



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服论坛

[关于](#) [招聘](#) [广告服务](#) [网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



原 从头认识多线程

2016年06月11日 21:45:25

版权声明：本文为博主原创文章，

这一章节我们来讨论一下对

其实在前两个章节我们都

1.发布（Publish）

当一个对象能够给其他代码

```
1 package com.ray.deepintothread.ch04.topic_1;
2
3 import java.util.HashMap;
4
5 public class Publish {
6     private HashMap<String, Object> map = null;
7
8     public HashMap<String, Object> getMap() {
9         return map;
10    }
11
12    public void setMap(HashMap<String, Object> map) {
13        this.map = map;
14    }
15
16    public Publish() {
17        map = new HashMap<String, Object>();
18    }
19 }
```

）、逸出（Escape）↓

逸出的解决方案

/article/details/51638201

题，我将会放到这个章节里面去。

例如上面的map，我们可以通
如果map是public，我们可

注意：map的发布，代表n

```
1 package com.ray.de
2
3 import java.util.H
4
5 public class Publi
6     private Ha
7
8     public Has
9         return map;
10
11
12     public void setMap(HashMap<String, Object> map) {
13         this.map = map;
14     }
15
16     public Publish2() {
17         map = new HashMap<String, Object>();
18         map.put("key", "value");
19     }
20
21     public void getItemOfMap() {
22         HashMap<String, Object> map = getMap();
23
24         Object object = map.get("key");
25     }
26 }
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 网站地图
©2018 CSDN版权所有 京ICP证09002463号
🔍 百度提供搜索支持

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

tatic，直接可以通过类来引用。

我可以通过引用map来引用里面所 2 象。

我们在上面的代码里面增加一

2.逸出（escape）

就是当一个不该发布的对象

```
1 package com.ray.de
2
3 public class Escap
4
5     private in
6     private St
7
8     public Esc
9         new Thread(new MyClass()).start();
10        new Thread(new MyClass()).start();
11        name = "ray";
12    }
13
14    private class MyClass implements Runnable {
15        @Override
16        public void run() {
17
18            System.out.println(Escape.this.name);
19            System.out.println(Escape.this.id);
20
21        }
22    }
23
24    public static void main(String[] args) {
25        new Escape();
26    }
27 }
```

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗣 客服论坛

[关于](#) [招聘](#) [广告服务](#) [网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0

2

输出：

null

0

ray

0

我们从输出可以看见，其实

在引用Escape.this这个对象

一般这种现象会出现在在构

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗨 QQ客服 🗨 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

其中一个输出跟我们的想象不符，

完成

0

2

因为对象的逸出和多线程的计算造成的。

3.逸出的解决方案

```

1 package com.ray.deepintothread.ch04.topic_1;
2
3 public class SolutionOfEscape {
4
5     private int id = 0;
6     private String name = null;
7
8     Thread thread1 = null;
9     Thread thread2 = null;
10
11     public SolutionOfEscape() {
12         thread1 = new Thread(new MyClass());
13         thread2 = new Thread(new MyClass());
14         name = "ray";
15     }
16

```

17
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```
public voi
    th
}
private cl
    @O
    pu

}
}

public sta
    ne

}
}
```

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

start();

0

2

pe.this.name);

pe.this.id);

输出：

ray

0

ray

0

我们只需要避免在构造函数里面启动线程。

其实我们上面的解决方案只是就上面的一个例子来说的，更深层次的解决方案是，在构造函数执行完之前，**要避免使用Object.this这种引用。**

https://blog.csdn.net/raylee2007/article/details/51638201

5/11

总结：这一章节我们讨论了

这一章节就到这里，谢谢

我的github: <https://github.com>

目录: <http://blog.csdn.net>

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

以及逸出的解决方案。

0
2



想对作者说点什么

zfgogo: 请问这个东西有啥具体的运用么? (07-20 09:50 #2楼)

szdphsw: 不错，解决了我的问题 (01-10 05:34 #1楼)

对象的共享（一）---发布和逸出

👁 493

“发布（publish）”一个对象的意思是，是对象能够在当前作用域之外的代码中使用。例如，将一个指向该对象的引用保存到其他代码可...

多线程5: 对象的发布与逸出（线程安全性）

👁 2517

一，发布：使对象能够在当前作用域之外的代码中使用 二，逸出：当某个不该被发布的对象被发布时，这种情况称为逸出 发布内部状...

