

# Bài 16: Lệnh break, continue

---

- ✓ Vòng lặp vô hạn
- ✓ Lệnh break
- ✓ Lệnh continue
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# Vòng lặp vô hạn

- Vòng lặp mà lặp vô tận không có điểm dừng gọi là vòng lặp vô hạn
- Vòng lặp vô hạn xảy ra khi điều kiện lặp luôn đúng
- Hai mặt của vòng lặp vô hạn:
  - Mặt lợi: cho phép ta chạy chương trình đến vô thời hạn. Do đó ta có cơ hội giám sát được thời điểm kết thúc của vòng lặp dựa trên dữ liệu nào đó
  - Mặt hại: chiếm CPU, RAM, các thành phần phần cứng liên quan->thiếu tài nguyên để phục vụ chương trình khác
- Các vòng lặp vô hạn thường sử dụng kèm với lệnh break để kết thúc lặp khi điều kiện nào đó xảy ra
- Thường sử dụng để kiểm soát dữ liệu đầu vào hoặc chờ đợi 1 sự kiện nào đó

# Ví dụ về vòng lặp vô hạn

➤ Sau đây là yêu cầu: Nhập vào các giá trị số nguyên từ bàn phím, việc nhập vào sẽ dừng khi gặp giá trị 0.

➤ Đoạn chương trình:

```
var input = new Scanner(System.in);
var n = 0;
while (true) {
    System.out.println("Nhập 0 để kết thúc. Nhập khác 0 để tiếp tục");
    n = input.nextInt();
    // ... Làm gì đó ở đây
    if(n == 0) { // nếu n == 0
        break; // thoát khỏi vòng Lặp
    }
}
```

# Lệnh break

- Khi cần chủ động thoát khỏi một cấu trúc lặp, ra quyết định như switch, ta dùng break
- Lệnh break rất hữu ích trong trường hợp muốn bỏ qua các lần lặp thừa, các đánh giá thừa vì ta đã đạt được mục đích
- Lệnh break sẽ thoát khỏi cấu trúc switch hoặc vòng lặp gần nhất chứa nó, không thoát hoàn toàn khỏi chương trình
- Khi gặp break, các lệnh phía dưới và cùng khối với nó sẽ không được thực hiện nữa

# Ví dụ về break

## ➤ Sử dụng break trong switch:

```
var choice = input.nextInt(); // nhập lựa chọn
// tiến hành đánh giá và thực hiện chương trình
switch (choice) {
    case 0:
        System.out.println("Xin chào và hẹn gặp lại!");
        break;
    case 1: // nếu giá trị của choice = 1
        var sum = a + b; // tính tổng hai số
        System.out.println(a + " + " + b + " = " + sum);
        break;
```

- Trong mỗi case độc lập của switch có thể chứa break để ngắt và thoát chương trình khỏi cấu trúc switch khi đã thực hiện xong nhiệm vụ

# Ví dụ về break

➤ Tìm giá trị đầu tiên chia hết cho k trong [a, b]:

```
var input = new Scanner(System.in);
System.out.println("Nhập số nguyên a < b và giá trị k: ");
var a = input.nextInt();
var b = input.nextInt();
var k = input.nextInt();
if(k != 0) {
    for (int i = a; i <= b; i++) {
        if(i % k == 0) {
            System.out.println("giá trị đầu tiên thỏa: " + i);
            break; // kết thúc vòng lặp
        }
    }
    // làm gì đó tiếp ở đây
} else {
    System.out.println("Nhập k != 0.");
}
```

# Lệnh continue

- Khác với break, chương trình không thoát khỏi một khối nào khi gặp continue
- Continue cho phép logic chương trình bỏ qua các lệnh phía dưới và cùng trong thân vòng lặp với nó
- Lệnh continue thường kết hợp với các cấu trúc điều khiển lặp và ra quyết định để hoàn tất nhiệm vụ

# Ví dụ về continue

- Ví dụ: liệt kê các giá trị trong đoạn  $[a, b]$  chia hết cho  $k \neq 0$

```
if(k != 0) {  
    for (int i = a; i <= b; i++) {  
        if(i % k != 0) { // nếu i không chia hết cho k  
            continue; // bỏ qua các lệnh phía dưới  
        }  
        // ngược lại, tức i mà chia hết cho k, thì thực hiện:  
        System.out.println(i + " ");  
    }  
    // làm gì đó tiếp ở đây  
} else {  
    System.out.println("Nhập k != 0.");  
}
```





## Các ví dụ:

- Ví dụ 1: Tìm giá trị đầu tiên trong đoạn  $[a, b]$  chia hết cho  $k$ .
- Ví dụ 2: Liệt kê các giá trị trong đoạn  $[a, b]$  chia hết cho  $k$ .

# Nội dung tiếp theo

Hướng dẫn gỡ lỗi - debugging