

Java for Base

GV: Bùi Quang Đăng

Contents

1

Introduction Java

2

Data type

3

Operators

4

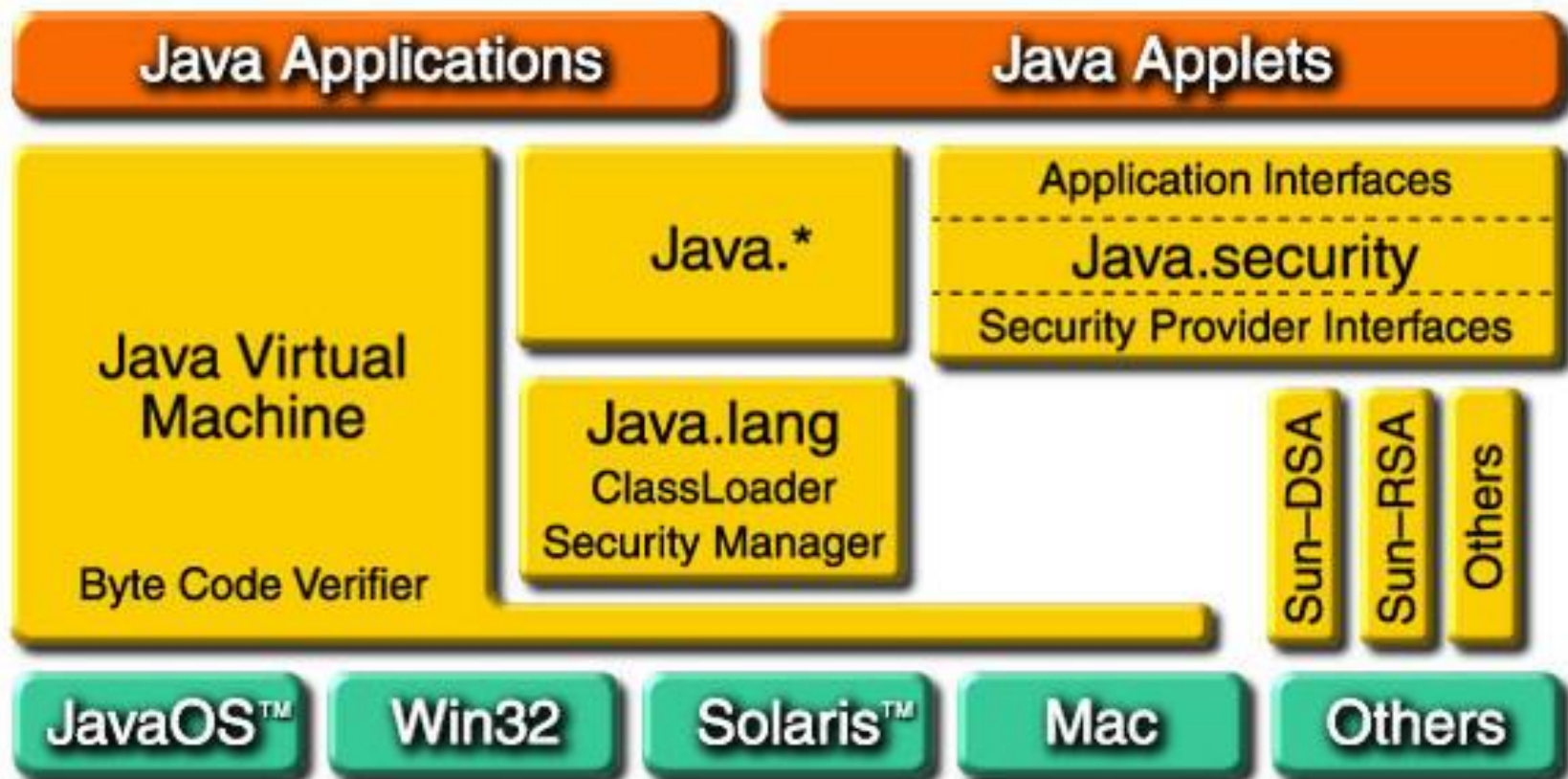
Structured programming

Java for Base

Introduction Java

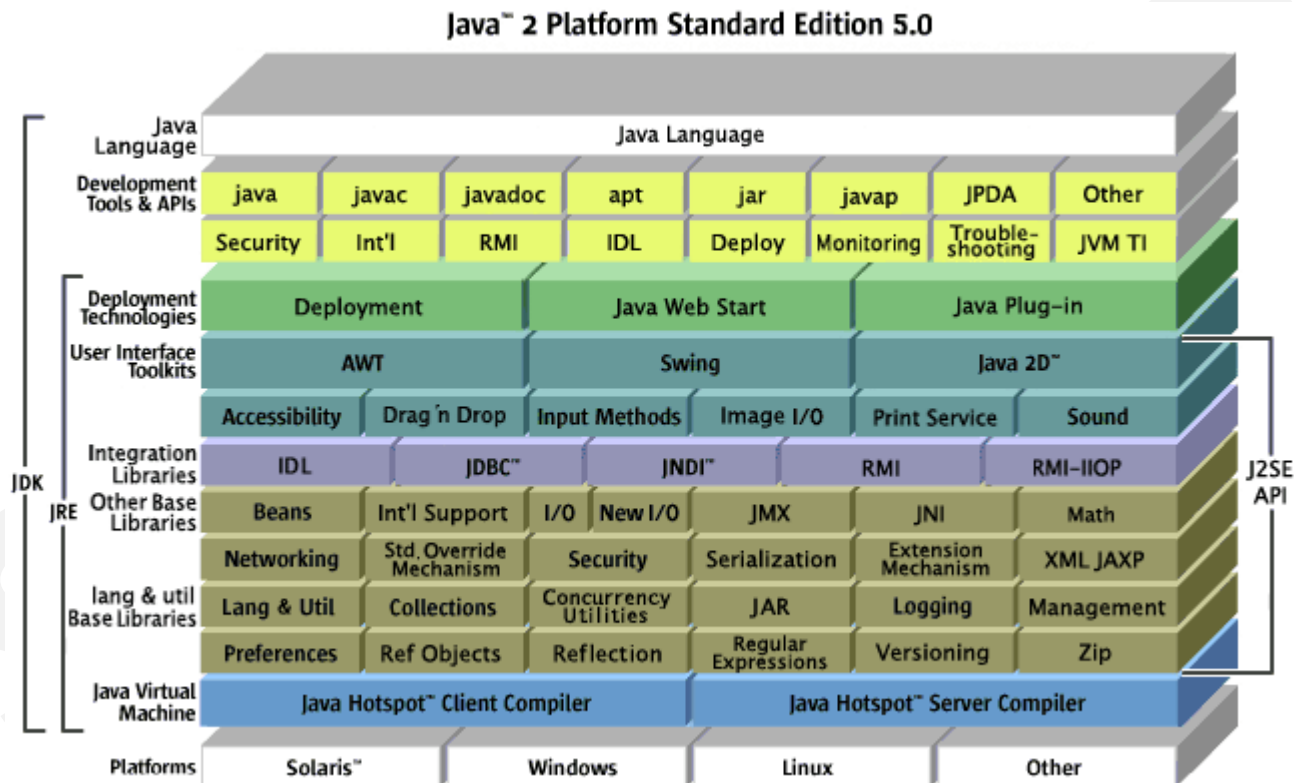
Java for Base

❖ Introduction Java



Java for Base

❖ Introduction Java



Java for Base

❖ Introduction Java

- Các file viết bằng ngôn ngữ lập trình Java có định dạng mở rộng là ***.java**.
- Để chạy được chương trình viết bằng ngôn ngữ Java cần phải cài đặt JDK (**J**ava **D**evelopment **K**it)
- Java có thể chạy được trên nhiều hệ điều hành khác nhau như Windows, Unix/Linux, Mobile,...
- Java có thể chia ra 3 loại chính:
 - **J2SE**: Lập trình trên Desktop
 - **J2EE**: Lập trình trên nền tảng Webbased
 - **J2ME**: Lập trình trên nền tảng di động (mobile)

