

Ngôn ngữ lập trình Java (Java Programming Language)

Môn học: Ngôn ngữ lập trình Java

Khoa: CNPM, Trường Đại học CNTT – ĐHQG TpHCM

GV: Th.S Huỳnh Tuấn Anh

Email: anhht@uit.edu.vn

Mục tiêu môn học

- ☐ Nắm vững các khái niệm, đặc điểm cơ bản của ngôn ngữ lập trình java.
- ☐ Hiểu rõ tư tưởng hướng đối tượng của java
- ☐ Có khả năng đọc hiểu tài liệu tiếng Anh về java.
- ☐ Có khả năng vận dụng các kỹ thuật lập trình cơ bản và nâng cao trong Java để xây dựng chương trình ứng dụng.
- ☐ Hiểu và có khả năng làm việc nhóm trong lập trình
- ☐ Hình thành được tư duy tổ chức kiến trúc chương trình một cách có hệ thống và tác phong lập trình chuyên nghiệp thông qua ngôn ngữ Java.
- ☐ Có khả năng tự học và sử dụng thành thạo một số công cụ phổ biến hỗ trợ lập trình java.

Nội dung môn học

Chương 1: Giới thiệu về NNLT Java

Chương 2: Hướng đối tượng trong Java

Chương 3: Các lớp tiện ích trong Java

Chương 4: Nhập xuất và quản lý Exception

Chương 5: Xử lý đa luồng

Chương 6: Kết nối và thao tác CSDL với JDBC

Chương 7: Lập trình GUI với AWT & Swing

Quy ước viết mã trong java

Hình thức đánh giá

- Lý thuyết cuối kỳ: 50%
- Thực hành: 30%
- Seminar/Bài tập: 20%

Danh sách đề tài

- Xây dựng Notepad++ cho Java/C++/C#, HTML ...
- Xây dựng chương trình Paint cho hình vẽ Vector
- Xây dựng chương trình mô phỏng cho các CTDL: danh sách liên kết, stack, queue
- Xây dựng chương trình mô phỏng cho các thuật toán sắp xếp
- Xây dựng chương trình mô phỏng cho CTDL cây
- Xây dựng chương trình mô phỏng cho Đồ thị

Danh sách đề tài

- Xây dựng chương trình mô phỏng các loại chuyển động
- Xây dựng chương trình mô phỏng mạch điện
- Xây dựng chương trình hỗ trợ học hình học như Carbi
- Xây dựng chương trình hỗ trợ hình học 3D
- Xây dựng chương trình từ điển Anh – Việt có hình ảnh, âm thanh

Danh sách đề tài

- Xây dựng chương trình Total Commander
- Xây dựng ứng dụng hỗ trợ học lập trình
- Xây dựng chương trình Lan Chat
- Xây dựng chương trình Mail Client
- Các ứng dụng tiện ích
- Các đề tài tự chọn: XNA, Multipoint
- ...

TÀI LIỆU THAM KHẢO

1. Herbert Schildt, Java, A Beginner's Guide, 5th Edition, McGraw-Hill, 2011
2. Herbert Schildt, Java The Complete Reference, 8th Edition, McGraw-Hill, 2011.
3. Stuart Reges, Marty Stepp, Building Java Programs: A Back to Basics Approach, Addison Wesley, 2010.
4. Barry Burd, Beginning Programming with Java For Dummies, For Dummies, 2012.
5. Madhusudhan Konda, What's New in Java 7, O'Reilly Media, 2011

TÀI LIỆU THAM KHẢO

- MIT Opencourseware

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/lecture-notes/>

- Oracle - Java

<http://www.oracle.com/us/technologies/java/overview/index.html>

CHƯƠNG 1

TỔNG QUAN VỀ NNLT 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
- Môi trường, công cụ: giới thiệu một số IDE phổ biến

Lịch sử phát triển

[Xem Video](#)

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



- **Java** là ngôn ngữ hướng đối tượng tựa C, C++

Lịch sử phát triển Java Development Kit (JDK)

Môi trường phát triển và thực thi do Sun Microsystems cung cấp (<http://java.sun.com>)
Bao gồm phần mềm và công cụ giúp compile, debug and execute ứng dụng.

JDK 1.0 - 1996

JDK 1.1 - 1997

JDK 1.2 (Java 2) - 1998

JDK 1.3 - 2000

Java 1.4 - 2002

Java 5 (1.5) - 2004

Java 6 - 2006

• Oracle mua Sun - April 20, 2009 - \$7.4 billion

James Gosling

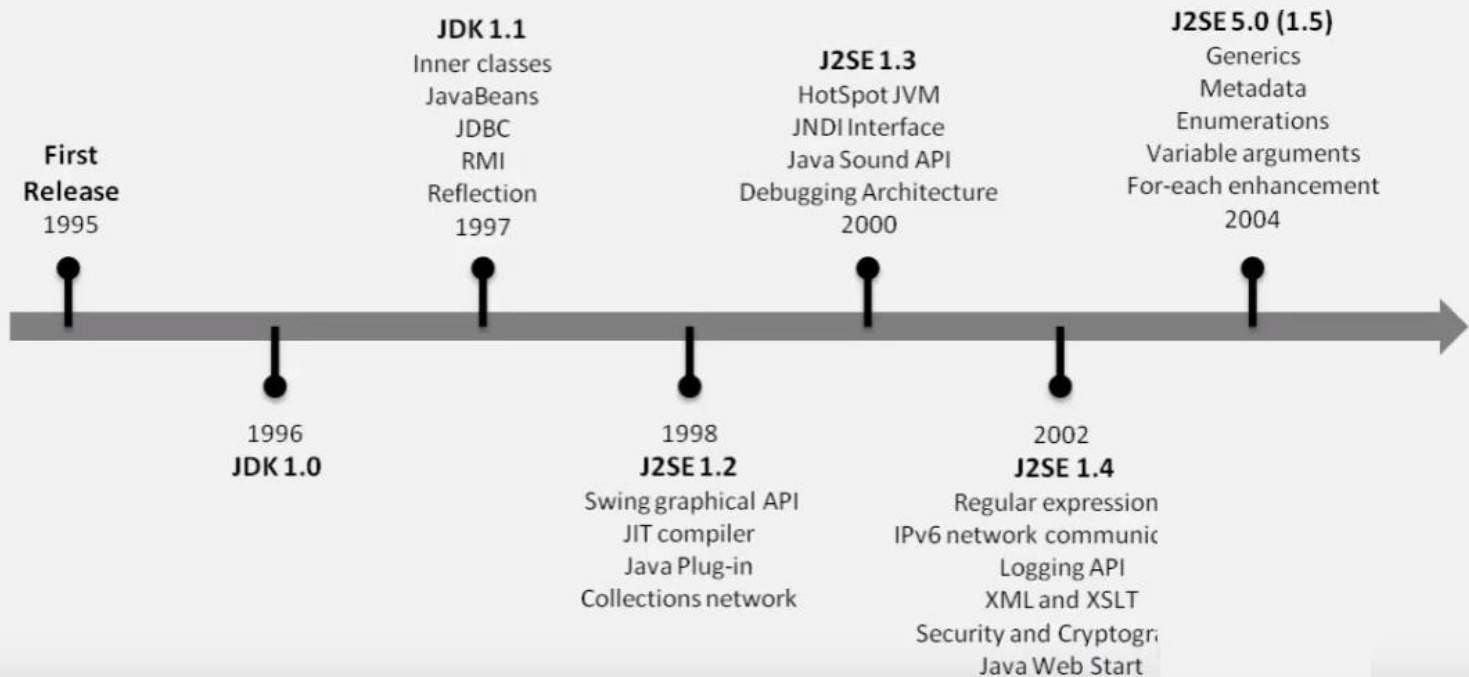


Born	James Arthur Gosling May 19, 1955 (age 58) Near Calgary, Alberta, Canada
Residence	San Francisco Bay Area, California, United States
Nationality	Canadian
Institutions	Sun Microsystems Oracle Corporation Google Liquid Robotics ^[1] Typesafe Inc.
Alma mater	Carnegie Mellon University University of Calgary
Thesis	<i>Algebraic Constraints</i>  (1983)
Doctoral advisor	Bob Sproull ^[2]
Known for	Java (programming language)
Notable awards	Officer of the Order of Canada
Website nighthacks.com 	

