

# 继承关系中this指代的是谁？

多重继承中，基类（父类）中的this指的就是最低的继承类的对象。

为什么会出现这样的结果？先看看下边的例子

```
1 package cn.itcast.basedao.demo3;
2
3 import java.lang.reflect.ParameterizedType;
4 import java.lang.reflect.Type;
5
6 import org.hibernate.Session;
7
8 /**
9  * 通用的DAO的编写：(泛型的反射的方式)
10 */
11 public class BaseDao<T> {
12
13     private Session session;
14     private Class<?> clazz;
15
16     public BaseDao(){
17         // 获得到子类的Class的对象。
18         Class<?> c = this.getClass();
19         System.out.println(c);
20         // 获得带有泛型的父类:BaseDao<Book>
21         Type type = c.getGenericSuperclass();
22         System.out.println(type);
23         ParameterizedType pType = (ParameterizedType) type;
24         // 通过参数化类型获得到具体实际类型参数Book
25         Type[] types = pType.getActualTypeArguments();
26         this.clazz = (Class<?>) types[0];
27     }
28
29     /**
30     * 通用的保存的方法
31     */
32     public void save(T t){
33         session.save(t);
34     }
35
36     /**
37     * 通用的修改的方法
38     */
39     public void update(T t){
40         session.update(t);
41     }
42
43     /**
44     * 通用的删除的方法：
45     */
```

```

46     public void delete(T t){
47         session.delete(t);
48     }
49
50     /**
51     * 通用的查询的方法:
52     */
53     public T findById(Integer id){
54         return (T)session.get(clazz, id);
55     }
56 }
57
1 package cn.itcast.basedao.demo3;
2
3 import cn.itcast.basedao.domain.Book;
4
5 /**
6 * 图书管理的DAO的类:
7 */
8 public class BookDao extends BaseDao<Book>{
9     public BookDao(){
10         super();
11     }
12 }
13

```

## 测试类

```

1 package cn.itcast.basedao.test;
2
3 import org.junit.Test;
4 import cn.itcast.basedao.demo3.BookDao;
5 /**
6 * 通用的DAO的测试类:
7 */
8 public class BaseDaoDemo1 {
9
10     @Test
11     public void demo1(){
12         BookDao bookDao = new BookDao();
13     }
14 }

```

现在来看看，在java中this调用谁，他就代表的是谁，由于子类在初始化的时候会先调用父类的构造器，这样就相当于，this代表的是子类的类。