



# LẬP TRÌNH JAVA

## CHƯƠNG 2: NỀN TẢNG CỦA JAVA

- Chương trình đơn giản viết bằng Java
- Các kiểu dữ liệu
- Các toán tử
- Định dạng xuất dữ liệu
- Các cấu trúc lập trình cơ bản



# Ví dụ một chương trình Java mẫu

// This is a simple program called “Ex1.java”

```
class Ex1
{
    public static void main(String args[])
    {
        System.out.println(“My first program in Java”);
    }
}
```



# Truyền đối số trong dòng lệnh

```
class Pass
{
    public static void main(String parameters[])
    {
        System.out.println("This is what the main method
        received");
        System.out.println(parameters[0]);
        System.out.println(parameters[1]);
        System.out.println(parameters[2]);
    }
}
```

# Truyền đối số trong dòng lệnh (tt)

MS-DOS C:\WINNT\System32\cmd.exe

```
D:\TEMP>java Pass We Change Lives
This is what the main method received
We
Change
Lives

D:\TEMP>_
```

# Các phần tử cơ bản của Java

---

- ✓ Kiểu dữ liệu
- ✓ Lớp và phương thức (Classes & Methods)
- ✓ Biến số
- ✓ Toán tử
- ✓ Cấu trúc điều khiển



- Kiểu dữ liệu cơ sở (Primitive Data Types)
- Kiểu dữ liệu tham chiếu (Reference data types)



# Kiểu dữ liệu cơ sở

Kiểu dữ liệu	Độ dài theo số bit	Phạm vi biểu diễn giá trị
byte	8	-128 đến 127
char	16	'\u0000' to '\uffff'
boolean	1	"True" hoặc "False"
short	16	-32768 đến 32767
int	32	-2,147,483,648 đến +2,147,483,648
long	64	-9,223,372,036,854,775,808 đến +9,223,372,036,854,775,808
float	32	-3.40292347E+38 đến +3.40292347E+38
double	64	-1,79769313486231570E+308 đến +1,79769313486231570E+308



# Ép kiểu

- ✓ Kiểu dữ liệu này được chuyển đổi sang một kiểu dữ liệu khác
- ✓ Ví dụ

```
float c = 34.89675f;
```

```
int b = (int)c + 10;
```



- ✓ Khai báo biến số gồm 3 thành phần:
  - Kiểu dữ liệu của biến số
  - Tên biến
  - Giá trị ban đầu của biến (không bắt buộc)

- ✓ Cú pháp

**datatype identifier [=value][,  
identifier[=value]...];**



# Những từ khóa của Java

abstract	boolean	break	byte
case	catch	char	class
const	continue	default	do
double	else	extends	final
finally	float	for	goto
if	implements	import	instanceof
int	interface	long	native
new	package	private	protected
public	return	short	static
super	switch	synchronized	this
throw	throws	transient	try
void	volatile	while	

# Kiểu dữ liệu tham chiếu

---

- ✓ Mảng (Array)
- ✓ Lớp (Class)
- ✓ Interface



# Khai báo mảng

- ✓ Ba cách để khai báo mảng:
  - **`datatype identifier [ ];`**
  - **`datatype identifier [ ] = new datatype[size];`**
  - **`datatype identifier [ ] = {value1,value2,...valueN};`**



# Lớp trong Java

## ✓ Cú pháp khai báo lớp (Class)

```
class Classname  
{  
    var_datatype variablename;  
    :  
    met_datatype methodname(parameter_list)  
    :  
}
```



# Phương thức

- ✓ Phương thức được định nghĩa như là một hành động hoặc một tác vụ thật sự của đối tượng
- ✓ Cú pháp

```
access_specifier modifier datatype  
    method_name(parameter_list)  
{  
    //body of method  
}
```



# Quyền truy xuất

---

- ✓ public
- ✓ private
- ✓ protected





# Từ khóa bổ sung phương thức

---

- ✓ static
- ✓ abstract
- ✓ Final
- ✓ synchronized
- ✓ volatile



# Ví dụ về sử dụng phương thức

```
class Temp {  
    static int x = 10;           // variable  
    public static void show( ) { // method  
        System.out.println(x);  
    }  
    public static void main(String args[ ]) {  
        Temp t = new Temp( );    // object 1  
        t.show( );               // method call  
        Temp t1 = new Temp( );   // object 2  
        t1.x = 20;  
        t1.show( );  
    }  
}
```

