	0
int	32	-2,147,483,648 to +2,147,483,647 (-2^{31} to $2^{31} - 1$)	0
long	64	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 (-2^{63} to $2^{63} - 1$)	0l
float	32	1.40129846432481707e-45 to 3.4028234663852886E+38	0.0f
double	64	4.94065645841246544e-324 to 1.7976931348623157E+308	0.0d

Chi tiết: docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

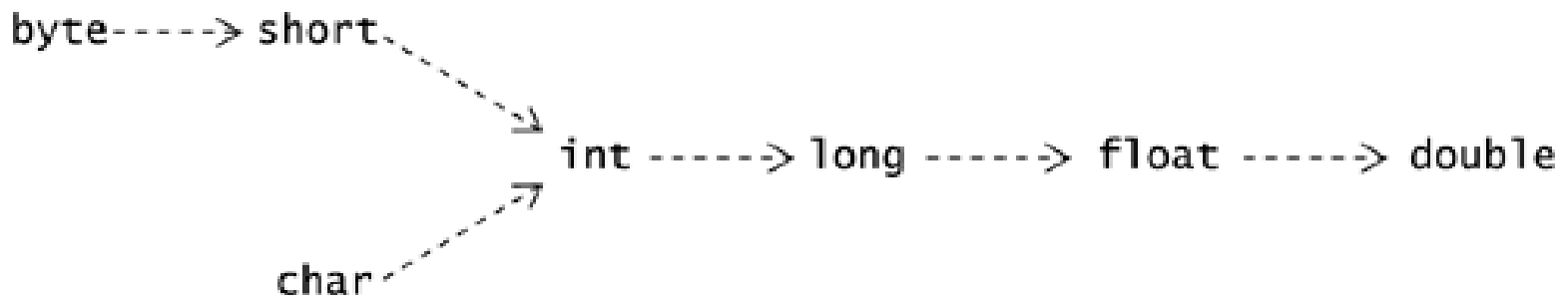
Kiểu dữ liệu cơ sở (tt)

- **Chuyển đổi kiểu dữ liệu:** khi có sự không tương thích về kiểu dữ liệu (gán, tính toán biểu thức, truyền đối số gọi phương thức)

- ✓ Chuyển kiểu hẹp (lớn → nhỏ): ***cần ép kiểu***

<tên biến 2> = (kiểu dữ liệu) <tên biến 1>;

- ✓ Chuyển kiểu rộng (nhỏ → lớn): ***tự động chuyển***



Kiểu dữ liệu cơ sở (tt)

- **Lưu ý**

1. Không thể chuyển đổi giữa kiểu boolean với int và ngược lại.
2. Nếu 1 toán hạng kiểu **double** thì

“Toán hạng kia chuyển thành **double**”

Nếu 1 toán hạng kiểu **float** thì

“Toán hạng kia chuyển thành **float**”

Nếu 1 toán hạng kiểu **long** thì

“Toán hạng kia chuyển thành **long**”

Ngược lại “Tất cả chuyển thành **int** để tính toán”

Kiểu dữ liệu cơ sở (tt)

- Ví dụ minh họa

1. *byte* $x = 5$;

2. *byte* $y = 10$;

3. *byte* $z = x + y$;

Chương trình có lỗi ko???

// Dòng lệnh thứ 3 báo lỗi chuyển kiểu cần sửa lại

// byte z = (byte) (x + y);

Kiểu dữ liệu tham chiếu

Khai báo biến tham chiếu

<Kiểu đối tượng> <biến ĐT>;

Khởi tạo đối tượng

*<Kiểu đối tượng> <biến ĐT> = **new** <Kiểu đối tượng>;*

Truy xuất thành phần đối tượng

<biến ĐT>.<thuộc tính>

<biến ĐT>.<phương thức>

Kiểu dữ liệu tham chiếu

- Kiểu mảng

- ✓ Mảng là tập hợp các phần tử có cùng tên và cùng kiểu dữ liệu.
- ✓ Mỗi phần tử được truy xuất thông qua chỉ số