为什么80%的人选择了区块链？

区块链DApp开发学习路线图，月薪4万很轻松

【多线程与并发】：发布与逸出

目录 目录 发布与逸出 发布与逸出

对象发布与逸出

对象发布 概念 使对象能够在当前作用域之外被其他代码访问

java 发布和逸出

【转载】：http://www.2cto.com

this引用逸出

《Java 并发编程实践》3.2章里



外语好课

这个学期做个学霸吧~

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 💬 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

发布与逸出 发布（publish） 一个对象

0

的引用保存到其他代码能够访问的地址

2

下，线程安全极为重要。往往一些问题

this引用的逸出。public class ThisEsc

public This...

👤 51

使对象能...

👤 164

私有的方...

👤 584

都是由于...

👤 391

《java并发编程实战》读书笔记二 对象的发布与逸出

👤 844

概念发布（publishing）：发布一个对象的意思是使它能够被当前范围之外的代码所使用。比如将一个引用存储到其他代码可以访问的...

java并发编程：发布与逸出

👤 182

发布与逸出的定义将线程的对象放在公共域，使其他线程可见，即为发布。错误的发布，即为逸出。安全的发布方式 将对象放在静态...

Java并发编程实践笔记之一发布和逸出(Publication and Escape)

👤 890

什么是发布对象——使对象在当前作用域之外的代码中使用 什么是逸出——当某个不应该发布的对象被发布时 对象如何逸出的 最简单...

Java 线程安全性中的对象发布和逸出

👤 42

Java 线程安全性中的对象发布和逸出“发布（Publish）”一个对象的意思是指，使对象能够在当前作用域之外的代码中使用。例如，将...

相关热词

从头计算 从头算法 从头ionic vps从头 从头算

个人资料



李灵晖

关注

原创

粉丝

喜欢

评论

494

3330

19

295

等级：

博客 1

访问：107万+

积分：1万+

排名：927

勋章：





最新文章

找到对的人，是你项目开展成功的第一步

从头认识多线程-4.4 ThreadLocal的实现原理

从头认识多线程-4.3 ThreadLocal使用时需要注意的地方

联系我们



请扫描二维码联系客服



webmaster@csdn.net



400-660-0108



QQ客服



客服论坛

关于

招聘

广告服务

网站地图

©2018 CSDN版权所有 京ICP证09002463号

 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0
2

从头认识多线程-4.2 对象逸出的其他解决方案-主要是ThreadLocal的应用

从头认识多线程-3.3 synchronized某些解决不了的可视性问题，只能使用volatile来解决

博主专栏

- 轻松python
阅读量：329195 152 篇
- java基础学习
阅读量：52960 31 篇
- 数据结构与算法
阅读量：16328 9 篇
- 轻松python-文本专题
阅读量：55453 18 篇
- 轻松python之文件专题

展开

个人分类

- JAVA基础208篇
- GXT1篇
- Spring55篇
- 数据结构与算法9篇
- python154篇

展开

归档

联系我们



请扫描二维码联系客服

✉webmaster@csdn.net

☎400-660-0108

🗨QQ客服 🗨客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0
2

2017年5月	1篇
2016年6月	7篇
2016年5月	18篇
2016年4月	28篇
2016年3月	10篇
展开	

热门文章

- 零基础学python-目录
阅读量：11193
- 轻松python文本专题-去掉字符串前后空格
阅读量：10802
- 零基础学python-1.7 第二个程序 猜数字小游戏
阅读量：10772
- 从头认识java-目录
阅读量：9364
- 从头认识多线程-2.1 局部变量与实例变量的线程安全
阅读量：7674

最新评论

- 从头认识java-7.3 接口怎样...
he1044838014：看了你的发现解耦的同时，也实现了代码的复用，感觉解耦可以分为两种，一种是方法解耦，就如你这种方法...
- 从头认识Spring-1.2 什么...
liuyachao110：这里的日志记录是指的可以共用的日志记录吧？如果是每个类的每个方法里面，这些个性化的东西不是都需要手...
- 从头认识多线程-4.1 对象的发布...

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗨 QQ客服 🗨 客服论坛

[关于](#) [招聘](#) [广告服务](#) [网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0
2

zfgogo: 请问这个东西有啥具体的运用么?

从头认识java-特辑-你不知道的...

N59yL: 这个怎么看都是重写..

轻松python文本专题-字符串逐...

u013054565: 按单词翻转有问题, 考虑下多个空格情况



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 💬 客服论坛

[关于](#) [招聘](#) [广告服务](#) [网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

🔍 百度提供搜索支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0
2