

# Các kiểu biến trong Java

Chương này, chúng ta cùng tìm hiểu về các kiểu biến trong Java. Biến là tên một khu nhớ. Có ba kiểu biến: biến cục bộ (còn gọi là biến local), biến instance và biến static.

Bạn phải khai báo tất cả các biến trước khi chúng có thể được sử dụng. Form cơ bản của một khai báo biến như sau:

```
kieu_du_lieu bien [= giatri][, bien [= giatri] ...] ;
```

Ở đây, *kieu\_du\_lieu* là một kiểu dữ liệu của Java và *bien* là tên của biến. Để khai báo nhiều hơn một biến với kiểu cụ thể, bạn có thể sử dụng một danh sách biến phân biệt nhau bởi dấu phẩy.

Dưới đây là ví dụ về khai báo và khởi tạo biến hợp lệ trong Java:

```
int a, b, c;           // Khai bao ba bien kieu int la a, b, va c. int a = 5, b = 7; // Vi du ve khai tao bien byte A = 11;           // Khai tao mot bien kieu byte ten la A. double pi = 3.14159; // Khai bao va gan mot gia tri cua PI. char nam = 'b';       // Bien nam duoc khai tao voi gia tri 'a'.
```

## Biến local trong Java

- Các biến được khai báo trong các phương thức, constructor, hoặc khối.
- Các biến được tạo khi phương thức, constructor hoặc khối được nhập và biến bị hủy khi phương thức, constructor hoặc khối kết thúc.
- Chỉ định truy cập (access modifier) có thể không được sử dụng cho các biến local.
- Các biến local chỉ nhìn thấy trong phương thức, constructor, hoặc khối được khai báo.
- Các biến local được thực thi nội bộ.
- Không có giá trị mặc định nào cho các biến local, vì thế các biến local nên được khai báo và một giá trị khởi tạo nên được gán trước khi sử dụng.

## Ví dụ:

Ở đây, *tuoi* là biến local. Nó được định nghĩa bên trong phương thức *tuoiCon()* và phạm vi của nó được giới hạn chỉ trong phương thức này.

<http://vietjack.com/>

Trang chia sẻ các bài học online miễn phí

```
public class Test{    public void tuoiCon(){        int tuoi = 0;        tuoi =  
tuoi + 10;        System.out.println("Tuoi con la : " + tuoi);    }  
public static void main(String args[]){        Test test = new Test();  
test.tuoiCon();    } }
```

## Ví dụ:

Ví dụ sau sử dụng *tuoi* mà không khởi tạo nó, vì thế nó sẽ tạo một lỗi tại thời gian biên dịch.

```
public class Test{    public void tuoiCon(){        int tuoi;        tuoi = tuoi  
+ 10;        System.out.println("Tuoi con la : " + tuoi);    }    public  
static void main(String args[]){        Test test = new Test();  
test.tuoiCon();    } }
```

Nó sẽ cho một lỗi sau trong khi biên dịch:

```
Test.java:4:variable number might not have been initialized tuoi = tuoi + 10;  
^ 1 error
```

## Biến instance (hay thuộc tính) trong Java

- Các biến instance được khai báo trong một lớp, nhưng ở bên ngoài một phương thức, constructor hoặc bất kỳ khối nào.
- Khi một không gian nhớ được cấp phát cho một đối tượng trong một heap (một dạng cấu trúc hàng đợi có thứ tự ưu tiên), một slot cho mỗi giá trị biến instance được tạo ra.
- Các biến instance được tạo khi một đối tượng được tạo bởi sử dụng từ khóa new và bị hủy khi đối tượng bị hủy.
- Các biến instance giữ các giá trị mà phải được tham chiếu bởi nhiều hơn một phương thức, constructor hoặc khối, hoặc các phần chủ yếu của trạng thái đối tượng mà phải có mặt xuyên suốt lớp đó.
- Các biến instance có thể được khai báo trong mức độ lớp trước hoặc sau khi sử dụng.
- Chỉ định truy cập (access modifier) có thể được cung cấp cho các biến instance.
- Các biến instance là nhìn thấy với tất cả các phương thức, constructor, và khối trong lớp. Tuy nhiên, tính nhìn thấy cho các lớp phụ có thể được cung cấp cho những biến này với sự sử dụng của chỉ định truy cập.