# Phương thức khởi tạo

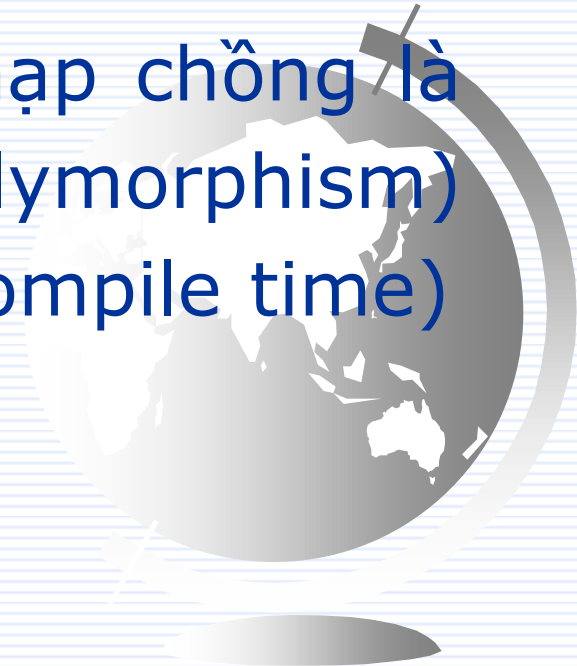
- ✓ Là một phương thức đặc biệt dùng để khởi tạo giá trị cho các biến thành viên của lớp đối tượng
- ✓ Có cùng tên với tên lớp và không có giá trị trả về
- ✓ Được gọi khi đối tượng được tạo ra
- ✓ Có 2 loại:
  - Tường minh (Explicit constructors)
  - Ngầm định (Implicit constructors)



# Những phương thức được nạp chồng

---

- ✓ Những phương thức được nạp chồng :
  - Cùng ở trong một lớp
  - Có cùng tên
  - Khác nhau về danh sách tham số
- ✓ Những phương thức được nạp chồng là một hình thức đa hình (polymorphism) trong quá trình biên dịch (compile time)



# Ghi đè phương thức

- ✓ Những phương thức được ghi đè:
  - Có mặt trong lớp cha (superclass) cũng như lớp kế thừa (subclass)
  - Được định nghĩa lại trong lớp kế thừa (subclass)
- ✓ Những phương thức được ghi đè là một hình thức đa hình (polymorphism) trong quá trình thực thi (Runtime)



# Phương thức khởi tạo của lớp dẫn xuất

---

- ✓ Có cùng tên với lớp dẫn xuất (subclass)
- ✓ Mệnh đề gọi constructor của lớp cha (superclass) phải là mệnh đề đầu tiên trong constructor của lớp dẫn xuất (subclass)
  - super();



## ✓ Các loại toán tử:

- Toán tử số học (Arithmetic operators)
- Toán tử dạng Bit (Bitwise operators)
- Toán tử so sánh (Relational operators)
- Toán tử logic (Logical operators)
- Toán tử điều kiện (Conditional operator)
- Toán tử gán (Assignment operator)



# Toán tử số học

+	Addition (Phép cộng)
-	Subtraction (Phép trừ)
*	Multiplication (Phép nhân)
/	Division (Phép chia)
%	Modulus (Lấy số dư)
++	Increment (Tăng dần)
--	Decrement (Giảm dần)





# Toán tử số học

$+=$

Phép cộng và gán

$-=$

Phép trừ và gán

$*=$

Phép nhân và gán

$/=$

Phép chia và gán

$\%=$

Phép lấy số dư và gán



# Toán tử Bit

~	Phủ định (NOT)
&	Và (AND)
	Hoặc (OR)
^	Exclusive OR
>>	Dịch sang phải (Shift right)
<<	Dịch sang trái (Shift left)



# Toán tử so sánh

$==$

So sánh bằng

$!=$

So sánh khác

$<$

Nhỏ hơn

$>$

Lớn hơn

$<=$

Nhỏ hơn hoặc bằng

$>=$

Lớn hơn hoặc bằng



# Toán tử Logic

---

&&

Logical AND

||

Logical OR

!

