

# Điều khiển vòng lặp trong Java

Có thể có một tình huống khi chúng ta cần thực hiện một khối code một vài lần, điều này có thể được xem như một vòng lặp.

Java có 3 kỹ thuật lặp linh động. Bạn có thể sử dụng một trong ba vòng lặp sau:

- Vòng lặp while
- Vòng lặp do...while
- Vòng lặp for

Trong Java 5, vòng lặp **foreach (hay enhanced for)** đã được giới thiệu. Vòng lặp này được sử dụng chủ yếu cho các mảng.

## Vòng lặp while trong Java

Một vòng lặp while là một cấu trúc điều khiển cho phép bạn lặp đi lặp lại một tác vụ một số lần nào đó.

## Cú pháp

Cú pháp của một vòng lặp while trong Java như sau:

```
while(Boolean_expression)
{
    //Statements
}
```

Khi thực thi, nếu kết quả của *boolean\_expression* là true, thì các hoạt động bên trong vòng lặp được thực thi. Nó sẽ tiếp tục khi nào mà kết quả expression này vẫn là true.

Ở đây, điểm chính của vòng lặp **while** là vòng lặp có thể chưa từng chạy lần nào. Đó là khi expression được kiểm tra và kết quả là false, thân vòng lặp sẽ bị bỏ qua và lệnh đầu tiên sau vòng lặp while sẽ được thực thi.

## Ví dụ:

```
public class Test {
```

```
public static void main(String args[]) {  
    int x = 10;  
  
    while( x < 20 ) {  
        System.out.print("value of x : " + x );  
        x++;  
        System.out.print("\n");  
    }  
}
```

Nó sẽ cho kết quả:

```
value of x : 10  
value of x : 11  
value of x : 12  
value of x : 13  
value of x : 14  
value of x : 15  
value of x : 16  
value of x : 17  
value of x : 18  
value of x : 19
```

## Vòng lặp do...while trong Java

Một vòng lặp do...while là tương tự như vòng lặp while, ngoại trừ rằng, một vòng lặp do...while được bảo đảm thực thi ít nhất một lần.

### Cú pháp:

Cú pháp của vòng lặp do...while trong Java là:

```
do  
{  
    //Statements
```

```
}while(Boolean_expression);
```

Ghi chú rằng, *Boolean\_expression* xuất hiện tại phần cuối của vòng lặp, vì thế các lệnh trong vòng lặp thực thi một lần trước khi expression được kiểm tra.

Nếu *Boolean\_expression* là true, nó quay trở lại và các lệnh được thực thi lần nữa. Tiến trình này lặp đi lặp lại tới khi *boolean\_expression* là false.

## Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        int x = 10;  
  
        do{  
            System.out.print("value of x : " + x );  
            x++;  
            System.out.print("\n");  
        }while( x < 20 );  
    }  
}
```

Nó sẽ cho kết quả sau:

```
value of x : 10  
value of x : 11  
value of x : 12  
value of x : 13  
value of x : 14  
value of x : 15  
value of x : 16  
value of x : 17  
value of x : 18  
value of x : 19
```

# Vòng lặp for trong Java

Một vòng lặp for cho phép bạn viết một vòng lặp một cách hiệu quả để cần thực thi một số lần lặp cụ thể nào đó.

Một vòng lặp for là hữu ích khi bạn biết số lần một tác vụ được lặp lại.

## Cú pháp:

Cú pháp của một vòng lặp for trong Java như sau:

```
for(initialization; Boolean_expression; update)
{
    //Statements
}
```

Dưới đây là luồng điều khiển trong một vòng lặp for:

- Bước *initialization* được thực thi đầu tiên, và chỉ một lần. Bước này cho phép bạn khai báo và khởi tạo bất kỳ biến điều khiển vòng lặp. Bạn không được yêu cầu đặt một lệnh ở đây, miễn là một dấu chấm phẩy xuất hiện.
- Sau đó, *Boolean\_expression* được ước lượng. Nếu nó là true, phần thân vòng lặp được thực thi. Nếu nó là false, phần thân vòng lặp không thực thi và luồng điều khiển nhảy tới lệnh tiếp theo sau vòng lặp for.
- Sau khi thân vòng lặp thực thi, luồng điều khiển nhảy trở lại lệnh update. Lệnh này cho phép bạn cập nhật bất kỳ biến điều khiển vòng lặp nào. Lệnh này có thể để trống, miễn là một dấu chấm phẩy xuất hiện sau *Boolean\_expression*.
- *Boolean\_expression* bây giờ được ước lượng lần nữa. Nếu là true, vòng lặp thực thi và tiến trình lặp lại như trên. Sau khi *Boolean\_expression* là false, vòng lặp kết thúc.

## Ví dụ:

