

Từ khóa super trong Java

Từ khóa super trong Java là một biến tham chiếu mà được sử dụng để tham chiếu đến đối tượng lớp cha gần nhất. Bất cứ khi nào bạn tạo instance (sự thể hiện) của lớp con, một instance của lớp cha được tạo ngầm định, được tham chiếu bởi biến super.

Sự sử dụng của từ khóa super trong Java

- super được sử dụng để tham chiếu biến instance của lớp cha gần nhất.
- super() được sử dụng để triệu hồi Constructor của lớp cha gần nhất.
- super được sử dụng để triệu hồi phương thức của lớp cha gần nhất.

Phần tiếp theo, chúng ta sẽ tìm hiểu chi tiết từng sự sử dụng của từ khóa super trong Java.

super được để tham chiếu biến instance của lớp cha gần nhất

Vấn đề xảy ra nếu không có từ khóa super:

```
class Vehicle{
    int speed=50;
}
class Bike3 extends Vehicle{
    int speed=100;
    void display(){
        System.out.println(speed); //se in speed của Bike
    }
    public static void main(String args[]){
        Bike3 b=new Bike3();
        b.display();
    }
}
```

Trong ví dụ trên, cả hai lớp Vehicle và Bike có cùng thuộc tính speed chung. Biến instance của lớp hiện tại được tham chiếu bởi instance theo mặc định, nhưng mình phải tham chiếu tới biến instance của lớp cha, và đó là tại sao chúng ta sử dụng từ khóa super để phân biệt giữa biến instance của lớp cha và biến instance của lớp hiện tại.

Xử lý vấn đề với từ khóa *super* trong Java:

```
//Chương trình ví dụ từ khóa super

class Vehicle{
    int speed=50;
}

class Bike4 extends Vehicle{
    int speed=100;

    void display(){
        System.out.println(super.speed); //bây giờ sẽ in speed của Vehicle
    }

    public static void main(String args[]){
        Bike4 b=new Bike4();
        b.display();
    }
}
```

`super()` được sử dụng để triệu hồi Constructor của lớp cha gần nhất

Bạn theo dõi ví dụ sau:

```
class Vehicle{
    Vehicle(){System.out.println("Vehicle được tạo");}
}

class Bike5 extends Vehicle{
    Bike5(){
        super();//sẽ triệu hồi constructor của lớp cha
        System.out.println("Bike được tạo");
    }
}
```

```
public static void main(String args[]){  
    Bike5 b=new Bike5();  
  
}  
}
```

Ghi chú: super() được tự động thêm vào trong mỗi Constructor của lớp bởi Compiler.

Qua các chương trước, chúng ta đã biết rằng constructor mặc định được cung cấp bởi Compiler nhưng nó cũng thêm super() cho lệnh đầu tiên. Nếu bạn đang tạo constructor cho riêng mình và bạn không có this() hoặc super() như là lệnh đầu tiên, thì Compiler sẽ cung cấp super() như là lệnh đầu tiên của Constructor đó.

Một ví dụ khác về từ khóa **super**, trong ví dụ này super() được cung cấp **ngầm định** bởi Compiler.

```
class Vehicle{  
    Vehicle(){System.out.println("Vehicle duoc tao");}  
}  
  
class Bike6 extends Vehicle{  
    int speed;  
    Bike6(int speed){  
        this.speed=speed;  
        System.out.println(speed);  
    }  
    public static void main(String args[]){  
        Bike6 b=new Bike6(10);  
    }  
}
```

super được sử dụng để triệu hồi phương thức của lớp cha gần nhất

Từ khóa super cũng có thể được sử dụng để triệu hồi phương thức lớp cha gần nhất. Nó nên được sử dụng trong lớp con mà có chứa cùng phương thức như lớp cha, như trong ví dụ sau:

```
class Person{
```

```
void message(){System.out.println("Chao mung");}

}

class Student16 extends Person{
void message(){System.out.println("Chao mung ban den voi java");}

void display(){
message();//se triệu hồi phương thức message() của lớp hiện tại
super.message();//se triệu hồi phương thức message() của lớp cha
}

public static void main(String args[]){
Student16 s=new Student16();
s.display();
}
}
```

Trong ví dụ trên, cả hai lớp Student và Person cùng có phương thức message(), nếu chúng ta gọi phương thức message() từ lớp Student, nó sẽ gọi phương thức message() của lớp Student chứ không phải của lớp Person bởi vì quyền ưu tiên cục bộ.

Trong tình huống mà không có phương thức nào của lớp con giống lớp cha, thì không cần sử dụng từ khóa super. Trong ví dụ dưới đây, phương thức message() được triệu hồi từ lớp Student, nhưng lớp Student không có phương thức message() này, vì thế bạn có thể trực tiếp gọi phương thức message().

Ví dụ chương trình *không cần* từ khóa super:

```
class Person{
void message(){System.out.println("Chao mung");}
}

class Student17 extends Person{

void display(){
```

```
message();//se trieu hoi phuong thuc message() cua lop cha
}

public static void main(String args[]){
    Student17 s=new Student17();
    s.display();
}
}
```