

Lập trình Java

Tuần 2: Các cấu trúc điều kiện, mảng và lệnh lặp

Nội dung

- Cấu trúc điều kiện
- Cấu trúc lệnh lặp
- Mảng

Cấu trúc điều kiện

Toán tử điều kiện

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

Ví dụ: tìm số lớn nhất của 2 số a và b

`Int max = a > b ? a : b`

Cấu trúc điều kiện if

Ví dụ:

```
If(diem>=5){  
System.out.print("đậu");  
}
```

```
if(< <điều kiện> >)  
{  
    < < Công việc > >  
}
```

Hoặc

```
String ketqua = diem>=5?"đậu":"rớt";  
System.out.println(ketqua);
```

Cấu trúc điều kiện if...else

Ví dụ:

```
If(diem>=5){  
System.out.print("đậu");  
} else {  
System.out.print("rớt");  
}
```

```
if (<<điều kiện>>)  
{  
    << công việc 1 >>  
}  
else  
{  
    << công việc 2 >>  
}
```

Cấu trúc điều kiện nhiều lệnh if

Cú pháp:

```
if (<<điều kiện 1>>){  
    << công việc 1 >>  
}  
else if (<<điều kiện 2>>){  
    << công việc 2 >>  
}  
...  
else {  
    << công việc N+1 >>  
}
```

Cấu trúc điều kiện switch...case

Cú pháp:

```
switch (<<biểu thức>>)  
{  
    case <<giá trị 1>>:  
        // Công việc 1  
        break;  
    case <<giá trị 2>>:  
        // Công việc 2  
        break;  
    ...  
    default:  
        // Công việc N+1  
        break;  
}
```

Cấu trúc lệnh lặp while

Cú pháp

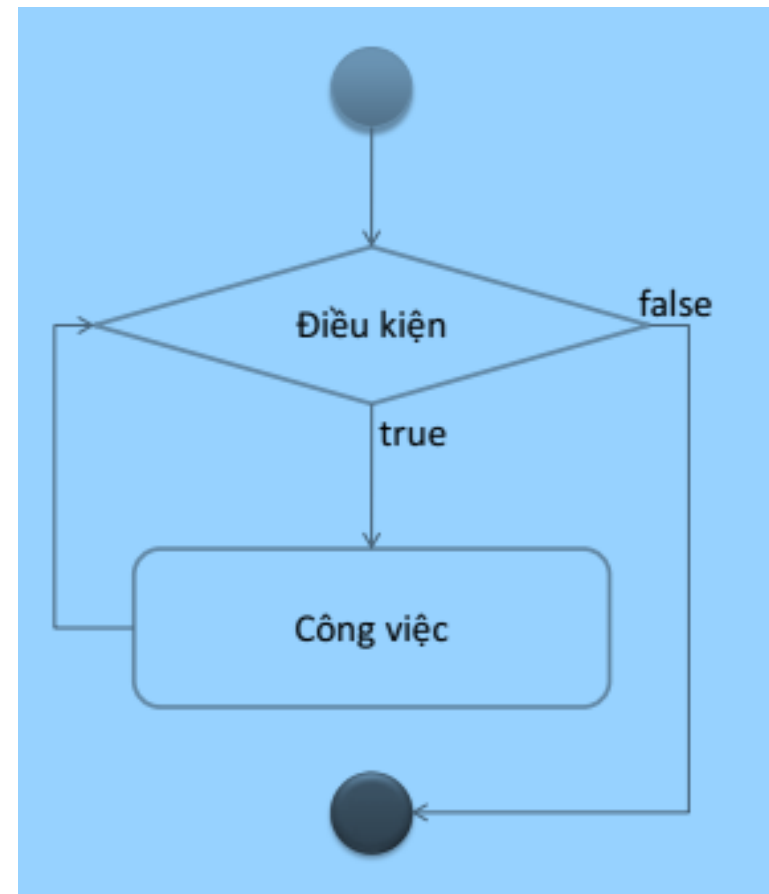
```
while (<<điều kiện>>) {  
    // công việc  
}
```

Ví dụ:

Viết chương trình có các lựa chọn:

1: xuất ra bảng cửu chương bất kì.