http://en.wikipedia.org/wiki/James_Gosling

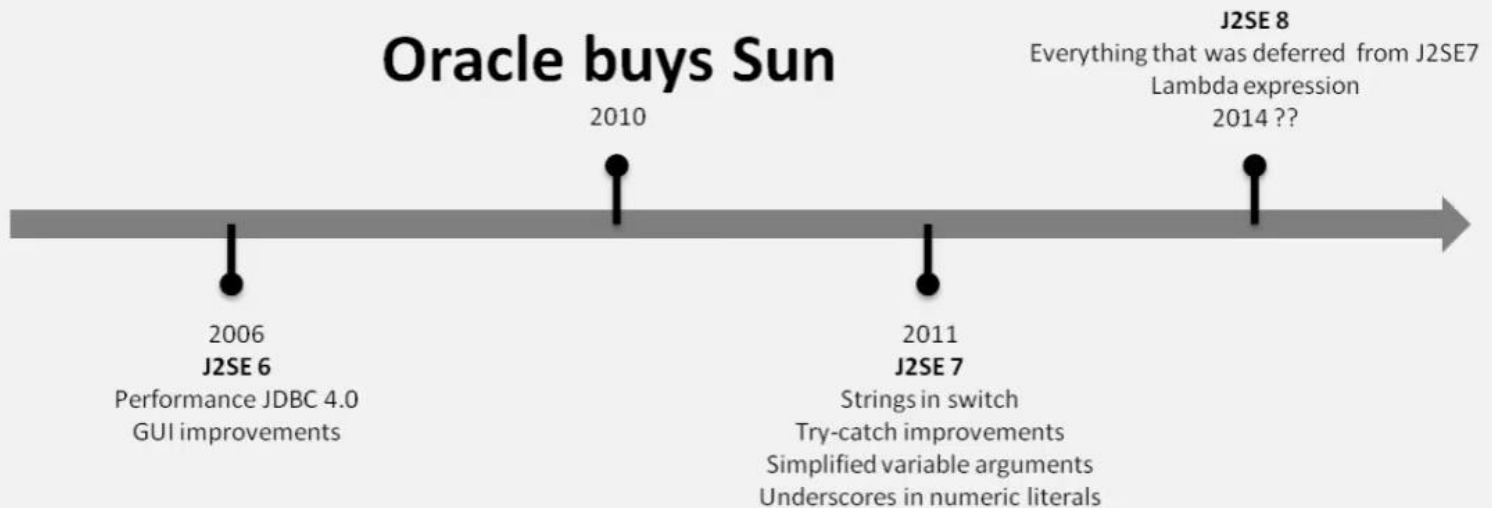
Lịch sử phát triển Java Development Kit (JDK)

HISTORY OF JAVA



Lịch sử phát triển Java Development Kit (JDK)

HISTORY OF JAVA



Lịch sử phát triển

New Name was Java



- The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA" etc.
- They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell and fun to say.
- According to James Gosling "**Java** and **Silk** were two top choices". Since java was so unique, most of the team members preferred java.
- Java:-Java is an island of Indonesia where first coffee was produced (called java coffee).
- Java coffee consumed in large quantities by the language's creators
- Java is just a name not an acronym.



Lịch sử phát triển

In the beginning...



There was C. But C was too hard. Well, for normal developers anyway. So Saint Gosling made C easier, made C better - Java was born.

- Memory management? Where Java was going, we didn't need memory management!
- Pointers? That's too old school for Java.
- Threads - built in!
- UI (AWT) built in!

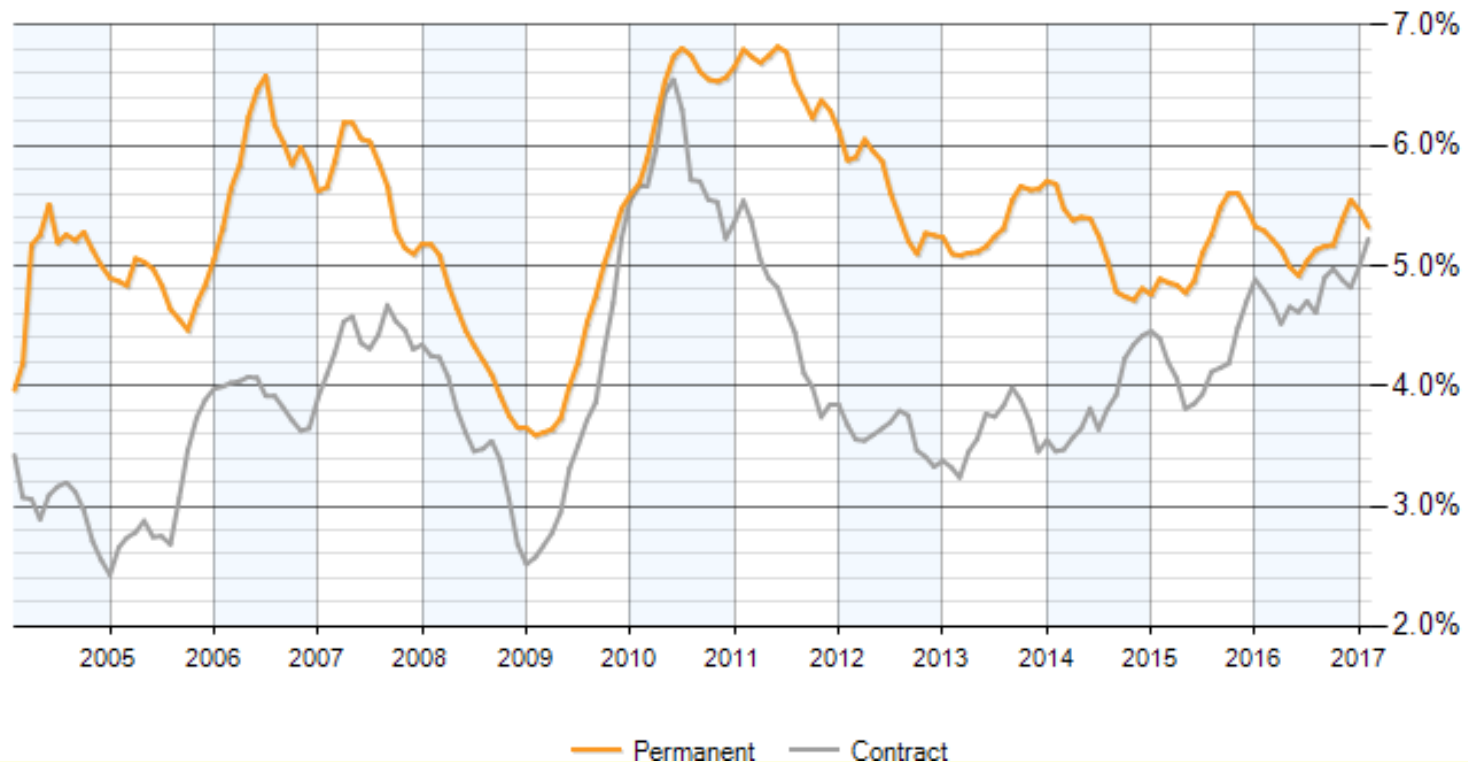
Các nền tảng Java

- Công nghệ:
 - ✓ Ngôn ngữ lập trình
 - ✓ Môi trường thực thi và triển khai
 - ✓ Môi trường phát triển
- Công nghệ J2SE (Java 2 Standard Edition)
- Công nghệ J2EE (Java 2 Enterprise Edition)
- Công nghệ J2ME (Java 2 Micro Edition)

Một số thống kê tham khảo

Java Developer Jobs Demand Trend

The demand trend of job ads that featured Java Developer in the job title.

