

Function

Tags

Progress

[Hàm trong java](#)

[Thao tác với hàm \(method\)](#)

[Hàm có trả về kết quả](#)

[Hàm không trả về kết quả](#)

[Tóm tắt cú pháp](#)

[Truyền tham số khi gọi hàm \(method\)](#)

[Truyền tham số theo giá trị \(Passing Parameters by Value\)](#)

Hàm hay phương thức (method) trong Java là một khối lệnh để thực hiện một hành động cụ thể. Ví dụ khi chúng ta gọi hàm `System.out.println()` thì hệ thống sẽ hiển thị dữ liệu của chúng ta truyền vào.

Hàm trong java

Cú pháp:

```
modifier returnType nameOfMethod (Parameter List) {  
    // method body  
}
```

Trong đó:

- **Modifiers:** Tạm dịch là phạm vi sửa đổi và truy cập
- **returnType:** Là kiểu dữ liệu trả về
- **nameOfMethod:** Là tên của hàm (method)
- **Parameter** là các tham số đầu vào của hàm (có thể có nhiều tham số với nhiều kiểu dữ liệu khác nhau)
- **method body:** là các mã code bên trong hàm

Ví dụ: Viết hàm tính tổng 2 chữ số a và b

```
public static int tongHaiSo(int a, int b) {  
    return a + b;  
}
```

Thao tác với hàm (method)

Trước khi chúng ta tìm hiểu về cách gọi hàm thì chúng ta tìm hiểu về các loại hàm. Java có 2 loại hàm, mỗi loại sẽ có cách gọi khác nhau:

- Hàm có trả về kết quả
- Hàm không trả về kết quả

Hàm có trả về kết quả

Đối với hàm có kết quả trả về, chúng ta cần dùng từ khoá `return` để trả về kết quả mà nó đã tính toán được. Khai báo biến có kiểu dữ liệu tương ứng với kết quả trả về của hàm để nhận giá trị trả về.

Như ví dụ tìm `tongHaiSo()` của chúng ta, nhiệm vụ của nó là tính tổng của 2 số nguyên **a**, **b** nhập vào thì kết quả sẽ ra một số nguyên. Tức là kiểu trả về (`returnType`) là một số nguyên (`int`) và kết quả của `return` cũng là một số nguyên (`int`). Chúng ta sẽ dùng biến có kiểu dữ liệu tương ứng để nhận kết quả trả về từ hàm.

```
public class Thaycacac {  
    public static int tongHaiSo(int a, int b) {  
        return a + b;  
    }  
    public static void main(String[] args) {  
        int sum = tongHaiSo(2,5);  
        System.out.println(sum);  
    }  
}
```

Hàm không trả về kết quả

Đối với loại hàm này chúng ta chỉ cần gọi để sử dụng.

```
public static void printHello() {  
    System.out.println("Hello");  
}
```

Hàm `printHello()` là một hàm không có kiểu trả về (returnType) nên được thay thế thành `void`. Từ khoá `void` cho chúng ta biết là method đó sẽ không có kết quả trả về.

Tóm tắt cú pháp

Hàm có kết quả trả về

```
modifier returnType nameOfMethod (Parameter List) {  
    // method body  
}
```

Hàm không có kết quả trả về

```
<modifier> void methodName(Danh sách các tham số) {  
    // body  
}
```

Truyền tham số khi gọi hàm (method)

Khi bạn gọi đến một hàm đã được định và nó có danh sách các thông số nhất định. Thì khi gọi các bạn sẽ phải truyền đúng **thứ tự** và **kiểu dữ liệu** cho từng tham số. Nếu sai sót trong quá trình này có thể dẫn đến biên dịch lỗi hoặc là kết quả không đúng như mong muốn.

Truyền tham số theo giá trị (Passing Parameters by Value)

```
public class Thaycacac {  
    public static void hanDoiViTri(int a, int b) {  
        int c = a;  
        a = b;
```

```
b = c;  
System.out.println("Sau khi hoán đổi: " + "a = " + a + " b="  
}  
public static void main(String[] args) {  
    int a = 5, b = 3;  
    System.out.println("Trước khi hoán đổi: " + "a = " + a + " |"  
    hanDoiViTri(a, b);  
}  
}
```

Kết quả

Sau khi hoán đổi: a = 3 b= 5

Trước khi hoán đổi: a = 5 b= 3

- Hàm `hoanDoiViTri` sẽ không làm ảnh hưởng đến giá trị của các tham số truyền vào.
- Trong một class nếu có 2 method cùng tên và cùng tham số đầu vào thì trình biên dịch sẽ báo lỗi
- Thông thường trong một lớp các phương thức nên được khai báo dùng từ khóa `public`, khác với vùng dữ liệu thường là dùng tiền tố `private` vì mục đích an toàn.