



TRƯỜNG ĐẠI HỌC  
**GIAO THÔNG VẬN TẢI**  
University of Transport and Communications

## HỌC PHẦN: CÔNG NGHỆ JAVA

Giảng viên: Đào Thị Lệ Thủy  
Khoa Công nghệ Thông tin



# NỘI DUNG HỌC PHẦN

**Chương 1. Tổng quan về ngôn ngữ Java**

**Chương 2. Nền tảng của ngôn ngữ Java**

**Chương 3. OOP – Lớp và đối tượng trong Java**

**Chương 4. Xử lý lỗi trong ngôn ngữ lập trình bậc cao**

**Chương 5. Làm việc với Collection trong Java**

**Chương 6. Giao diện trong Java với AWT - SWING**

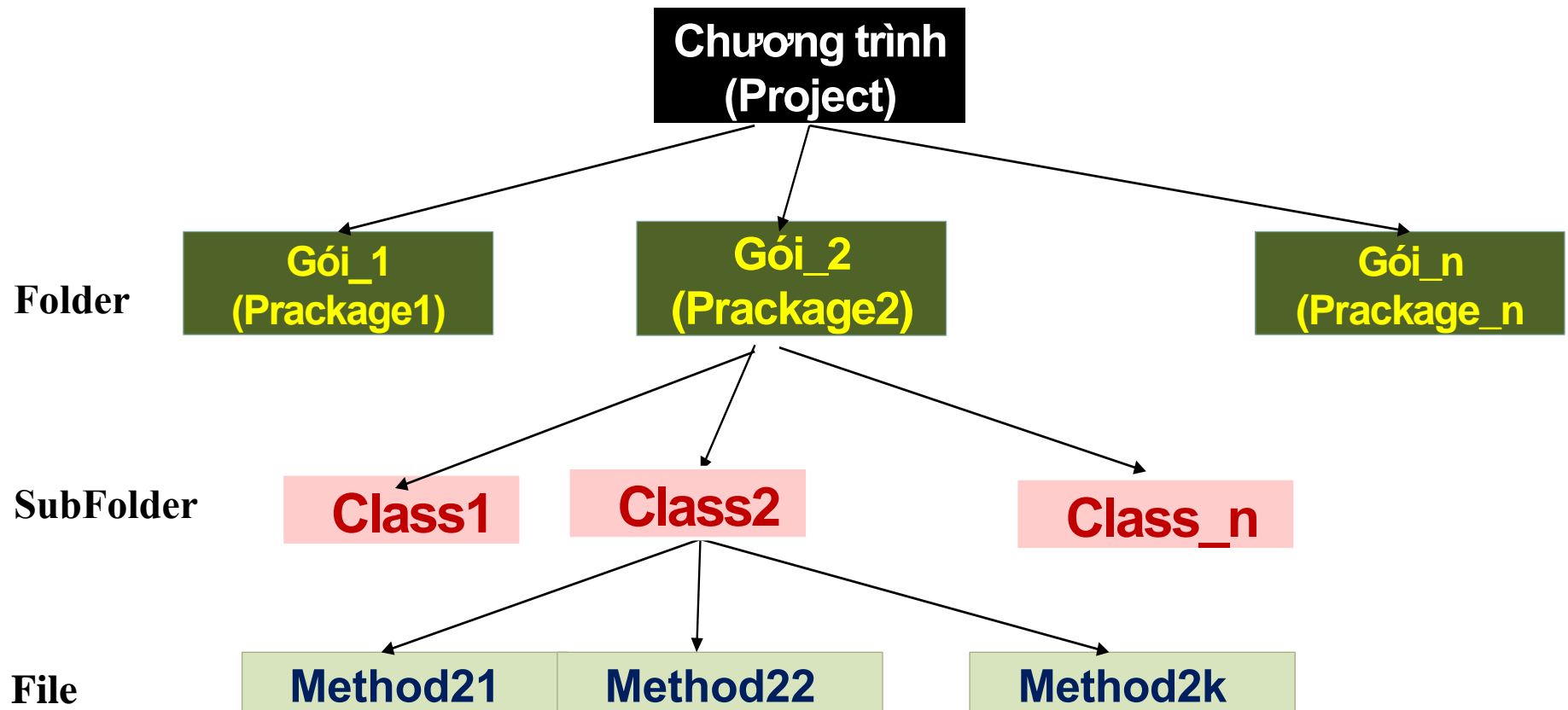
**Chương 7. Vào/ra dữ liệu trong Java**

**Chương 8. Làm việc với cơ sở dữ liệu**

# NỘI DUNG

- **Cấu trúc một chương trình java**
- **Kiểu dữ liệu**
- **Các toán tử trong ngôn ngữ Java.**
- **Biến, hằng**
- **Nhập, xuất dữ liệu**
- **Các cấu trúc điều khiển**
- **Mảng**
- **Chuỗi**

# Cấu trúc chương trình Java



# Cấu trúc chương trình Java

## CẤU TRÚC CƠ BẢN CỦA MỘT FILE MÃ NGUỒN JAVA:

[**Package** packageName; ]      //Khai báo tên gói, tùy chọn  
[**import** java.awt.\*;]      //Khai báo tên thư viện sẵn có, nếu cần dùng

*//Khai báo và định nghĩa các lớp dạng:*

[**Bổ nghĩa**] **class** ClassName [**extends** AnyClass] [**implements** InterfaceList] {  
    */\*Ghi chú, nếu cần\*/*      }  
    *//Khai báo biến*