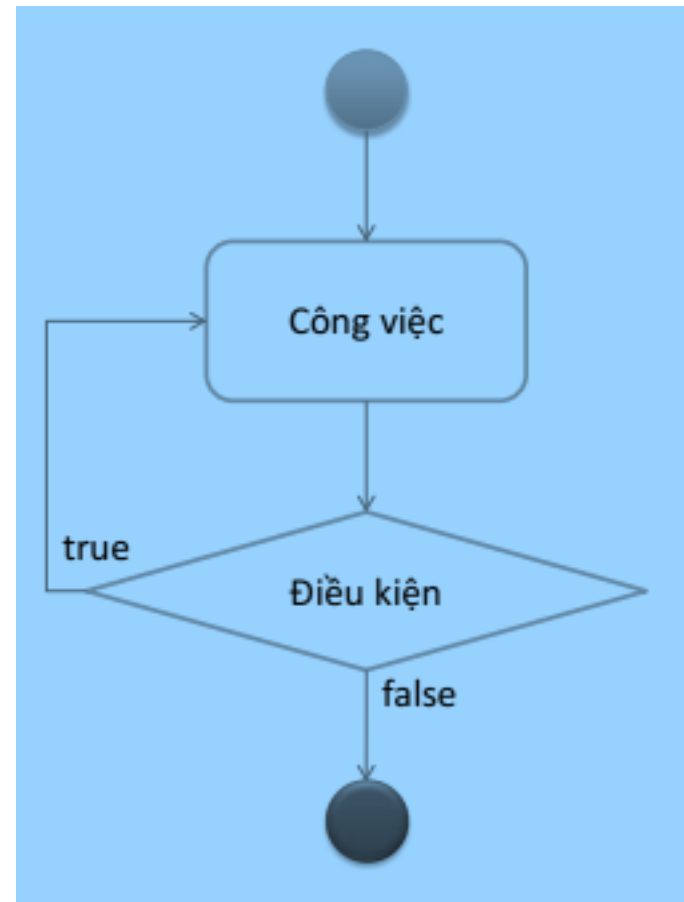
2: tính trung bình cộng các số chia hết cho 3 từ 27 đến 250



Cấu trúc lệnh lặp do...while

Cú pháp

```
do {  
    // công việc  
}  
while (<<điều kiện>>);
```



Cấu trúc lệnh lặp do...while

Câu hỏi:

```
int a=6;
```

```
int b=6;
```

```
System.out.println(a++);
```

```
System.out.println(++b);
```



Output?

Cấu trúc lệnh lặp do...while

Câu hỏi:

```
int a=6;
```

```
int b=6;
```

```
System.out.println(a++);
```

```
System.out.println(++b);
```

```
System.out.println(a++);
```

```
System.out.println(++b);
```

`++i`: thực hiện tăng biến `i` lên 1 đơn vị trước.

`i++`: thực hiện biểu thức trước sau đó cộng `i` lên 1 đơn vị.



Cấu trúc lệnh lặp for

Cú pháp

```
for (khởi đầu ; điều kiện; bước nhảy){  
    // công việc  
}
```

Ví dụ

```
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
}
```

Cấu trúc lệnh lặp for

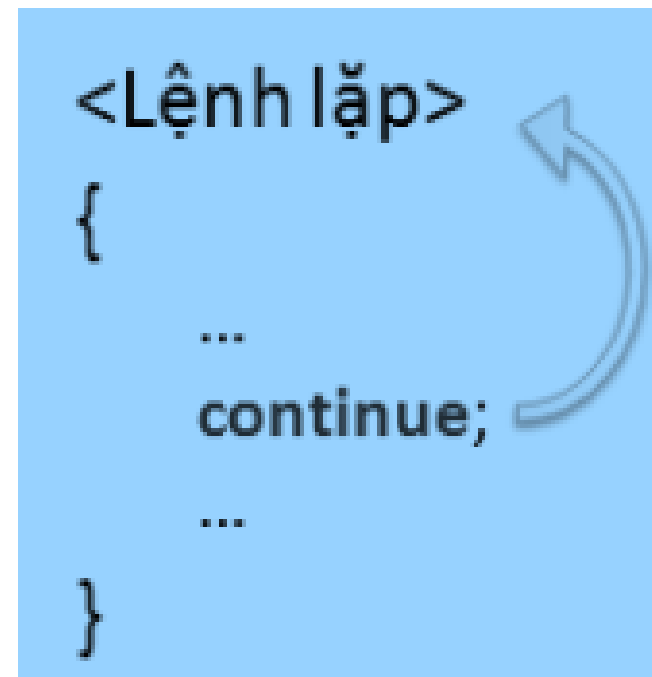
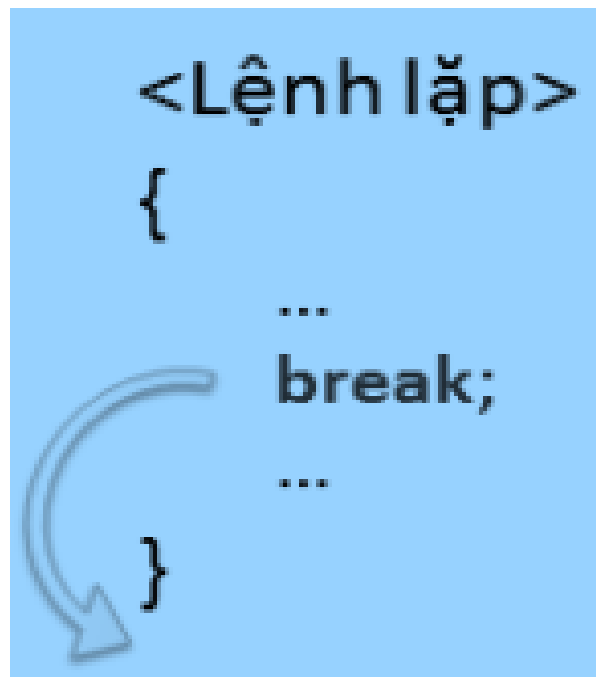
Cú pháp duyệt mảng foreach

```
for (Kiểu dữ liệu: tên mảng) {  
    // câu lệnh thực thi  
}
```