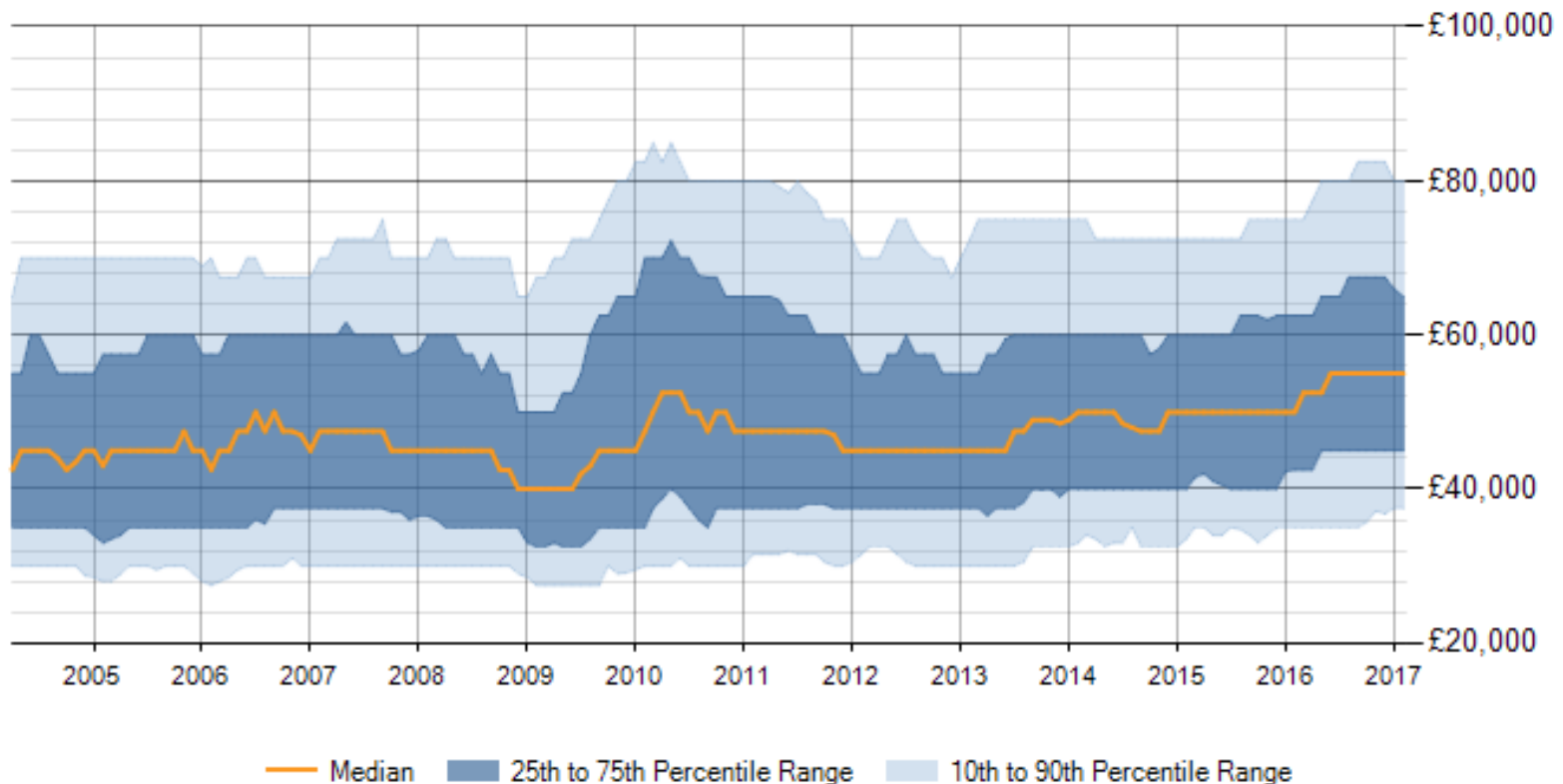


<http://www.itjobswatch.co.uk/jobs/uk/java%20developer.do>

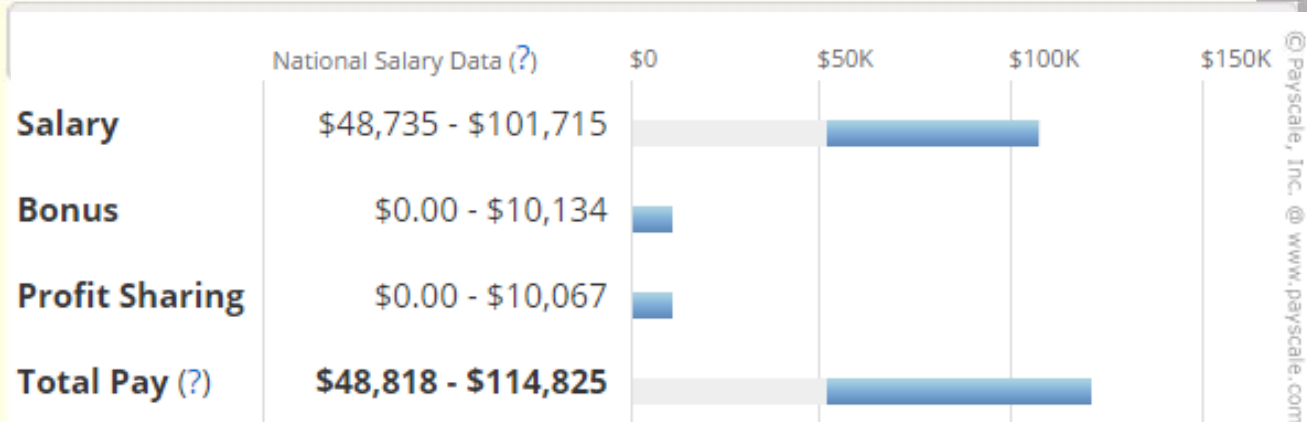
Một số thống kê tham khảo

Java Developer Salary Trend

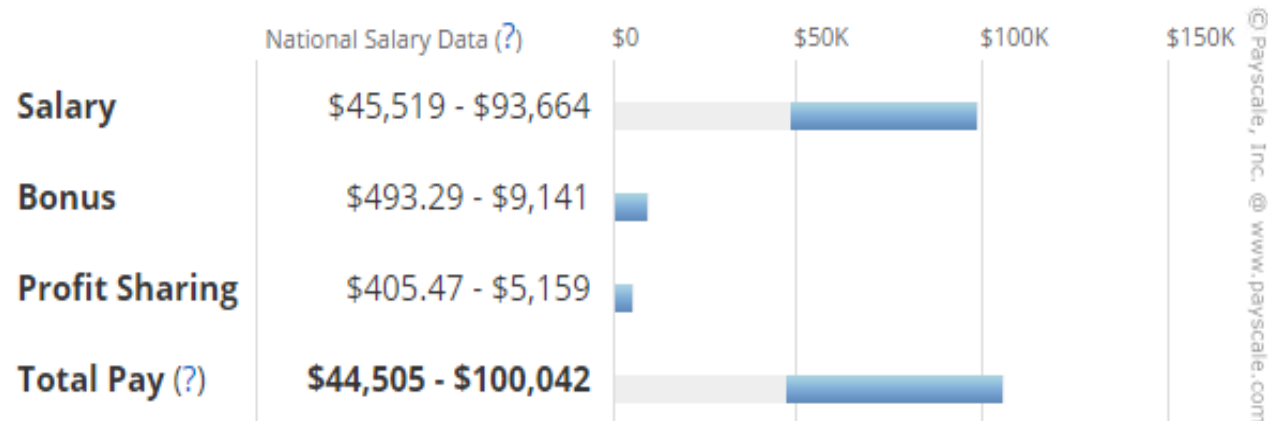
This chart provides the 3-month moving average for salaries quoted in permanent IT jobs citing Java Developer within the UK.



