

笑哼

[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅](#)[管理](#)

随笔 - 242 文章 - 0 评论 - 6

内部类访问局部变量的时候，为什么变量必须加上final修饰

这里的局部变量就是在类方法中的变量，能访问方法中变量的类当然也是局部内部类了。

我们都知道，局部变量在所处的函数执行完之后就释放了，但是内部类对象如果还有引用指向的话它是还存在的。例如下面的代码：

```
class Outer{
    public static void main(String[] args){
        Outer out = new Outer();
        Object obj = out.method();
    }

    Object method(){
        int locvar = 1;
        class Inner{
            void displayLocvar(){
                System.out.println("locvar = " + locvar);
            }
        }
        Object in = new Inner();
```

公告

昵称：笑哼

园龄：2年3个月

粉丝：24

关注：8

[+加关注](#)

2018年4月						
<	一	二	三	四	五	六
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

搜索

```
        return in;
    }
}
```

当out.method()方法执行结束后，局部变量 locvar 就消失了，但是在method () 方法中 obj in = new Inner() 产生的 in 对象还存在引用obj，这样对象就访问了一个不存在的变量，是不允许的。这种矛盾是由局部内部类可以访问局部变量但是局部内部类对象和局部变量的生命周期不同而引起的。

局部内部类访问局部变量的机制

在java中，类是封装的，内部类也不例外。我们知道，非静态内部类能够访问外部类成员是因为它持有外部类对象的引用 Outer.this，就像子类对象能够访问父类成员是持有父类对象引用super一样。局部内部类也和一般内部类一样，只持有了Outer.this，能够访问外部类成员，但是它又是如何访问到局部变量的呢？

实际上java是将局部变量作为参数传给了局部内部类的构造函数，而将其作为内部类的成员属性封装在了类中。我们看到的内部类访问局部变量实际上只是访问了自己的成员属性而已，这和类的封装性是一致的。那么上面的代码实际上是这样：

```
Object method() {
    int locvar = 1;
    class Inner{
        private int obj;
        public Inner(int obj) {
            this.obj = obj;
        }
        void displayLocvar() {
            System.out.println("locvar = " + locvar);
        }
    }
    Object in = new Inner(locvar); //将locvar作为参数传给构造，以初始话成员
    return in;
}
```

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

58集团(22)
android(20)
dataStructure(2)
html_css_javascript
java(13)
javaWeb
java垃圾回收
JDBC(4)
jsp(8)
mysql(1)
others(7)
ssh(3)
办公软件(6)
编码规范
产品(15)
动漫(4)
多线程(4)
鸡汤(18)

那么问题又来了，我们写代码的目的是在内部类中直接控制局部变量和引用，但是java这么整我们就不高兴了，我在内部类中整天想着是在操作外部变量，结果你给整个副本给我，我搞半天丫是整我自己的东西啊？要是java不这么整吧，由破坏了封装性-----你个局部内部类牛啊，啥都没有还能看局部变量呢。这不是java风格，肯定不能这么干。这咋整呢？

想想，类的封装性咱们一定是要遵守的，不能破坏大局啊。但又要保证两个东西是一模一样的，包括对象和普通变量，那就使用final嘛，当传递普通变量的之前我把它变成一个常量给你，当传递引用对象的时候加上final就声明了这个引用就只能指着这一个对象了。这样就保证了内外统一。

转：<https://segmentfault.com/a/1190000002735907>

路漫漫其修远兮，吾将上下而求索

分类: java

好文要顶

关注我

收藏该文



笑哼

关注 - 8

粉丝 - 24

+加关注

« 上一篇：[Web 测试经验总结](#)

» 下一篇：[泛型的内部原理：类型擦除以及类型擦除带来的问题](#)

posted @ 2016-08-02 15:12 笑哼 阅读(3173) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- “第4届高校云计算大赛”来袭！超10万总奖金、50万创业支持等你来拿
- 投资方解读美团摩拜收购案 创业公司面临巨头如何生存
- 盲人到底如何使用智能手机？来看看小米、腾讯做出的努力

- 健康(24)
- 理财(5)
- 骑行(1)
- 软件测试(19)
- 设计模式
- 摄影(5)
- 算法(15)
- 无线通信(11)
- 武汉(8)
- 性能测试(1)
- 一道试题的学习(7)
- 英雄杀(1)
- 影视(4)
- 转转测试开发
- 自动化测试(7)

随笔档案

- 2017年7月 (3)
- 2017年4月 (2)
- 2017年2月 (26)
- 2017年1月 (8)
- 2016年12月 (10)
- 2016年11月 (29)
- 2016年10月 (25)
- 2016年9月 (18)
- 2016年8月 (32)
- 2016年7月 (42)
- 2016年6月 (2)
- 2016年5月 (1)
- 2016年4月 (8)
- 2016年3月 (26)
- 2016年2月 (10)

- YouTube准备推出全新儿童版App：人类员工推荐内容
- 去世后，你的QQ微信淘宝支付宝网游账号怎么办？
- » 更多新闻...

最新知识库文章:

- 写给自学者的入门指南
- 和程序员谈恋爱
- 学会学习
- 优秀技术人的管理陷阱
- 作为一个程序员，数学对你到底有多重要
- » 更多知识库文章...

文章分类

程序员面试宝典

最新评论

1. Re:java中两个Integer类型的值相比较的问题

@clearbug其实感觉最主要的就是这些包装类都是不可变的，用final修饰的，导致传引用时的效果相当于传值。...

--笑哼

2. Re:java中两个Integer类型的值相比较的问题

@笑哼看到了这个解释。。涨姿势了，原来如此：...

--clearbug

3. Re:java中两个Integer类型的值相比较的问题

@clearbug，那你可以先写下你期望的输出结果，然后在ide中运行一下试试，看看是否与你预期的结果相同。

--笑哼

4. Re:java中两个Integer类型的值相比较的问题

@笑哼我去，这里不是传址调用吗？Integer类型做方法参数只是传值调用？？？...

--clearbug

5. Re:java中两个Integer类型的值相比较的问题

@clearbug那么对于Integer作方法中的形参进行传递时，又会有怎样的变化呢？如：

```
public class Test{    public static void  
main(String[] arg.....
```

阅读排行榜

1. 电影发烧友必备知识-720P、1080P、4K的区别(20489)
2. java中两个Integer类型的值相比较的问题(15958)
3. 线程通信与进程通信的区别(6888)
4. 内部类访问局部变量的时候，为什么变量必须加上final修饰(3173)
5. （原）list中null的谨慎使用(3137)

评论排行榜

1. java中两个Integer类型的值相比较的问题(6)

推荐排行榜

1. 测试开发之路--一个小小工程师的回首(4)
2. 线程通信与进程通信的区别(2)
3. 如何规范小开发公司的测试流程(1)
4. Java Web基础——Action+Service +Dao 三层的功能划分(1)
5. 移动APP测试实战(书籍)学习(1)