Ví dụ

```
String[] arr = {"A", "B", "C", "D"};  
for (String i : arr) {  
    System.out.println(i);    }
```

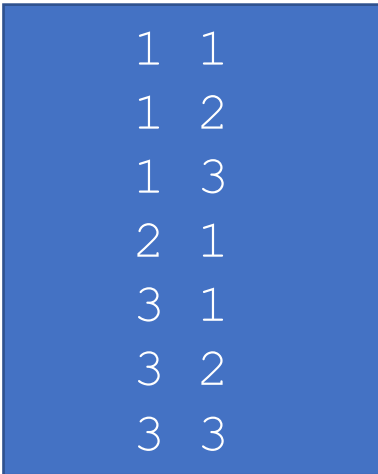
Lệnh ngắt **break** & **continue**



Lệnh ngắt break & continue

Ví dụ

```
for (int i = 1; i <= 3; i++) {  
    for (int j = 1; j <= 3; j++) {  
        if (i == 2 && j == 2) {  
            break;  
        }  
        System.out.println(i + " " + j);  
    }  
}
```



1	1
1	2
1	3
2	1
3	1
3	2
3	3

Nếu break nằm trong vòng lặp khác thì nó chỉ stop vòng lặp bên trong.

Lệnh ngắt break & continue

Ví dụ

```
for (int i = 1; i <= 10; i++) {  
    if (i == 5) {  
        continue;  
    }  
    System.out.println(i);  
}
```

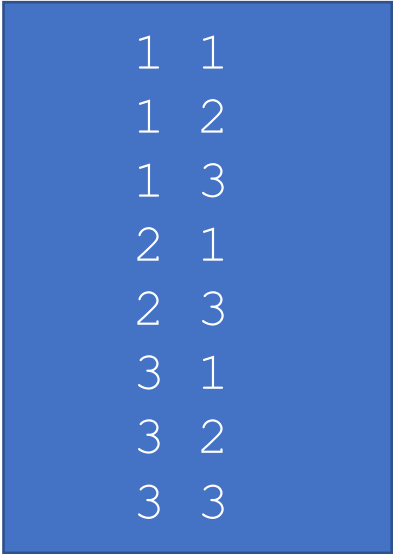


1
2
3
4
6
7
8
9
10

Lệnh ngắt break & continue

Ví dụ

```
for (int i = 1; i <= 3; i++) {  
    for (int j = 1; j <= 3; j++) {  
        if (i == 2 && j == 2) {  
            continue;  
        }  
        System.out.println(i + " " + j);  
    }  
}
```



1	1
1	2
1	3
2	1
2	3
3	1
3	2
3	3

Nếu continue nằm trong vòng lặp khác thì nó chỉ có tác dụng với vòng lặp bên trong.

Mảng

Mảng là cấu trúc lưu trữ nhiều phần tử có cùng kiểu dữ liệu.

Trong java chỉ số (index) được đánh dấu từ 0.

