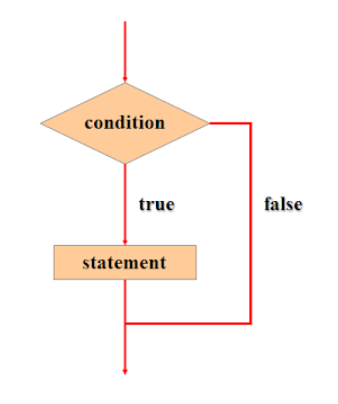
**If – else**

**Dạng 1:**

if (expression) {

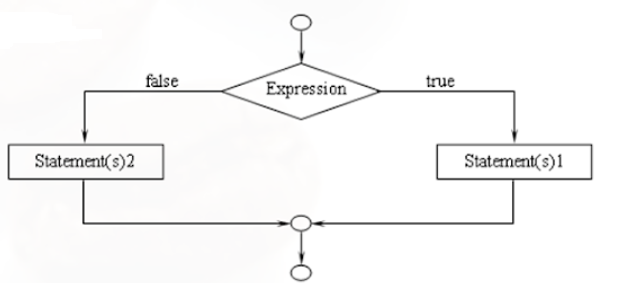
statement(s);

}

expression: là biểu thức điều kiện

statement(s): là các câu lệnh, chúng sẽ được thực thi nếu biểu thức điều kiện trả về đúng <-> true

**Dạng 2:**

 if (expression) {

statement(s)1;

}

else{

statement(s)2;

}

Nếu biểu thức điều kiện đúng, sẽ thực hiện khối lệnh thứ nhất, ngược lại sẽ thực hiện khối lệnh thứ 2

**Switch… case…**

**switch** (integer expression) {

**case** integer expression:

statement(s)

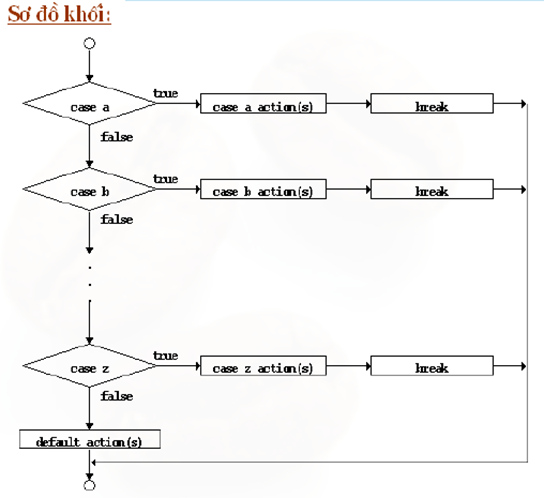
**break;**

…

**default:**

statement(s);

**break;**



**For**

**for** (initialization; loopContinuationTest; increment) {

**Statements**;

}

Initialization: biểu thức điều kiện

loopContinuationTest: điều kiện tăng giảm

increment: tăng hoặc giảm biểu thức

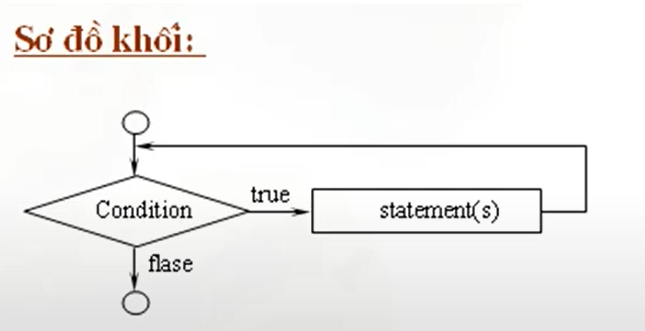


**While**

**while** (Condition){

Statement(s)

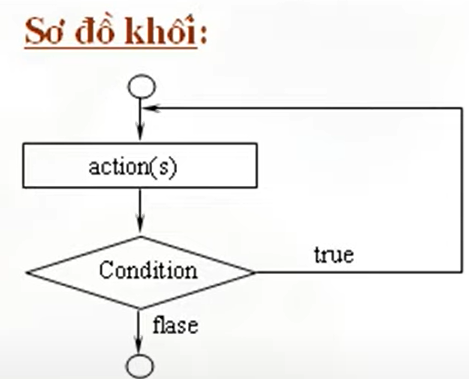
}



Thoát vòng lặp nếu không đáp ứng điều kiện

**Phân biệt vòng lặp for và vòng lặp while**

* Vòng lặp **for** dùng cho những mảng hoặc giá trị xác định được điểm dừng (đi từ điểm A đến điểm B xác định)
* Vòng lặp **while** dùng khi không biết khi nào sẽ xảy ra tình huống dừng lại