- Các biến instance có các giá trị mặc định. Với các số, giá trị mặc định là 0, với Boolean là false và với đối tượng là null. Các giá trị có thể được gán trong khi khai báo hoặc trong constructor.
- Các biến instance có thể được truy cập một cách trực tiếp bởi việc gọi tên biến bên trong lớp đó. Tuy nhiên, với các phương thức static và các lớp khác nhau (khi biến instance được cung cấp khả năng truy cập), nó nên được gọi bởi sử dụng tên đầy đủ hợp lệ như sau: *ObjectReference.VariableName*.

## Ví dụ:

```
import java.io.*; public class Student{ // Bien instance nay la nhin thay
cho bat ky lop con nao. public String ten; // Bien hocphi la chi nhin
thay cho lop Student. private double hocphi; // Bien ten duoc gan
trong constructor. public Student (String tenSV){ ten = tenSV; }
// Bien hocphi duoc gan mot gia tri. public void setHocPhi(double hp){
hocphi = hp; } // Phuong thuc nay in chi tiet ve Student. public
void inThongTin(){ System.out.println("Ho va ten: " + ten );
System.out.println("Hoc phi: " + hocphi); } public static void
main(String args[]){ Student sv1 = new Student("Nguyen Van Doan");
sv1.setHocPhi(4000); sv1.inThongTin(); } }
```

Nó sẽ cho kết quả sau:

```
Ho va ten: Nguyen Van Doan Hoc phi: 4000.0
```

## Các biến Class/static trong Java:

- Các biến class cũng được biết như là các biến static được khai báo với từ khóa static trong một lớp, nhưng ở bên ngoài một phương thức, constructor hoặc một khối.
- Sẽ chỉ có một bản sao của mỗi biến class cho mỗi lớp, bất chấp việc bao nhiêu đối tượng được tạo từ nó.
- Các biến static hiếm khi được sử dụng, ngoài việc được khai báo như là các hằng số. Các hằng số là các biến, mà được khai báo như là các biến static, biến final, biến chung/riêng. Các biến hằng số không bao giờ thay đổi từ giá trị khởi tạo của chúng.
- Các biến static được lưu giữ trong bộ nhớ static.

- Các biến static được tạo khi chương trình bắt đầu và bị hủy khi chương trình kết thúc.
- Tính nhìn thấy là tương tự như các biến instance. Tuy nhiên, hầu hết biến static được khai báo chung khi chúng phải là có sẵn cho việc sử dụng của lớp.
- Giá trị mặc định là giống với các biến instance. Với các số, giá trị mặc định là 0; với Boolean là false, và với đối tượng là null. Các giá trị có thể được gán trong khi khai báo hoặc trong constructor. Ngoài ra, các giá trị có thể được gán trong các khối khởi tạo static đặc biệt.
- Các biến static có thể được truy cập bởi việc gọi tên lớp: *ClassName.VariableName*.
- Khi khai báo các biến class như public static final, thì các tên biến ở dạng chữ hoa. Nếu các biến static không là public và final, thì cú pháp đặt tên giống với các biến local và biến instance.

## Ví dụ:

```
import java.io.*; public class Student{ // Bien hocphi la mot bien private
static private static double hocphi; // FACULTY la mot constant
public static final String FACULTY = "KhoaIT "; public static void
main(String args[]){ hocphi = 4000;
System.out.println(FACULTY+"hocphi trung binh:"+hocphi); } }
```

Nó sẽ cho kết quả sau:

```
KhoaIT hocphi trung binh:4000
```

**Ghi chú:** Nếu các biến được truy cập từ lớp bên ngoài, hằng số nên được truy cập như ở dạng: Student.FACULTY