*//Khai báo hoặc định nghĩa các hàm, phương thức*

*//Khai báo các giao diện dạng:*

[**Bổ nghĩa**] **interface** InterfaceName [**extends** AnyInterface] {  
    *//Định nghĩa các đại lượng hằng*  
    *//Chỉ khai báo các hàm (prototype): chỉ có định danh mà không có thân hàm*  
}

# Cấu trúc chương trình Java

Ví dụ 1: HelloWorld.java

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello You!");
    } //end method
} //end class
```

Hello You!

# CHÚ THÍCH

- **Chú thích (comment)**

- ✓ Là những dòng văn bản do người lập trình viết ra để chú giải cho những gì mình viết trong chương trình.
- ✓ Những dòng chú thích không được biên dịch trong quá trình thực thi chương trình.
- ✓ Có thể phục vụ trong việc sửa lỗi (debug) chương trình.
- ✓ Một dòng chú thích bắt đầu bằng dấu //; nhiều dòng bắt đầu bằng /\* và kết thúc bằng \*/.

```
// Lệnh hiển thị dòng Hello World! ra màn hình
```

```
System.out.println("Hello World!");
```

```
/* Lệnh hiển thị  
dòng Hello World!  
ra màn hình */
```

```
System.out.println("Hello World!");
```

# ĐỊNH DANH

Định danh là tên được đặt trong chương trình như tên biến, tên hằng cần lưu ý:

- Phân biệt hoa thường. Có thể chứa ký tự, số, \_, \$
- Không bắt đầu bằng số và không được đặt tên trùng từ khóa
- Không chứa khoảng trắng, ký hiệu toán học
- Khuyến nghị

- ✓ Đặt tên có nghĩa
- ✓ Bắt đầu bằng chữ thường.

Ví dụ: `dateOfBirth`, `address`, `description`, `placeOfBirth`

- ✓ Biến tạm trong phạm vi hẹp có thể đặt `a`, `i`, `j`....
- ✓ Tránh kết hợp nhiều ngôn ngữ (Tiếng Anh + Việt...)



# CÁC KIỂU DỮ LIỆU

- **Kiểu dữ liệu cơ bản:** `byte`, `float`, `short`, `double`, `int`, `char`, `long`, `boolean`
- **Kiểu dữ liệu đối tượng:**

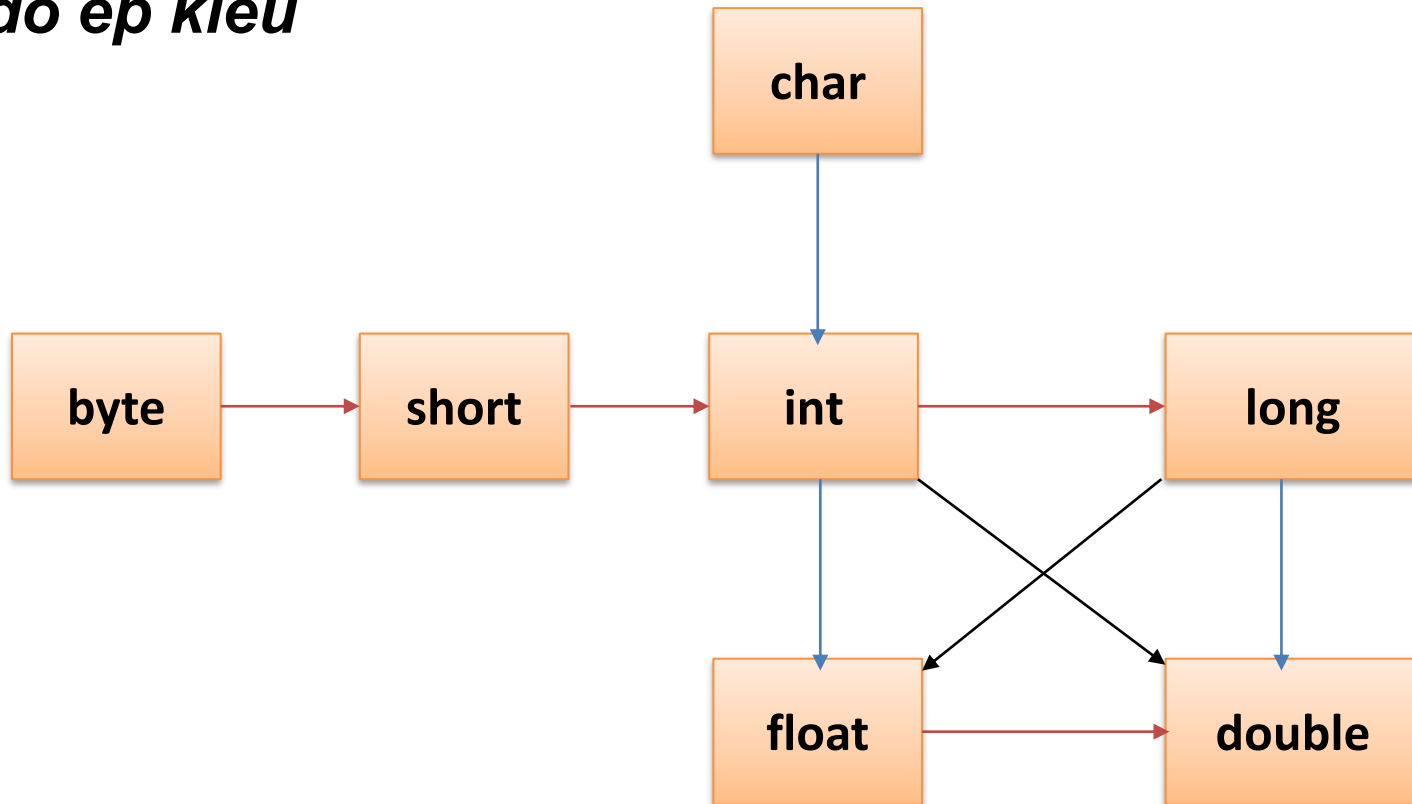
<b>array</b>	Một mảng của các dữ liệu cùng kiểu.
<b>class</b>	Dữ liệu kiểu lớp đối tượng do người dùng định nghĩa. Chứa tập các thuộc tính và phương thức..
<b>interface</b>	Dữ liệu kiểu lớp giao tiếp do người dùng định nghĩa. Chứa các phương thức của giao tiếp.

# CÁC KIỂU DỮ LIỆU

Kiểu	Kích thước	Trị mặc định	Khoảng giá trị	Ví dụ
byte	1 byte	0	[-128, 127]	byte x = 56;
short	2 byte	0	[-32768..32767]	short x = 1000;
int	4 byte	0	[-2,147,483,648, 2,147,483,647]	int x = 150000;
