


lambda表达式或者匿名函数中为什么要求外部变量为final

转载指尖的星光xl 于 2019-03-09 23:09:54 发布 12339 收藏 15

分类专栏: corejava

 corejava 专栏收录该内容

0 订阅 7 篇文章 订阅专栏

1、参考博客

- 1. 关于Lambda表达式里面修改外部变量问题
- 2. JDK8之前，匿名内部类访问的局部变量为什么必须要用final修饰

2、匿名内部类

在jdk7之前，匿名内部类访问外部类的局部变量时，那么这个局部变量必须用final修饰符修饰，如下图1所示。jdk8则不需要，但是我们在使用这个局部变量时，无法改变局部变量的值，否则编译会报错。

```
12 public class TestClassFinal {
13
14     public static void main(String[] args){
15         String str="testFinal";
16         Thread thread = new Thread() {
17             @Override
18             public void run() {
19                 System.out.println(str);
20             }
21         };
22     }
23 }
```

Cannot refer to a non-final variable str inside an inner class defined in a different method
1 quick fix available:
Change modifier of 'str' to final

这个特点为什么要求是final类的呢？
因为在java设计之初为了保护数据的一致性而规定的。对引用变量来说是引用地址的一致性，对基本类型来说就是值的一致性。

注意：在JDK8之后，匿名内部类引用外部变量时虽然不用显式的用final修饰，但是这个外部变量必须和final一样，不能被修改（这是一个坑）。解决方案：可以通过定义一个相同类型的变量b，然后将该外部变量赋值给b，匿名内部类引用b就行了，然后就可以继续修改外部变量。

3、lambda 表达式

这是因为：Java会将result的值作为参数传递 给Lambda表达式，为Lambda表达式建立一个副本，它的代码访问的是这个副本，而不是外部声明result变量。可能很多同学会问为什么非要建立副本呢，直接访问外部的result变量得多方便呢。答案是：这是不可能滴，因为result定义在栈中，当Lambda表达式被执行的时候，result可能已经被释放掉了。

当然啦，你要是一一定要在Lambda 表达式 里面修改外部变量的值也是可以的，可以将变量定义为实例变量或者将变量定义为数组。

① 情形一

如下图所示：在使用外部类的局部变量时，如果试图修改值，就会编译报错。

```
21      @Test
22      public void test2() {
23          String str = "adc";
24          ArrayList<String> list = new ArrayList<>();
25          list.add("1");
26          list.add("2");
27          list.add("3");
28          list.add("4");
29          list.forEach(x-> {
30              str = str+"1";
31          });
32      }
```

Variable used in lambda expression should be final or effectively final
https://blog.csdn.net/qq_39111325

② 情形二

如果局部变量是对象类型，则对象的引用地址不可改变，如下图所示。但是如果在局部变量中修改对象是没有问题的（第二篇博客有详细解释）。

```
21      @Test
22      public void test2() {
23          String str = "adc";
24          ArrayList<String> list1 = new ArrayList<>();
25          ArrayList<String> list2 = new ArrayList<>();
26          ArrayList<String> list = new ArrayList<>();
27          list.add("1");
28          list.add("2");
29          list.add("3");
30          list.add("4");
31          list.forEach(x-> {
32              list2 = list1;
33          });
34          list1.forEach(System.out::println);
35      }
```

Variable used in lambda expression should be final or effectively final
https://blog.csdn.net/qq_39111325

③ 情形三

在项目中，我遇到过这种情况：有两个集合，我先对一个集合进行遍历，在这里面又需要对另外一个集合遍历。当时出现了编译报错：variable used in lambda expression should be final or effectively final（lambda表达式中使用的变量应该是final或effective final），位置在下图的红框内。如下图所示。但是下图当我工作之余究其缘由调试时的demo，此时编译又不报错了。我能力有限，暂时还不清楚问题出现的原因。当时我的解决方案有两个：一个是在将lambda表达式换成了for下标遍历，就OK了。另一种方案是创建一个报错的同类型新对象，将list1的引用赋值给新对象，编译也不报错了（如下下图所示）。

```
20
21      @Test
22      public void test2() {
23          String str = "adc";
24          ArrayList<String> list1 = new ArrayList<>();
25          list1.add("a");
26          list1.add("b");
27          list1.add("c");
28          list1.add("d");
29          ArrayList<String> list = new ArrayList<>();
30          list.add("1");
31          list.add("2");
32          list.add("3");
33          list.add("4");
34          list.forEach(x -> {
35              list1.forEach(y->{
36
37              });
38          });
39          list1.forEach(System.out::println);
40      }
41  }
```

https://blog.csdn.net/qq_39111325

```
@Test
public void test2() {
    String str = "adc";
    ArrayList<String> list1 = new ArrayList<>();
    ArrayList<String> list2 = list1;
    list1.add("a");
    list1.add("b");
    list1.add("c");
    list1.add("d");
    ArrayList<String> list = new ArrayList<>();
    list.add("1");
    list.add("2");
    list.add("3");
    list.add("4");
    list.forEach(x -> {
        list2.forEach(y->{
            //业务代码
        });
    });
    list1.forEach(System.out::println);
}
```

https://blog.csdn.net/qq_39111325

4、总结

本文简单介绍了匿名内部类以及lambda表达式中，引入外部变量时要求为final类型，做了简单的总结。LZ借鉴了他人的博客，加上自己的理解、测试，整理了这篇文章。在开头已经贴出原文地址，若有错误，万望留言指正；如果涉及侵权，请联系本人删除。

如果没有项目经验怎么办？

合理规划自己的时间，提升自己，实习、高薪



指尖的星光xl

关注

编译器说 **Lambda 表达式中的变量**必须是 **final** 的，我偏不信12-22

偶尔，我们需要在 **Lambda 表达式中修改变量**的值，但如果直接尝试修改的话，编译器不会...

