http://blog.csdn.net/edisonlg/article/details/7183373

**[html]** [view plain](http://blog.csdn.net/edisonlg/article/details/7183373) [copy](http://blog.csdn.net/edisonlg/article/details/7183373)

1. //只有定义成 final String obj 才不会报错
2. public void test(final String obj)
3. {
4. Thread myThread = new Thread() {
5. public void run() {
6. System.out.println(obj);
7. }
8. };
9. myThread.start();
10. }

**[html]** [view plain](http://blog.csdn.net/edisonlg/article/details/7183373) [copy](http://blog.csdn.net/edisonlg/article/details/7183373)

1. public class OuterClass {
2. public void outerDoSomething() {
3. final String localObj = "hello";
4. InnerClass innerObj = new class InnerClass {
5. public void innerDoSomething() {
6. // do with localObj
7. }
8. };
9. }
10. }

如果将一个访问了final的局部变量的内部类进行反编译，可以发现该变量是被作为构造函数的参数传入进去的，当然与之一起传入的参数还有外部类引用this。

实际经过编译，生成的字节码如下：

**[html]** [view plain](http://blog.csdn.net/edisonlg/article/details/7183373) [copy](http://blog.csdn.net/edisonlg/article/details/7183373)

1. public class OuterClass {
2. public void outerDoSomething(){
3. final String localObj = "hello";
4. class InnerClass{
5. private String obj;
6. public InnerClass(String obj) {
7. this.obj = obj;
8. }
9. public void innerDoSomething(){
10. // do with obj
11. }
12. };
13. InnerClass innerObj = new InnerClass(localObj);
14. }
15. }

**荒唐原因：Java竟然允许内部类对象访问局部变量！**

要知道局部变量的生命周期与局部内部类的对象的生命周期的不一致。局部变量当所处的函数执行结束后就已经死亡了，不存在了，但是局部内部类对象还可能一直存在（只要有人还引用该对象），这是就会出现了一个悲剧的结果，局部内部类对象访问一个已不存在的局部变量。

Java为了避免上述情况，才发明了上述机制，偷偷地将局部变量的引用放在内部类对象的成员变量中。

但是解决了生命周期的问题之后，又引出了另一个问题，因为局部变量和内部类中使用的变量是同一个对象呢？

比如下面的例子，而且如果不强制使用final，我们根本就不知道执行时，obj到底是哪个对象：

**[html]** [view plain](http://blog.csdn.net/edisonlg/article/details/7183373) [copy](http://blog.csdn.net/edisonlg/article/details/7183373)

1. public class helloWorld
2. {
3. public void test()
4. {
5. String obj = "hello"; // 不使用final
6. Thread myThread = new Thread() {
7. public void run() {
8. System.out.println(obj); // ojb到底是哪个？
9. }
10. };
11. myThread.start();
12. }
14. public static void main(String [] argv)
15. {
16. helloWorld hello = new helloWorld();
17. hello.test();
18. }
19. }

又比如下面的例子，

**[html]** [view plain](http://blog.csdn.net/edisonlg/article/details/7183373) [copy](http://blog.csdn.net/edisonlg/article/details/7183373)

1. public class helloWorld
2. {
3. public void test()
4. {
5. String obj = "hello"; // 不使用final
6. Thread myThread = new Thread() {
7. public void run() {
8. }
9. };
10. myThread.start();
11. System.out.println(obj); // obj到底是什么？
12. }
14. public static void main(String [] argv)
15. {
16. helloWorld hello = new helloWorld();
17. hello.test();
18. }
19. }