


Kiểm thu nhập bằng việc làm bạn yêu thích

 Google AdSense

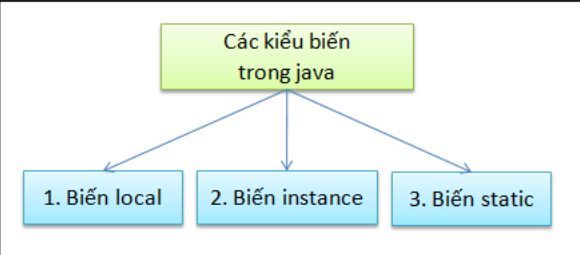
TÌM HIỂU THÊM

Biến trong java

☯ Sự khác nhau giữa JVM, JRE và JDK

Ép kiểu trong java ☯

Trong java, biến là tên của vùng nhớ. Có 3 kiểu biến trong java, bao gồm biến local (biến địa phương), biến instance (biến toàn cục) và biến static.



Khai báo biến trong java

Cú pháp khai báo biến:

```
DataType varName [ = value] [, varName2] [ = value2]...;
```

Trong đó, DataType là kiểu dữ liệu của biến, varName là tên biến.

Quy tắc đặt tên biến trong java:

- Chỉ được bắt đầu bằng một ký tự(chữ), hoặc một dấu gạch dưới(_), hoặc một ký tự dollar(\$)
- Tên biến không được chứa khoảng trắng
- Bắt đầu từ ký tự thứ hai, có thể dùng ký tự(chữ), dấu gạch dưới(_), hoặc ký tự dollar(\$)
- Không được trùng với các từ khóa
- Có phân biệt chữ hoa và chữ thường

Ví dụ về khai báo biến trong java:

```
package vn.viettuts.bienvadulieu;

public class Bien {
    public static float PI = 3.14f;    // Đây là biến static
    int n;                             // Đây là biến instance
}
```

Recent Updates

- ☯ Liệt kê số Fibonacci nhỏ hơn n và là số nguyên tố trong C++
- ☯ Tìm số thuận nghịch trong C++
- ☯ Tính tổng của các chữ số của một số nguyên n trong C++
- ☯ Phân tích số nguyên n thành tích các số nguyên tố trong C++
- ☯ Liệt kê tất cả số nguyên tố có 5 chữ số trong C++
- ☯ Liệt kê n số nguyên tố đầu tiên trong C++
- ☯ Liệt kê tất cả các số nguyên tố nhỏ hơn n trong C++
- ☯ Tìm USCLN và BSCNN của 2 số trong C++
- ☯ Giải phương trình bậc 2 trong C++
- ☯ Chuyển đổi hệ cơ số trong C++
- ☯ Tính giai thừa trong C++
- ☯ Dãy số Fibonacci trong C++

```
public Bien () {  
    char c = 'c';           // Đây là biến local  
}  
}
```

1. Biến local trong java

- 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.

Ví dụ 1: Khởi tạo biến local:

```
package vn.viettuts.bienvadulieu;  
  
public class Bien {  
  
    public void sayHello() {  
        int n = 10;           // Đây là biến local  
        System.out.println("Giá trị của n là: " + n);  
    }  
  
    public static void main(String[] args) {  
        Bien bienLocal = new Bien();  
        bienLocal.sayHello();  
    }  
}
```

Kết quả:

```
Gia tri cua n la: 10
```

Ví dụ 2: Không khởi tạo biến local:

```
package vn.viettuts.bienvadulieu;  
  
public class Bien {  
  
    public void sayHello() {  
        int n;               // Đây là biến local  
        System.out.println("Giá trị của n là: " + n);  
    }  
  
    public static void main(String[] args) {  
        Bien bienLocal = new Bien();  
        bienLocal.sayHello();  
    }  
}
```

Kết quả:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
The local variable n may not have been initialized
```

VietTuts on facebook





Khi không khởi tạo biến local, chương trình java sẽ báo lỗi khi biên dịch.

2. Biến biến instance (biến toàn cục) trong java

- 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.

Ví dụ về biến instance trong java:

```
package vn.vietttuts.bienvadulieu;

public class Sinhvien {
    // biến instance "ten" kiểu String, có giá trị mặc định là null
    public String ten;

    // biến instance "tuoi" kiểu Integer, có giá trị mặc định là 0
    private int tuoi;

    // sử dụng biến ten trong một constructor
    public Sinhvien(String ten) {
        this.ten = ten;
    }

    // sử dụng biến tuoi trong phương thức setTuoi
    public void setTuoi(int tuoi) {
        this.tuoi = tuoi;
    }

    public void showStudent() {
        System.out.println("Ten   : " + ten);
        System.out.println("Tuoi  : " + tuoi);
    }

    public static void main(String args[]) {
        Sinhvien sv = new Sinhvien("Nguyen Van A");
        sv.setTuoi(21);
        sv.showStudent();
    }
}
```

Kết quả:

```
Ten   : Nguyen Van A
Tuoi  : 21
```

3. Biến static trong java

- 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".

Ví dụ về biến static trong java:

```
package vn.viettuts.bienvadulieu;

public class Sinhvien {
    // biến static 'ten'
    public static String ten = "Nguyen Van A";

    // biến static 'tuoi'
    public static int tuoi = 21;

    public static void main(String args[]) {
        // Sử dụng biến static bằng cách gọi trực tiếp
        System.out.println("Ten : " + ten);

        // Sử dụng biến static bằng cách gọi thông qua tên class
        System.out.println("Ten : " + Sinhvien.tuoi);
    }
}
```

Kết quả:

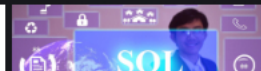
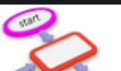
```
Ten : Nguyen Van A
Ten : 21
```

☯ Sự khác nhau giữa JVM, JRE và JDK

Ép kiểu trong java ☯

Bài viết liên quan:

- Biến trong java
- Các kiểu dữ liệu trong java
- Biến và kiểu dữ liệu trong java
- Ép kiểu trong Java
- Toán tử trong java
- Bộ nhớ stack và heap trong java





Phản Xạ Tiếng Anh Free

Bài tập java có lời giải -
rèn luyện tư duy logic
và oop - Viettuts

499k - Trọn Bộ VBA
Excel

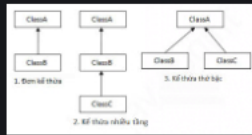
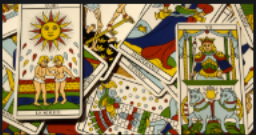
Nạp chồng phương
thức trong java (method
overloading in java

QC X3 English

viettuts.vn

QC Học Excel Online

viettuts.vn



Free 2019 Tarot Reading

Bộ nhớ stack và heap
trong java - học Java cơ
bản đến nâng cao

Tính đa hình trong java -
học Java miễn phí hay
nhất

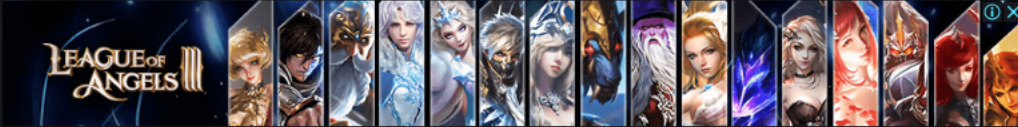
Tính kế thừa trong java -
học Java cơ bản đến
nâng cao

QC AstroMary

viettuts.vn

viettuts.vn

viettuts.vn



1 bình luận

Sắp xếp theo **Cũ nhất**



Thêm bình luận...



Tài Seul Bi

Khó hiểu quá

Thích · Phản hồi · 4 tuần

Plugin bình luận trên Facebook

Câu hỏi phỏng vấn

- 201 câu hỏi phỏng vấn java
- 25 câu hỏi phỏng vấn servlet
- 75 câu hỏi phỏng vấn jsp
- 52 câu hỏi phỏng vấn Hibernate
- 70 câu hỏi phỏng vấn Struts2
- 70 câu hỏi phỏng vấn Spring
- 57 câu hỏi phỏng vấn SQL

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...