评论 4

- 请发表有价值的评论， 博客评论不欢迎灌水，良好的社区氛围需大家一起 评论
- 「已注销」2020.09.14

1

list2 = list 会报错是因为 = 相当于把list的内存地址给了 list2 也就是改变了list2的值，所以编译会报错。 如果想实现这个功能可以使用list2.addAll(list)
- nailsoul2020.05.14

1

之所以在内部类里面修改变量会报错 是因为其实修改的是内部类对象所持有的final对象 因为被private修饰所以在外部通过对象看不到 如果真想修改通过反射可以修改 至于内部类外面的局部变量必须不可变的原因我猜测应该是 内部类持有的和局部变量不是同一个 当创建完对象以后 在修改局部变量 在运行内部类中的方法 此时获取的值还是原来值 容易参数误区
- 吃琰了2020.01.06

1

我的理解是：匿名内部类接收到的局部变量是基本类型的副本或引用类型的地址值。当栈中的引用被销毁时，副本与地址值仍旧存在。但此时若对其进行修改，则会导致该方法在该方法外的范围影响了内存中的值，这边会导致意想不到的异常，因此局部变量应该允许访问但不允许修改。并且博主你第一个图例中的代码在IDEA中是没有问题的。

<12>

Java 8 **lambda表达式中为什么要求外部变量为final**?(相...2-25

==注意**lambda表达式**,在反编译的代码中,不是一个**匿名内部类**,而是一个**匿名的函数**,其原理是...

Java 8 **lambda表达式中为什么要求外部变量为final_遨游...2-18**

明白这两者的区别后,你就会明白 Java 8 **lambda表达式中为什么要求外部变量为final**。 **函数入...**

Java 8 **lambda表达式中为什么要求外部变量为final**杨逸斐的博客4万+

我的理解如下，如有错误欢迎指正。 这是由Java对**lambda表达式**的实现决定的，在Java中la...

你知道**匿名内部类**、**Lambda表达式**为嘛只能使用**外部final的变...Java方向盘3768**

1、概述 各位都知道， **匿名内部类**在使用的时候需要使用**外部**的**变量**，该**变量**必须被**final**修饰...

为什么**lambda**里面要访问**局部变量**必须是**final**_hello world2-3

这个forEach是一个**lambda表达式**,实际上是一个**匿名内部类**, 这里访问mappedFile的时候,必...

【有奖问答】 - 为什么 **Lambda 表达式**要用 **final** 关键字修饰**变量**?2-8

为什么 **Lambda 表达式**(匿名类) 不能访问非 **final** 的**局部变量**呢? 答:因为实例**变量**存在堆中,而...

Variable used in **lambda** expression should be **final**... 热门推荐 我有一个梦10万+

我们在使用 Java8 **lambda 表达式**的时候常常会遇到这样的编译报错 Variable used in **lambda**...

为什么**lambda**里面要访问**局部变量**必须是**final**hello world2564

MappedFile mappedFile = mappedFiles.get(current); msgMap.forEach((k, v) -> { map...

你知道**匿名内部类**、**Lambda表达式**为嘛只能使用**外部final的变量**吗?2-9

各位小伙伴大家好,我是A哥。各位都知道,**匿名内部类**在使用的时候需要使用**外部**的**变量**,该变...

Lambda表达式下访问**外部变量**勤快的小蚂蚁的博客5717

Lambda表达式可以访问给它传递的**变量**，访问自己内部定义的**变量**，同时也能访问它**外部**的...

Lambda表达式里面修改**外部变量**问题08-19

主要介绍了 **Lambda表达式**里面修改**外部变量**的相关知识，本文给大家介绍的非常详细，对大...

为什么**lambda表达式**使用的**局部变量**要是**final**的阳阳的博客1935

为什么 **Lambda 表达式**(匿名类) 不能访问非 **final** 的**局部变量**呢? 因为实例**变量**存在堆中，而...

为什么在**lambda**中使用的**局部变量final**必须是fi...最新发布 qfchenjunbo的博客41

直接的答案 JLS给我们了一些提示，它对有效的最终**变量**的限制对动态变化的本地**变量**的...

为什么**lambda**中用到的**局部变量**需要为**final**weixin_30883777的博客109

局部**变量** 局部**变量**是存储在栈上的，而栈上的

为什么Java **Lambda表达式**引用的局部3指尖的星光xl 关注