long	8 byte	0	[-9,223,372,036,854,775,808 ( $2^{63}$ ), 9,223,372,036,854,775,807 ( $2^{63} - 1$ )]	long x = 653535535632l;
float	4 byte	0.0	$1.4E^{-45}$ - $3.4028235E^{38}$	float x = 5.421f
double	8 byte	0.0	$4.9E^{-34}$ - $1.7976931348623157E^{308}$	double x = 5.421d
char	2 byte		0 đến 65,535	char c = 'b';
boolean	1 bit	false	True (1), false (0)	boolean b = true;

# CÁC KIỂU DỮ LIỆU

## □ Sơ đồ ép kiểu



- Dữ liệu có thể ép kiểu theo nguyên tắc từ kiểu nhỏ sang kiểu lớn hơn

# BIỂU THỨC VÀ TOÁN TỬ

## ○ Biểu thức:

- + Biểu thức là sự kết hợp giữa các toán tử, toán hạng
- + Toán hạng có thể là hằng số, các biến, lời gọi các hàm.
- + Kết quả của biểu thức cho ta một giá trị.

## ○ Ví dụ:

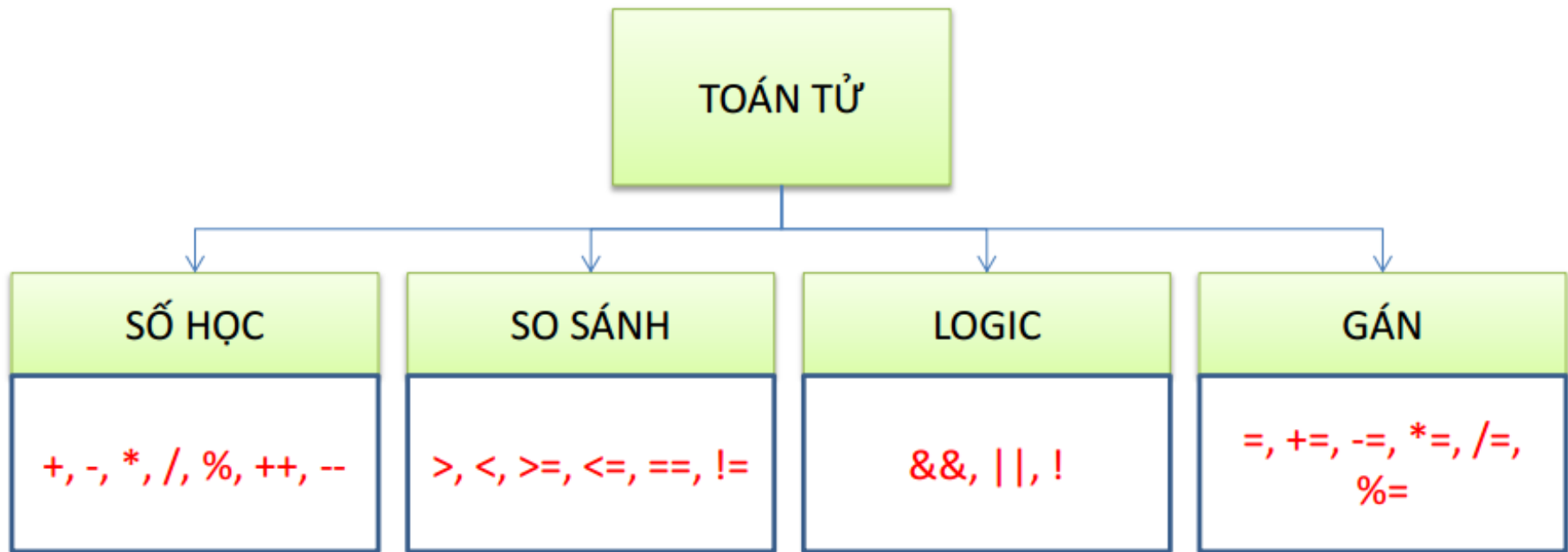
### ✓ Biểu thức: $10 + 6 * 5$

- 10, 6, 5 là các toán hạng
- + và \* là các toán tử
- Giá trị của biểu thức là 40

### ✓ Biểu thức: $3 > (7 \% 2)$

- 3, 7, 2 là các toán hạng
- >, % là các toán tử
- Giá trị của biểu thức là true

# BIỂU THỨC VÀ TOÁN TỬ



# BIỂU THỨC VÀ TOÁN TỬ

Toán tử số học: là các phép toán thao tác trên các số nguyên và số thực

Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Tính tổng của 2 số	$5 + 4$	9
-	Tính hiệu của 2 số	$5 - 4$	1
*	Tính tích của 2 số	$5 * 4$	20
/	Tính thương của 2 số	$5 / 4$	1.25
%	Chia có dư của 2 số	$5 \% 4$	1
++	Tăng giá trị của biến lên 1 đơn vị	$5++$	6
--	Giảm giá trị của biến xuống 1 đơn vị	$5--$	4

Cho biết kết quả của các biểu thức sau:

$$\diamond 4++ + 2++ = 6$$

$$\diamond ++4 - 2++ = 3$$

$$\diamond ++4 + ++2 = 8$$

$$\diamond --4 + ++2 = 6$$

# BIỂU THỨC VÀ TOÁN TỬ

Toán tử so sánh: là các phép toán so sánh 2 toán hạng

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<b>==</b>	So sánh bằng	$5 == 4$	false
<b>&gt;</b>	So sánh lớn hơn	$5 > 4$	true
<b>&gt;=</b>	So sánh lớn hơn hoặc bằng	$5 >= 4$	true
<b>&lt;</b>	So sánh nhỏ hơn	$5 < 4$	false
<b>&lt;=</b>	So sánh nhỏ hơn hoặc bằng	$5 <= 4$	false
<b>!=</b>	So sánh khác	$5 != 4$	true

Cho biết kết quả của các biểu thức sau:

- ❖  $5/2 > 3 * 2$
- ❖  $7 != 2+2$
- ❖  $4 * 6 == 25$
- ❖  $5-6 <= 3$
- ❖  $4 < 2*3$

# BIỂU THỨC VÀ TOÁN TỬ

## Toán tử logic: các phép toán thao tác trên các toán hạng logic

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<b>&amp;&amp;</b>	Trả về giá trị true khi tất cả biểu thức tham gia biểu thức có giá trị true	$5 > 4 \ \&\& \ 6 \neq 7$ $5 < 4 \ \&\& \ 6 \neq 7$	?
