

 Cập nhật tháng 8 năm 2024

[Bài đọc] So sánh và ứng dụng câu lệnh điều kiện

1. Tại sao cần câu lệnh điều kiện

- Trong thực tế, chương trình thường phải ra quyết định dựa vào dữ liệu đầu vào từ người dùng hoặc từ hệ thống
 - Nếu người dùng nhập đúng mật khẩu → cho đăng nhập
 - Nếu điểm ≥ 5 → học sinh đạt, ngược lại → không đạt
 - Nếu số lượng hàng trong kho < 10 → cảnh báo tồn kho thấp
- Những tình huống như vậy đòi hỏi chương trình kiểm tra điều kiện, từ đó thực hiện hành động phù hợp

2. Các tình huống ứng dụng phổ biến

- Xây dựng menu lựa chọn
 - Sử dụng `if...else if` hoặc `switch...case` để cho phép người dùng chọn chức năng:

```
Scanner sc = new Scanner(System.in);
System.out.println("1. Xem điểm\n2. Thoát");
int chon = sc.nextInt();

if (chon == 1) {
    // xử lý xem điểm
} else {
    System.out.println("Tạm biệt!");
}
```

- Phân loại đầu vào
 - Dùng `if...else` để xử lý dữ liệu đầu vào và phân loại
 - Ví dụ: Xếp loại học sinh theo điểm

```
int diem = 8;

if (diem >= 9) {
    System.out.println("Giỏi");
} else if (diem >= 7) {
    System.out.println("Khá");
} else if (diem >= 5) {
    System.out.println("Trung bình");
} else {
    System.out.println("Yếu");
}
```

- Kiểm tra hợp lệ (Validate)
 - Trước khi xử lý, cần kiểm tra xem đầu vào có hợp lệ không

```
int tuoi = 17;

if (tuoi >= 18) {
    System.out.println("Đủ điều kiện thi bằng lái xe");
} else {
    System.out.println("Chưa đủ tuổi");
}
```

- Kết hợp với vòng lặp để làm chương trình thông minh hơn

```
for (int i = 1; i <= 10; i++) {
    if (i % 2 == 0) {
        System.out.println(i + " là số chẵn");
    } else {
        System.out.println(i + " là số lẻ");
    }
}
```

- Phát triển ứng dụng thực tế
 - Trong các hệ thống như:
 - **Ứng dụng ngân hàng:** Kiểm tra số dư trước khi rút tiền
 - **Game:** Xác định thắng/thua tùy theo điểm số hoặc tình huống

- **Ứng dụng học tập:** Tự động đánh giá kết quả người học

- Câu lệnh điều kiện là **bắt buộc phải có** để xử lý các luồng logic khác nhau

3. Lợi ích khi thành thạo câu lệnh điều kiện

- Viết chương trình linh hoạt, thông minh.
- Tạo ra các phần mềm có **khả năng tương tác** cao.
- Làm nền tảng để học các khái niệm cao hơn như: vòng lặp, mảng, class, OOP...

4. Tổng kết

| Tình huống ứng dụng | Loại câu lệnh thường dùng |
|--------------------------------|-----------------------------|
| Chọn chức năng menu | if... else, switch... case |
| Xếp loại, phân loại | if... else if... else |
| Kiểm tra dữ liệu đầu vào | if, else |
| Tạo logic trong vòng lặp | kết hợp if, trong for/while |
| Xử lý nhiều tình huống cố định | switch... case |

Danh sách các bài học