```
public class Test {

    public static void main(String args[]) {

        for(int x = 10; x < 20; x = x+1) {
```

```
        System.out.print("value of x : " + x );  
        System.out.print("\n");  
    }  
}  
}
```

Nó sẽ cho kết quả sau:

```
value of x : 10  
value of x : 11  
value of x : 12  
value of x : 13  
value of x : 14  
value of x : 15  
value of x : 16  
value of x : 17  
value of x : 18  
value of x : 19
```

## Vòng lặp foreach trong Java

Trong Java 5, vòng lặp foreach (hay enhanced for) đã được giới thiệu. Nó được sử dụng chủ yếu với các mảng.

### Cú pháp:

Cú pháp của vòng lặp foreach như sau:

```
for(declaration : expression)  
{  
    //Statements  
}
```

- **Declaration - Khai báo:** Biến khối được khai báo mới, mà là một kiểu tương thích với các phần tử của mảng bạn đang truy cập. Biến này sẽ có sẵn trong khối for và giá trị của nó sẽ giống như phần tử mảng hiện tại.

- **Expression:** Nó ước lượng mảng bạn cần lặp. Expression này có thể là một biến mảng hoặc gọi phương thức mà trả về một mảng.

## Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        int [] numbers = {10, 20, 30, 40, 50};  
  
        for(int x : numbers ){  
            System.out.print( x );  
            System.out.print(",");  
        }  
        System.out.print("\n");  
        String [] names ={"James", "Larry", "Tom", "Lacy"};  
        for( String name : names ) {  
            System.out.print( name );  
            System.out.print(",");  
        }  
    }  
}
```

Nó sẽ cho kết quả sau:

```
10,20,30,40,50,  
James,Larry,Tom,Lacy,
```

## Từ khóa break trong Java

Từ khóa *break* được sử dụng để dừng toàn bộ vòng lặp. Từ khóa *break* phải được sử dụng bên trong bất kỳ vòng lặp nào hoặc một lệnh *switch*.

Từ khóa *break* sẽ dừng sự thực thi của vòng lặp trong cùng và bắt đầu thực thi dòng code tiếp theo sau khối đó.

## Cú pháp:

Cú pháp của một lệnh break là một lệnh đơn bên trong bất kỳ vòng lặp nào:

```
break;
```

## Ví dụ:

```
public class Test {  
  
    public static void main(String args[]) {  
        int [] numbers = {10, 20, 30, 40, 50};  
  
        for(int x : numbers ) {  
            if( x == 30 ) {  
                break;  
            }  
            System.out.print( x );  
            System.out.print("\n");  
        }  
    }  
}
```

Nó sẽ cho kết quả sau:

```
10  
20
```

## Từ khóa continue trong Java

Từ khóa *continue* có thể được sử dụng trong bất kỳ cấu trúc điều khiển vòng lặp nào. Nó làm cho vòng lặp ngay lập tức tiếp tục tiến trình lặp tiếp theo của vòng lặp.

- Trong một vòng lặp for, từ khóa continue làm luồng điều khiển ngay lập tức nhảy tới lệnh update.
- Trong vòng lặp while hoặc do...while, luồng điều khiển ngay lập tức nhảy tới Boolean-expression.

## Cú pháp:

Cú pháp của một continue là một lệnh đơn bên trong bất kỳ vòng lặp nào:

```
continue;
```

## Ví dụ:

```
public class Test {  
  
    public static void main(String args[]) {  
        int [] numbers = {10, 20, 30, 40, 50};  
  
        for(int x : numbers ) {  
            if( x == 30 ) {  
                continue;  
            }  
            System.out.print( x );  
            System.out.print("\n");  
        }  
    }  
}
```

Nó sẽ cho kết quả sau:

```
10  
20  
40  
50
```

## Chương sau bàn về chủ đề gì trong Java?

Trong chương tới, chúng ta sẽ nghiên cứu về các lệnh điều khiển luồng trong ngôn ngữ Java.