链滴

java 中关键字 super 表示的真正对象

作者：huihui
原文链接：https://ld246.com/article/1482395717052
来源网站：链滴

Java中关键字 super表示的真正对象

java中的super，大家都知道是表示一个父类的引用。上次群里见到一个网友询问 super.getClass().getName()的输出问题，大部分都知道输出的是当前这个类的类名。而不是父类的名称。关于这个问题解释很多，基本都是说getClass()是一个final方法，说这个方法都是调用超父类Object的方法。这个释很好，也容易理解，不过，我们从super这个关键词的本质入手，就能更清楚，为什么super.getClass().getName()会输出当前类的名称了。

先定义两个类，一个父类，一个继承的子类。

父类：

[java] view plain copy

```
1. public class Parent {
2.
 public String name;
3.
 private int code;
4.
 public Parent parent;
5.
 //定义几个父类成员属性，parent将指向this父类对象
6.
 public Parent()
7.
 {
8.
    //parent指向this，就是当前实例的父类对象，并输出hashcode和给code属性赋值
9.
    parent=this;
10.
    code=1001;
```

```
11.
    System.out.println("Parent's code is:"+code);
12.
    System.out.println("Parent hashcode is:"+this.hashCode());
13.
 }
14.
 //定义一个重载的构造方法，用于测试使用了哪个父类构造方法
15.
 public Parent(String name){
16.
    parent=this;
17.
    this.name=name;
18.
    System.out.println("Parent's name is"+name);
19.
    code=1002;
20.
    System.out.println("Parent's code is"+code);
21.
22.
    System.out.println("Parent hashcode is:"+this.hashCode());
23.
 }
24.
```

```
 public int getCode()
25.

 {
26.

    //打印父类code属性值，
27.

    System.out.println("Parent :print code :"+code);
28.

    return code;
29.

 }
30. }
```

子类：

[java] view plain copy

```
1. public class Child extends Parent{
2.

 public String childName;
3.

 private int childCode;
4.

 //定义两个类的不同修饰符的属性
5.

 public Child(String childName)
6.

 {
7.
```

```
    //给属性赋值
8.
    this.childName=childName;
9.
    //输出属性值，和类的hashcode值
10.
    System.out.println("child's childName is:"+childName);
11.
    System.out.println("child hashcode is:"+this.hashCode());
12.
}
13.
//测试方法
14.
public void test()
15.
{
16.
    //通过super获取父类中的parent，这个成员就表示了父类对象。
17.
    Child testChild=(Child)super.parent;
18.
    //强转parent为子类类型。并输出子类中定义的属性，和获取父类getCode()方法
19.
    System.out.println("testChild name is:"+testChild.childName);
20.
```

```
    testChild.getCode();
21.

 }
22.

 public static void main(String[] args) {
23.

    //实例化
24.

    Child c=new Child("window");
25.

    c.test();
26.

 }
27. }
```

运行输出结果：

Parent's code is:1001

//首先输出这句，可以表示，父类默认的构造方法执行了。

Parent hashcode is:11985823

//这里输出的父类this对象的hashcode码

child's childName is:window

//调用子类的构造方法，输出childName和hashcode码值，大家可以看到，父类的this对象和子类实的对象，hashcode码是相同的。

child hashcode is:11985823

//这里执行test()方法的结果。

//使用Child testChild=(Child)super.parent 强制转换父类中指向父类this对象的parent对象为子类Cild对象，并输出子类childName属性值。

testChild name is:window

//输出结果上可以看出，这个parent其实就是子类的实例对象。应该说内存中的对象是同一个，只是同的表示方式。

//执行testChild.getCode(); 调用父类的getCode()方法，输出的和父类构造方法中的code值是相同。

Parent :print code :1001

从上面可以看出super这个关键字所表示的引用父类对象，其实和当前实例化的子类对象在内存中是一个对象，虽然不知道sun 的虚拟机是如何实现这个关键字的，但是这个super的等同效果就是(Paren)this,这里的this是当前代表当前实例化的子类对象。

通过在父类中添加一个Parent类型的成员，来指向父类实例化的那个this对象，达到引用父类对象的的，使用super.parent来获得父类parent对象的引用。

运行的结果表明，这个parent应用的其实就是当前子类实例对象，通过强制转换为子类类型，这个父Parent类型的parent也可以读取成员childName属性值。就表明了他们在内存中是同一个对象。而他们的hashcode值也是相同的。

java说明中指出，super是代表父类对象的引用，而super()表示应用父类的默认构造方法，看起来这个uper和c中的define有点相同作用的效果。在这个例子中，super代表的父类对象的引用，和父类中paent代表的都是父类对象的引用，parent其实就是当前内存中子类对象的引用，如同(Parent)this一样这样就可以解释为什么使用(Child)super.parent可以得到子类成员childName。

回到开始的问题，super.getClass().getName()，这样，就很容易解释他为什么是输出当前类的名称，因为构造一个实例化对象，其中，在父类和子类构造方法中，引用的对象都是同一个，都是当前实化的子类对象。super关键字应该充当了一个类型转化的作用。

熟悉c的知道，c中经常使用强制转换指针类型来引用一些结构或变量的部分数据，如通过强制转换为同结构类型，来引用不同数据大小的结构体。这里的应用效果应该等同，通过(Parent)Child可以通过换为父类类型而只引用父类类型的那一部分数据。

在java中new一个对象，和c或c++这些和内存打交道的语言一样，都是会分配内存，在c中可能更直一点，我们在此不讨论到底是分配了多大的内存问题。

在new一个子类的时候，上面例子看到，父类的成员同样也有赋值初始化，说明，同样在内存中也有存父类的信息空间，（Object类的不讨论）。