

Điều khiển luồng trong Java

Có hai loại lệnh điều khiển luồng trong Java. Chúng là:

- Lệnh if
- Lệnh switch

Lệnh if trong Java

Một lệnh if trong Java bao gồm một Boolean expression được theo sau bởi một hoặc nhiều lệnh.

Cú pháp:

Cú pháp của lệnh if như sau:

```
if(Boolean_expression)
{
    //Statements will execute if the Boolean expression is true
}
```

Nếu *Boolean expression* ước lượng là true thì khi đó khối code bên trong lệnh if sẽ được thực thi. Nếu không, code ở sau phần cuối của lệnh if (sau dấu ngoặc móc đóng) sẽ được thực thi.

Ví dụ:

```
public class Test {

    public static void main(String args[]){
        int x = 10;

        if( x < 20 ){
            System.out.print("This is if statement");
        }
    }
}
```

Nó sẽ cho kết quả:

This is if statement

Lệnh if...else trong Java

Một lệnh if có thể được theo sau bởi một lệnh *else* tùy ý, mà thực thi khi Boolean expression là false.

Cú pháp:

Cú pháp của một lệnh if...else trong Java là:

```
if(Boolean_expression){  
    //Executes when the Boolean expression is true  
}else{  
    //Executes when the Boolean expression is false  
}
```

Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        int x = 30;  
  
        if( x < 20 ){  
            System.out.print("This is if statement");  
        }else{  
            System.out.print("This is else statement");  
        }  
    }  
}
```

Nó sẽ cho kết quả:

This is else statement

Lệnh if...else if...else trong Java

Một lệnh if có thể được theo sau bởi một lệnh *else if...else* tùy ý, mà rất hữu ích để kiểm tra các điều kiện đa dạng bởi sử dụng lệnh if...else if đơn.

Khi sử dụng các lệnh if, else if, else, có một số điểm chính cần ghi nhớ:

- Một lệnh if có 0 hoặc một lệnh else và nó phải theo sau bởi bất kỳ lệnh else if nào.
- Một lệnh if có từ 0 tới nhiều lệnh else if và chúng phải ở trước lệnh else.
- Một khi lệnh else if thành công, các phần else if hoặc else còn lại sẽ không được kiểm tra.

Cú pháp:

Cú pháp của một if...else là:

```
if(Boolean_expression 1){  
    //Executes when the Boolean expression 1 is true  
}else if(Boolean_expression 2){  
    //Executes when the Boolean expression 2 is true  
}else if(Boolean_expression 3){  
    //Executes when the Boolean expression 3 is true  
}else {  
    //Executes when the none of the above condition is true.  
}
```

Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        int x = 30;  
  
        if( x == 10 ){  
            System.out.print("Value of X is 10");  
        }else if( x == 20 ){  
            System.out.print("Value of X is 20");  
        }  
    }  
}
```

```
    }else if( x == 30 ){  
        System.out.print("Value of X is 30");  
    }else{  
        System.out.print("This is else statement");  
    }  
}  
}
```

Nó sẽ cho kết quả sau:

```
Value of X is 30
```

Lồng các lệnh if...else trong Java

Nó là hợp lệ để lồng các lệnh if-else, nghĩa là bạn có thể sử dụng một lệnh if hoặc else if bên trong lệnh if hoặc else if khác.

Cú pháp:

Cú pháp cho một lệnh if...else được lồng vào nhau như sau:

```
if(Boolean_expression 1){  
    //Executes when the Boolean expression 1 is true  
    if(Boolean_expression 2){  
        //Executes when the Boolean expression 2 is true  
    }  
}
```

Bạn có thể lồng *else if...else* theo cách tương tự như chúng ta đã lồng lệnh *if*.

Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        int x = 30;  
        int y = 10;
```

```
if( x == 30 ){  
    if( y == 10 ){  
        System.out.print("X = 30 and Y = 10");  
    }  
}  
}
```

Nó sẽ cho kết quả sau:

```
X = 30 and Y = 10
```

Lệnh switch trong Java

Lệnh switch cho phép một biến để được kiểm tra bình đẳng với một danh sách các giá trị. Mỗi giá trị được gọi là một case – trường hợp.

Cú pháp:

Cú pháp cho lệnh switch trong Java là:

```
switch(expression){  
    case value :  
        //Statements  
        break; //optional  
    case value :  
        //Statements  
        break; //optional  
    //You can have any number of case statements.  
    default : //Optional  
        //Statements  
}
```

Các qui tắc sau áp dụng cho một lệnh switch:

- Biến được sử dụng trong một lệnh switch chỉ có thể là byte, short, int hoặc char.

- Bạn có thể có nhiều lệnh case bên trong một lệnh switch. Mỗi case được theo sau bởi giá trị để được so sánh và một dấu hai chấm.
- Giá trị cho một case phải giống kiểu dữ liệu của biến trong switch và nó phải là hằng số hoặc literal.
- Khi biến đang được switch là tương đương với một case, các lệnh theo sau case đó sẽ thực thi tới khi gặp lệnh *break*.
- Khi gặp một lệnh *break* thì switch kết thúc, và luồng điều khiển nhảy tới dòng tiếp theo lệnh switch.
- Không phải mọi case đều cần một break. Nếu không có lệnh break xuất hiện, luồng điều khiển sẽ đi qua các case sau đó tới khi gặp một lệnh break.
- Một lệnh *switch* có thể có một case mặc định, mà phải xuất hiện ở cuối lệnh switch. Case mặc định này có thể được sử dụng để thực thi một tác vụ trong trường hợp không có case nào là true. Trong trường hợp này, chúng ta không cần lệnh break.

Ví dụ:

```
public class Test {  
  
    public static void main(String args[]){  
        //char grade = args[0].charAt(0);  
        char grade = 'C';  
  
        switch(grade)  
        {  
            case 'A' :  
                System.out.println("Excellent!");  
                break;  
            case 'B' :  
            case 'C' :  
                System.out.println("Well done");  
                break;  
            case 'D' :
```

```
        System.out.println("You passed");  
    case 'F' :  
        System.out.println("Better try again");  
        break;  
    default :  
        System.out.println("Invalid grade");  
    }  
    System.out.println("Your grade is " + grade);  
}  
}
```

Biên dịch và chạy chương trình trên sẽ cho kết quả sau:

```
$ java Test  
Well done  
Your grade is a C  
$
```

Chương tiếp theo bàn về chủ đề gì trong Java?

Chương tới bàn về lớp Number (trong java.lang package) và các lớp phụ của nó trong ngôn ngữ Java.

Chúng ta sẽ xem xét một số tình huống mà bạn sẽ sử dụng các thuyết minh của các lớp ngoài các kiểu dữ liệu gốc, cũng như định dạng, các hàm toán học mà bạn cần biết khi làm việc với Number.