

Từ khóa super trong java

☺ Từ khóa this trong java

Từ khóa final trong java ☺

Từ khóa **super** trong java là một biến tham chiếu được sử dụng để tham chiếu trực tiếp đến đối tượng của lớp cha gần nhất.

Bất cứ khi nào bạn tạo ra instance(thể hiện) của lớp con, một instance của lớp cha được tạo ra ngầm định, nghĩa là được tham chiếu bởi biến **super**.

Trong java, từ khóa **super** có 3 cách sử dụng như sau:

1. Từ khóa super được sử dụng để tham chiếu trực tiếp đến biến instance của lớp cha gần nhất.
2. super() được sử dụng để gọi trực tiếp Constructor của lớp cha.
3. Từ khóa super được sử dụng để gọi trực tiếp phương thức của lớp cha.

1. Tham chiếu trực tiếp đến biến instance của lớp cha.

Từ khóa super được sử dụng để tham chiếu trực tiếp đến biến instance của lớp cha.

Ví dụ: khi không sử dụng từ khóa super

```

class Vehicle {
    int speed = 50;
}

public class Bike extends Vehicle {
    int speed = 100;

    void display() {
        System.out.println(speed); //in speed của lớp Bike
    }

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

Kết quả:

Recent Updates

- ☞ Hàm WeekDayName() trong VBA
- ☞ Hàm WeekDay() trong VBA
- ☞ Hàm MonthName() trong VBA
- ☞ Hàm Year() trong VBA
- ☞ Hàm Month() trong VBA
- ☞ Hàm Day() trong VBA
- ☞ Hàm IsDate() trong VBA
- ☞ Hàm FormatDateTime() trong VBA
- ☞ Hàm DateSerial() trong VBA
- ☞ Hàm PartDate() trong VBA
- ☞ Hàm DateDiff() trong VBA
- ☞ Hàm DateAdd() trong VBA

VietTuts on facebook



Học Java Cơ Bản Đến
 2.136 lượt thích
 it works..... Why?

Ví dụ: khi sử dụng từ khóa super

```
class Vehicle {
    int speed = 50;
}

public class Bike2 extends Vehicle {
    int speed = 100;

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

    public static void main(String args[]) {
        Bike2 b = new Bike2();
        b.display();
    }
}
```

Kết quả:

50

2. Sử dụng super()

Trong java, super() được sử dụng để gọi trực tiếp Constructor của lớp cha.


```
class Vehicle {
    Vehicle() {
        System.out.println("Vehicle is created");
    }
}

class Bike2 extends Vehicle {
    Bike2() {
        super(); // gọi Constructor của lớp cha
        System.out.println("Bike is created");
    }

    public static void main(String args[]) {
        Bike2 b = new Bike2();
    }
}
```

Kết quả:

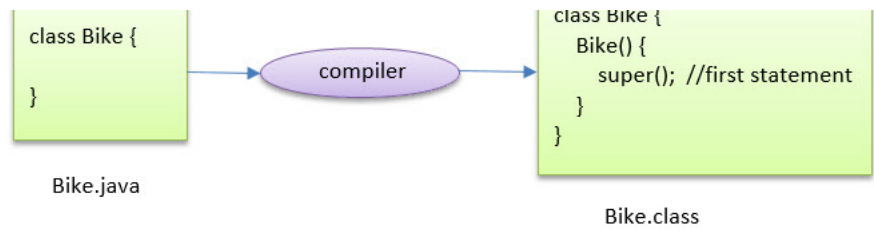
Vehicle is created
Bike is created

 Note: super() được tự động thêm vào mỗi Constructor của class bởi trình biên dịch.



1 người bạn thích nội dung này





Như chúng ta đã biết Constructor được tạo ra tự động bởi trình biên dịch nhưng nó cũng thêm `super()` vào câu lệnh đầu tiên. Nếu bạn tạo Constructor và bạn không có `this()` hoặc `super()` ở dòng lệnh đầu tiên, trình biên dịch sẽ cung cấp `super()` của Constructor.

Một ví dụ khác của từ khóa `super` nơi `super()` được cung cấp ngầm định bởi trình biên dịch.

```
class Vehicle {
    Vehicle() {
        System.out.println("Vehicle is created");
    }
}

public 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);
    }
}
```

Kết quả:

```
Vehicle is created
10
```

3. `super` được sử dụng để gọi trực tiếp phương thức của lớp cha.

Từ khóa `super` cũng có thể được sử dụng để gọi phương thức của lớp cha. Nó nên được sử dụng trong trường hợp lớp chứa các phương thức tương tự như lớp cha như trong ví dụ dưới đây:

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

public class Student16 extends Person {
    void message() {
        System.out.println("welcome to java");
    }

    void display() {
```

```
        message();// gọi phương thức message() của lớp hiện tại
        super.message();// gọi phương thức message() của lớp cha
    }

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

Kết quả:

```
welcome to java
welcome
```

Trong ví dụ trên cả hai lớp Student và Person đều có phương thức message(). Nếu bạn gọi phương thức message() từ lớp Student thì phương thức message của Student sẽ được thực thi vì phương thức ở local sẽ được ưu tiên.

Trong TH không có phương thức ở class con giống class cha thì không cần phải sử dụng super. Ví dụ:

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

public class Student17 extends Person {

    void display() {
        message();// will invoke parent class message() method
    }

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

Kết quả:

```
welcome
```

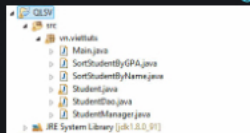
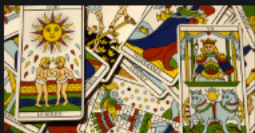
Từ khóa this trong java

Từ khóa final trong java

Bài viết liên quan:

- Từ khóa this trong java
- Từ khóa super trong java
- Từ khóa final trong java
- Từ khóa static trong java
- Access Modifier trong Java
- Tính đóng gói trong java
- Tính kế thừa trong java

- Tính đa hình trong java



Free 2019 Tarot Reading

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

DellG3 3579: mạnh mẽ, giá rẻ

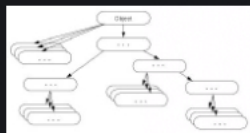
Bài tập quản lý sinh viên trong java - bài tập java có lời giải

QC AstroMary

viettuts.vn

QC prolap.vn

viettuts.vn



Học Nói Tiếng Anh Miễn Phí

Interface trong java - học Java cơ bản đến nâng cao

Hibernate là gì

Lớp Object trong java - học Java miễn phí hay nhất

QC x3english

viettuts.vn

viettuts.vn

viettuts.vn



0 bình luận

Sắp xếp theo

Cũ nhất



Thêm bình luận...

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>