<b>  </b>	Trả về giá trị true khi có 1 biểu thức tham gia biểu thức có giá trị là true	$5 > 4 \    \ 6 \neq 7$ $5 < 4 \    \ 6 \neq 7$	?
<b>!</b>	Lấy giá trị phủ định của biểu thức	$!(5 < 4)$	?



# BIỂU THỨC VÀ TOÁN TỬ

## Các phép gán:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<b>=</b>	Phép gán bằng	$a=5$	5
<b>+=</b>	Phép cộng bằng	$a+=2$	$a=a+2=7$
<b>-=</b>	Phép trừ bằng	$a-=2$	$a=a-2=3$
<b>*=</b>	Phép nhân bằng	$a*=2$	$a=a*2=10$
<b>/=</b>	Phép chia bằng	$a/=2$	$a=a/2=2.5$
<b>%=</b>	Phép lấy dư bằng	$a\%=2$	$a=a\%2=1$

# Độ ưu tiên toán tử

Thứ tự	Toán tử
1	Các toán tử đơn như: +, -, ++, --
2	Các toán tử số học *, /, +, -
3	Các toán tử quan hệ >, <, >=, <=, ==, !=
4	Các toán tử luận lý &&,   , ?:
5	Các toán tử gán =, *=, /=, +=, -=

# Các hàm toán học

❑ Java cung cấp các hàm tiện ích giúp chúng ta thực hiện các phép tính khó một cách dễ dàng như:

- ❖ Làm tròn số
- ❖ Tính căn bậc 2
- ❖ Tính lũy thừa
- ❖ ...

❑ Ví dụ sau đây tính căn bậc 2 của 7

❖ `double a = Math.sqrt(7)`

❑ Ngoài `Math.sqrt()` còn rất nhiều hàm khác được trình bày ở slide sau.

# Các hàm toán học

Hàm	Diễn giải	Ví dụ
Math.min(a, b)	Lấy số nhỏ nhất của 2 số a và b	$x = \text{Math.min}(5, 3.5) \Rightarrow x = 3.5$
Math.max(a, b)	Lấy số lớn nhất của 2 số a và b	$x = \text{Math.max}(5, 3.5) \Rightarrow x = 5$
Math.pow(a, n)	Tính $a^n$ (a lũy thừa n)	$x = \text{Math.pow}(5, 3) \Rightarrow x = 75$
Math.sqrt(a)	Tính $\sqrt{a}$ (căn bậc 2 của a)	$x = \text{Math.sqrt}(16) \Rightarrow x = 4$
Math.abs(a)	Lấy giá trị tuyệt đối của a	$x = \text{Math.abs}(-5) \Rightarrow x = 5$
Math.ceil(a)	Lấy số nguyên trên của a	$x = \text{Math.ceil}(3.5) \Rightarrow x = 4$
Math.floor(a)	Lấy số nguyên dưới của a	$x = \text{Math.floor}(3.5) \Rightarrow x = 3$
Math.round(a)	Làm tròn số của a	$x = \text{Math.round}(3.5) \Rightarrow x = 4$
Math.random()	Sinh số ngẫu nhiên từ 0 đến 1	$x = \text{Math.random()} \Rightarrow x = 0..1$

# BIỂU THỨC VÀ TOÁN TỬ

Toán tử điều kiện là toán tử 3 ngôi duy nhất trong ngôn ngữ Java

- Cú pháp:

<điều kiện> ? <giá trị đúng> : <giá trị sai>

- Diễn giải: Nếu biểu thức <điều kiện> có giá trị là **true** thì kết quả của biểu thức là <giá trị đúng>, ngược lại là <giá trị sai>

- Ví dụ 1: Tìm số lớn nhất của 2 số a và b

