

Từ khóa this trong Java

Có nhiều cách sử dụng từ khóa this trong Java. Trong Java, this là một biến tham chiếu mà tham chiếu tới đối tượng hiện tại.

Sự sử dụng của từ khóa this trong Java

Trong Java, từ khóa this có 6 cách sử dụng, như sau:

- Từ khóa this có thể được sử dụng để tham chiếu biến instance của lớp.
- this() có thể được sử dụng để triệu hồi Constructor của lớp hiện tại.
- Từ khóa this có thể được sử dụng để triệu hồi ngầm định phương thức lớp hiện tại.
- Từ khóa this có thể được truyền như là một tham số trong lời gọi phương thức.
- Từ khóa this có thể được truyền như là một tham số trong lời gọi Constructor.
- Từ khóa this cũng có thể được sử dụng để trả về instance của lớp hiện tại.

Đề nghị: Nếu bạn mới học về Java thì bạn nên chỉ theo dõi hai cách sử dụng của từ khóa this.

Từ khóa this có thể được sử dụng để tham chiếu biến instance của lớp hiện tại

Nếu có tính lưỡng nghĩa giữa biến instance và tham số, thì từ khóa this xử lý vấn đề này. Đầu tiên, chúng ta theo dõi ví dụ để hiểu vấn đề xảy ra khi không sử dụng từ khóa this.

Ví dụ

```
class Student10{
    int id;
    String name;

    Student10(int id,String name){
        id = id;
        name = name;
    }

    void display(){System.out.println(id+" "+name);}
}
```

```
public static void main(String args[]){
    Student10 s1 = new Student10(111,"Hoang");
    Student10 s2 = new Student10(321,"Thanh");
    s1.display();
    s2.display();
}
}
```

Trong ví dụ này, tham số và biến instance là giống nhau, và đó là lý do tại sao chúng ta sử dụng từ khóa this để phân biệt giữa các biến cục bộ và các biến instance.

Xử lý vấn đề trên bởi từ khóa this trong Java

```
//Vi du ve tu khoa this trong Java
class Student11{
    int id;
    String name;

    Student11(int id,String name){
        this.id = id;
        this.name = name;
    }

    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
        Student11 s1 = new Student11(111,"Hoang");
        Student11 s2 = new Student11(222,"Thanh");
        s1.display();
        s2.display();
    }
}
```

Nếu biến cục bộ và biến instance là khác nhau, thì không cần thiết sử dụng từ khóa this như trong chương trình sau đây:

```
class Student12{
```

```
int id;
String name;

Student12(int i,String n){
    id = i;
    name = n;
}

void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
    Student12 e1 = new Student12(111,"Hoang");
    Student12 e2 = new Student12(222,"Thanh");
    e1.display();
    e2.display();
}
}
```

this() có thể được sử dụng để triệu hồi constructor của lớp hiện tại

Lời gọi this() constructor có thể được sử dụng để triệu hồi Constructor của lớp hiện tại. Hướng tiếp cận này là tốt hơn nếu bạn có nhiều Constructor trong lớp và muốn tái sử dụng Constructor đó.

```
//Chương trình ví dụ lời gọi this() constructor (constructor chaining)

class Student13{
    int id;
    String name;
    Student13(){System.out.println("Constructor mặc định được gọi");}

    Student13(int id,String name){
        this (); //no được sử dụng để gọi constructor của lớp hiện tại.
        this.id = id;
        this.name = name;
    }
}
```

```
void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
    Student13 e1 = new Student13(111,"Hoang");
    Student13 e2 = new Student13(222,"Thanh");
    e1.display();
    e2.display();
}
}
```

Câu hỏi: Nơi để sử dụng lời gọi this() constructor?

