



LỆNH ĐIỀU KHIỂN

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



NỘI DUNG

- Lệnh ghi chú
- Lệnh rẽ nhánh (if)
- Lệnh lựa chọn (switch)
- Lặp
 - ✱ While
 - ✱ Do...while
 - ✱ For
- Ngắt





LỆNH GHI CHÚ

- Ghi chú (comment) là lệnh đặc biệt, không ảnh hưởng mã nguồn khi chạy, chỉ chú thích mã

- Có 3 dạng comment

★ //

- Ghi chú mã trên 1 hàng

★ /* ... */

- Ghi chú mã trên nhiều hàng

★ /** ... */

- Diễn giải cách sử dụng của phương thức, trường hay lớp

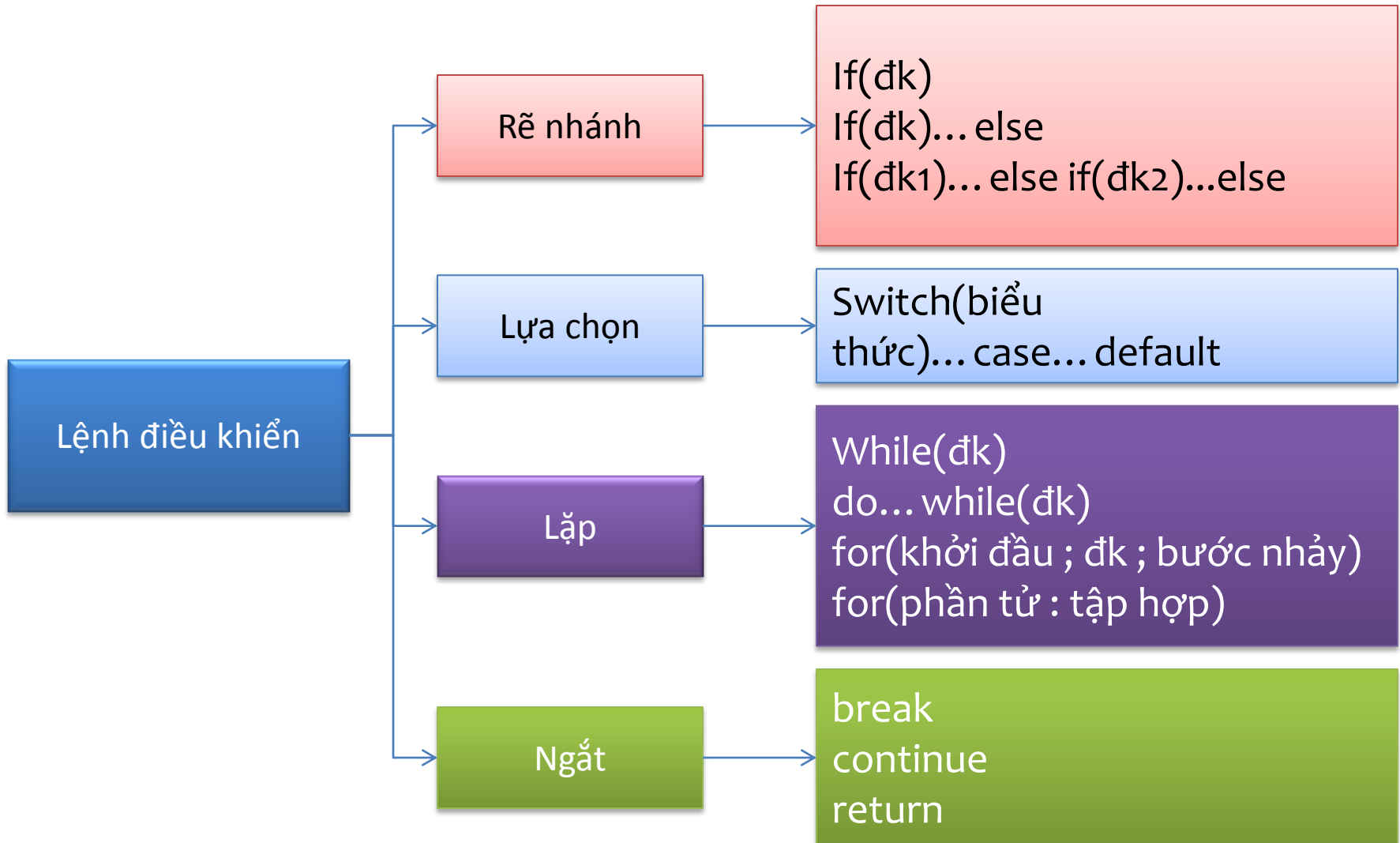


LỆNH GHI CHÚ

```
/**
 * Đọc số nguyên từ bàn phím
 * @param thôngBao thông báo yêu cầu nhập
 * @return số nguyên nhập từ bàn phím
 */
public static int readInt(String thôngBao) {
    /*
     * Gọi phương thức readString() để
     * đọc chuỗi từ bàn phím. Phương thức
     * này đã được xây dựng trước đó
     */
    String chuoai = Keyboard.readString(thôngBao);
    // Chuyển chuỗi đọc được sang số nguyên
    Integer soNguyen = Integer.parseInt(chuoai);
    return soNguyen; // trả về số nguyên đọc từ bàn phím
}
```