```
int a = 1, b = 9;
```

```
int max = a > b ? a : b;
```

- Ví dụ 2: Tìm số lớn nhất của 3 số a và b, c

```
int max3so = (a>b?a:b)>c?(a>b?a:b):c;
```

Hoặc: 

```
max2so = a>b?a:b
```

```
max3so = max2so > c ? Max2so : c
```

# KHAI BÁO BIẾN

- **Biến là một đại lượng có giá trị thay đổi trong quá trình thực hiện chương trình. Mỗi biến có kiểu dữ liệu riêng.**
- Cú pháp khai báo biến:  
**<kiểu dữ liệu> <tên biến> [= <giá trị khởi đầu>];**
- Ví dụ:
  - ✓ **int a;** // khai báo biến không có giá trị khởi đầu
  - ✓ **double b = 5;** // khai báo biến có giá trị khởi đầu
  - ✓ **String name = "Anna";**
- Khai báo nhiều biến cùng kiểu: **int a, b=5, c;**

Chú ý: Trong java nếu lúc khai báo không khởi tạo giá trị cho biến thì nó sẽ nhận 1 giá trị mặc định. Mỗi kiểu dữ liệu có 1 giá trị mặc định khác nhau.

- 0 nếu kiểu dữ liệu là kiểu số
- '\0' nếu kiểu dữ liệu là ký tự
- false nếu kiểu dữ liệu là boolean
- null nếu kiểu dữ liệu là lớp

# KHAI BÁO BIẾN

- **Cách đặt tên biến:**

- ✓ Sử dụng ký tự alphabet, số, \$ hoặc gạch dưới (\_).
- ✓ Tên có phân biệt HOA/thường
- ✓ Không bắt đầu bởi số, không dùng từ khóa

*\* Từ khóa là các từ được sử dụng để xây dựng ra ngôn ngữ lập trình java*

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

# KHAI BÁO HẲNG

- Hằng là một đại lượng có giá trị không thay đổi trong suốt quá trình thực hiện chương trình. Mỗi hằng có kiểu dữ liệu riêng.
- Tên đặt theo qui ước như tên biến
- Khai báo dùng từ khóa **final**
- Khai báo hằng số:

**final** <kiểu dữ liệu> <tên hằng số> = <giá trị>;

- Ví dụ:
  - ✓ final **int** a=20;
  - ✓ final **String** hoten = "Nguyễn Văn Dũng";
- Chú ý:
  - ✓ Hằng ký tự: đặt giữa cặp nháy đơn "
  - ✓ Hằng chuỗi: là một dãy ký tự đặt giữa cặp nháy đôi ""



# KHAI BÁO HẰNG

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

Hằng ký tự đặc biệt

# XUẤT DỮ LIỆU TRONG JAVA

- Có thể sử dụng 3 hàm sau để xuất dữ liệu:

`System.out.print()`: Xuất xong không xuống dòng

`System.out.println()`: Xuất xong có xuống dòng

`System.out.printf()`: Xuất có định dạng, các ký tự định dạng

✓ `%d`: số nguyên

✓ `%f`: số thực, mặc định là 6 số lẻ

`%.3f` định dạng 3 số lẻ

✓ `%s`: chuỗi

- Ví dụ:

✓ `System.out.print("UTC - ");`

✓ `System.out.println("Đại học giao thông vận tải ");`

✓ `System.out.printf("Đào tạo %d ngành", 20);`

UTC – Đại học giao thông vận tải  
Đào tạo 20 ngành

# NHẬP DỮ LIỆU TRONG JAVA - `java.util.Scanner`

- B1: Khai báo trước class: `java.util.Scanner` để cho phép nhận dữ liệu từ bàn phím
- B2: Tạo đối tượng **Scanner**  
`Scanner <tên_đối_tượng> = new Scanner(System.in)`
- B3: Sử dụng các phương thức thường dùng để nhập:  
`Scanner sc = new Scanner(System.in)`  
**`sc.nextLine()`**: nhập chuỗi  
**`sc.nextInt()`**: nhập số nguyên  
**`sc.nextDouble()`**: nhập số thực

# NHẬP DỮ LIỆU TRONG JAVA - java.util.Scanner

- Ví dụ:

```
package Lab01;
import java.util.Scanner;
public class Lab1Bai1 {
    public static void main(String[] args){
//Nhập dữ liệu
        Scanner sc = new Scanner(System.in);
        System.out.print("Họ và tên : "); String hoten = sc.nextLine();
        System.out.print("Điểm TB : "); double dtb = sc.nextDouble();
//Xuất dữ liệu
        System.out.print(hoten + ", điểm trung bình là : " + dtb + "\n");
        System.out.println(hoten + ",điểm trung bình là : " + dtb);
        System.out.printf("%s %.2f điểm\n",hoten, dtb);
    }
}
```

# Nhập dữ liệu từ bàn phím – java.util.Scanner

CongHaiSo.java

```
5 import java.util.Scanner;
6 public class CongHaiSo{
7     public static void main(String[] args){
8         Scanner inputData = new Scanner(System.in);
9         int a, b, s;
10        System.out.print("Nhap vao so nguyen thu nhat: ");
11        a = inputData.nextInt();
12
13        System.out.print("Nhap vao so nguyen thu hai: ");
14        b = inputData.nextInt();
15        inputData.close();
16
17        s = a+b;
18        System.out.println("Tong cua hai so = " + s);
19    }
20 }
```

# Nhập dữ liệu từ bàn phím – java.io.\*

CongHaiSo.java

