

# LẬP TRÌNH JAVA

## Bài 2: Biến và các kiểu dữ liệu trong Java

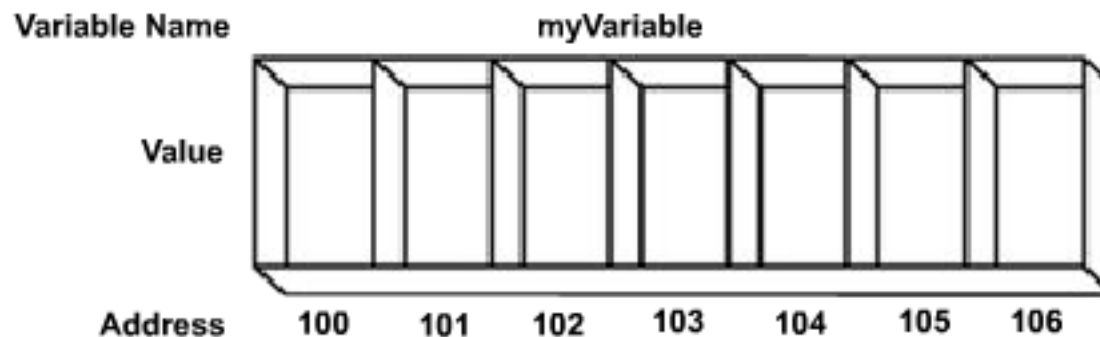
# NỘI DUNG

- ▣ Biến trong Java
- ▣ Kiểu dữ liệu
- ▣ Định dạng I/O
- ▣ Toán tử
- ▣ Ép kiểu

# Biến trong Java

# Biến là gì?

- ❑ Biến là địa chỉ của ô nhớ trong máy tính, địa chỉ này xác định vị trí mà một giá trị nào đó được lưu trong bộ nhớ (RAM)
- ❑ Biến được sử dụng nhiều trong Java và các ngôn ngữ lập trình khác. Nó là đơn vị cơ bản để lưu trữ trong Lập trình
- ❑ Trong Java, chúng ta cần khai báo biến trước khi sử dụng.



# Biến là gì?

## ▣ Cú pháp

```
datatype variableName;
```

\* **datatype**: kiểu dữ liệu

\* **variableName**: tên biến

## ▣ Ví dụ

```
int age;
```

```
char gender;
```

# Cách đặt tên Biến

- ▣ Tên biến cần có ý nghĩa và dễ nhớ
- ▣ Tên biến trong Java phải tuân theo một số nguyên tắc, nếu vi phạm sẽ có lỗi cú pháp được ném ra.

# Cách đặt tên Biến

## Nguyên tắc

- ❑ Tên biến chỉ chứa ký tự, số, dấu gạch dưới \_ và dấu đô la \$
- ❑ Tên biến phải được bắt đầu bằng ký tự, dấu \_ hoặc dấu \$
- ❑ Tên biến không được trùng với keyword của Java
- ❑ Tên biến trong Java có phân biệt hoa thường (case-sensitive)
- ❑ Tên biến được viết thường nếu chỉ có 1 từ, nếu 2 từ trở lên, chữ cái đầu tiên của từ thứ 2 trở đi sẽ được viết hoa

# Cách đặt tên Biến

Tên biến	Hợp lệ / Không hợp lệ
roolNumber	OK
a2x5_w7t3	OK
\$yearly_salary	OK
_2010_tax	OK
double	Error
amount#Balance	Error
4short	Error



# Kiểu dữ liệu

# Kiểu dữ liệu

**Kiểu dữ liệu trong Java được chia thành 2 nhóm**

- ▣ Kiểu dữ liệu nguyên thủy - primitive data types**
- ▣ Kiểu dữ liệu tham chiếu – reference data type**

# Dữ liệu nguyên thủy

Data Type	Description	Range
<b>byte</b>	8-bit signed integer	-128 to 127
<b>short</b>	16-bit signed integer	-32768 to 32767
<b>long</b>	64-bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<b>int</b>	32-bit signed integer	-2,147,483,648 to 2,147,483,647
<b>float</b>	32-bit floating-point variable	-3.40292347E+38 to +3.40292347E+38
<b>boolean</b>	Stores a true or false value	true or false
<b>char</b>	16-bit Unicode character	0 to 65535
<b>double</b>	64-bit floating-point variable	-1.79769313486231570E+308 to 1.79769313486231570E+308

# Dữ liệu tham chiếu

Data Type	Description
<b>Array</b>	A collection of several items of the same data type such as names of students.
<b>Class</b>	A collection of variables and methods. For example, class Student containing the complete details of the students and the methods that operate on the details.
<b>Interface</b>	A type of class in Java used to implement multiple inheritance.