LỆNH ĐIỀU KHIỂN





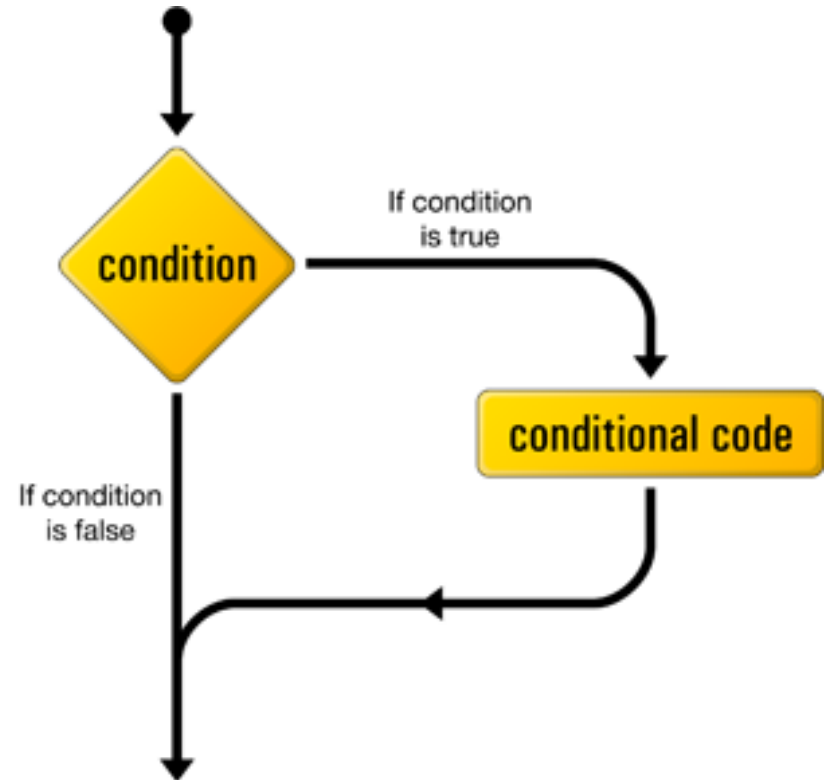
LỆNH IF

- Cú pháp

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

- Diễn giải:

- ✱ Công việc được thực hiện khi và chỉ khi biểu thức **<điều kiện>** có giá trị **true**.





VÍ DỤ LỆNH IF

- Ví dụ: ví dụ sau xuất những gì ra màn hình

```
double diem = 4;
```

```
if (diem >= 5) {
```

```
    System.out.println("Đậu");
```

```
}
```

- Diễn giải:

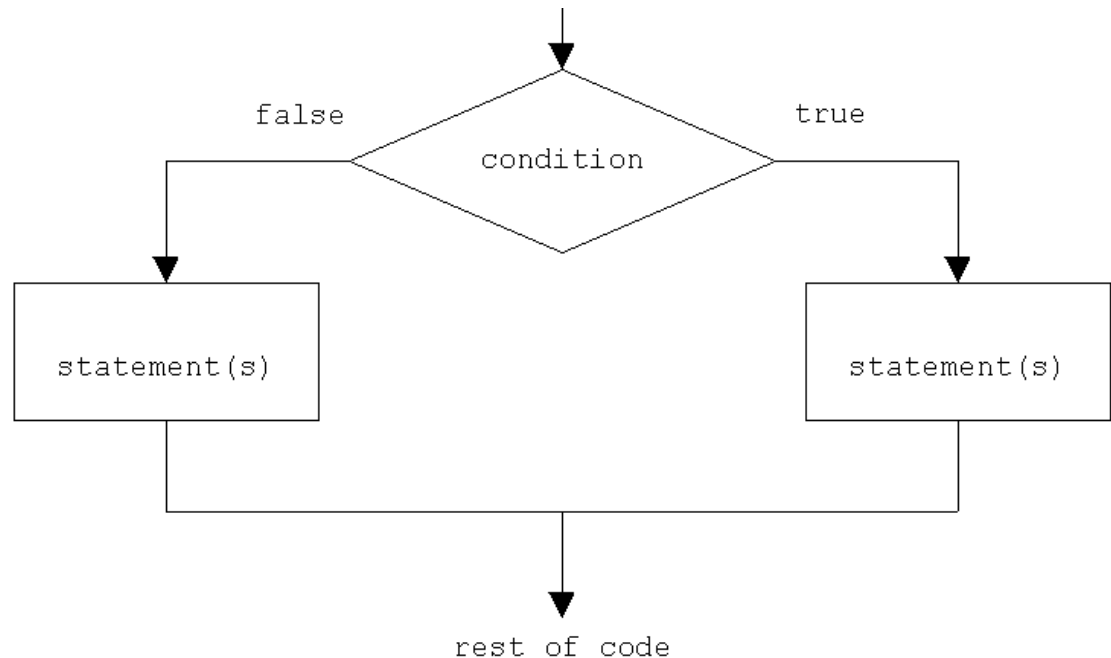
✱ Không xuất gì ra màn hình cả vì biểu thức điều kiện **diem >= 5** có giá trị **false**.



