

# LẬP TRÌNH JAVA

## Bài 3: Cấu trúc điều khiển

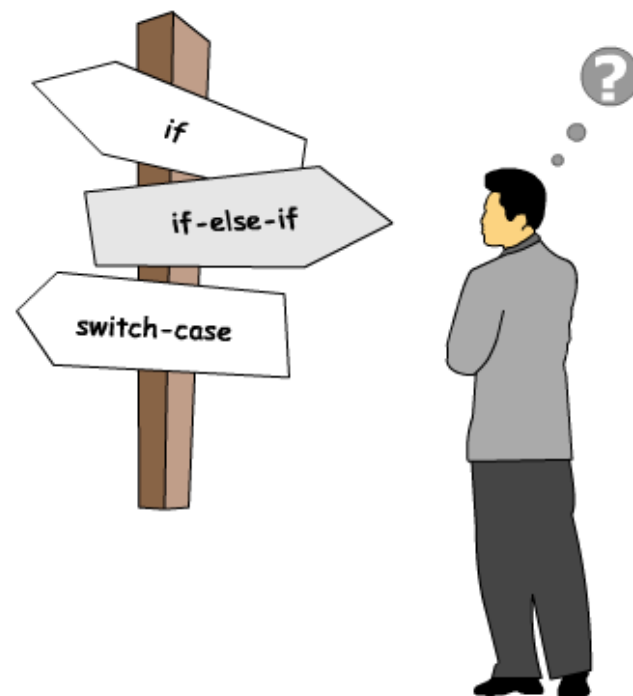
# NỘI DUNG

- ▣ Cấu trúc rẽ nhánh
- ▣ Cấu trúc lặp
- ▣ Câu lệnh nhảy

# Cấu trúc rẽ nhánh

# Cấu trúc rẽ nhánh

- ▣ Trong một số trường hợp, chúng ta không mong muốn câu lệnh chạy theo thứ tự, mà sẽ thực hiện những câu lệnh theo một điều kiện nào đó
- ▣ Cấu trúc rẽ nhánh có các loại:
  - ▣ `if`
  - ▣ `if-else-if`
  - ▣ `switch-case`



# Câu lệnh if

- ❑ Câu lệnh if sẽ thực hiện kiểm tra một biểu thức điều kiện, nếu biểu thức điều kiện trả về kết quả đúng, khối lệnh bên trong if sẽ được thực hiện
- ❑ Có thể chia thành 3 cách sử dụng
  - ❑ `if đơn`
  - ❑ `if - else`
  - ❑ `if - else - if`

# Câu lệnh if đơn giản

## ▣ Cú pháp

```
if (biểu_thức_điều_kiện) {  
    khối_lệnh  
}
```

## ▣ Ví dụ

```
if (age < 18) {  
    System.out.print("Không đủ tuổi đăng ký")  
}
```

# Câu lệnh if-else

## ▣ Cú pháp

```
if (biểu_thức_điều_kiện) {  
    khối_lệnh  
} else {  
    khối_lệnh  
}
```

## ▣ Ví dụ

```
if (a < b) {  
    min = a;  
} else {  
    min = b;  
}
```

# Câu lệnh if-else-if

## ▣ Cú pháp

```
if (biểu_thức_điều_kiện) {  
    khối_lệnh  
} else if (biểu_thức_điều_kiện) {  
    khối_lệnh  
} else {  
    khối_lệnh  
}
```



# Câu lệnh if-else-if

## ▣ Ví dụ

```
double mark = 8;
if (mark > 8.5) {
    System.out.print("Grade = A");
} else if (mark > 6.5){
    System.out.print("Grade = B");
} else if (mark > 5) {
    System.out.print("Grade = C");
} else if (mark > 4) {
    System.out.print("Grade = D");
} else {
    System.out.print("Grade = F");
}
```

# Câu lệnh switch - case

- ▣ Cấu trúc switch - case có tác dụng tương tự như khối if-else
- ▣ Thường dùng trong các bài toán có nhiều case khác nhau, biểu thức điều kiện đơn giản là so sánh bằng.