# Định dạng I/O

# Định dạng I/O

- ▣ Khi hiển thị một nội dung ra màn hình trong Java, cần phải định dạng kiểu cho nó.
- ▣ Sử dụng phương thức **printf()**

# Định dạng I/O

Định dạng	Mô tả
%d	định dạng số nguyên
%f	định dạng số thực
%o	định dạng số hệ bát phân
%e	định dạng kiểu số khoa học
%n	định dạng 1 dòng mới

# Scanner Class

- ▣ Scanner là lớp cho phép nhập dữ liệu vào chương trình từ bàn phím
- ▣ Hỗ trợ nhập nhiều các định dạng khác nhau: số nguyên, số thực, ký tự , chuỗi,...



# Scanner Class

## ▣ Khởi tạo đối tượng Scanner

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

# Scanner Class

## ▣ Một số phương thức cơ bản

Phương thức	Mô tả
<code>nextInt()</code>	Trả về giá trị kiểu số nguyên
<code>nextByte()</code>	Trả về giá trị kiểu số byte
<code>nextLong()</code>	Trả về giá trị kiểu long
<code>nextFloat()</code>	Trả về giá trị kiểu float
<code>nextDouble()</code>	Trả về giá trị kiểu double

# Escape Sequences

Ký tự	Mô tả
\n	Newline - 1 dòng mới
\b	Backspace - lùi 1 khoảng trắng
\t	Tab - tương đương 1 tab
\\	Ký tự \
\"	Ký tự “

# Toán tử

# Toán tử

- ▣ Ngôn ngữ lập trình nào cũng phải có toán tử
- ▣ Nó giúp ta gán giá trị cho một biến, hay đơn giản hơn là thực hiện các biểu thức toán học, biểu thức logic,...
- ▣ Ví dụ

$$Y = A + B$$

# Toán tử gán

- ▣ Trong Java, chúng ta sử dụng dấu '=' để thực hiện phép gán
- ▣ Ví dụ

**X = 5**

# Toán tử gán

- ▣ Hỗ trợ Toán tử '+=', '\*=', '\_=', '/=', '%='

$X += 5$

- ▣ Tương đương

$X = X + 5$

# Toán tử Toán học

Toán tử	Mô tả
+	Thể hiện phép toán cộng
-	Thể hiện phép toán trừ
*	Thể hiện phép toán nhân
/	Thể hiện phép toán chia
%	Thể hiện phép toán chia lấy dư



# Toán tử Một ngôi

Toán tử	Mô tả
+	Lấy giá trị dương
-	Lấy giá trị âm
++	Tăng giá trị thêm 1
--	Giảm giá trị 1
!	Lấy giá trị phủ định

# Toán tử Quan hệ

Toán tử	Mô tả
$==$	So sánh bằng
$!=$	So sánh khác
$>$	So sánh lớn hơn
$<$	So sánh nhỏ hơn
$>=$	So sánh lớn hơn bằng
$<=$	So sánh nhỏ hơn bằng

# Toán tử Điều kiện

Toán tử	Mô tả
&&	AND
	OR

# Toán tử Ba ngôi

- ▣ Cú pháp

`expression1 ? expression2 : expression3`

- ▣ `expression1`: biểu thức điều kiện

- ▣ `expression2`: thực hiện khi biểu thức 1 đúng

- ▣ `expression3`: thực hiện khi biểu thức 1 sai

# Toán tử Dịch bit và so sánh bit

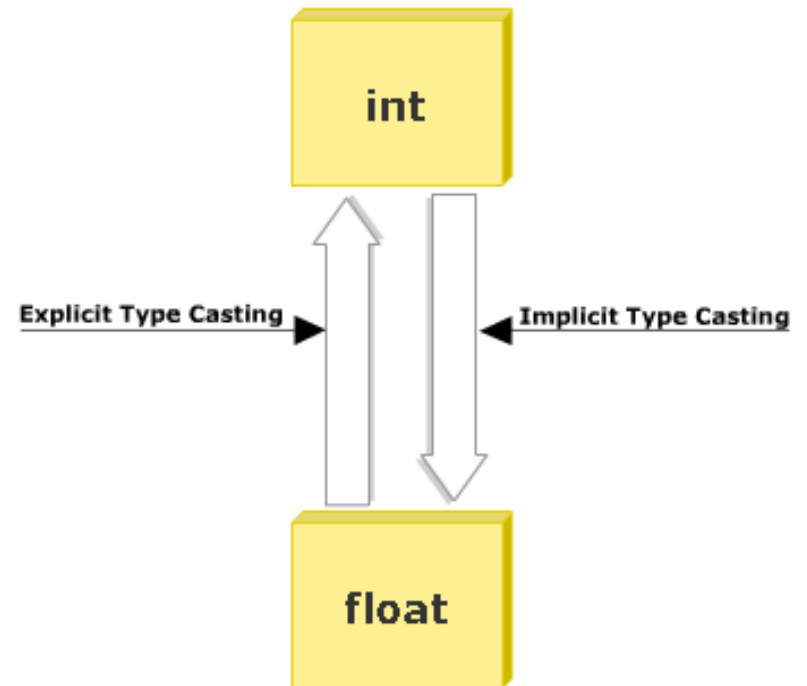
Toán tử	Mô tả
&	toán tử and
	toán tử or
^	toán tử exclusive or
~	toán tử invert
>>	toán tử dịch bit phải
<<	Toán tử dịch bit trái

