

 Cập nhật tháng 8 năm 2024

## [Bài đọc] Câu lệnh Switch Case

Câu lệnh switch-case trong Java là một cấu trúc điều kiện cho phép kiểm tra nhiều giá trị khác nhau của một biến và thực hiện các hành động tương ứng với từng giá trị đó. Câu lệnh switch có thể giúp thay thế nhiều câu lệnh if-else khi cần kiểm tra giá trị của một biến so với một số lựa chọn cố định

### 1. Cấu trúc cơ bản của switch-case:

Cấu trúc cơ bản của câu lệnh **switch-case** như sau:

```
switch (expression) {  
    case value1:  
        // Block of code executed if expression equals value1  
        break;  
    case value2:  
        // Block of code executed if expression equals value2  
        break;  
    case value3:  
        // Block of code executed if expression equals value3  
        break;  
    default:  
        // Block of code executed if none of the above cases match  
}
```

- **expression:** Là biểu thức cần kiểm tra giá trị. Biểu thức này có thể là một biến hoặc kết quả của một phép toán trả về một giá trị.
- **case value1:** Là một nhánh của switch, kiểm tra xem biểu thức có bằng value1 hay không.
- **break:** Được sử dụng để thoát khỏi câu lệnh switch sau khi một nhánh case được thực thi. Nếu không có break, các nhánh tiếp theo sẽ tiếp tục được thực thi (gọi là "fall-through").
- **default:** Là nhánh mặc định, sẽ được thực thi nếu không có nhánh nào khớp với giá trị của biểu thức. Đây là phần tùy chọn, có thể có hoặc không.

### 2. Nguyên lý hoạt động:

- **Kiểm tra giá trị của biểu thức:** Câu lệnh switch sẽ lấy giá trị của biểu thức và so sánh với các giá trị trong các nhánh case.
- **Chạy mã phù hợp:** Nếu tìm thấy một nhánh case có giá trị bằng với giá trị của biểu thức, chương trình sẽ thực thi mã trong nhánh đó.
- **Thực hiện break:** Sau khi mã trong một nhánh case được thực thi, chương trình sẽ thoát khỏi cấu trúc switch (nếu có break).
- **Nếu không có nhánh nào khớp:** Nếu không có nhánh case nào khớp với giá trị của biểu thức, mã trong phần default (nếu có) sẽ được thực thi.

### 3. Ví dụ sử dụng switch-case:

Ví dụ về việc sử dụng switch-case để kiểm tra ngày trong tuần và in ra tên của ngày:

```
int day = 3;
switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    case 4:
        System.out.println("Thursday");
        break;
    case 5:
        System.out.println("Friday");
        break;
    case 6:
        System.out.println("Saturday");
        break;
    case 7:
        System.out.println("Sunday");
        break;
    default:
        System.out.println("Invalid day");
}
```

Giải thích:

- Biến day có giá trị là 3, do đó, chương trình sẽ in ra Wednesday.
- Mỗi case kiểm tra giá trị của day, nếu giá trị khớp với một nhánh case, nó sẽ thực thi mã trong nhánh đó và sau đó thoát khỏi câu lệnh switch với break.
- Nếu không có giá trị nào khớp, phần default sẽ được thực thi.

### 4. Lưu ý khi sử dụng switch-case:

- **Kiểu dữ liệu của biểu thức:** Biểu thức trong câu lệnh switch có thể là một byte, short, int, char, String, hoặc các kiểu dữ liệu đối tượng của Java. Tuy nhiên, không thể sử dụng kiểu dữ liệu float, long, hoặc double.

- **Không cần break:** Nếu không có break, chương trình sẽ tiếp tục kiểm tra các case tiếp theo, ngay cả khi chúng không khớp. Điều này được gọi là "fall-through" và có thể hữu ích trong một số trường hợp.
- **Câu lệnh default:** Câu lệnh default không bắt buộc, nhưng nó rất hữu ích khi bạn muốn đảm bảo rằng một hành động mặc định được thực hiện khi không có case nào khớp.

## 5. Ví dụ về Fall-through:

```
int num = 2;
switch (num) {
    case 1:
        System.out.println("One");
        break;
    case 2:
    case 3:
        System.out.println("Two or Three");
        break;
    case 4:
        System.out.println("Four");
        break;
    default:
        System.out.println("Invalid");
}
```

Giải thích:

- Khi num là 2, chương trình sẽ không thực thi mã trong case 2 mà sẽ tiếp tục vào case 3 (do không có break), và in ra "Two or Three".
- Đây là một ví dụ về "fall-through", nơi một nhánh case có thể thực thi nhiều hành động mà không cần phải lặp lại mã.

Link tài nguyên đọc thêm: <https://data-flair.training/blogs/switch-statement-in-java/>