# Câu lệnh switch - case

## ▣ Cú pháp

```
switch (biểu_thức_điều_kiện) {  
    case value0: câu_lệnh;  
                break;  
    Case value1: câu_lệnh;  
                break;  
    . . .  
    case valueN: câu_lệnh;  
                break;  
    default: câu_lệnh;  
}
```

# Câu lệnh switch - case

## ▣ Ví dụ

```
switch (day) {  
    case 0: System.out.print("Monday") ;  
            break;  
    case 1: System.out.print("Tuesday") ;  
            break;  
    . . .  
    case 6: System.out.print("Sunday") ;  
            break;  
    default: System.out.print("Day not valid") ;  
}
```

# if-else VS switch-case

<code>if - else</code>	<code>switch - case</code>
mỗi khối if-else đều có logic riêng của nó, không nhất thiết phụ thuộc nhau	mỗi case trong switch-case phụ thuộc vào giá trị ban đầu trong biểu thức điều kiện
kiểu dữ liệu của biến trong biểu thức điều kiện có thể là bất cứ loại nào	kiểu dữ liệu của biến chỉ có thể là: <code>byte</code> , <code>short</code> , <code>int</code> hoặc <code>char</code>
Chỉ 1 khối lệnh được thực hiện	nếu không có <code>break</code> có thể thực hiện nhiều khối lệnh cùng lúc

# Cấu trúc lặp

# Cấu trúc lặp

- ▣ Trong một số trường hợp, chương trình của chúng ta cần thực hiện nhiều lần một thao tác nào đó.
- ▣ Ví dụ: Tính tổng từ 1-10
  - ▣  $1+2=3$
  - ▣  $3+3=6$
  - ▣  $6+4=10$
  - ▣  $10+5=15$
  - ▣ . . .

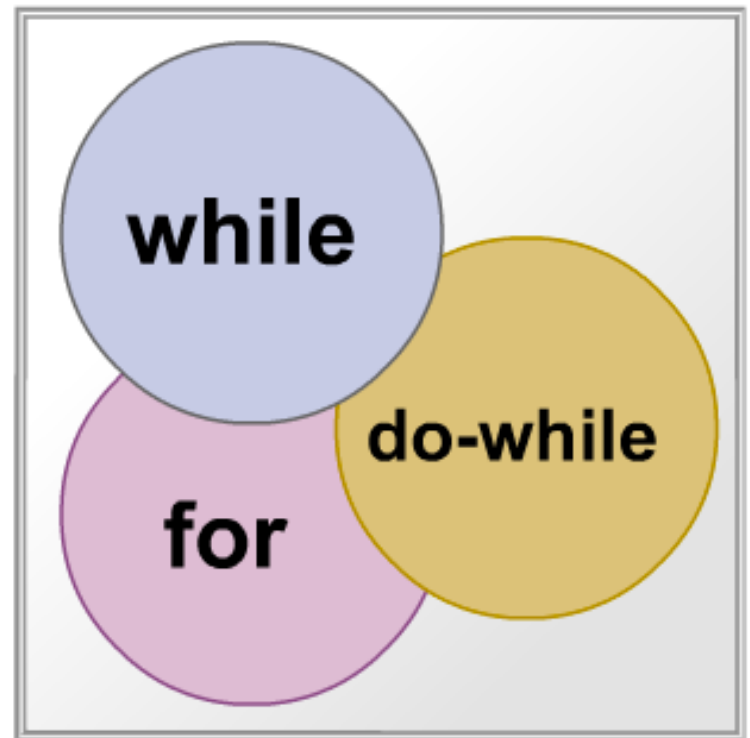
```
BEGIN
  COMPUTE result1 AS 1 * 10
  COMPUTE result2 AS 2 * 10
  COMPUTE result3 AS 3 * 10
  COMPUTE result4 AS 4 * 10
  COMPUTE result5 AS 5 * 10
END
```

**Output** →

10  
20  
30  
40  
50

# Cấu trúc lặp

- ▣ Trong Java, chúng ta có 3 loại cấu trúc lặp:
  - ▣ `for`
  - ▣ `while`
  - ▣ `do-while`