LỆNH IF...ELSE

● Cú pháp

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



● Diễn giải

- ✳ Nếu <điều kiện> có giá trị true thì thực hiện <công việc 1> ngược lại thực hiện <công việc 2>



VÍ DỤ LỆNH IF...ELSE

- Ví dụ

```
double diem = 6;
```

```
if (diem < 5) {
```

```
    System.out.println("Rớt");
```

```
}
```

```
else {
```

```
    System.out.println("Đậu");
```

```
}
```

- Diễn giải

✳ Ví dụ trên xuất chữ “Đậu” ra màn hình vì điều kiện **<diem < 5>** có giá trị là **false**.



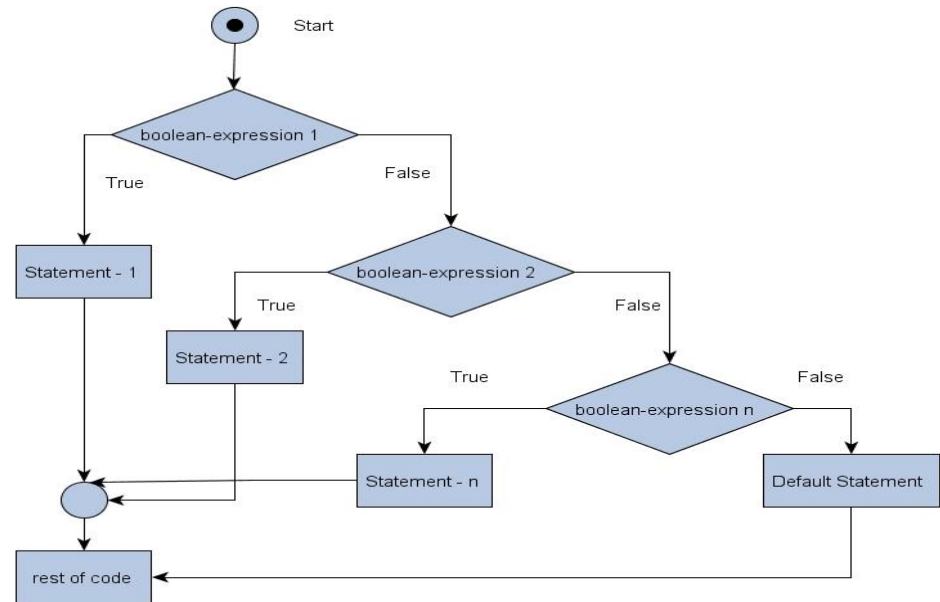
NHIỀU LỆNH IF

Cú pháp

```
if (ĐK1) {  
    công việc 1;  
}  
else if (ĐK2) {  
    công việc 2;  
}  
...  
else {  
    công việc N+1  
}
```

Diễn giải

- ✳ Nếu ĐK<i> có giá trị true thì thực hiện <công việc i> ngược lại thực hiện <công việc n+1>





VÍ DỤ NHIỀU LỆNH IF

- Ví dụ

```
if(delta < 0) {  
    System.out.println("Vô nghiệm");  
}  
else if(delta == 0) {  
    System.out.println("Nghiệm kép");  
}  
else {  
    System.out.println("2 nghiệm");  
}  
...
```

- Diễn giải

- ✳ Biện luận phương trình bậc 2



TÍNH THUẾ THU NHẬP



- Thu nhập gồm lương và thưởng. Hãy tính thuế thu nhập theo phương pháp lũy tiến như sau:
 - ✱ Thu nhập dưới 9 triệu: không đóng thuế
 - ✱ Thu nhập từ 9 đến 15: thuế 10%
 - ✱ Thu nhập trên 15 triệu: thuế 15%