- Khai báo mảng

<kiểu dữ liệu>[] <tên mảng>; // mảng 1 chiều

<kiểu dữ liệu> <tên mảng>[]; // mảng 1 chiều

<kiểu dữ liệu>[][] <tên mảng>; // mảng 2 chiều

<kiểu dữ liệu> <tên mảng>[][]; // mảng 2 chiều

Kiểu dữ liệu tham chiếu (tt)

- Khởi tạo

```
int    arrInt[] = {1, 2, 3};
```

```
char    arrChar[]      = {'a', 'b', 'c'};
```

```
String  arrString[]    = {"ABC", "EFG", "GHI"};
```

```
int[][] 2dimensionArrInt =
```

```
{
```

```
    {16, 3, 2},
```

```
    {5, 10, 11, 8},
```

```
    {9, 6, 7, 12},
```

```
    {4}
```

```
};
```

Kiểu dữ liệu tham chiếu (tt)

- **Arrays class**

java.util.Arrays

public static int **binarySearch**(Object[] a, Object key)

public static boolean **equals**(long[] a, long[] a2)

public static void **fill**(int[] a, int val)

public static void **sort**(Object[] a)

Kiểu dữ liệu tham chiếu (tt)

- **Cấp phát & truy cập mảng**

*int [] arrInt = **new** int[100];*

int arrInt[100]; // Khai báo này trong Java sẽ bị báo lỗi

Chỉ số mảng ***n*** phần tử: từ ***0*** đến ***n-1***

- **Mảng có số phần tử bằng 0**

Int [] arrInt=new int[0];

Array Copying

int[] arr2= Arrays.copyOf(arr1, size);

In mảng: *arrInt.toString();*

Kiểu dữ liệu tham chiếu (tt)

- Kiểu enum

```
enum Size { SMALL, MEDIUM, LARGE, EXTRA_LARGE };
```

```
Size s = Size.MEDIUM;
```

```
switch(s)
```

```
{
```

```
    case SMALL:
```

```
        ...
```

```
    case MEDIUM:
```

```
        ...
```

```
}
```

Kiểu dữ liệu tham chiếu (tt)

- **String**

- Chuỗi các kí tự Unicode
- '==' kiểm tra 2 String có cùng địa chỉ → `str1.equals(str2);`
- null vs length = 0
- Code unit
- StringBuilder (String vs StringBuilder:
<https://www.geeksforgeeks.org/c-sharp-string-vs-stringbuilder/>)

Kiểu dữ liệu tham chiếu (tt)

- **String**

```
String s1 = "Hello";
```

```
String s2 = "Hello";
```

```
String s3 = s1;
```

```
String s4 = new String("Hello");
```

```
String s5 = new String("Hello");
```

```
s1 == s1;      // true
```

```
s1 == s2;      // true
```

```
s1 == s3;      // true
```

```
s1 == s4;      // false
```

```
s4 == s5;      // false
```

Modifier Types

Access Control Modifiers

- default
- private
- public
- protected

Non Access Modifiers

- static
- final
- abstract
- synchronized

Toán tử, biểu thức

- Toán tử số học

Toán tử	Ý nghĩa
+	Cộng
-	Trừ
*	Nhân
/	Chia nguyên
%	Chia dư
++	Tăng 1
--	Giảm 1

Toán tử, biểu thức (tt)

- **Phép toán trên bit**

Toán tử	Ý nghĩa
&	AND
	OR
^	XOR
<<	Dịch trái
>>	Dịch phải
~	Bù bit

Toán tử, biểu thức (tt)