Một số thống kê tham khảo



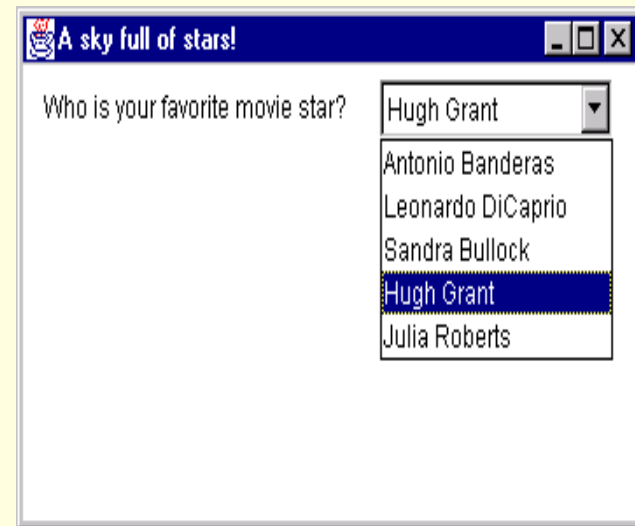
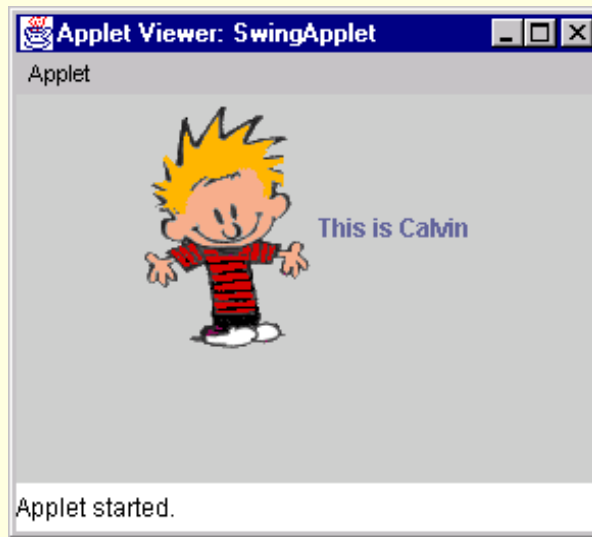
Country: United States | Currency: USD | Updated: 28 Oct 2016 | Individuals Reporting: 1,646



Country: United States | Currency: USD | Updated: 28 Oct 2016 | Individuals Reporting: 2,056

Các dạng chương trình java

- Applets:



Các dạng chương trình java

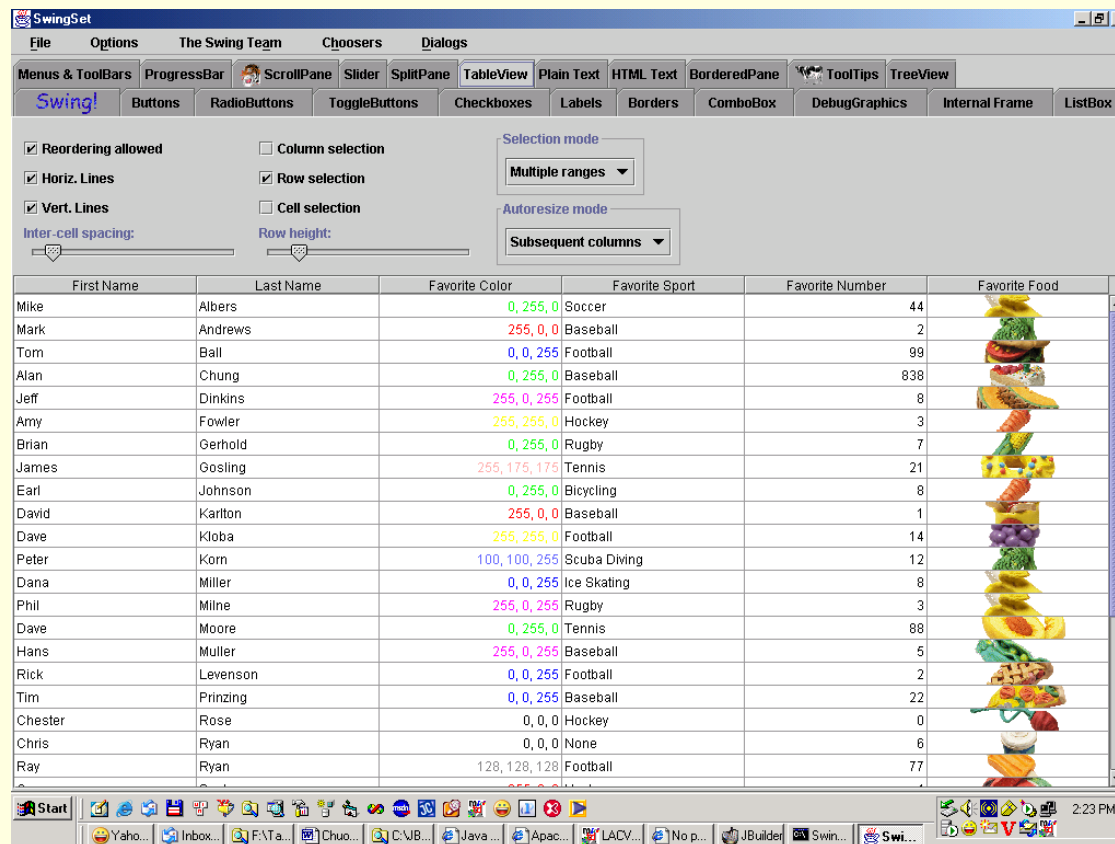
- Console Applications



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


Các dạng chương trình java

- Ứng dụng Desktop



Các dạng chương trình java

- Ứng dụng Web



Các dạng chương trình java

- Một dạng phần mềm trên thiết bị di động



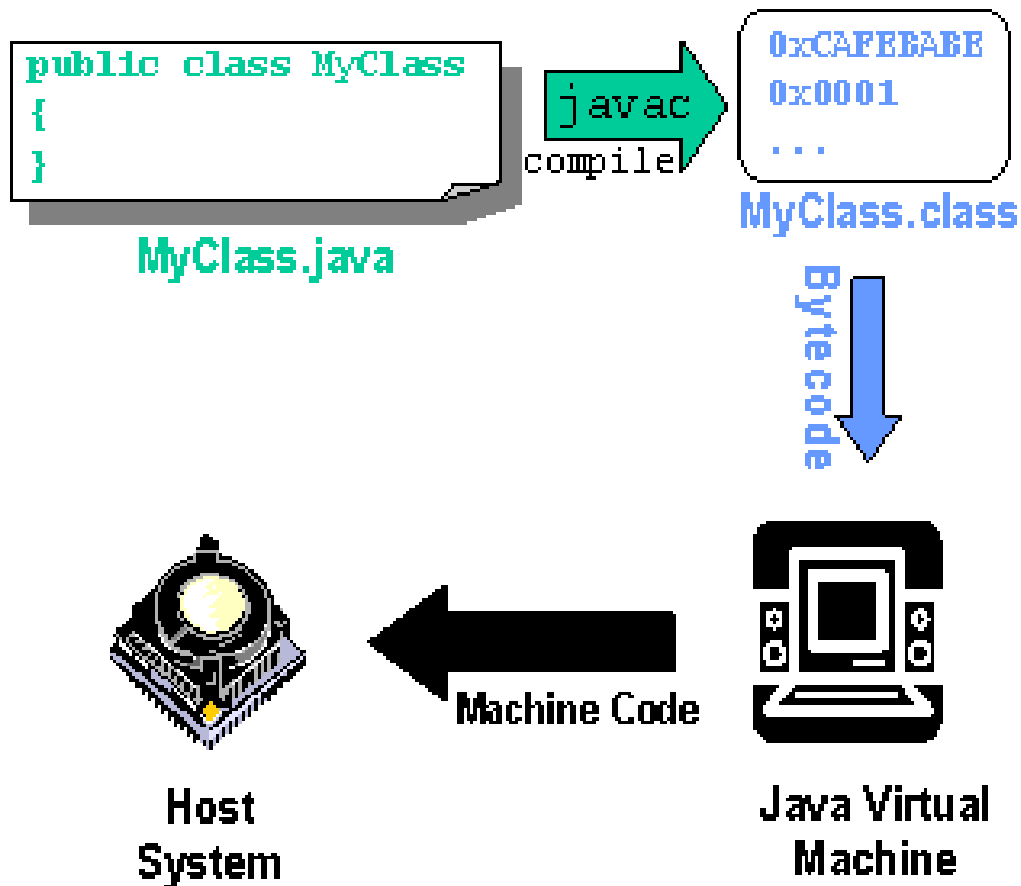
<http://www.tiobe.com/index.php/content/perinfo/tpci/index.html>

Feb 2017	Feb 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.676%	-4.47%
2	2		C	8.445%	-7.15%
3	3		C++	5.429%	-1.48%
4	4		C#	4.902%	+0.50%
5	5		Python	4.043%	-0.14%
6	6		PHP	3.072%	+0.30%
7	9	⬆	JavaScript	2.872%	+0.67%
8	7	⬇	Visual Basic .NET	2.824%	+0.37%
9	10	⬆	Delphi/Object Pascal	2.479%	+0.32%
10	8	⬇	Perl	2.171%	-0.08%
11	11		Ruby	2.153%	+0.10%
12	16	⬆	Swift	2.125%	+0.75%
13	13		Assembly language	2.107%	+0.28%
14	38	⬆	Go	2.105%	+1.81%
15	17	⬆	R	1.922%	+0.73%
16	12	⬇	Visual Basic	1.875%	+0.02%
17	18	⬆	MATLAB	1.723%	+0.63%
18	19	⬆	PL/SQL	1.549%	+0.49%
19	14	⬇	Objective-C	1.536%	+0.13%
20	23	⬆	Scratch	1.500%	+0.71%

Đặc điểm java

- Tựa C++, hướng đối tượng hoàn toàn.
- **Khả chuyển, độc lập nền.**
- Thông dịch (vừa biên dịch vừa thông dịch).
- Cơ chế giải phóng bộ nhớ tự động.
- An toàn, bảo mật.

ĐỘC LẬP VỚI NỀN



Java Virtual Machine

- Là phần mềm dựa trên cơ sở máy tính ảo.
- Có thể xem như 1 hệ điều hành thu nhỏ.
- Cung cấp môi trường thực thi cho chương trình java (độc lập nền)
- Hình thành 1 lớp trừu tượng:

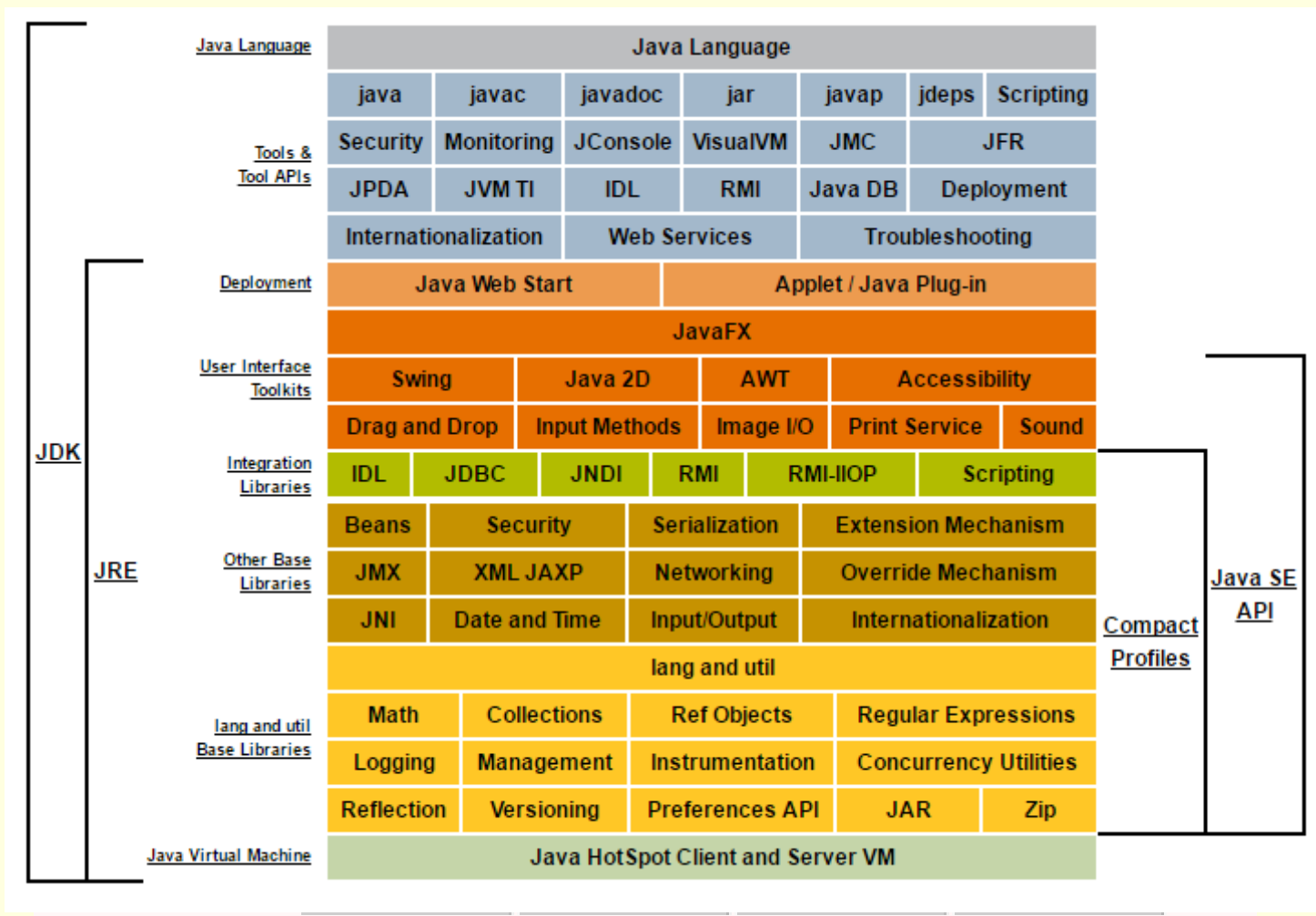
Phần cứng máy tính bên dưới

Hệ điều hành

Mã đã biên dịch

- Chương trình java chỉ chạy khi có JVM
- JVM đọc và thực thi từng câu lệnh java
- ...