**Ép kiểu**

# Ép kiểu

Java có 2 cách ép kiểu (chuyển đổi kiểu) dữ liệu

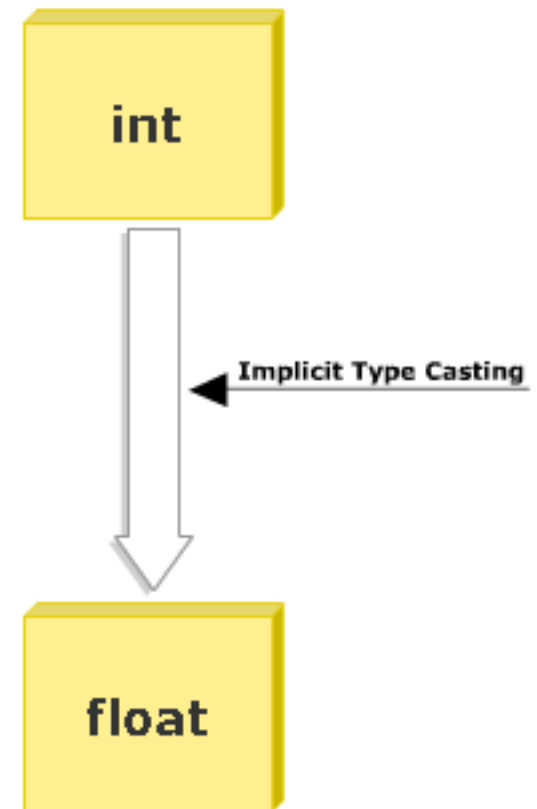
- ▣ implicit type casting
- ▣ explicit type casting



# Ép kiểu ngầm định

- ▣ được thực hiện tự động bởi JVM
- ▣ được thực hiện khi chuyển từ kiểu dữ liệu nhỏ sang kiểu dữ liệu lớn hơn

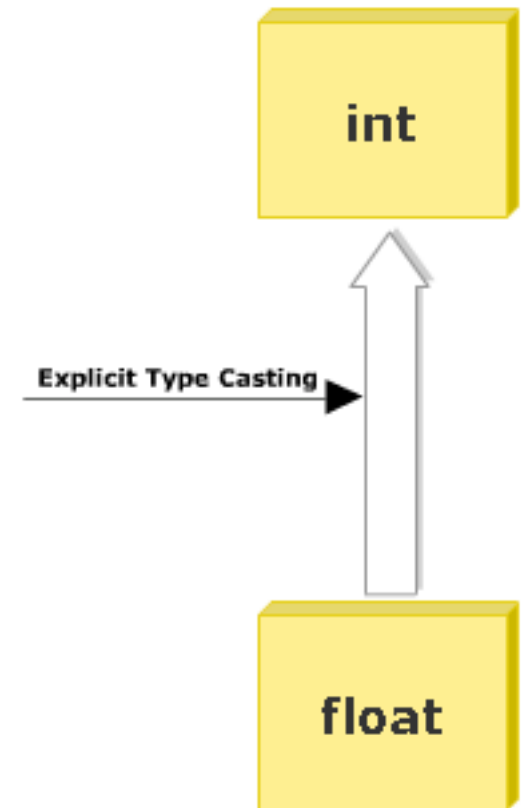
**byte > short > int > long > float > double**





# Ép kiểu trực tiếp

- ▣ được thực hiện khi được chỉ định
- ▣ thường ép kiểu từ kiểu dữ liệu lớn hơn về dữ liệu nhỏ hơn
- ▣ có thể sẽ xảy ra mất mát dữ liệu



# Ví dụ ép kiểu

```
long p = (long) 12345.56; // p == 12345
```

```
int g = p; // không hợp lệ dù kiểu int
```

```
        //có thể lưu giá trị 12345
```

```
char c = 't';
```

```
int j = c; // tự động chuyển đổi
```

```
short k = c; // không hợp lệ
```

```
short k = (short) c; // ép kiểu trực tiếp float
```

```
f = 12.35; // không hợp lệ
```