

# Java cơ bản

# Mục tiêu môn học

- Học cách sử dụng ngôn ngữ lập trình java
- Làm quen với lập trình hướng đối tượng trên java

# Nội dung

1. Giới thiệu về ngôn ngữ lập trình java
2. Java platform
3. Quá trình biên dịch và thực thi
4. Công cụ để lập trình với java
5. Quy tắc đặt tên trong java
6. Java identifiers
7. Chương trình java đơn giản
8. Toán tử

# 1. Giới thiệu về ngôn ngữ lập trình java

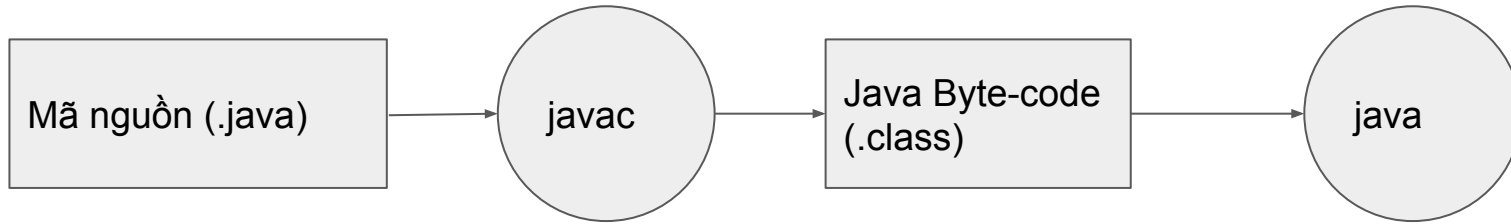
- Java là một ngôn ngữ lập trình được phát triển bởi Sun Microsystems (Oracle).
- Java là một ngôn ngữ lập trình khá trẻ. Phiên bản đầu tiên được phát hành năm 1995.
- Tiêu chí phát triển của java là “Write Once, Run Anywhere”.
- Mục đích của java là sử dụng cho lập trình hướng đối tượng dựa theo lớp (class base).

## 2. Java platform

Các thành phần của java platform - nền tảng java

- Các API
- Java virtual machine (JVM)

### 3. Quá trình biên dịch và thực thi



## 4. Công cụ để lập trình với java

IDEs:

- NetBeans
- IntelliJ IDEA
- Eclipse

## 5. Quy tắc đặt tên trong java

Các cách đặt tên:

- PascalCase (Upper Camel Case)
- camelCase (Lower Camel Case)
- UPPERCASE\_WITH\_UNDERSCORE
- lowercase
- lowercase\_separated\_by\_underscores



## 5. Quy tắc đặt tên trong java (tiếp)

Packages	lowercase
Class, Interface	PascalCase
Method	camelCase
Variables	camelCase
Constants	UPPERCASE_WITH_UNDERSCORE

## 6. Java Identifiers

- Định danh trong java (java identifiers) được sử dụng để đặt tên cho các thành phần trong java như: tên biến, hằng, lớp, giao diện, phương thức.
- Định danh trong java:
  - Bắt đầu bằng ký tự A - Z, a - z, \_ (underscore), \$
  - Tiếp theo sau có thể là các ký tự A - Z, a - z, \_, \$, 0 - 9
  - Không được phép trùng với các từ khóa trong java
- Định danh trong java phân biệt chữ hoa và chữ thường.

## 6. Java Identifiers (tiếp)

Literals: null, true, false

Các từ khóa (keywords): abstract, continue, for, new, switch, assert, default, goto, package, synchronized, boolean, do, if, private, this, break, double, implements, protected, throw, byte, else, import, public, throws, case, enum, instanceof, return, transient, catch, extends, int, short, try, char, final, interface, static, void, class, finally, long, strictfp, volatile, const, float, native, super, while.

