

# Java Variables - Kiểu biến trong Java

## Khái niệm

Các ứng dụng sử dụng các biến để lưu trữ các dữ liệu cần thiết hoặc các dữ liệu được tạo ra trong quá trình thực thi chương trình. Các biến được xác định bởi một tên biến và có một phạm vi tác động. Phạm vi tác động của biến được xác định một cách rõ ràng trong chương trình. Mỗi biến được khai báo trong một khối chương trình chỉ có tác động trong phạm vi khối đó, không có ý nghĩa và không được phép truy nhập từ bên ngoài khối.

 or type unknown

```
int data=50; //Here data is variable
```

Việc khai báo một biến bao gồm 3 thành phần: kiểu biến, tên của nó và giá trị ban đầu được gán cho biến (không bắt buộc). Để khai báo nhiều biến ta sử dụng dấu phẩy để phân cách các biến. Khi khai báo biến, luôn nhớ rằng Java phân biệt chữ thường và chữ in hoa (case-sensitive).

## Cú pháp

```
Datatype indentifier [=value] [, indentifier[=value]... ];
```

Để khai báo một biến nguyên (int) có tên là counter dùng để lưu giá trị ban đầu là 1, ta có thể thực hiện phát biểu sau đây:

```
int counter = 1;
```

Java có những yêu cầu hạn chế đặt tên biến mà bạn có thể gán giá trị vào. Những hạn chế này cũng giống các hạn chế khi đặt tên cho các định danh mà ta đã thảo luận ở các phần trước của chương này.

## Các loại biến

 or type unknown

### 1. Biến local - Local variable

- Biến local được khai báo trong các phương thức, hàm constructor hoặc trong các block.

- Biến local được tạo bên trong các phương thức, constructor, block và sẽ bị phá hủy khi kết thúc các phương thức, constructor và block.
- Không được sử dụng "access modifier" khi khai báo biến local.
- Các biến local được lưu trên vùng nhớ stack của bộ nhớ.
- Bạn cần khởi tạo giá trị mặc định cho biến local trước khi có thể sử dụng.

```
class Simple {
    public static void main(String[] args) {
        int a = 10; // local variable
        float f = a; // local variable
        System.out.println(a);
        System.out.println(f);
    }
}
```

## 2. Biến toàn cục - Instance variable

- Biến instance được khai báo trong một lớp(class), bên ngoài các phương thức, constructor và các block.
- Biến instance được lưu trong bộ nhớ heap.
- Biến instance được tạo khi một đối tượng được tạo bằng việc sử dụng từ khóa "new" và sẽ bị phá hủy khi đối tượng bị phá hủy.
- Biến instance có thể được sử dụng bởi các phương thức, constructor, block... Nhưng nó phải được sử dụng thông qua một đối tượng cụ thể.
- Bạn được phép sử dụng "access modifier" khi khai báo biến instance, mặc định là "default".
- Biến instance có giá trị mặc định phụ thuộc vào kiểu dữ liệu của nó. Ví dụ nếu là kiểu int, short, byte thì giá trị mặc định là 0, kiểu double thì là 0.0d... Vì vậy, bạn sẽ không cần khởi tạo giá trị cho biến instance trước khi sử dụng.
- Bên trong class mà bạn khai báo biến instance, bạn có thể gọi nó trực tiếp bằng tên khi sử dụng ở khắp nơi bên trong class đó.

```
class A {
    int data = 50; //instance variable
    void method() {
        System.out.println(data);
    }
} //end of class
```

## 3. Biến static - Static variable

- Biến static được khai báo trong một class với từ khóa "static", phía bên ngoài các phương thức, constructor và block.

- Sẽ chỉ có duy nhất một bản sao của các biến static được tạo ra, dù bạn tạo bao nhiêu đối tượng từ lớp tương ứng.
- Biến static được lưu trữ trong bộ nhớ static riêng.
- Biến static được tạo khi chương trình bắt đầu chạy và chỉ bị phá hủy khi chương trình dừng.
- Giá trị mặc định của biến static phụ thuộc vào kiểu dữ liệu bạn khai báo tương tự biến instance.
- Biến static được truy cập thông qua tên của class chứa nó, với cú pháp: TenClass.tenBien.
- Trong class, các phương thức sử dụng biến static bằng cách gọi tên của nó khi phương thức đó cũng được khai báo với từ khóa "static".

```
class A {  
    static int m = 100; //static variable  
    void method() {  
        System.out.println(m);  
    }  
} //end of class
```

---

Revision #2

Created 26 September 2019 15:49:32 by Laptrinh.vn

Updated 12 April 2020 14:38:39 by Laptrinh.vn