LỆNH SWITCH

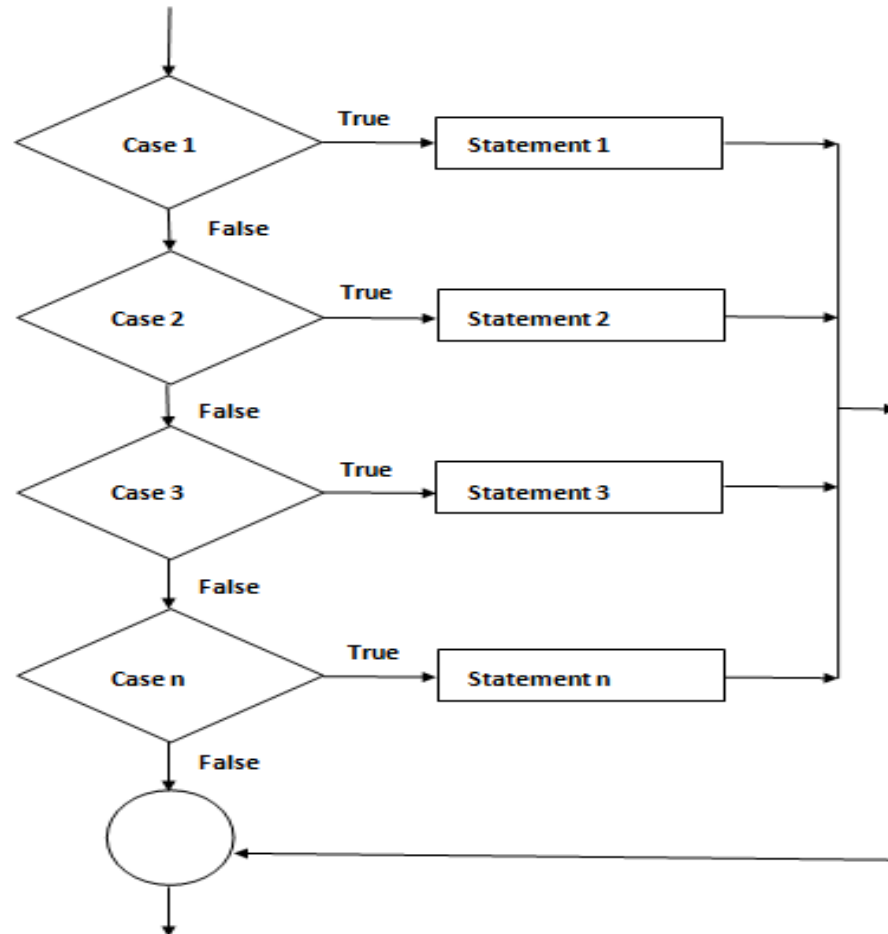
Cú pháp

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;  
}
```

Diễn giải

- ✱ Nếu giá trị <biểu thức> bằng <giá trị i> thì <công việc i> được thực hiện, ngược lại <công việc n+1> được thực hiện





VÍ DỤ LỆNH SWITCH

```
double toanHang1 = 5;
double toanHang2 = 9;
char toanTu = '+';

switch (toanTu) {
case '+':
    double tong = toanHang1 + toanHang2;
    System.out.printf("Tổng: %f + %f = %f", tong);
    break;
case '-':
    double hieu = toanHang1 - toanHang2;
    System.out.printf("Hiệu: %f - %f = %f", hieu);
    break;
default:
    System.out.printf("Không hỗ trợ phép %s", toanTu);
    break;
}
```



VÍ DỤ LỆNH SWITCH

- Tìm số ngày của tháng

```
int year = 2000;
int month = 6;