Một số thao tác với mảng:

- Khai báo
- Truy xuất phần tử
- Lấy số phần tử
- Duyệt mảng
- Sắp xếp các phần tử mảng

0	1	2	3	4	5	6	7	8	← Indices
5	7	9	1	45	1	9	9	2	← Elements

Khai báo mảng

Khai báo không khởi tạo

```
int[] a; // mảng số nguyên chưa biết số phần tử  
int b[]; // mảng số nguyên chưa biết số phần tử  
String[] c = new String[5]; // mảng chứa 5 chuỗi
```

Khai báo có khởi tạo

```
double[] d1 = new double[]{2, 3, 4, 5, 6}; // mảng số  
thực, 5 phần tử, đã được khởi tạo  
double[] d2 = {2, 3, 4, 5, 6}; // mảng số thực, 5 phần  
tử, đã được khởi tạo
```

Truy xuất các phần tử

```
int a[] = {1,2,3,4}
```

```
a[2] = a[1]*4
```

Sau phép gán trên thì mảng thu được sẽ là 1,2,8,4.

Sử dụng thuộc tính **length** để lấy số phần tử của mảng `a.length` sẽ cho ra giá trị là 4.

Duyệt mảng

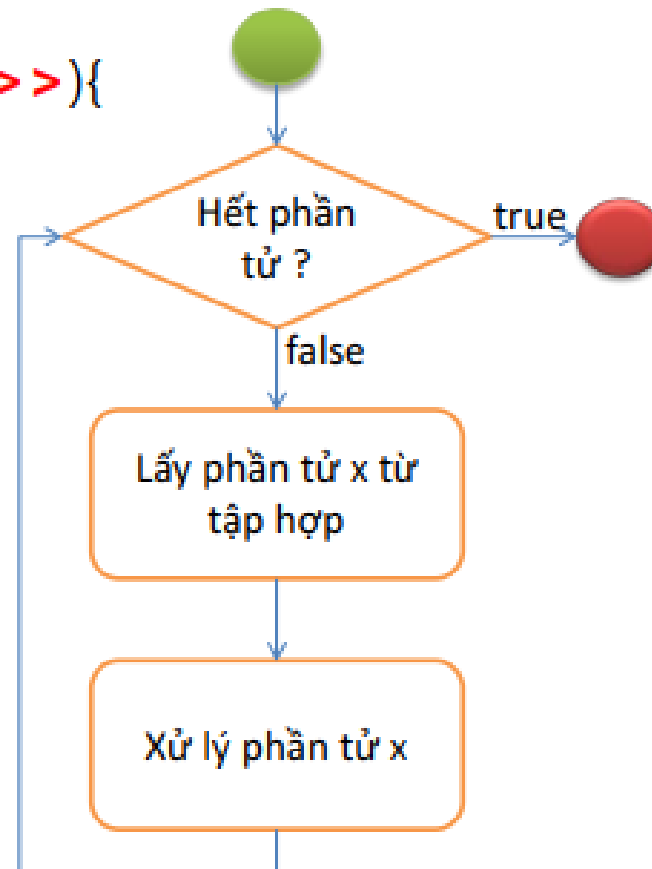
Duyệt mảng với vòng lặp for each và for

Cú pháp

```
for (<<kiểu>> x : <<tập hợp>>){  
    // Xử lý phần tử x  
}
```

Diễn giải:

- ❖ For each được sử dụng để duyệt tập hợp. Mỗi lần lấy 1 phần tử từ tập hợp và xử lý phần tử đó.



Duyệt mảng

```
int[] a = {4, 3, 5, 9};  
for(int i=0; i<a.length; i++){  
    System.out.println(a[i]);  
}
```

← **for(;;)**

for-each →

```
int[] a = {4, 3, 5, 9};  
for (int x : a){  
    System.out.println(x);  
}
```

Thao tác mảng

<code>void sort(Object[] a)</code>	Sắp xếp các phần tử theo thứ tự tăng dần. Ví dụ: Arrays.sort(a);
<code>String toString(Object[] a)</code>	Chuyển mảng thành chuỗi được bọc giữ cặp dấu [] và các phần tử mảng cách nhau dấu phẩy. Ví dụ: String s = Arrays.toString(a);
<code>void fill(Object[] a, Object val)</code>	Gán 1 giá trị cho tất cả các phần tử mảng. Ví dụ: Arrays.fill(a, 9);

Bài tập tuần 2

Nhập vào mảng số nguyên gồm n phần tử, in ra hoán vị của mảng đó.

Ví dụ:

Nhập mảng: 1 2 3

In ra:

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1