Lời gọi this() constructor nên được sử dụng để tái sử dụng constructor. Nó duy trì dây chuyền giữa các Constructor, và được sử dụng cho Constructor Chaining. Bạn theo dõi ví dụ sau để hiểu hơn về điều này.

```
class Student14{
    int id;
    String name;
    String city;

    Student14(int id,String name){
        this.id = id;
        this.name = name;
    }
    Student14(int id,String name,String city){
        this(id,name); //Bay gio khong can khoi tao id va name
        this.city=city;
    }
    void display(){System.out.println(id+" "+name+" "+city);}

    public static void main(String args[]){
        Student14 e1 = new Student14(111,"Hoang");
        Student14 e2 = new Student14(222,"Thanh","NamDinh");
    }
}
```

```
e1.display();
e2.display();
}
}
```

Qui tắc: Lời gọi tới this() phải là lệnh đầu tiên trong Constructor.

Trong chương trình ví dụ sau sẽ cho một thông báo lỗi tại compile time.

```
class Student15{
    int id;
    String name;
    Student15(){System.out.println("Constructor mac dinh duoc trieu hoi");}

    Student15(int id,String name){
        id = id;
        name = name;
        this (); //phai la lenh dau tien
    }
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
        Student15 e1 = new Student15(111,"Hoang");
        Student15 e2 = new Student15(222,"Thanh");
        e1.display();
        e2.display();
    }
}
```

Từ khóa this() có thể được sử dụng để triệu hồi ngầm định phương thức lớp hiện tại

Bạn có thể triệu hồi phương thức của lớp hiện tại bởi sử dụng từ khóa this. Nếu bạn không sử dụng từ khóa this, trình biên dịch Compiler tự động thêm từ khóa this trong khi gọi phương thức. Bạn theo dõi ví dụ sau:

```
class S{
    void m(){
        System.out.println("Phuong thuc duoc trieu hoi");
    }
    void n(){
        this.m(); //khong can boi vi compiler thuc hien no cho ban.
    }
    void p(){
        n(); //compiler se them this de trieu hoi phuong thuc n() duoi dang this.n()
    }
    public static void main(String args[]){
        S s1 = new S();
        s1.p();
    }
}
```

Từ khóa this có thể được truyền như là một tham số trong phương thức

Cách này chủ yếu được sử dụng trong xử lý sự kiện. Bạn theo dõi ví dụ:

```
class S2{
    void m(S2 obj){
        System.out.println("Phuong thuc duoc trieu hoi");
    }
    void p(){
        m(this);
    }

    public static void main(String args[]){
        S2 s1 = new S2();
        s1.p();
    }
}
```

Từ khóa this có thể được truyền như là tham số trong lời gọi constructor

Bạn cũng có thể truyền từ khóa this trong constructor. Nó là hữu ích nếu chúng ta phải sử dụng một đối tượng trong nhiều lớp. Bạn theo dõi ví dụ:

```
class B{
    A4 obj;
    B(A4 obj){
        this.obj=obj;
    }
    void display(){
        System.out.println(obj.data); //su dung thanh vien du lieu cua lop A4
    }
}

class A4{
    int data=10;
    A4(){
        B b=new B(this);
        b.display();
    }
    public static void main(String args[]){
        A4 a=new A4();
    }
}
```

Từ khóa this có thể được sử dụng để trả về instance của lớp hiện tại

Chúng ta có thể trả về từ khóa this như một lệnh từ phương thức. Trong tình huống này, kiểu trả về của phương thức phải là kiểu class (không là kiểu gốc). Bạn theo dõi ví dụ:

Cú pháp của this mà có thể được trả về như là một lệnh.

```
kiểu_tra_ve ten_phuong_thuc(){
```

```
return this;
}
```

Ví dụ

```
class A{
A getA(){
return this;
}
void msg(){System.out.println("Hello World");}
}

class Test1{
public static void main(String args[]){
new A().getA().msg();
}
}
```

Chúng ta cùng chứng minh rằng từ khóa `this` tham chiếu tới biến instance của lớp hiện tại. Trong chương trình sau, chúng ta in biến tham chiếu và `this`, kết quả của chúng là giống nhau.

```
class A5{
void m(){
System.out.println(this);//in ra cùng tham chiếu ID
}

public static void main(String args[]){
A5 obj=new A5();
System.out.println(obj);//in tham chiếu ID

obj.m();
}
}
```