# Câu lệnh for

## ▣ Cú pháp

```
for(khởi_tạo;điều_kiện;điều_khiển) {  
    Khối_lệnh  
}
```

## ▣ Ví dụ

```
for (int i = 1; i < 10; i++) {  
    sum += i;  
}
```

# Câu lệnh for-lồng

## ▣ Cú pháp

```
for(khởi_tạo;điều_kiện;điều_khiển) {  
    for(khởi_tạo;điều_kiện;điều_khiển) {  
        khối_lệnh;  
        . . .  
    }  
    khối_lệnh;  
    . . .  
}
```

# Câu lệnh for-lồng

## ▣ Ví dụ

```
for(int i = 0;i<10;i++) {  
    for(int j = 0;j<10;j++){  
        System.out.print("*") ;  
    }  
    System.out.println("")  
}
```

# Câu lệnh while

## ▣ Cú pháp

```
While(biểu_thứckiềukiện) {  
    Khối_lệnh  
}
```

## ▣ Ví dụ

```
while (i < 10) {  
    sum += i;  
}
```

# Câu lệnh do-while

## ▣ Cú pháp

```
do {  
    Khối_lệnh  
} while(biểu_thức_điều_kiện);
```

## ▣ Ví dụ

```
do {  
    sum += i;  
} while(i < 10);
```

# while/for VS do-while

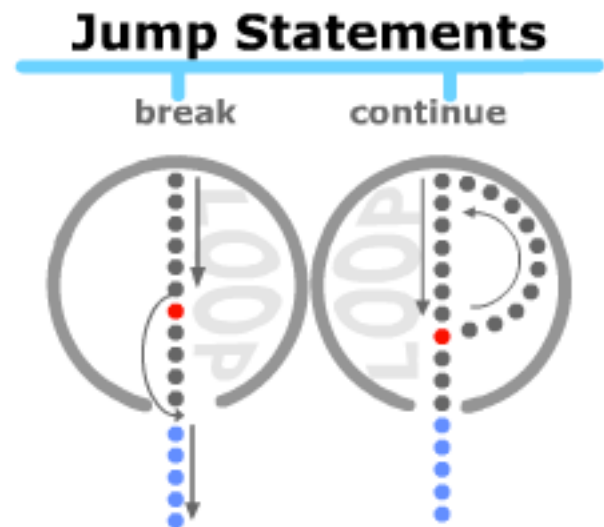
while/for	do-while
kiểm tra điều kiện trước khi lặp	kiểm tra điều kiện sau khi thực hiện khối lệnh
khối lệnh có thể không được thực hiện lần nào	khối lệnh chắc chắn được thực hiện ít nhất 1 lần

# Câu lệnh nhảy

# Câu lệnh nhảy

- Thông thường, một vòng lặp chỉ dừng khi điều kiện lặp kết thúc hay hết số lần lặp. Nhưng trong 1 số trường hợp, chúng ta cần dừng vòng lặp giữa chừng, hoặc bỏ qua 1 số vòng lặp nhất định. Chúng ta sẽ sử dụng **Câu lệnh nhảy-Jump Statement.**

- Chúng ta có 2 loại:
  - **break;**
  - **continue;**





# Break

- Câu lệnh break trong java, có 2 cách để sử dụng. Cách thứ 1, là trong cấu trúc swith-case và cách thứ 2 là sử dụng trong vòng lặp.
- Tác dụng làm dừng vòng lặp ngay lập tức, kể cả khi điều kiện lặp vẫn còn đúng.

# Break

## ▣ Ví dụ

```
Scanner input = new Scanner(System.in);  
for(int i = 0; i < 100; i++) {  
    System.out.print("Enter a number");  
    number = input.nextInt();  
    if (number == 500)  
        break;  
}
```

# Continue

- Giống như **break**, **continue** chỉ có thể sử dụng bên trong các cấu trúc lặp như **while**, **for**, **do-while**
- Tác dụng bỏ qua vòng lặp hiện tại và thực hiện ngay vòng lặp tiếp theo, mà không cần thực hiện hết khối lệnh.

# Break

## ▣ Ví dụ

```
for(int i = 0; i < 10; i++) {  
    if (i % 2 != 0) continue;  
    sum += i;  
}
```