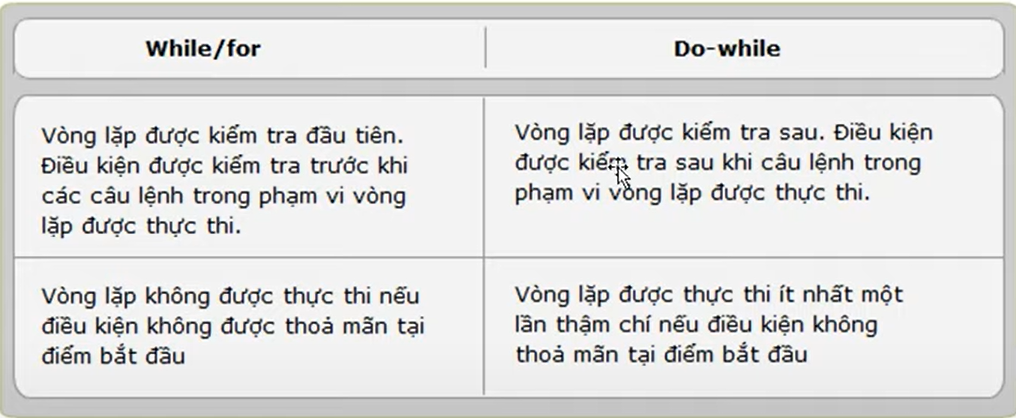
**Do… While**

**do** (Condition){

Statement(s)

} **while** (Condition);

Do… while sẽ thực hiện ít nhất 1 lần rồi mới xét điều kiện đúng hay sai để thoát hoặc chạy tiếp



**Break, continue, return**

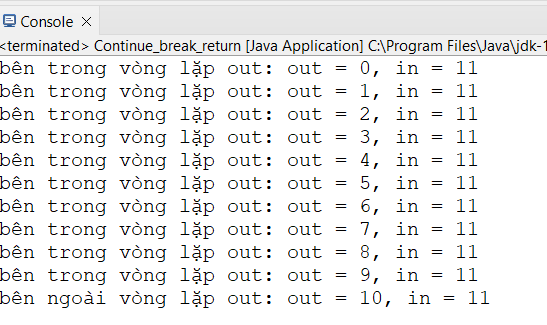
- Thường dùng cho vòng lặp for

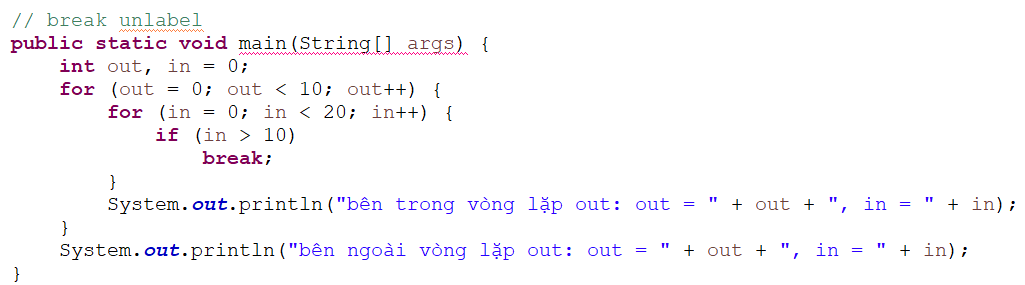
- **continue** và **break** có thể dùng cùng với một **label** hoặc không