switch(month){
case 2:
    if(year % 4 == 0 && year % 100 != 0){
        System.out.println(29);
    }
    else{
        System.out.println(28);
    }
    break;
case 4:
case 6:
case 9:
case 11:
    System.out.println(30);
    break;
default:
    System.out.println(31);
    break;
}
```



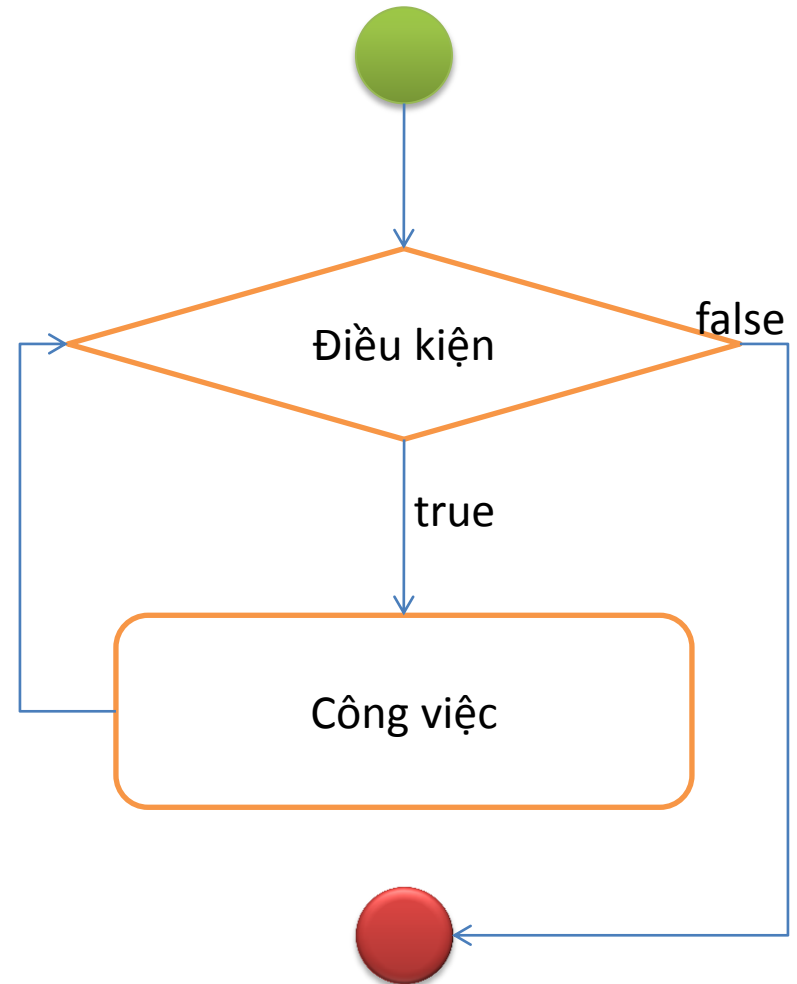
LỆNH LẶP WHILE

- Cú pháp

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

- Diễn giải

✱ Thực hiện công việc trong khi điều kiện còn thỏa mãn (điều kiện có giá trị true).





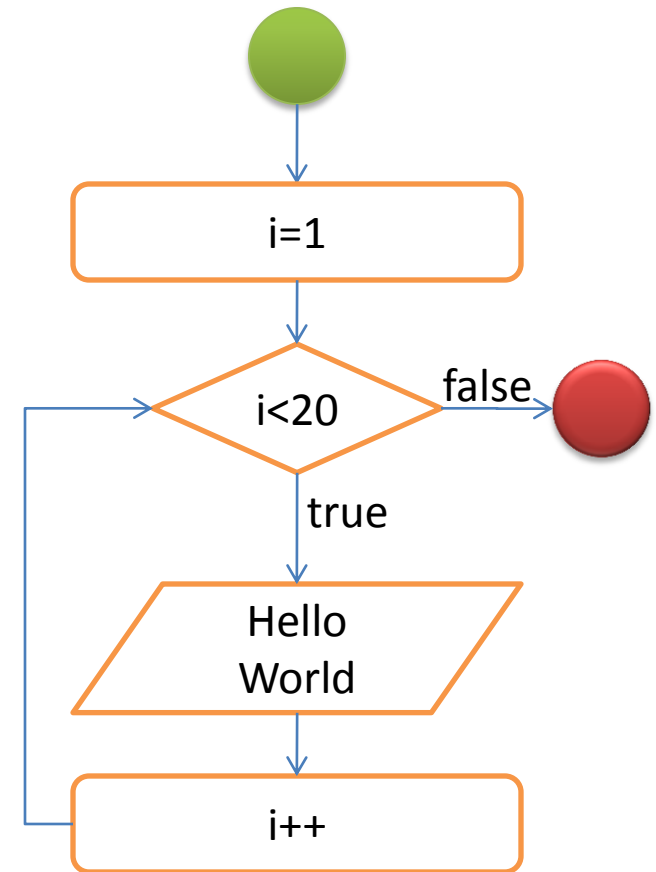
LỆNH LẶP WHILE

● Ví dụ