Java for Base

❖ Introduction Java

Package	Description
java.awt	Java Abstract Window Toolkit Package chứa các lớp và giao diện để tạo ra các GUI trong Java 1.0 và 1.1.
java.awt.event	Java Abstract Window Toolkit Event Package chứa các lớp và giao diện quản lý các sự kiện trong thành phần GUI của cả 2 package java.awt và javax.swing
java.io	Java Input/Out Package chứa các lớp và giao diện cho phép chương trình làm việc với input và output của dữ liệu
java.lang	Java Language Package chứa các lớp và giao diện cơ bản để thiết kế của ngôn ngữ lập trình java
java.net	Java Networking Package chứa các lớp và giao diện cho phép kết nối và làm việc với networks trong internet

Java for Base

❖ Introduction Java

Package	Description
java.text	Java Text Package chứa các lớp và giao diện cho phép chương trình làm việc với numbers, dates, characters và strings.
java.util	Java Utilities Package chứa các lớp và giao diện tiện ích cho phép tương tác với date và time, số ngẫu nhiên, lưu trữ và xử lý dữ liệu lớn,...
javax.swing	Java Swing GUI Components Package chứa các lớp và giao diện của Java's Swing GUI components cung cấp và hỗ trợ portable GUIs.
javax.swing.event	Java Swing Event Package chứa các lớp và giao diện cho phép quản lý sự kiện của các thành phần GUI trong package javax.swing

Java for Base

❖ Introduction Java

Keyword	Contents
Abstract	+ Phương thức hoặc lớp đối tượng là trừu tượng
Break	+ Ngắt dòng lệnh để thoát khỏi vòng lặp
Class	+ Lớp đối tượng
Extends	+ Dẫn xuất (kế thừa)
Final	+ Thông báo biến đi sau là hằng số
Implements	+ Thực thi
Import	+ Khai báo sử dụng các thư viện
Package	+ Gói thư viện chứa các lớp đối tượng

Java for Base

Data types

Java for Base

❖ Data types

Kiểu	Kích thước (byte)	Mô tả
byte	1	Giá trị từ -128 đến 127, giá trị mặc định là 0, ví dụ: <code>b = 125;</code>
short	2	Giá trị từ -32,768 đến 32,767, ví dụ: <code>short s = 1000;</code>
int	4	Giá trị từ - 2,147,483,648 đến 2,147,483,647, ví dụ: <code>int a = 2000;</code>
long	8	Giá trị từ -2^{63} – $(2^{63} - 1)$, ví dụ: <code>long a = 100000L;</code>
float	4	ví dụ: <code>float f1 = 234.5f</code>
double	8	ví dụ: <code>double d1 = 123.4</code>

Java for Base

❖ Data types

Kiểu	Kích thước (byte)	Mô tả
boolean	1	Kiểu trả về true, false
char	1	Mã kí tự dạng Unicode, ví dụ: char letterA ='A', với Unicode cần 2 byte

Ví dụ: String strCompany = “Stanford – Dạy kinh nghiệm lập trình”

String strWebsite = “Stanford.com.vn”

int code = 25;

