

Từ khóa final trong Java

Từ khóa final trong Java được sử dụng để hạn chế người dùng. Từ khóa final có thể được sử dụng trong nhiều ngữ cảnh: với biến, với phương thức và với lớp.

Từ khóa final có thể được áp dụng với các biến, một biến final mà không có giá trị nào được gọi là biến final trống hoặc biến final không được khởi tạo. Nó chỉ có thể được khởi tạo trong Constructor. Biến final trống cũng có thể là static mà sẽ chỉ được khởi tạo tổng khối static.

Biến final trong Java

Nếu bạn tạo bất cứ biến nào là final, bạn không thể thay đổi giá trị của biến final (nó sẽ là hằng số).

Ví dụ của biến final trong Java

Giả sử có một biến final có tên là speedlimit, chúng ta đang thay đổi giá trị của biến này, nhưng nó không thể bị thay đổi bởi vì một khi biến final đã được gán giá trị thì không bao giờ bị thay đổi.

```
class Bike9{  
    final int speedlimit=90;//bien final  
    void run(){  
        speedlimit=400;  
    }  
    public static void main(String args[]){  
        Bike9 obj=new Bike9();  
        obj.run();  
    }  
} //phan cuoi cua lop
```

Phương thức final trong Java

Nếu bạn tạo bất cứ phương thức nào là final, thì bạn không thể ghi đè nó.

Ví dụ của phương thức final

```
class Bike{  
    final void run(){System.out.println("running");}  
}
```

```
class Honda extends Bike{
    void run(){System.out.println("Chay an toan voi 100kmph");}

    public static void main(String args[]){
        Honda honda= new Honda();
        honda.run();
    }
}
```

Lớp final trong Java

Nếu bạn tạo bất cứ lớp nào là final thì bạn không thể kế thừa nó.

```
final class Bike{}

class Honda1 extends Bike{
    void run(){System.out.println("Chay an toan voi 100kmph");}

    public static void main(String args[]){
        Honda1 honda= new Honda();
        honda.run();
    }
}
```

Câu hỏi: Phương thức final có được kế thừa không?

Có, phương thức final được kế thừa nhưng bạn không thể ghi đè nó. Ví dụ:

```
class Bike{
    final void run(){System.out.println("dang chay...");}
}

class Honda2 extends Bike{
    public static void main(String args[]){
        new Honda2().run();
    }
}
```

Câu hỏi: Biến final trống hoặc không được khởi tạo là gì?

Một biến final mà không được khởi tạo tại thời điểm khai báo được gọi là biến final trống. Nếu bạn muốn tạo một biến mà được khởi tạo tại thời điểm tạo đối tượng và một khi nó đã được khởi tạo thì không thể bị thay đổi, thì biến final trống là hữu ích trong trường hợp này. Ví dụ như số thẻ PAN CARD của một nhân viên.

Nó chỉ có thể được khởi tạo trong Constructor. Sau đây là ví dụ về biến final trống:

```
class Student{
    int id;
    String name;
    final String PAN_CARD_NUMBER;
    ...
}
```

Câu hỏi: Chúng ta có thể khởi tạo biến final trống không?

Có, nhưng chỉ trong Constructor. Ví dụ:

```
class Bike10{
    final int speedlimit;//biến final trống

    Bike10(){
        speedlimit=70;
        System.out.println(speedlimit);
    }

    public static void main(String args[]){
        new Bike10();
    }
}
```

Biến static final trống trong Java

Một biến static final mà không được khởi tạo tại thời điểm khai báo thì đó là biến static final trống. Nó chỉ có thể được khởi tạo trong khối static.

Dưới đây là ví dụ về biến static final trong Java:

```
class A{
    static final int data;//bien static final trong
    static{ data=50;}
    public static void main(String args[]){
        System.out.println(A.data);
    }
}
```

Câu hỏi: Tham số final là gì?

Nếu bạn khai báo bất cứ tham số nào là final, thì bạn không thể thay đổi giá trị của nó.

```
class Bike11{
    int cube(final int n){
        n=n+2;//khong the duoc thay doi khi n la final
        n*n*n;
    }
    public static void main(String args[]){
        Bike11 b=new Bike11();
        b.cube(5);
    }
}
```

Chạy chương trình trên sẽ cho Compile Time Error.

Câu hỏi: Chúng ta có thể khai báo một constructor final không?

Không, bởi vì constructor không bao giờ được kế thừa.