```
int i = 1;  
while (i < 20) {  
    System.out.println("Hello World !");  
    i++;  
}
```

● Diễn giải

✱ sau sẽ xuất 19 dòng Hello World ra màn hình





- ✓ Xuất bảng cửu chương
- ✓ Tính trung bình cộng các số chia hết cho 3 từ 27 đến 232.



LỆNH LẶP DO...WHILE

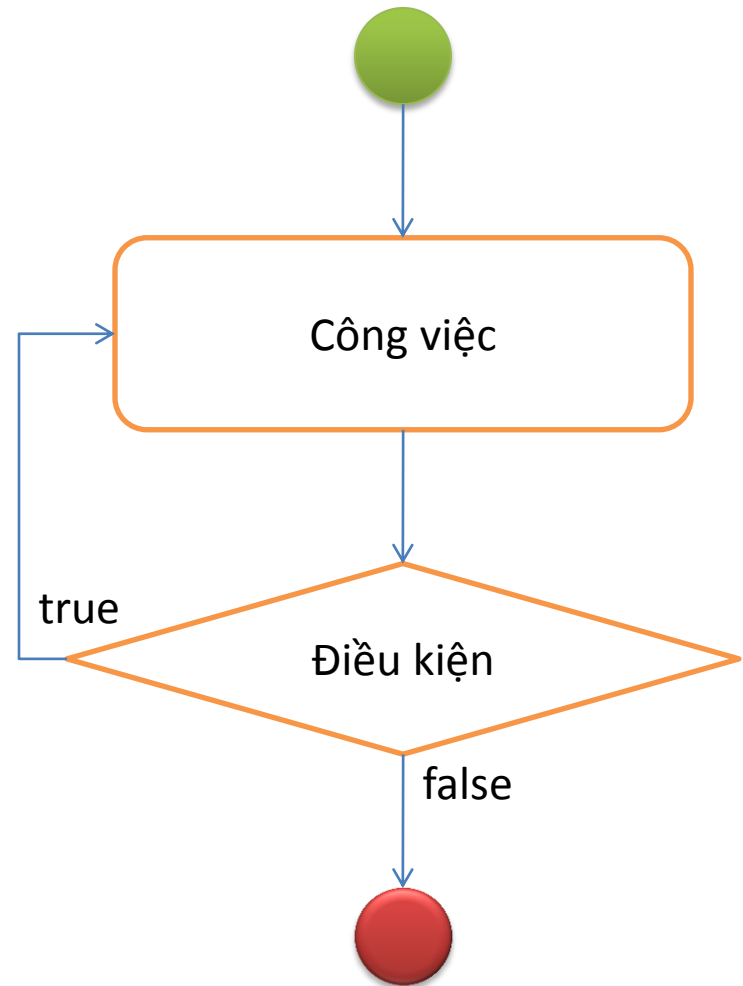
- Cú pháp

```
do {  
    // công việc  
}
```

```
while (điều kiện);
```

- Diễn giải

✱ Tương tự lệnh lặp while chỉ khác ở chỗ điều kiện được kiểm tra sau nghĩa là công việc được thực hiện ít nhất 1 lần.





LỆNH LẶP DO...WHILE

● Ví dụ

```
int so = -1;
```

```
do {
```

```
    so = Keyboard.readInt("Nhập số nguyên dương: ");
```

```
}
```

```
while (so < 0);
```

● Diễn giải

- ✳ Ví dụ trên cho phép nhập số nguyên dương từ bàn phím.



✓ **Nhập điểm từ 0 đến 10**



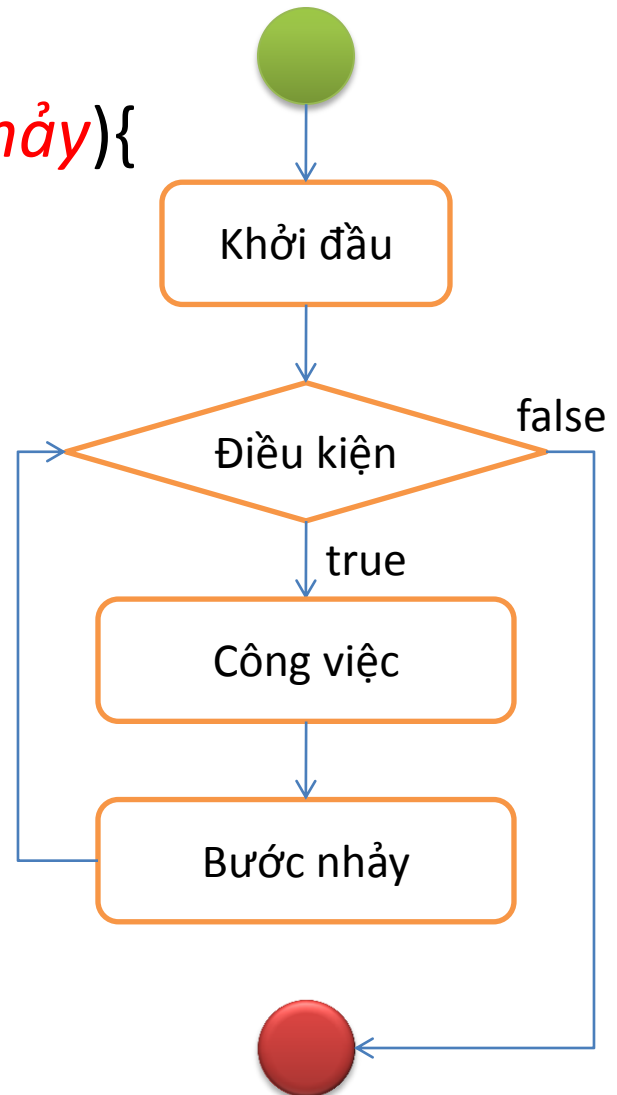
LỆNH LẶP FOR

● Cú pháp

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

● Diễn giải

- ✱ B1: Thực hiện <khởi đầu>
- ✱ B2: Kiểm tra <điều kiện>
 - Nếu là false thì kết thúc vòng lặp
- ✱ B3: Thực hiện <công việc>
- ✱ B4: Thực hiện <bước nhảy>
- ✱ B5: Trở lại bước 2





✓ **Bảng cửu chương với for**



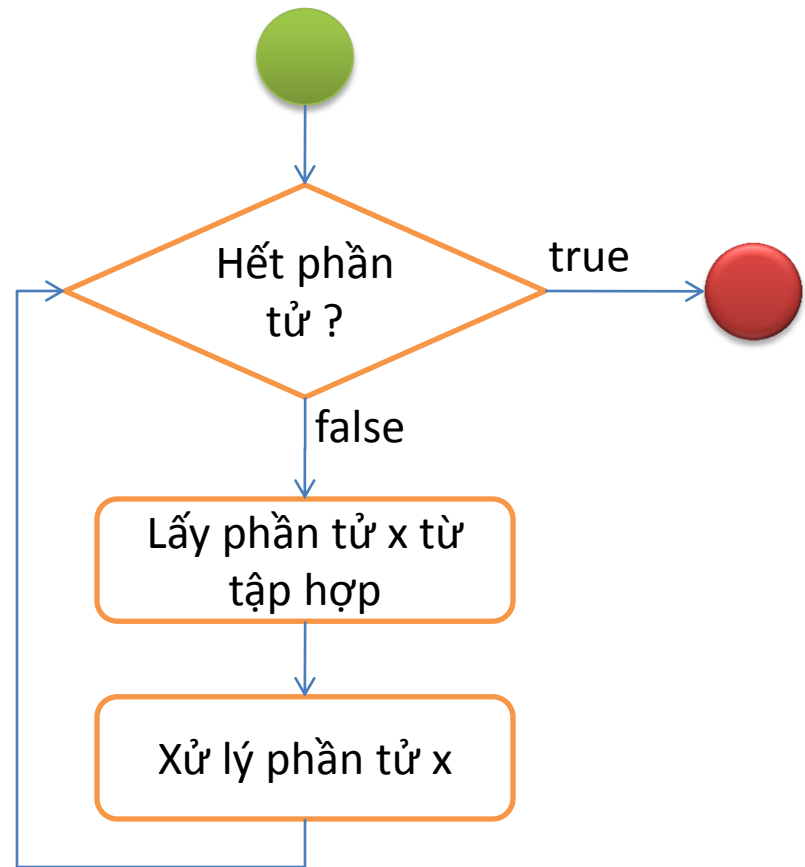
FOR EACH

● 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.





VÍ DỤ FOR VÀ FOR-EACH