Logical NOT



# Toán tử điều kiện

✓ Cú pháp

**Biểu thức 1 ? Biểu thức 2 : Biểu thức 3;**

✓ **Biểu thức 1**

Điều kiện kiểu Boolean trả về giá trị True hoặc False

✓ **Biểu thức 2**

Trả về giá trị nếu kết quả của mệnh đề 1 là True

✓ **Biểu thức 3**

Trả về giá trị nếu kết quả của mệnh đề 1 là False



# Toán tử gán

## ✓ Phép gán

Giá trị có thể được gán cho nhiều biến số

## ✓ Ví dụ

**$a = b = c = d = 90;$**



## Thứ tự ưu tiên của các toán tử

Thứ tự	Toán tử
1.	trong ngoặc tính trước
2.	Các toán tử đơn như $+$ , $-$ , $++$ , $--$
3.	Các toán tử số học và các toán tử dịch như $*$ , $/$ , $+$ , $-$ , $<<$ , $>>$
4.	Các toán tử quan hệ như $>$ , $<$ , $>=$ , $<=$ , $=$ , $!=$
5.	Các toán tử logic và Bit như $\&\&$ , $\ \ $ , $\&$ , $\ $ , $\wedge$
5.	Các toán tử gán như $=$ , $*=$ , $/=$ , $+=$ , $-=$

# Các kí tự định dạng xuất dữ liệu

Escape Sequence	Mô tả
\n	Xuống dòng mới
\r	Chuyển con trỏ đến đầu dòng hiện hành
\t	Chuyển con trỏ đến vị trí dừng Tab kế tiếp (ký tự Tab)
\\	In dấu \
\'	In dấu nháy đơn (')
\''	In dấu nháy kép (")



# Các lệnh điều khiển

## ✓ Điều khiển rẽ nhánh:

- Mệnh đề **if-else**
- Mệnh đề **switch-case**

## ✓ Vòng lặp (Loops):

- Vòng lặp **while**
- Vòng lặp **do-while**
- Vòng lặp **for**



# Lệnh if-else

## ✓ Cú pháp

```
if (condition)  
{  
action1 statements;  
}  
else  
{  
action2 statements;  
}
```



# Lệnh switch-case

✓ Cú pháp

**switch (expression)**

**{**

**case 'value1': action1 statement(s);  
break;**

**case 'value2': action2 statement(s);  
break;**

**:**

**:**

**case 'valueN': actionN statement(s);  
break;**

**default: default\_action statement(s);**

**}**



# Lệnh lặp while

✓ Cú pháp

```
while(condition)
```

```
{
```

```
    action statements;
```

```
    :
```

```
    :
```

```
}
```



# Lệnh lặp do-while

✓ Cú pháp

**do**

**{**

**action statements;**

**:**

**:**

**} while(condition);**



# Vòng lặp for

## ✓ Cú pháp

**for(initialization statements; condition;  
increment statements)**

**{**

**action statements;**

**:**

**:**

**}**



# Xử lý dữ liệu nhập

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

- `InputStreamReader io = new  
InputStreamReader(System.in);`
  - `BufferedReader console = new BufferedReader(io);`
  - try{**
    - `System.out.print("Nhap chuoi str = ");`
    - `String str = console.readLine();`
    - `System.out.print(str);`
  - }**
  - catch(Exception ex){}**
- ```
}
```



# Bài tập

- ✓ 1. Viết một chương trình hiển thị tổng các bội số của 7 nằm giữa 1 và 100.
- ✓ 2. Viết chương trình để cộng n số hạng của dãy sau:  $1!+2!+3!...$
- ✓ 3. Viết chương trình nhập vào chuỗi, xuất ra các từ được tách từ chuỗi.
- ✓ 4. Viết chương trình nhập danh sách sinh viên (MSSV, HoTen, NgaySinh), sắp xếp sinh viên theo thứ tự tăng dần HoTen

