

# Bài 9: CẤU TRÚC RỄ NHÁNH TRONG JAVA

Xem bài học trên website để ủng hộ Kteam: [Cấu trúc rẽ nhánh trong Java](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Sau khi tìm hiểu các khái niệm cơ bản về một ngôn ngữ lập trình. Bây giờ ta sẽ tìm hiểu điều cơ bản nhất trong lập trình là **Cấu trúc rẽ nhánh**.

## Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- CÁC [BIẾN](#) & [HẲNG](#) TRONG JAVA.
- [CÁC KIỂU DỮ LIỆU TRONG JAVA](#)
- [CÁC HANG TOÁN TỬ TRONG JAVA](#)

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Cấu trúc rẽ nhánh là gì? Phân loại
- Cách sử dụng cấu trúc rẽ nhánh
- Lời khuyên

## Cấu trúc rẽ nhánh là gì? Phân loại

Trong lập trình, có những lúc ta cần phân chia các trường hợp và mỗi hoàn cảnh sẽ thực hiện những đoạn chương trình khác nhau. Như vậy dựa vào điều kiện ta sẽ rẽ nhánh cho chương trình chạy câu lệnh tương ứng.

Phân loại: Có 2 loại cấu trúc rẽ nhánh là **dạng thiếu** và **dạng đủ**

### Ví dụ:

- Dạng thiếu: Nếu biến age trên 18 thì ta sẽ in ra 'Bạn đã đủ tuổi để đăng kí'.
- Dạng đủ: Nếu biến age trên 18 thì ta sẽ in ra 'Bạn đã đủ tuổi để đăng kí', ngược lại thì in ra là 'Bạn chưa đủ tuổi để đăng kí'

## Cách sử dụng cấu trúc rẽ nhánh

### Cấu trúc rẽ nhánh dạng thiếu

#### Cú pháp:

**if (<Biểu thức điều kiện>) <Câu lệnh thực hiện>**

**Ý nghĩa:** Nếu **<Biểu thức điều kiện>** trả về **true** thì sẽ thực hiện **<Câu lệnh thực hiện>**.

#### Ví dụ:

```
public class HelloWorld{  
  
    public static void main(String []args){  
        String s = "Kteam";  
        if (s == "Kteam")  
            System.out.print("How Kteam");  
        }  
    }  
}
```

```
$javac HelloWorld.java  
$java HelloWorld  
How Kteam
```



## Cấu trúc rẽ nhánh dạng đủ

Dạng đủ sẽ chia ra 2 loại: Gồm **if..else..** và **if..else if..else..**

### Dạng if..else..

#### Cú pháp:

```
if (<Biểu thức điều kiện>)  
    <Câu lệnh thực hiện điều kiện đúng>  
  
else  
    <Câu lệnh thực hiện điều kiện sai>
```

#### Ý nghĩa:

Nếu **<Biểu thức điều kiện>** trả về **true** thì sẽ thực hiện **<Câu lệnh thực hiện điều kiện đúng>**. Ngược lại nếu trả về **false** sẽ thực hiện **<Câu lệnh thực hiện điều kiện sai>**

#### Ví dụ:

```
public class HelloWorld{  
  
    public static void main(String []args){  
        int age = 18;  
        if (age > 18)  
            System.out.print("Bạn đủ tuổi để đăng ký");  
        else  
            System.out.print("Bạn chưa đủ tuổi để đăng ký");  
    }  
}
```

```
}
```

```
$javac HelloWorld.java  
$java HelloWorld  
Bạn chưa đủ tuổi để đăng ký
```



## Dạng if..else if..else

### Cú pháp:

**if** (<Biểu thức điều kiện 1>)

<Câu lệnh thực hiện điều kiện 1>

**else if** (<Biểu thức điều kiện 2>)

<Câu lệnh thực hiện điều kiện 2>

(Nhiều câu điều kiện khác nếu cần)

**else**

<Câu lệnh thực hiện không đúng trong tất các điều kiện trên>

### Ý nghĩa:

Đây là dạng cấu trúc rẽ nhiều nhánh nếu ta muốn xét nhiều trường hợp để thực hiện rõ ràng hơn.

### Ví dụ:

```
public class HelloWorld{  
  
    public static void main(String []args){  
        String job = "Sinh viên";  
        if (job == "Học sinh")  
            System.out.print("Bạn còn lứa tuổi học sinh");  
    }  
}
```

```
else if (job == "Sinh viên")
    System.out.print("Sinh viên có thể tham gia");
else
    System.out.print("Không rõ công việc của bạn");
}
```

```
$javac HelloWorld.java
$java HelloWorld
Sinh viên có thể tham gia
```

**Lưu ý:** Nếu câu lệnh thực hiện gồm nhiều câu lệnh thì ta phải để vào trong cặp { }

```
public class HelloWorld{

    public static void main(String []args){
        int age = 18;
        if (age >= 18) {
            System.out.println("Bạn đủ tuổi để đăng ký");
            System.out.println("Mời bạn đăng ký");
        }
        else
            System.out.print("Bạn chưa đủ tuổi để đăng ký");
    }
}
```

```
$javac HelloWorld.java
$java HelloWorld
Bạn đủ tuổi để đăng ký
Mời bạn đăng ký
```

## Lời khuyên

Có thể không cần thực hiện những lời khuyên sau đây gần như là “**Luật bất thành văn**” trong lập trình cấu trúc rẽ nhánh không riêng ngôn ngữ Java.

### Luôn dùng cặp { } và thụt lề dòng code vào trong

Mặc dù cặp { } chỉ dùng cho nhóm câu lệnh, nhưng việc sử dụng { } sẽ giúp ta dễ sửa đổi trong trường hợp sau này ta muốn thêm câu lệnh khác. Ngoài ra, ta sử dụng thụt lề để dễ nhìn hơn.

```
1 public class HelloWorld{
2
3     public static void main(String []args){
4         String job = "Sinh viên";
5         if (job == "Học sinh") {
6             System.out.print("Bạn còn lứa tuổi học sinh");
7         } else if (job == "Sinh viên") {
8             System.out.print("Sinh viên có thể tham gia");
9         } else {
10            System.out.print("Không rõ công việc của bạn");
11        }
12    }
13 }
14 |
```

### Không nên đưa câu lệnh if..else trong if..else quá nhiều lần

Có những trường hợp các bạn viết như thế này

```
1 public class HelloWorld{
2
3     public static void main(String []args){
4         if (true) {
5             if (true) {
6                 if (true){
7
8                 } else if (true) {
9
10                } else {
11
12                }
13            } else {
14
15            }
16        }
17    }
18 }
```

Việc đưa quá nhiều câu rẽ nhánh vào trong trong câu rẽ nhánh là điều tối kỵ trong lập trình. Nó giống như tạo nhiều đường mê cung vậy. Có những lúc

bạn sẽ không rõ vì sao kết quả trả về như vậy. Đặc biệt là khi có **bug** phát sinh. Lời khuyên là bạn nên để 1-2 câu điều kiện bên trong là hợp lý.

---

## Kết

Như vậy chúng ta đã tìm hiểu cấu trúc rẽ nhánh trong Java

Ở bài sau, Kteam sẽ giới thiệu đến bạn về [VÒNG LẶP WHILE TRONG JAVA](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.