```
String[] dssv = {"Tuấn", "Hoa", "Hạnh", "Phương"};
```

```
for(int i=0;i<dssv.length;i++){  
    String x = dssv[i];  
    System.out.println(x);  
}
```

Mảng chứa danh
sách sinh viên

Duyệt với for
bình thường

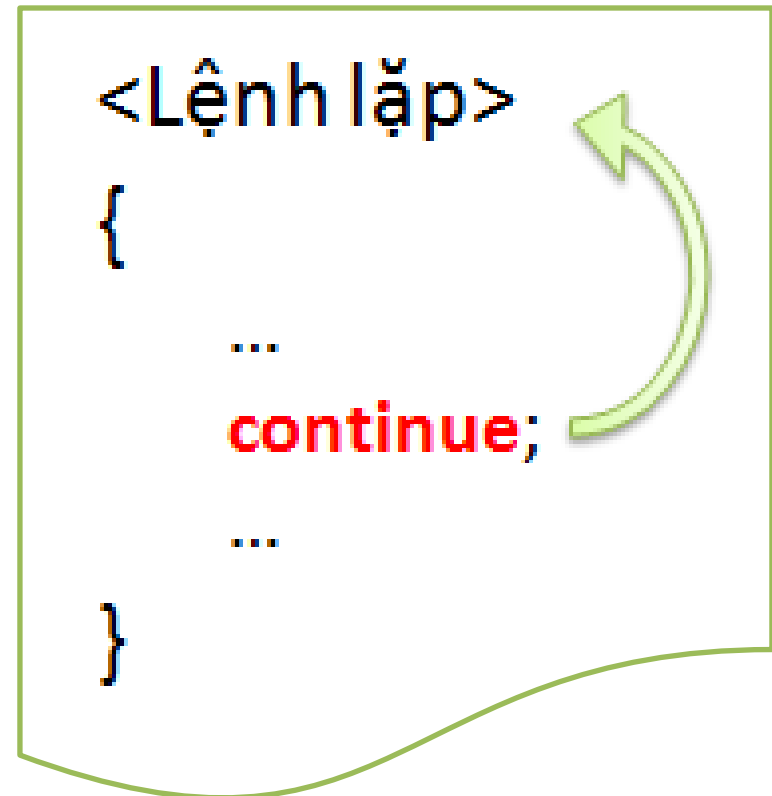
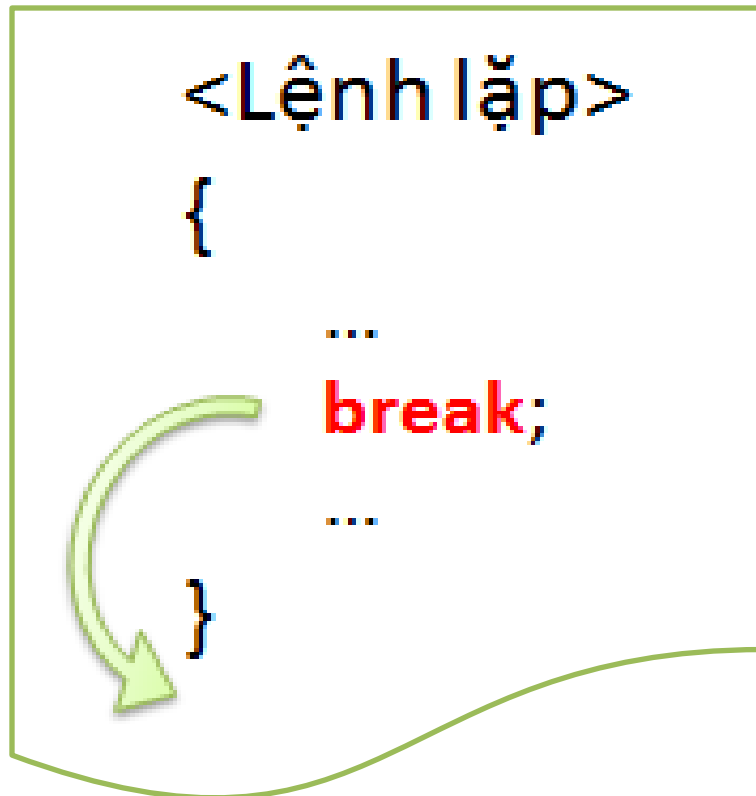
```
for(String x : dssv){  
    System.out.println(x);  
}
```

Duyệt với for-each



LỆNH BREAK & CONTINUE

- **Break** dùng để ngắt lệnh lặp
- **Continue** dùng để thực hiện lần lặp tiếp theo ngay lập tức





VÍ DỤ BREAK

- Nhập điểm hợp lệ (từ 0 đến 10)