Từ dành riêng: goto, const

## 7. Chương trình java đơn giản

```
class Hello {  
  
    public static void main(String[] arguments){  
  
        // Program execution begins here  
  
        System.out.println("Hello world!");  
  
    }  
  
}
```

# Kiểu dữ liệu

Kiểu dữ liệu nguyên thủy (primitive data types)

- Số nguyên (integer)
- Số thực (float)
- Ký tự (char)
- Giá trị logic (boolean)

Kiểu dữ liệu tham chiếu (reference data types)

# Kiểu dữ liệu (tiếp)

	Kiểu dữ liệu	Kích thước (byte)	Giá trị mặc định
integer	byte	1	0
	short	2	0
	int	4	0
	long	8	0L
float	float	4	0.0f
	double	8	0.0d
boolean	boolean	NA	false
char	char	2	'\u0000'

# Kiểu dữ liệu (tiếp)

Biểu diễn các ký tự đặc biệt:

`\b` backspace

`\n` newline

`\r` return

`\t` tab

`\"` quotation mark

`\'` apostrophe

`\\` backslash

# Khai báo biến

Cú pháp:

```
Kiểu_dữ_liệu tên_biến;
```

```
Kiểu_dữ_liệu tên_biến = giá_trị_khởi_tạo;
```

Ví dụ:

```
int i, j;
```

```
float pi = 3.1413f;
```



# Chuyển đổi kiểu dữ liệu

- Java là ngôn ngữ định kiểu chặt (strict data types)
- Java ngầm định chuyển đổi từ kiểu dữ liệu hẹp hơn sang kiểu dữ liệu rộng hơn
- Để chuyển đổi sang kiểu dữ liệu hẹp hơn cần phải định kiểu rõ ràng.
  - Cú pháp: (kiểu\_dữ\_liệu) tên\_biến;

# Chú thích

Java hỗ trợ các kiểu chú thích sau:

// Chú thích trên một dòng

/\* Chú thích cho một đoạn

Dòng thứ 2 trong chú thích \*/

/\*\* Chú thích dạng javadoc \*/

# Toán tử

## Toán tử số học

- Cộng: +
- Trừ: -
- Nhân: \*
- Chia: /
- Lấy phần dư: %
- Tăng một đơn vị: ++
- Giảm một đơn vị: --

# Toán tử

Toán tử quan hệ:

- Bằng:  $==$
- Không bằng:  $!=$
- Lớn hơn:  $>$
- Nhỏ hơn:  $<$
- Lớn hơn hoặc bằng:  $>=$
- Nhỏ hơn hoặc bằng:  $<=$

# Toán tử

## Toán tử bit:

- AND: &
- OR: |
- XOR: ^
- Đảo bit: ~
- Dịch trái: <<
- Dịch phải: >>
- Dịch phải (điền số không vào đầu): >>>

# Toán tử

Toán tử logic:

- AND: &&
- OR: ||
- NOT: !

# Toán tử

## Toán tử gán

=	$A = B$	Gán giá trị A bằng giá trị B
+=	$A += B$	$A = A + B$
-=	$A -= B$	$A = A - B$
*=	$A *= B$	$A = A * B$
/=	$A /= B$	$A = A / B$
%=	$A \% = B$	$A = A \% B$

# Toán tử

## Toán tử gán (tiếp)

$\ll=$	$A \ll= B$	$A = A \ll B$
$\gg=$	$A \gg= B$	$A = A \gg B$
$\ggg=$	$A \ggg= B$	$A = A \ggg B$
$\&=$	$A \&= B$	$A = A \& B$
$ =$	$A  = B$	$A = A   B$
$\wedge=$	$A \wedge= B$	$A = A \wedge B$



# Toán tử

Toán tử 3 ngôi: ? :

- Cú pháp: (expression) ? true\_expression: false\_expression
- Ví dụ: (a > b) ? “a lớn hơn b”: “a nhỏ hơn hoặc bằng b”

Toán tử instanceof

- Cú pháp: (Object reference variable) instanceof (class/interface type)

# Toán tử

## Thứ tự ưu tiên của các toán tử

postfix	expr++ expr--
unary	++expr --expr +expr -expr ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=

# Toán tử

## Thứ tự ưu tiên của các toán tử

bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
logical AND	&&
logical OR	
ternary	? :
assignment	= += -= *= /= %= &= ^=  = <<= >>= >>>=