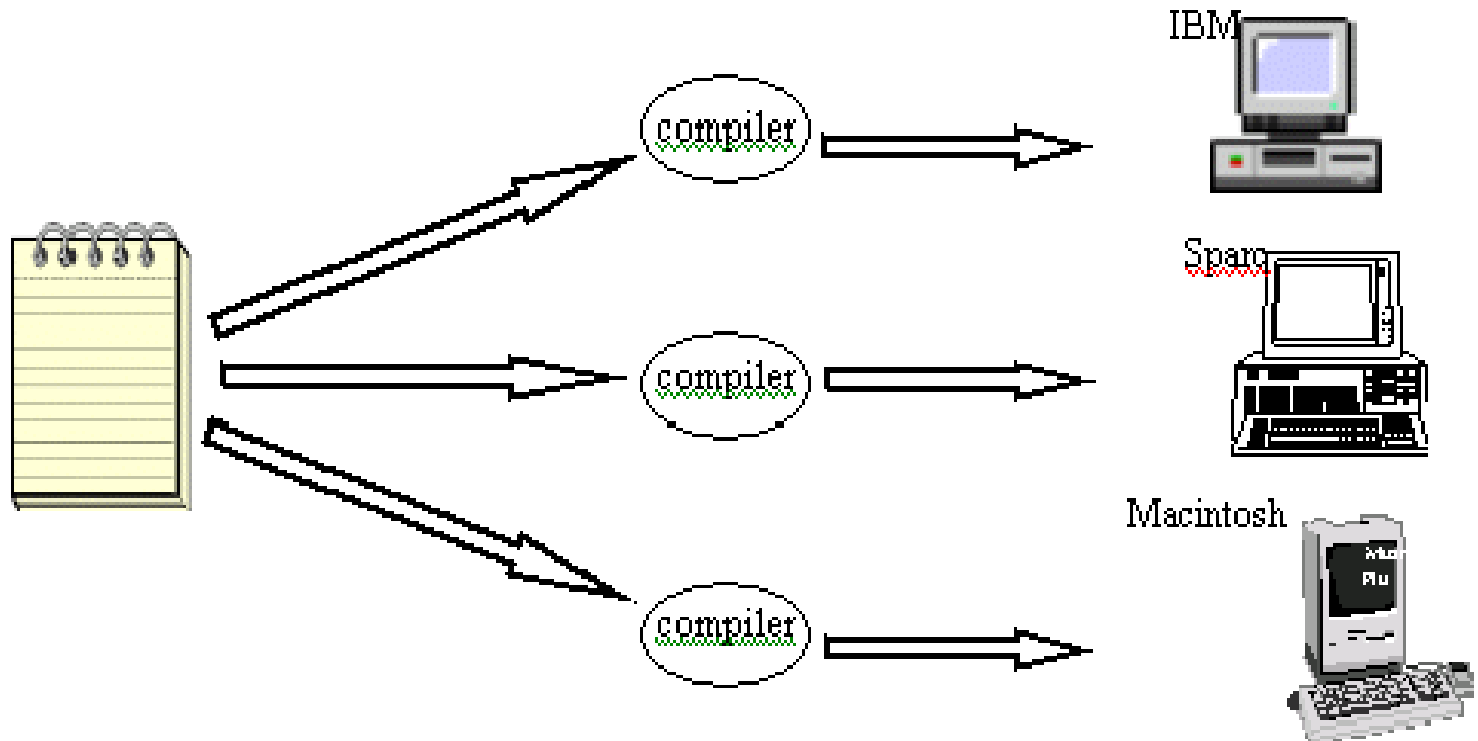
KIẾN TRÚC J2SE



Đặc điểm java

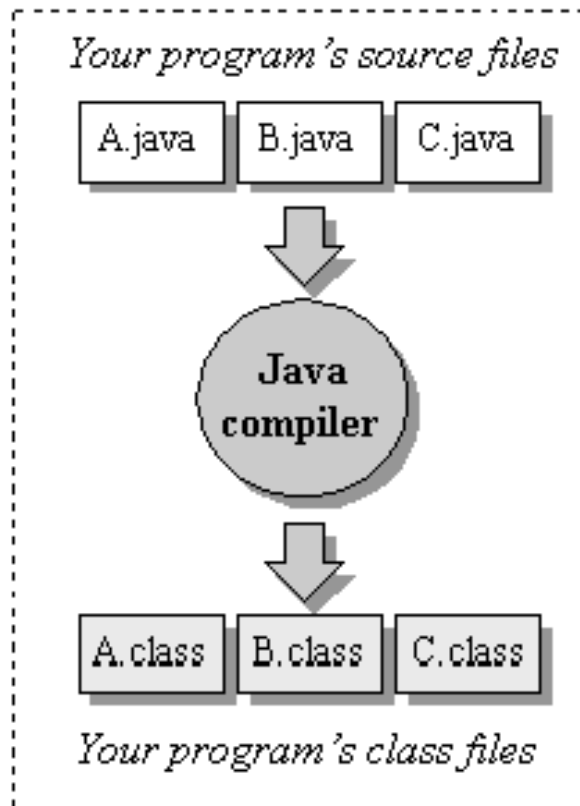
- Tựa C++, hướng đối tượng hoàn toàn.
- Khả chuyển, độc lập nền.
- **Thông dịch (vừa biên dịch vừa thông dịch).**
- Cơ chế giải phóng bộ nhớ tự động.
- An toàn, bảo mật.