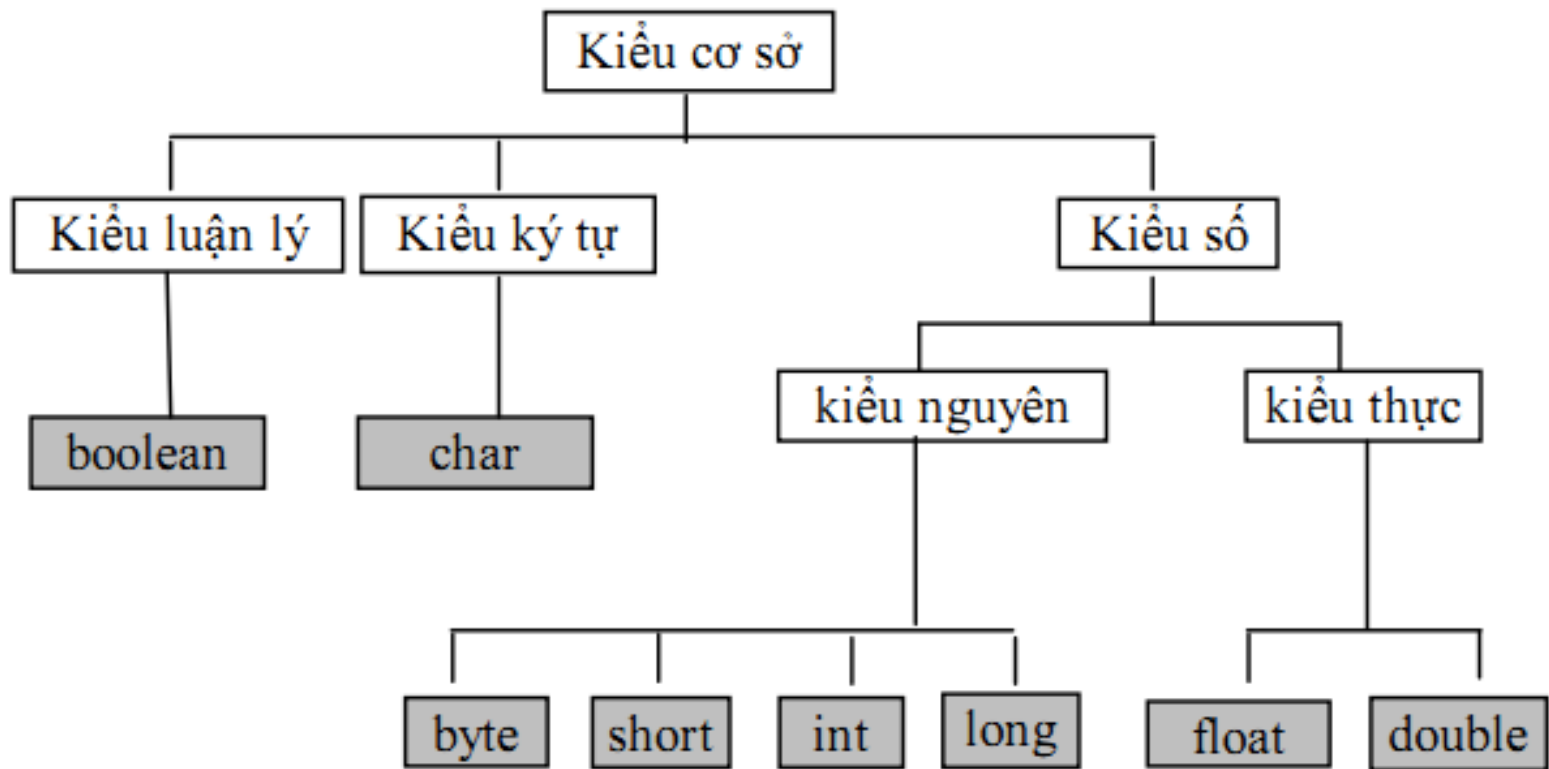
Java for Base

❖ Data types

- Không thể chuyển đổi kiểu dữ liệu có giá trị lớn → nhỏ
- Kiểu boolean có giá trị true, false tương ứng với giá trị là 1 và 0
- Không thể chuyển đổi kiểu Char → Boolean. Và ngược lại
- Chỉ có thể chuyển đổi qua lại giữa kiểu Char với tất cả các kiểu khác. và ngược lại (tham khảo bảng ASCII)
- Kiểu float khi khởi tạo phải đi với ký tự F. Ví dụ: a=1.234F
- Kiểu decimal khi khởi tạo phải đi với ký tự M. Ví dụ: a=1.234M
- Chuyển đổi về kiểu Boolean giá trị khác 0 là True còn = 0 là False