- Toán tử quan hệ & logic

Toán tử	Ý nghĩa
<code>==</code>	So sánh bằng
<code>!=</code>	So sánh khác
<code>></code>	So sánh lớn hơn
<code><</code>	So sánh nhỏ hơn
<code>>=</code>	So sánh lớn hơn hay bằng
<code><=</code>	So sánh nhỏ hơn hay bằng
<code> </code>	OR (biểu thức logic)
<code>&&</code>	AND (biểu thức logic)
<code>!</code>	NOT (biểu thức logic)

Toán tử, biểu thức (tt)

- Toán tử gán

Toán tử	Ví dụ	Ý nghĩa
=	$a = b$	gán $a = b$
+=	$a += 5$	$a = a + 5$
-=	$b -= 10$	$b = b - 10$
*=	$c *= 3$	$c = c * 3$
/=	$d /= 2$	$d = d/2$
%=	$e \% = 4$	$e = e \% 4$

Độ ưu tiên

Operators	Associativity
[] . () (method call)	Left to right
! ~ ++ -- + (unary) - (unary) () (cast) new	Right to left
* / %	Left to right
+ -	Left to right
<< >> >>>	Left to right
< <= > >= instanceof	Left to right
== !=	Left to right
&	Left to right
^	Left to right
	Left to right
&&	Left to right
	Left to right
?:	Right to left
= += -= *= /= %= &= = ^= <<= >>= >>>=	Right to left

Toán tử, biểu thức (tt)

- Toán tử điều kiện

Cú pháp: *<điều kiện> ? <biểu thức 1> : <biểu thức 2>*

Ví dụ:

```
int x = 10;
```

```
int y = 20;
```

```
int Z = (x < y) ? 30 : 40;
```

```
// Kết quả z = 30 do biểu thức (x < y) là đúng.
```

Cấu trúc điều khiển

- Cấu trúc *if ... else*

Dạng 1: *if* (<điều_kiện>) {
 <khởi_lệnh>;
 }

Dạng 2: *if* (<điều_kiện>) {
 <khởi_lệnh1>;
 }
 else {
 <khởi_lệnh2>;
 }

Cấu trúc điều khiển

- Cấu trúc ***switch ... case***

```
switch (<biến>) {  
    case <giá trị_1>:  
        <khởi_lệnh_1>;  
        break;  
  
    ....  
    case <giá trị_n>:  
        <khởi_lệnh_n>;  
        break;  
    default:  
        <khởi_lệnh default>;  
}
```


Cấu trúc điều khiển

- **Cấu trúc lặp**

- **Dạng 1:** *while (<điều_kiện_lặp>) {...}*

- **Dạng 2:** *do {...*
} while (điều_kiện);

- **Dạng 3:** *for (khởi_tạo_biến_đếm;đk_lặp;tăng_biến) {...}*

- **Dạng 4 (từ Java 5)**

- int arr[]={1, 2, 3};*

- for(int i: arr){....}*

Cấu trúc điều khiển

- **Cấu trúc lệnh nhảy jump:** dùng kết hợp nhãn (label) với từ khóa ***break*** và ***continue*** để thay thế cho lệnh ***goto*** (trong C).

Ví dụ:

```
label:
for (...) {
    for (...) {
        if (<biểu thức điều kiện>)
            break label;
        else
            continue label;
    }
}
```

Input

Lớp java.util.Scanner

public boolean

nextBoolean() [Details](#)

public byte

nextByte() [Details](#)

public byte

nextByte(int radix) [Details](#)

public double

nextDouble() [Details](#)

public float

nextFloat() [Details](#)

public int

nextInt() [Details](#)

public int

nextInt(int radix) [Details](#)

public String

nextLine() [Details](#)

public long

nextLong() [Details](#)

public long

nextLong(int radix) [Details](#)

public short

nextShort() [Details](#)

public short

nextShort(int radix) [Details](#)

Input

```
Scanner in = new Scanner(System.in);
```

```
System.out.print("What is your name? ");
```

```
String name = in.nextLine();
```

```
System.out.print("How old are you? ");
```

```
int age = in.nextInt();
```

Output

System.out.print(...);