Chương trình truyền thông

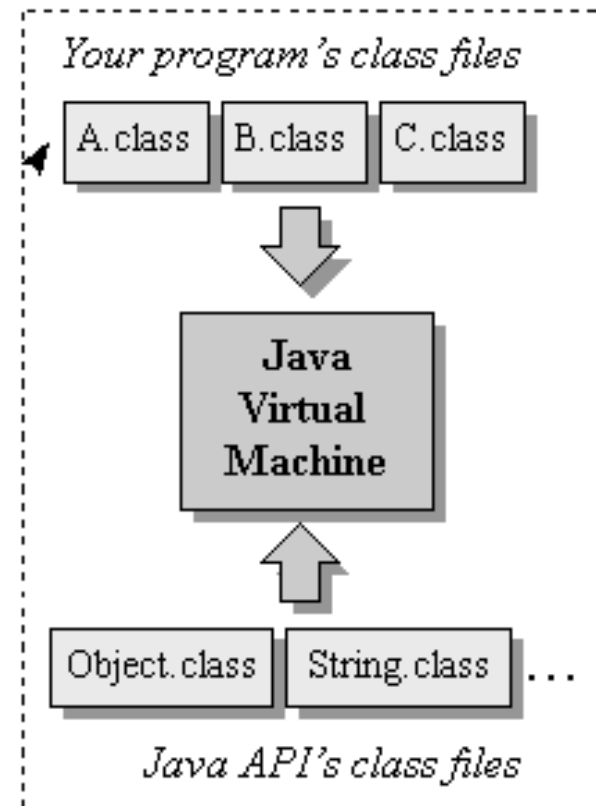


MÔI TRƯỜNG LẬP TRÌNH JAVA

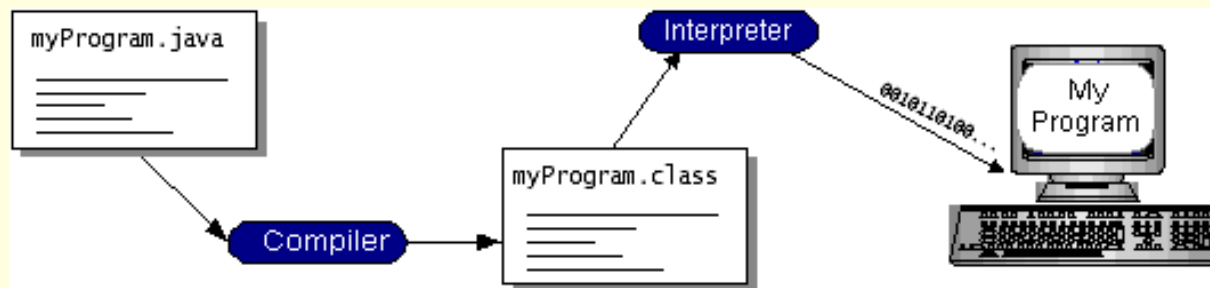
compile-time environment



run-time environment



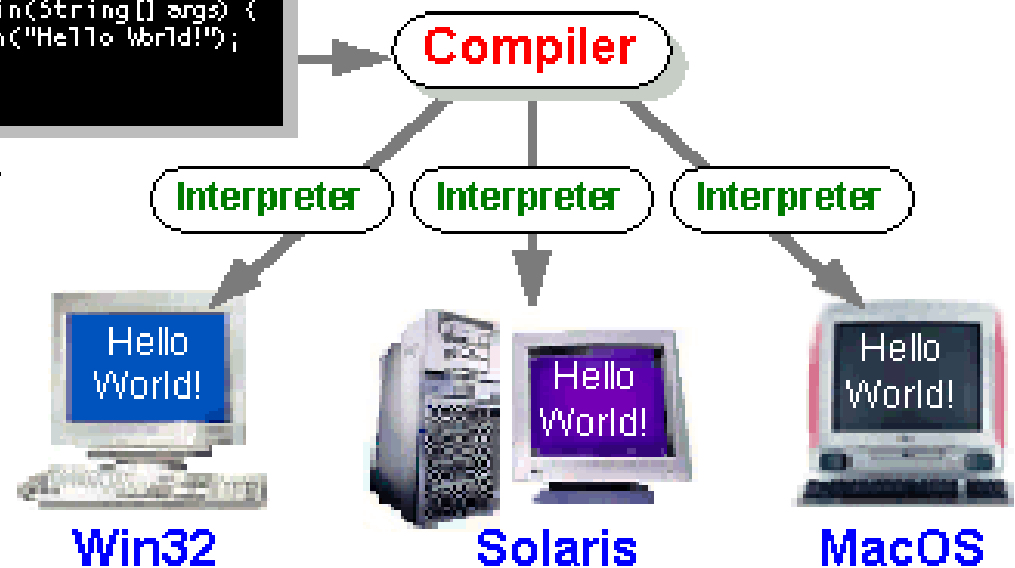
Dịch và thực thi chương trình java



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

import java.io.*;

Khai báo thư viện java.io

class HelloWorld

Định nghĩa lớp tên “**HelloWorld**”

{

Bắt đầu đoạn lệnh

public static void main(String args[])

Phương thức
main

{

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

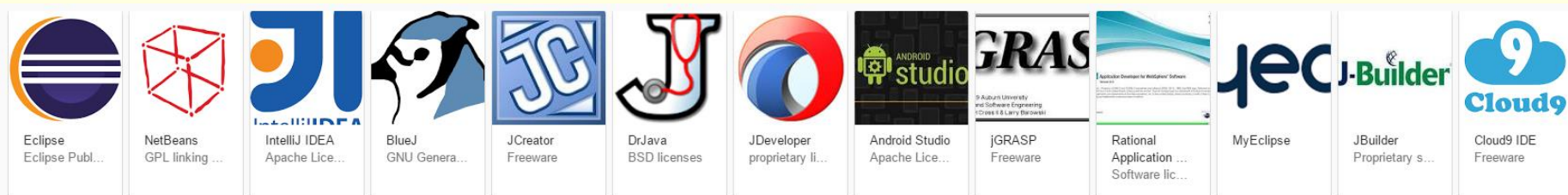
Ví dụ:

```
C:\> set path=C:\jdk1.5\bin\
```

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

Môi trường, công cụ

- Môi trường phát triển và thực thi của Java – JDK 1.8
- IDE (Integrated Development Enviroment)
 - ✓ NetBean 8.2
 - ✓ Eclipse Neon
 - ✓ Jcreator Pro 5.10
 - ✓ Jbuilder
 - ✓ ...



CHƯƠNG 2

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
- Phương thức và cách sử dụng
- Một số ví dụ minh họa

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: instance variable và local variable.
- Đối với instance variable, có thể được sử dụng mà không cần khởi tạo giá trị (được tự động gán giá trị mặc định).
- Đối với local variable, Java bắt buộc phải khởi tạo giá trị trước khi sử dụng. Nếu không sẽ tạo ra lỗi khi biên dịch.

Hằng

- Là một giá trị bất biến trong chương trình
- Tên đặt theo qui ước như tên biến
- Được khai báo dùng từ khóa **final**, và thường dùng tiếp vĩ ngữ đối với các hằng số (l, L, d, D, f, F)
- 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 ""

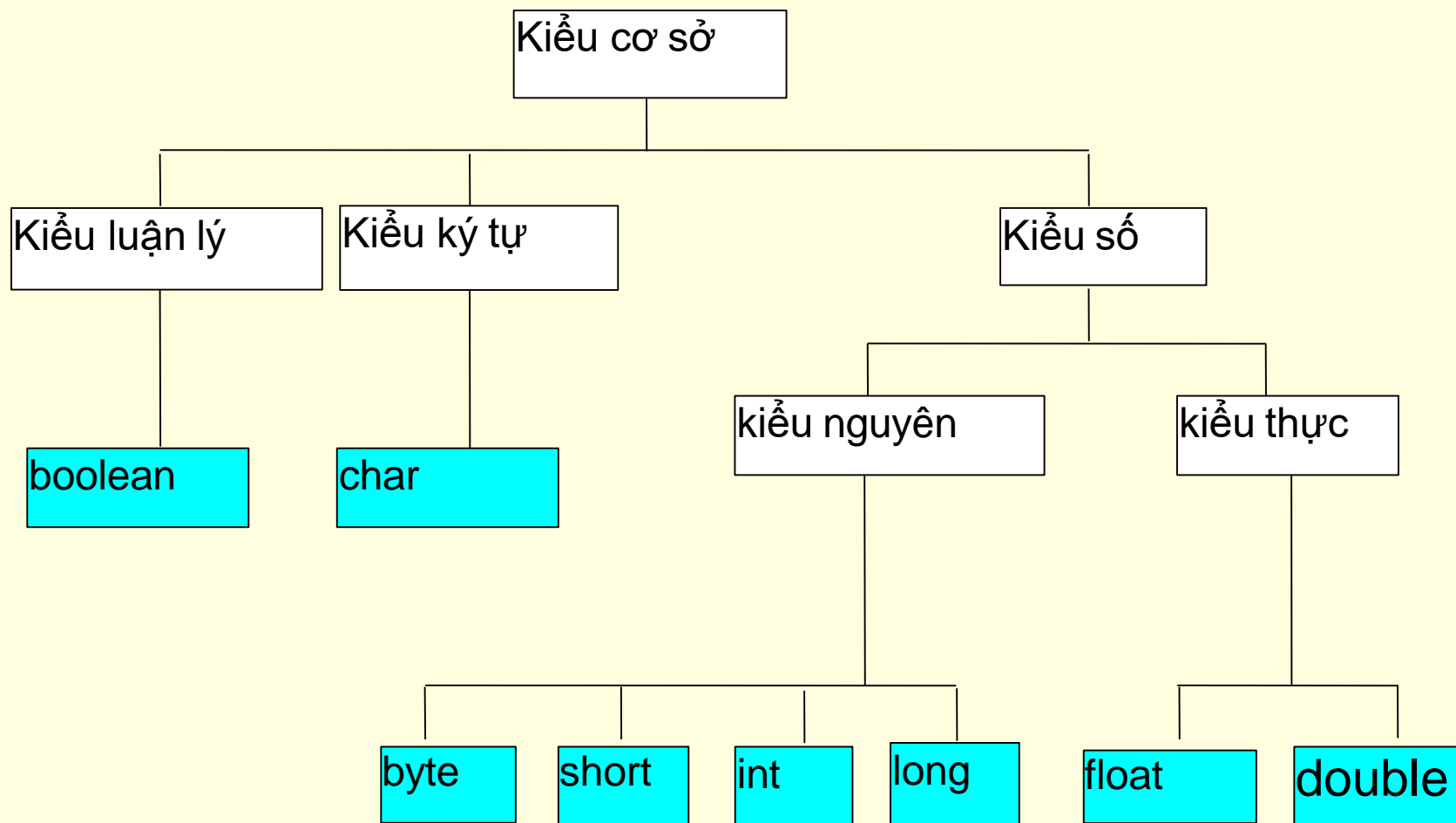
Hàng ký tự đặc biệt

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



Kiểu dữ liệu cơ sở (tt)