```
1 /*Ví dụ minh hoạ nhập 2 số nguyên từ bàn phím,  
2  * tính tổng và xuất kết quả  
3  */  
4 package minhthai.oop;  
5 import java.io.*;  
6 public class CongHaiSo {  
7     public static void main(String[] args) throws IOException {  
8         int a, b, s;  
9         String str;  
10        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
11        System.out.print("Nhập vào số nguyên thu nhất: ");  
12        str = br.readLine();  
13        a = Integer.parseInt(str);  
14  
15        System.out.print("Nhập vào số nguyên thu hai: ");  
16        str=br.readLine();  
17        b=Integer.parseInt(str);  
18  
19        s = a+b;  
20        System.out.println("Tong cua hai so = " + s);  
21    }  
22 }
```

# BÀI THỰC HÀNH 1

## *Hướng dẫn bài 2: Nhập dữ liệu sử dụng đối tượng Scanner trong thư viện java.util.\**

```
import java.io.*; //thư viện chứa các lớp cho việc xuất, nhập
import java.util.*; //thư viện chứa lệnh nhập sử dụng Scanner
class vd
{
    public static void main (String[] args)
    {

        Scanner x=new Scanner(System.in);
        System.out.print("Nhap ma sinh vien:");
        String masv=x.nextLine();

        //nhập họ tên: tương tự
        //...

        System.out.print("Nhap tuoi:");
        byte tuoi=x.nextByte();

        //tương tự với năm sinh, điểm trung bình
        //...

    }
}
```

# BÀI THỰC HÀNH 1

**Bài tập 3. Viết chương trình: Nhập vào hai số nguyên x, y. Xuất ra màn hình tổng, hiệu, tích, thương của hai số trên.**

**Bài tập 4: Viết chương trình nhập từ bàn phím 2 cạnh của hình chữ nhật. Tính và xuất chu vi, diện tích và cạnh nhỏ nhất của hình chữ nhật.**

*Hướng dẫn:*

**Chu vi = (dai+rong)\*2**

**Dientich=Dai\*rong**

**Canhhonhat=Math.min(dai,rong)**



# BÀI THỰC HÀNH 1

## Bài tập 5:

Viết chương trình nhập các hệ số của phương trình bậc 2. Tính delta và xuất căn delta ra màn hình.

### HƯỚNG DẪN:

- ✓  $\Delta = \text{Math.pow}(b, 2) - 4 * a * c$
- ✓ Sử dụng `Math.sqrt(delta)` để tính căn delta

# CÁC CẤU TRÚC ĐIỀU KHIỂN

## ❑ Rẽ nhánh

- ✓ if
- ✓ switch

## ❑ Lặp:

- ✓ while
- ✓ do...while
- ✓ for

## ❑ Lệnh ngắt:

- ✓ break
- ✓ continue

# CẤU TRÚC IF

- Cấu trúc **if** được sử dụng để ra quyết định thực hiện một công việc nào đó tùy vào điều kiện đúng hay sai
- Ví dụ: Cho một số nguyên  $a$ , kiểm tra xem  $a$  là số chẵn hay số lẻ
  - Nếu  $a$  chia hết cho 2 thì  $a$  là số chẵn
  - Nếu  $a$  không chia hết cho 2 thì  $a$  là số lẻ

# CẤU TRÚC IF

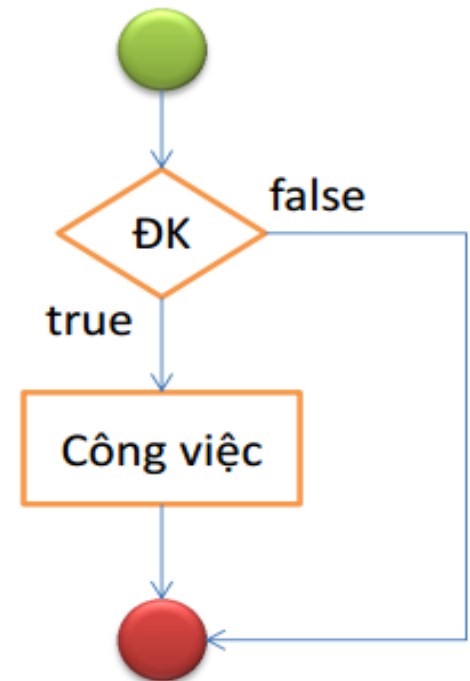
## ❖ Có 3 dạng rẽ nhánh của **if**

Dạng 1:

```
if (condition) {  
    //Statements;  
}
```

Ví dụ:

```
int a=8;  
if (a % 2 == 0) {  
    System.out.println(a + “ là một số chẵn”);  
}
```



# CẤU TRÚC IF

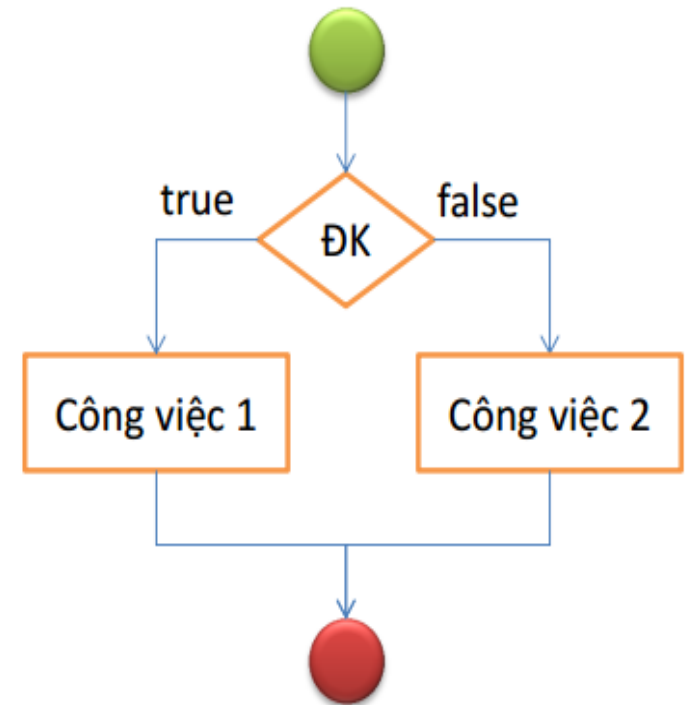
❖ Có 3 dạng rẽ nhánh của **if**

Dạng 2:

```
if (condition) {  
    //Statements;  
}  
else {  
    //Statement;  
}
```

Ví dụ:

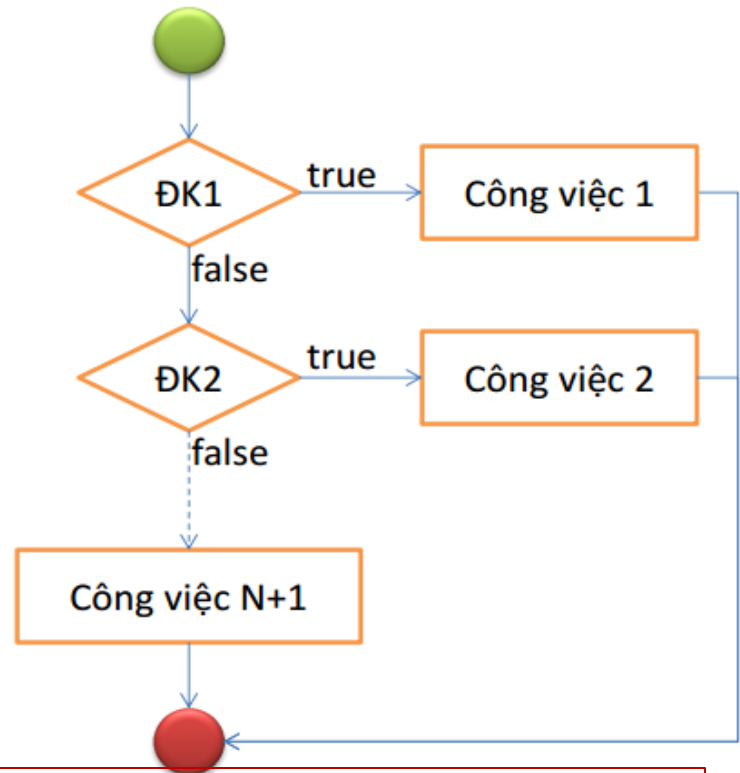
```
int a=8;  
if (a % 2 == 0) {  
    System.out.println(a + “ là một số chẵn”);}  
else {  
    System.out.println(a + “ là một số lẻ”); }
```



# CẤU TRÚC IF

## Dạng 3: Có nhiều điều kiện

```
if(condition_1) {  
    //Statements_1;  
} else if(condition_2){  
    //Statements_2;  
} else if(condition_3){  
    //Statements_3;  
}  
...  
else{  
    //Statements_N+1;  
}
```



```
int a = -10;  
if (a > 0) {  
    System.out.println(a + " là một số lớn hơn 0");  
else if (a == 0) {  
    System.out.println("Đây là số 0");  
}  
else {  
    System.out.println(a + " là một số nhỏ hơn 0");  
}
```

# CẤU TRÚC SWITCH

Cú pháp:

```
switch (Expression)
{
    case Cons1: Statement_1; break;
    case Cons2: Statement_2; break;
    ...
    default: Statement_N+1; break;
}
```

Diễn giải: Nếu <biểu thức> = <giá trị i> thì thực hiện <công việc i> ngược lại sẽ thực hiện <công việc N+1>.

Nếu case không chứa **break** thì case tiếp sau sẽ được thực hiện.

Khi gặp lệnh **break** chương trình sẽ **thoát** khỏi switch

# CẤU TRÚC SWITCH

Ví dụ:

```
int a=10, b=4;
char dau='+';
switch (dau){
    case '+': System.out.printf("Tổng = %d\n", a+b); break;
    case '-': System.out.printf("Hiệu = %d\n", a-b); break;
    case '*': System.out.printf("Tích = %d\n", a*b); break;
    case '/': System.out.printf("Thương = %.2f\n", (float)
a/b); break;
    default:
        System.out.println('Phép tính bạn nhập chưa đúng!');
        break;
}
```



# BÀI TẬP SỬ DỤNG IF VÀ SWITCH

1. Viết chương trình giải bài toán phương trình bậc nhất  $ax^2+bx+c=0$  với a, b, c được nhập vào từ bàn phím
2. Viết chương trình nhập vào a, b, c. Tìm số lớn nhất, số nhỏ nhất
3. Viết chương trình nhập bậc lương (BacLuong), ngày công (NgayCong), phụ cấp (PhuCap). Tính tiền thực lĩnh (TienLinh) =  $BacLuong * 1500000 * NCTL + PhuCap$

Với:  $NCTL = NgayCong$  nếu  $NgayCong < 25$

$NCTL = (NgayCong - 25) * 2 + 25$  nếu  $NgayCong \geq 25$

5. Nhập vào độ dài 3 cạnh của tam giác, kiểm tra độ dài 3 cạnh lập thành tam giác thì tính chu vi, diện tích của tam giác đó.

# BÀI TẬP SỬ DỤNG IF VÀ SWITCH

6. Viết chương trình nhập vào tháng, năm. Thông báo ra màn hình số ngày của tháng năm đó. (Biết **tháng 1,3,5,7,8,10,12 có 31 ngày**. Tháng **4,6,9,11 có 30 ngày**. Năm chia hết cho 4 thì tháng 2 có 29 ngày)
7. Viết một chương trình trắc nghiệm tính cách qua **màu sắc** mà bạn thích:
- Nhập vào màu sắc: xanh, đỏ, tím, vàng, hồng, trắng
  - Thông báo ra màn hình tính cách của người thích màu sắc đó

Nếu nhập vào màu sắc khác thì chương trình thông báo: “Chương trình đang cập nhật!”

# CẤU TRÚC LẶP

Lệnh lặp được sử dụng để điều khiển các công việc được lặp đi lặp lại nhiều lần. Mỗi lệnh lặp luôn có điều kiện dừng.

- **while**
- **do...while**
- **for**

# CẤU TRÚC LẶP

- Cú pháp:

```
while (Condition){  
    Statements;  
}
```

```
do{  
    Statements;  
}  
while (condition);
```

- Ví dụ:

```
int i=1;  
while (i<=20) {  
    System.out.print(i + ",");  
    i++;  
}
```

- Ví dụ:

```
int i=1;  
do{  
    System.out.println(i + ",");  
    i++; }  
while (i<=20);
```

# CẤU TRÚC LẶP

```
for(varInit; Condition; Step){  
    Statements;  
}
```

```
for(varInit:List){  
    Statements;  
}
```

- Ví dụ 1:

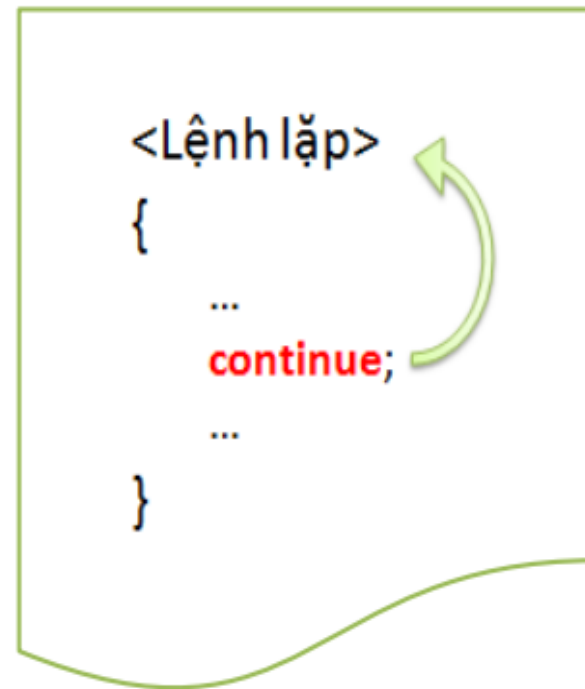
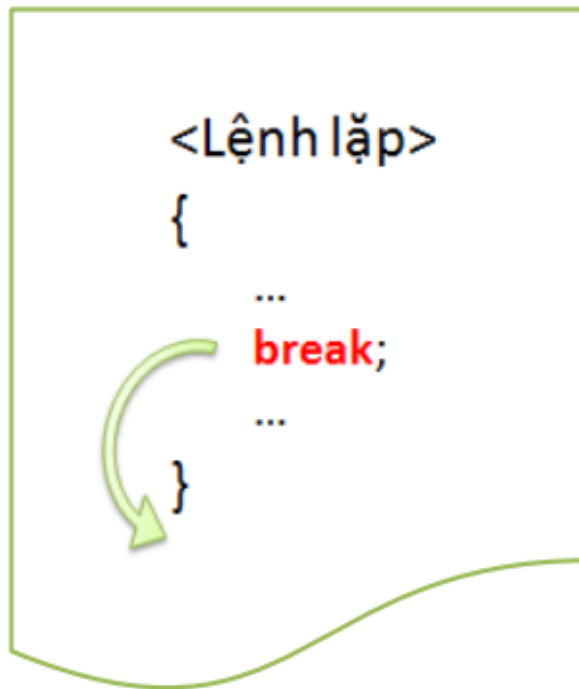
```
int[] a = new int[] { 1, 2, 3, 4, 5 };  
for (int i = 0; i < a.length; i++) {  
    System.out.println(a[i]);  
}
```

- Ví dụ 2:

```
int[] a = new int[] { 1, 2, 3, 4, 5 };  
for (int n : a) {  
    System.out.println(n);  
}
```

# LỆNH NGẮT BREAK, CONTINUE

- Lệnh break: thực hiện ngắt vòng lặp
- Lệnh continue: thực hiện lần lặp tiếp theo



# LỆNH NGẮT BREAK, CONTINUE

- Ví dụ break: Nhập vào điểm toán hợp lệ (phải từ 0 đến 10)

```
Scanner sc = new Scanner(System.in);
int diemtoan = 0;
while(true){
    diemtoan = sc.nextInt();
    if(diemtoan >= 0 && diemtoan <= 10){
        break;
    }
    System.out.println("Điểm phải từ 0 đến 10");
}
```

# LỆNH NGẮT BREAK, CONTINUE

- Ví dụ continue: In ra các số không chia hết cho 4 nằm trong khoảng [1,10]

```
for(int i=1;i<=10;i++){  
    if(i%4 ==0){  
        continue;  
    }  
    System.out.print(i + ",");  
}
```

Kết quả màn hình: 1,2,3,5,6,7,9,10