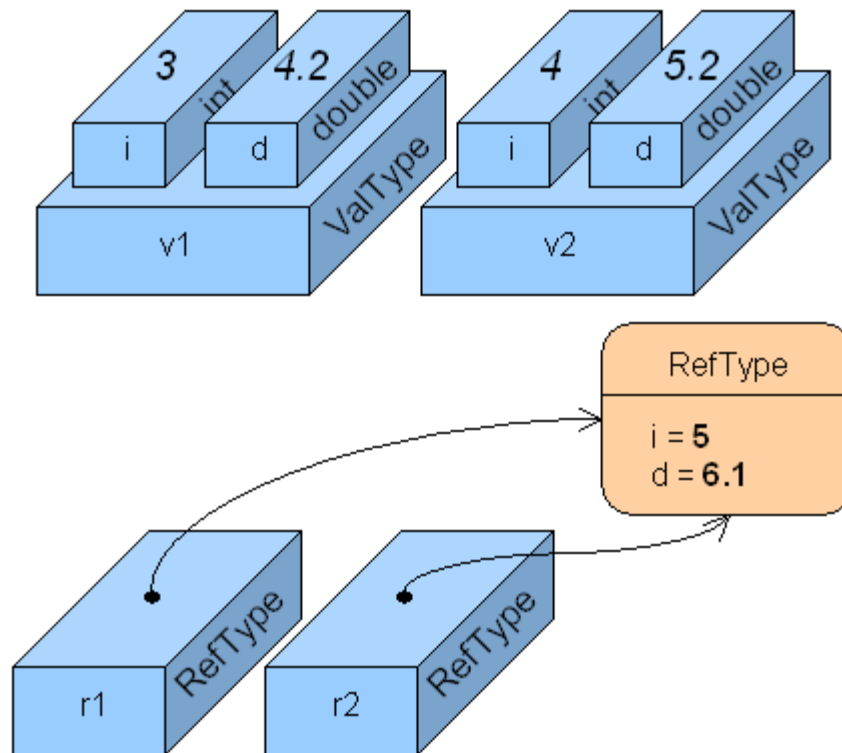
Java for Base

❖ Data types



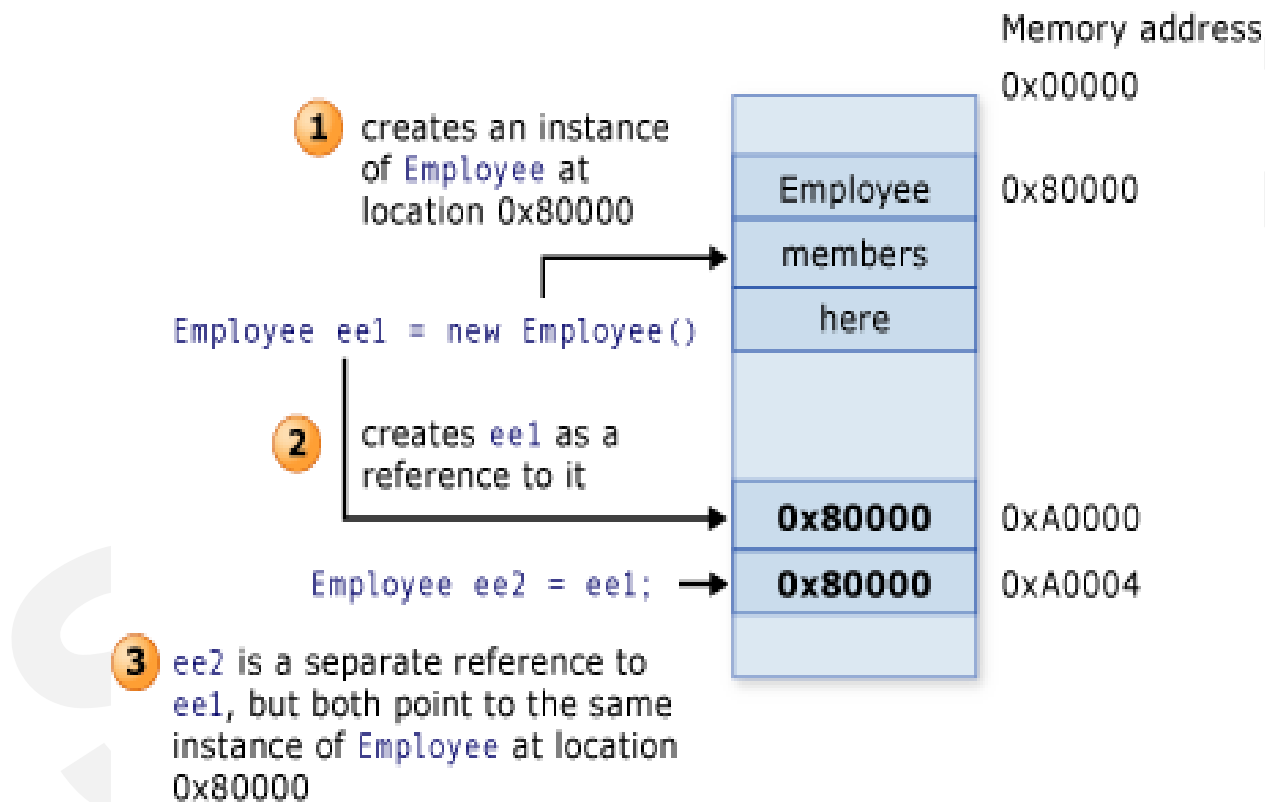
Java for Base

❖ Data types



Java for Base

❖ Data types



Java for Base

Operators

Java for Base

❖ Operators

Toán tử	Ý nghĩa
<code>+=</code>	Cộng thêm giá trị toán hạng bên phải vào giá trị toán hạng bên trái
<code>-=</code>	Toán hạng bên trái được trừ bớt đi một lượng bằng giá trị của toán hạng bên phải
<code>*=</code>	Toán hạng bên trái được nhân với một lượng bằng giá trị của toán hạng bên phải.
<code>/=</code>	Toán hạng bên trái được chia với một lượng bằng giá trị của toán hạng bên phải.
<code>%=</code>	Toán hạng bên trái được chia lấy dư với một lượng bằng giá trị của toán hạng bên phải.

Toán tử đặc biệt là tăng một (`++`) hoặc giảm một (`--`)

Java for Base

❖ Operators

▪ Toán tử so sánh

Tên toán tử	Kí hiệu	Biểu thức so sánh	Kết quả so sánh
So sánh bằng	==	value1 == 100 value1 == 50	true false
Không bằng	!=	value1 != 100 value2 != 90	false true
Lớn hơn	>	value1 > value2 value2 > value1	true false
Lớn hơn hay bằng	>=	value2 >= 50	true
Nhỏ hơn	<	value1 < value2 value2 < value1	false true
Nhỏ hơn hay bằng	<=	value1 <= value2	false

Java for Base

❖ Operators

▪ Toán tử logic

Tên toán tử	Ký hiệu	Biểu thức logic	Giá trị	Logic
and	&&	$(x == 3) \&\& (y == 7)$	false	Cả hai điều kiện phải đúng
or		$(x == 3) (y == 7)$	true	Chỉ cần một điều kiện đúng
not	!	$! (x == 3)$	true	Biểu thức trong ngoặc phải sai.