```
int diem = 0;
while(true){
    diem = Keyboard.readInt("Điểm: ");
    if(diem >= 0 && diem <= 10){
        break;
    }
    System.out.println("Vui lòng nhập điểm hợp lệ !");
}
```



VÍ DỤ: TRUNG BÌNH CỘNG

- Tính trung bình cộng các số chia hết cho 3 từ 27 đến 232

```
double tong = 0, dem = 0;
for(int i=27;i<=232;i++){
    if(i % 3 != 0){
        continue;
    }
    tong += i;
    diem++;
}
double tb = tong/diem;
```



LỆNH RETURN

- Lệnh return dùng để kết thúc hoặc trả về kết quả của phương thức.
 - ✱ Nếu kiểu trả về của phương thức là void thì return chỉ có ý nghĩa kết thúc phương thức.
 - ✱ Ngược lại return sẽ trả về kết quả của phương thức



KẾT THÚC HÀM

```
public static void menu() {  
    while(true){  
        System.out.println("1. Nhập");  
        System.out.println("2. Xuất");  
        System.out.println("3. Kết thúc");  
        int so = Keyboard.readInt("Chọn chức năng?");  
        switch (so) {  
            case 1:  
                nhap();  
                break;  
            case 2:  
                xuat();  
                break;  
            case 3:  
                return;  
            default:  
                System.out.println("Vui lòng nhập 1, 2 hoặc 3");  
                break;  
        }  
    }  
}
```

Kết thúc phương thức
(không trả về kết quả)



TRẢ KẾT QUẢ

```
public static int readInt(String thôngBao) {  
    String chuoì = Keyboard.readString(thôngBao);  
    int soNguyên = Integer.parseInt(chuoì);  
    return soNguyên;  
}
```

Trả kết quả về



LỆNH TRY...CATCH

- Cấu trúc lệnh

```
try{
```

```
    // Khối lệnh có khả năng xảy ra ngoại lệ
```

```
}
```

```
catch(Exception ex){
```

```
    // Xử lý ngoại lệ
```

```
}
```




LỆNH TRY...CATCH

- Ví dụ: chuyển chuỗi sang số

```
try{  
    int a = Integer.parseInt(s);  
}  
catch(Exception ex){  
    System.out.println("Không chuyển số");  
}
```



TÓM TẮT

- Comments
- If, if...else, if...else if...else
- Switch...case
- Loop
 - ✱ While
 - ✱ Do...while
 - ✱ For(;điều kiện;)
 - ✱ For(phần tử: tập hợp)
- Ngắt
 - ✱ Break
 - ✱ Continue
 - ✱ return