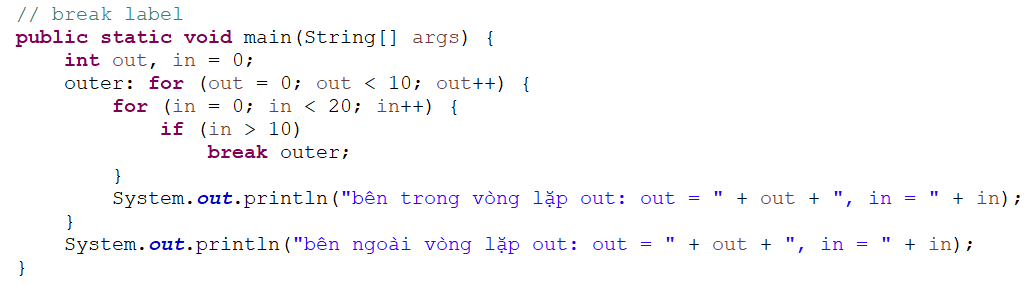
**Label**: Dùng để chỉ định chỗ cho các lệnh **continue** và **break**

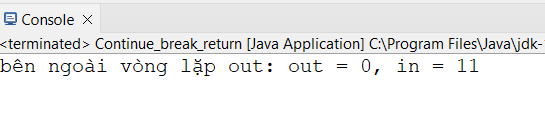
**Lệnh break**

Lệnh break có 2 dạng:

- unlabeled form: dùng để thoát khỏi vòng lặp như switch, for, while or do-while



- labeled form: di chuyển tới vùng đặt label

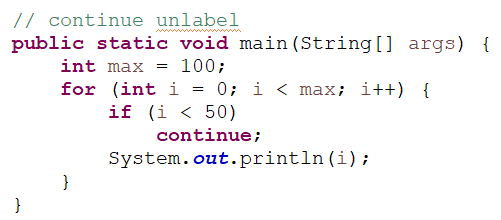
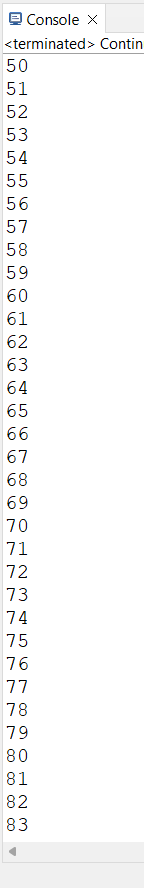


**Lệnh continue**

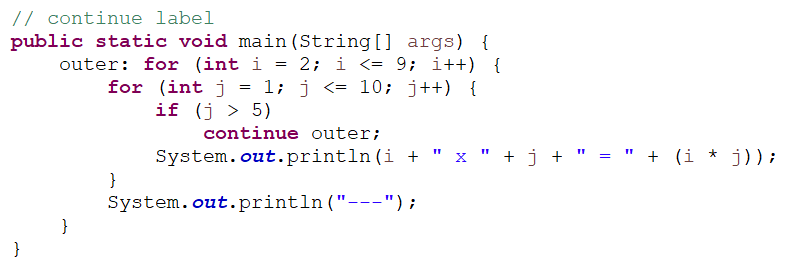
Dùng để nhảy qua (skip) vòng lặp (for, while, do-while)

Lệnh continue có 2 dạng:

- unlabeled form



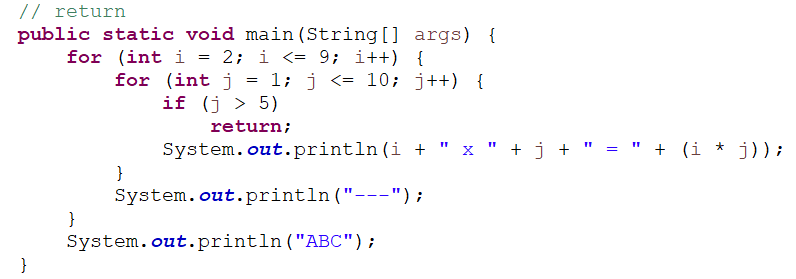
- labeled form

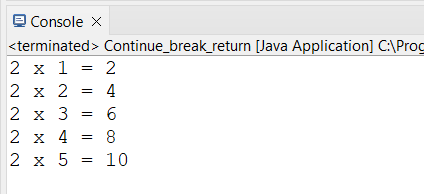




**Lệnh return**

* Lệnh return có 2 dạng:
  + Trả về 1 giá trị:
    - return ++count;
    - Giá trị trả về phải phù hợp với kiểu trả về của hàm
  + Không trả về giá trị nào cả:
    - return;
    - Kiểu trả về của hàm phải là void

****

****