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	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
---------------	----	---	-------------

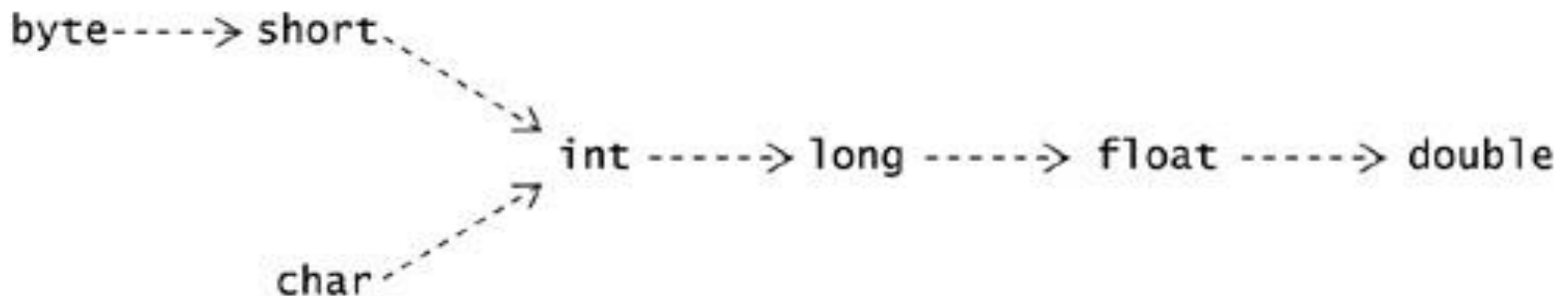
Kiểu dữ liệu cơ sở (tt)

• **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* $z = x + y$;

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

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

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'};

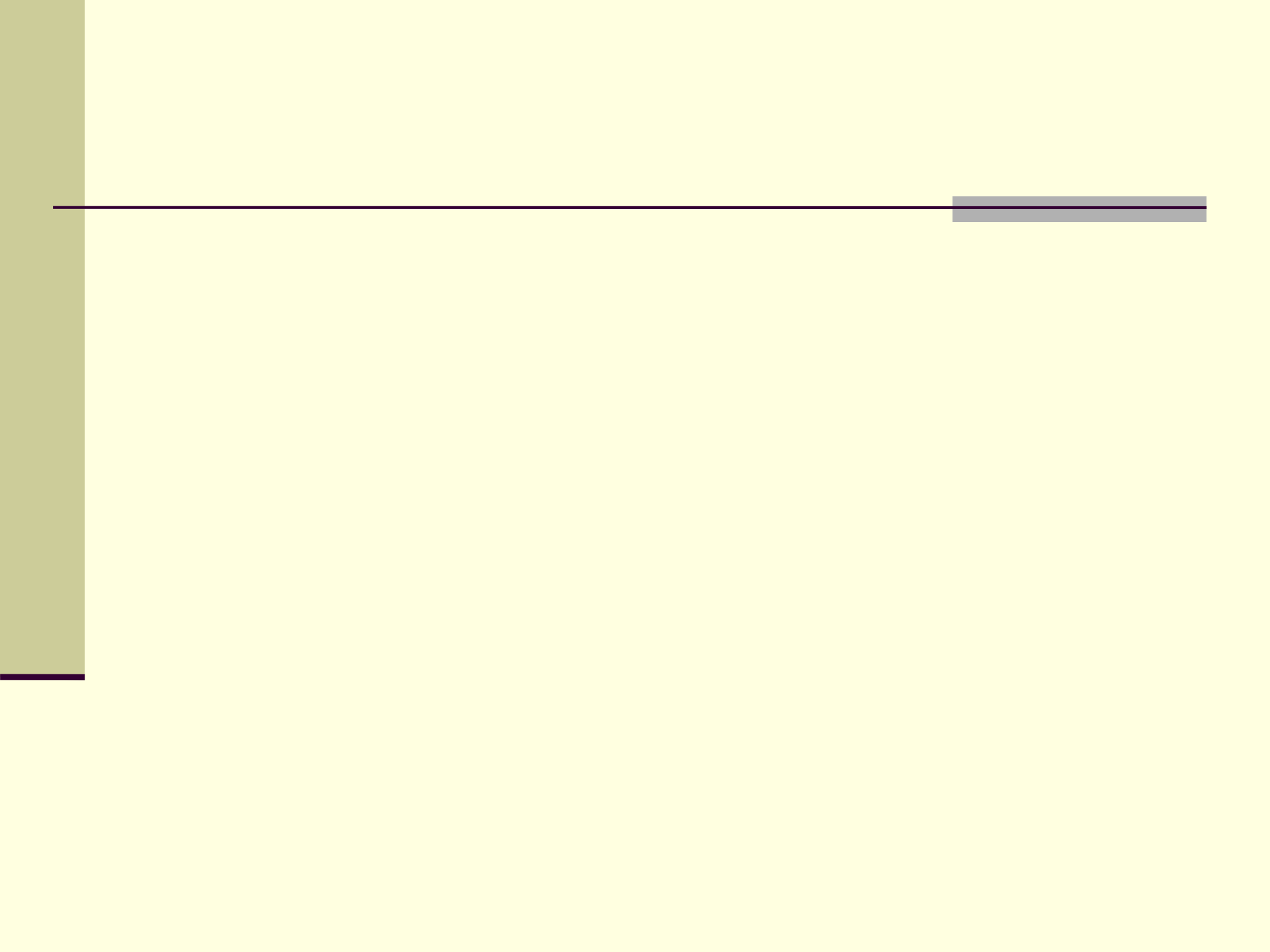
String *arrString[]* = {"ABC", "EFG", "GHI"};

- 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***



Kiểu dữ liệu tham chiếu (tt)

- Kiểu đối tượng

Khai báo đối tượng

<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>

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
==	So sánh bằng
!=	So sánh khác
>	So sánh lớn hơn
<	So sánh nhỏ hơn
>=	So sánh lớn hơn hay bằng
<=	So sánh nhỏ hơn hay bằng
	OR (biểu thức logic)
&&	AND (biểu thức logic)
!	NOT (biểu thức logic)

Toán tử, biểu thức (tt)

- Toán tử gán

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

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>) {*
<khởi_lệnh>;
}

- **Dạng 2:** *do {*
<khởi_lệnh>;
} while (điều_kiện);

- **Dạng 3:** *for (khởi_tạo_biến_đếm; đk_lặp; tăng_biến)*
{
<khởi_lệnh>;
}

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;
    }
}
```


NHẬP DỮ LIỆU TỪ CONSOLE

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](#)

Lớp bao kiểu dữ liệu

Data type	Wrapper Class (java.lang.*)	Ghi chú
boolean	Boolean	- Gói (package): chứa
byte	Byte	
short	Short	
char	Character	
int	Integer	
long	Long	
float	Float	
double	Double	