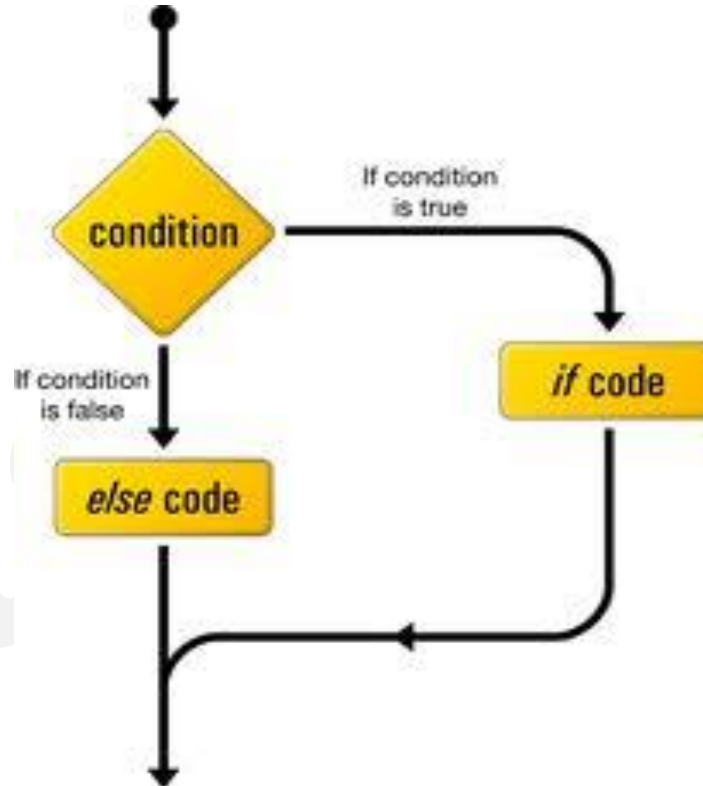
Java for Base

Structured programming

Java for Base

❖ Structured programming

▪ If...Else



Java for Base

❖ Structured programming

▪ If...Else

if (biểu thức điều kiện)

```
{  
    <lệnh 1>  
    <lệnh 2>  
    ....  
}  
[else  
    {  
        <lệnh 1>  
        <lệnh 2>  
        ...  
    }  
}]
```

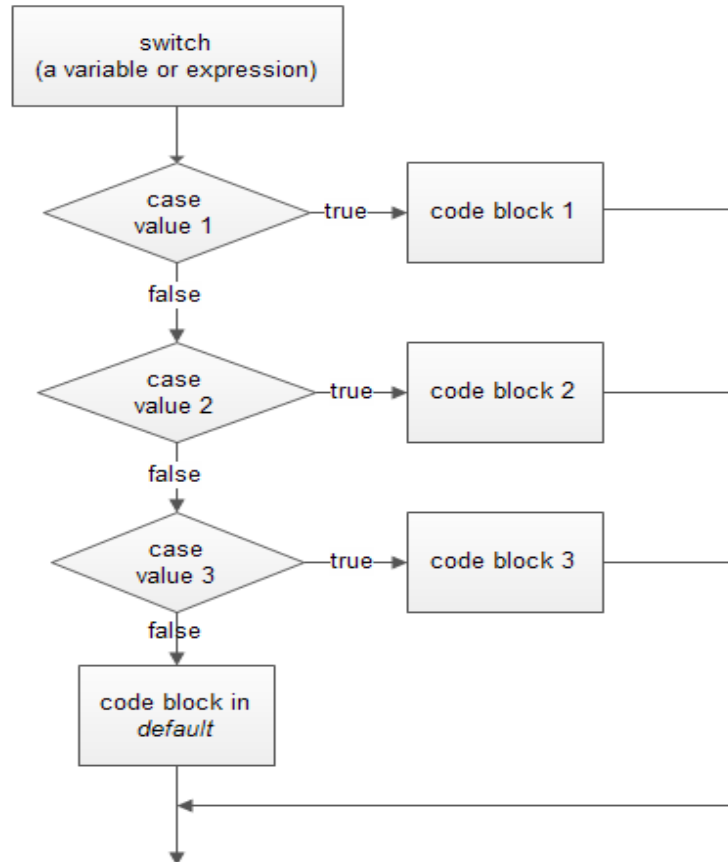
* Câu lệnh If lồng nhau :

```
if(biểu thức điều kiện)  
{  
    if(biểu thức điều kiện)  
    {  
        <lệnh n>  
    }  
    else  
    {  
        <lệnh n>  
    }  
}  
else  
{  
    <lệnh n>  
}
```

Java for Base

❖ Structured programming

▪ Switch...Case



Java for Base

❖ Structured programming

▪ Switch...Case

switch (biểu thức điều kiện)

{

case <giá trị>:

<Các câu lệnh thực hiện>

<lệnh nhảy>

[default:

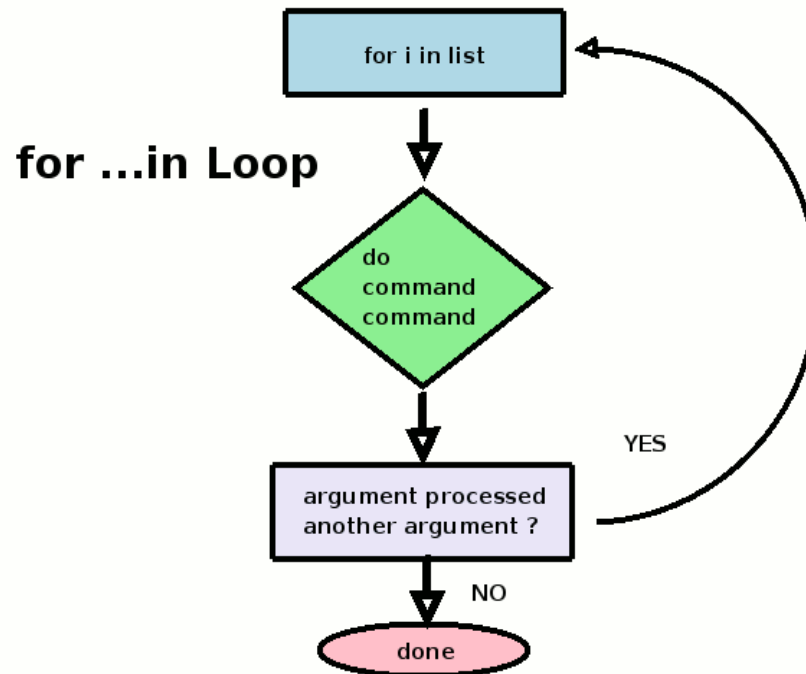
<Các câu lệnh thực hiện mặc định>]

}

Java for Base

❖ Structured programming

▪ Vòng lặp



Java for Base

❖ Structured programming

▪ Vòng lặp

- Sử dụng để thực hiện một số công việc giống nhau lặp đi lặp lại nhiều lần theo một khuôn mẫu.
- Sử dụng các cấu trúc while, do...while, for trong lập trình Java khi dùng vòng lặp.

Java for Base

❖ Structured programming

- **while:** Thực hiện công việc trong vòng lặp nếu biểu thức điều kiện đúng.

```
while (Biểu thức điều kiện)
{
    <Câu lệnh thực hiện>
}
```

Ví dụ: while(i<5)

```
{
    i++;
    System.out.println ("Số thứ tự là: " + i);
}
```

Java for Base

❖ Structured programming

- **Do...while:** Thực hiện công việc trong vòng lặp ít nhất một lần

```
do
{
    <Câu lệnh thực hiện>
}
```

while (Biểu thức điều kiện)

Ví dụ:

```
do {System.out.println("Số thứ tự là:" + i);
}
while (i<0)
```

Java for Base

❖ Structured programming

- **for:** Sử dụng để duyệt một tập phần tử theo các biểu thức điều kiện, bước lặp quy định

for ([phần khởi tạo] ; [biểu thức điều kiện]; [bước lặp])
<Câu lệnh thực hiện>

- **for(Kieu_du_lieu Ten_bien : Ten_mang)**
{
}

=> Duyệt lần lượt các phần tử của mảng, danh sách

Java for Base

❖ Structured programming

- **for:** Sử dụng để duyệt một tập phần tử theo các biểu thức điều kiện, bước lặp quy định

Ví dụ:

```
//Sử dụng vòng lặp for
for(i=3; i<5; i+=2)
{
    System.out.println("Số thứ tự là: " + i);
}
```

```
//Sử dụng vòng for để duyệt tất cả các phần tử trong mảng hoặc danh sách
int arr[] = new int[]{10,40,20,50,60,70,90,80};

System.out.println("Số phần tử là: ");
for(int a : arr)
{
    System.out.println(a);
}
```



Thank You !