

 Cập nhật tháng 8 năm 2024

[Bài đọc] Các câu lệnh nhảy break, continue, return

Trong Java, vòng lặp có thể kết hợp với các câu lệnh nhảy như **break** và **continue** để điều khiển luồng thực thi một cách linh hoạt hơn. Các câu lệnh này cho phép bạn bỏ qua hoặc kết thúc vòng lặp dựa trên một điều kiện nhất định, giúp tối ưu hóa logic của chương trình.

1. Câu lệnh break trong Java:

Câu lệnh break được sử dụng để **kết thúc vòng lặp sớm** và chuyển chương trình ra ngoài vòng lặp. Khi câu lệnh break được gặp trong vòng lặp, vòng lặp sẽ dừng lại ngay lập tức, và chương trình tiếp tục thực thi mã bên ngoài vòng lặp.

Cú pháp của break:

```
public static void main(String[] args) new *
{
    for (int i = 0; i < 10; i++) {
        if (i == 5) {
            break; // Dừng vòng lặp khi i == 5
        }
        System.out.println(i);
    }
}
```

Giải thích:

- Trong ví dụ trên, vòng lặp sẽ in các số từ 0 đến 4.
- Khi i đạt giá trị 5, câu lệnh break sẽ được thực thi, làm cho vòng lặp kết thúc ngay lập tức.

Kết quả:

```
0
1
2
3
4
```

Ứng dụng:

- break thường được sử dụng khi bạn cần thoát khỏi vòng lặp ngay lập tức khi đã đạt được điều kiện nhất định (ví dụ: tìm thấy giá trị trong một danh sách).

2. Câu lệnh continue trong Java:

Câu lệnh continue được sử dụng để **bỏ qua phần còn lại của vòng lặp** trong một lần lặp hiện tại và tiếp tục thực thi vòng lặp với lần lặp tiếp theo. Khi gặp continue, vòng lặp sẽ bỏ qua phần mã còn lại (từ câu lệnh continue đến cuối vòng lặp) và quay lại điều kiện kiểm tra để quyết định liệu vòng lặp có tiếp tục hay không.

Cú pháp của continue:

```
public static void main(String[] args) {
    for (int i = 0; i < 10; i++) {
        if (i == 5) {
            continue; // Bỏ qua phần còn lại của vòng lặp khi i == 5
        }
        System.out.println(i);
    }
}
```

Giải thích:

- Trong ví dụ trên, khi i bằng 5, câu lệnh continue sẽ được thực thi, khiến vòng lặp bỏ qua câu lệnh System.out.println(i) và tiếp tục với giá trị i là 6.

Kết quả:

```
0
1
2
3
4
6
7
8
9
```

Ứng dụng:

- continue thường được sử dụng khi bạn muốn bỏ qua một vòng lặp khi một điều kiện nào đó được thỏa mãn, nhưng không muốn kết thúc vòng lặp.

3. Sử dụng break và continue trong các vòng lặp khác nhau:

Vòng lặp for:

```

public static void main(String[] args) {
    for (int i = 0; i < 10; i++) {
        if (i == 3) {
            break; // Dừng vòng lặp khi i == 3
        }
        if (i == 1) {
            continue; // Bỏ qua in khi i == 1
        }
        System.out.println(i);
    }
}

```

Kết quả:

```

0
2

```

- Khi `i == 1`, câu lệnh `continue` bỏ qua `System.out.println(i)` và chuyển đến vòng lặp tiếp theo (với `i == 2`).
- Khi `i == 3`, câu lệnh `break` kết thúc vòng lặp.

Vòng lặp `while`:

```

public static void main(String[] args) {
    int i = 0;
    while (i < 10) {
        if (i == 5) {
            break; // Dừng vòng lặp khi i == 5
        }
        if (i == 2) {
            i++;
            continue; // Bỏ qua in khi i == 2
        }
        System.out.println(i);
        i++;
    }
}

```

Kết quả:

```

0
1
3
4

```

- Khi `i == 2`, câu lệnh `continue` bỏ qua in `i`, và vòng lặp tiếp tục với `i == 3`.
- Khi `i == 5`, câu lệnh `break` dừng vòng lặp.

Vòng lặp `do-while`:

```

public static void main(String[] args) new *
{
    int i = 0;
    do {
        if (i == 3) {
            break; // Dừng vòng lặp khi i == 3
        }
        if (i == 1) {
            i++;
            continue; // Bỏ qua in khi i == 1
        }
        System.out.println(i);
        i++;
    } while (i < 5);
}

```

Kết quả:

```

0
2

```

- Tương tự như các vòng lặp khác, câu lệnh continue sẽ bỏ qua phần còn lại của vòng lặp khi `i == 1`.
- Khi `i == 3`, câu lệnh break dừng vòng lặp.

4. Tóm tắt về break và continue:

- **break:** Dừng vòng lặp ngay lập tức và thoát khỏi vòng lặp.
- **continue:** Bỏ qua phần còn lại của vòng lặp trong lần lặp hiện tại và tiếp tục với lần lặp tiếp theo.

Khi nào nên sử dụng break và continue?

- **Sử dụng break:** Khi bạn muốn dừng vòng lặp ngay lập tức khi đạt được điều kiện nào đó, chẳng hạn như tìm thấy một phần tử trong danh sách hoặc khi một điều kiện ngoại lệ xảy ra.
- **Sử dụng continue:** Khi bạn muốn bỏ qua một lần lặp mà không muốn kết thúc vòng lặp, chẳng hạn như bỏ qua giá trị không hợp lệ trong khi duyệt qua một dãy số.

Link tài nguyên đọc thêm: https://www.w3schools.com/java/java_break.asp