System.out.println(...);

System.out.printf(...);

Output

printf

Conversion character	Type	Example
d	Decimal integer	159
x	Hexadecimal integer	9f
o	Octal integer	237
f	Fixed-point floating-point	15.9
e	Exponential floating-point	1.59e+01
g	General floating-point (the shorter of e and f)	—
a	Hexadecimal floating-point	0x1.fccdp3
s	String	Hello
c	Character	H
b	boolean	true
h	Hash code	42628b2
tx	Date and time	See Table 3.7
%	The percent symbol	%
n	The platform-dependent line separator	—

Output

printf flags

Flag	Purpose	Example
+	Prints sign for positive and negative numbers	+3333.33
space	Adds a space before positive numbers	3333.33
0	Adds leading zeroes	003333.33
-	Left-justifies field	3333.33
(Encloses negative numbers in parentheses	(3333.33)
,	Adds group separators	3,333.33
# (for f format)	Always includes a decimal point	3,333.
# (for x or o format)	Adds 0x or 0 prefix	0xcafe
\$	Specifies the index of the argument to be formatted; for example, %1\$d %1\$x prints the first argument in decimal and hexadecimal	159 9F
<	Formats the same value as the previous specification; for example, %d %<x prints the same number in decimal and hexadecimal	159 9F

Lớp bao kiểu dữ liệu

Data type	Wrapper Class (java.lang.*)	Ghi chú
boolean	Boolean	<ul style="list-style-type: none">- Gói (package): chứa nhóm nhiều class.- Ngoài các Wrapper Class, gói java.lang còn cung cấp các lớp nền tảng cho việc thiết kế ngôn ngữ java như: String, Math, ...
byte	Byte	
short	Short	
char	Character	
int	Integer	
long	Long	
float	Float	
double	Double	

Tại sao cần lớp bao kiểu dữ liệu???

- Chuyển kiểu số (primitive) thành kiểu object hỗ trợ việc truyền tham chiếu cho phương thức
- Trong Java hầu hết đều là lớp (sự tương thích)
- Các cấu trúc trong Collection (Array, Vector,...) chỉ thao tác trên các biến là đối tượng
- Chỉ các đối tượng mới được hỗ trợ đồng bộ (đa luồng)

Bài tập

1. Viết và chạy chương trình xuất ra dòng "Hello World" với Java (bằng notepad và IDE)
2. Giải phương trình bậc hai $ax^2 + bx + c = 0$.
3. Nhập n số nguyên. Hãy sắp xếp giá trị của các số nguyên này theo thứ tự tăng dần.
4. Nhập vào tháng và năm. Cho biết tháng đó có bao nhiêu ngày.

Bài tập

5. Viết chương trình quản lý một dãy số nguyên gồm các tính năng:
nhập, xuất dãy; cho phép thêm, xóa, sửa các số trong dãy; sắp xếp
dãy số tăng dần; tính giá trị trung bình của dãy và cho biết phần tử
nào gần với giá trị trung bình nhất.
6. Viết chương trình xuất ra lịch của một năm (do người dùng nhập vào)
7. Viết chương trình nhập vào mảng phân số (n phần tử) và xuất ra phân
số nhỏ nhất, lớn nhất của mảng vừa nhập.

Câu hỏi ôn tập

1. Các loại biến trong Java là gì? Khác nhau giữa chúng?
2. Java có mấy kiểu dữ liệu cơ sở? Gồm những kiểu nào? String có phải là kiểu dữ liệu cơ sở?
3. Giá trị mặc định của các biến (member) thuộc kiểu dữ liệu cơ sở là gì?
4. Những kiểu tham chiếu trong Java là gì? Kể tên?
5. Cách khai báo mảng có mấy cách? Gồm những cách nào?
6. Cách nhập, xuất dữ liệu với màn hình console là gì?
7. Lớp bao (wrapper class) nhằm để làm gì?

QA???
