

# 林加欣

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

随笔 - 120 文章 - 1 评论 - 15

## 单例模式中为什么用枚举更好

枚举单例 (Enum Singleton) 是实现单例模式的一种新方式，尽管单例模式在java中已经存在很长时间了，但是枚举单例相对来说是一种比较新的概念，枚举这个特性是在Java5才出现的，这篇文章主要讲解关于为什么我们应该使用枚举来实现单例模式，它与传统方式实现的单例模式相比较又有哪些优势？

### 1. 枚举写法简单

写法简单这是它最大的优点，如果你先前写过单例模式，你应该知道即使有DCL (double checked locking) 也可能会创建不止一个实例，尽管在Java5这个问题修复了 (jdk1.5在内存模型上做了大量的改善，提供了volatile关键字来修饰变量)，但是仍然对新手来说还是比较棘手。对比通过double checked locking 实现同步，枚举单例那实在是太简单了。如果你不相信那么对比下面代码，分别为传统的用double checked locking 实现的单例和枚举单例。

#### 枚举实现：

下面这段代码就是声明枚举实例的通常做法，它可能还包含实例变量和实例方法，但是为了简单起见，我并没有使用这些东西，仅仅需要小心的是如果你正在使用实例方法，那么你需要确保线程安全（如果它影响到其他对象的状态的话）。默认枚举实例的创建是线程安全的，但是在枚举中的其他任何方法由程序员自己负责。

```
1  /**
2   * Singleton pattern example using Java Enumj
3   */
4   public enum EasySingleton{
5       INSTANCE;
6   }
```

公告

昵称：林加欣  
园龄：2年4个月  
粉丝：32  
关注：1  
[+加关注](#)

< 2018年9月 >						
日	一	二	三	四	五	六
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	6

搜索

 找找看 谷歌搜索

你可以通过EasySingleton.INSTANCE来访问，这比调用getInstance()方法简单多了。

### double checked locking 实现法:

下面代码就是用double checked locking 方法实现的单例，这里的getInstance()方法要检查两次，确保是否实例INSTANCE是否为null或者已经实例化了，这也是为什么叫double checked locking 模式。

```
1  /**
2   * Singleton pattern example with Double checked Locking
3   */
4  public class DoubleCheckedLockingSingleton{
5      private volatile DoubleCheckedLockingSingleton INSTANCE;
6
7      private DoubleCheckedLockingSingleton(){}
8
9      public DoubleCheckedLockingSingleton getInstance(){
10         if(INSTANCE == null){
11             synchronized(DoubleCheckedLockingSingleton.class){
12                 //double checking Singleton instance
13                 if(INSTANCE == null){
14                     INSTANCE = new DoubleCheckedLockingSingleton();
15                 }
16             }
17         }
18         return INSTANCE;
19     }
20 }
```

你可以使用 DoubleCheckedLockingSingleton.getInstance()来获取实例。

从创建一个lazy loaded thread-safe单例来看，它的代码行数与枚举相比，后者可以全部在一行内完成，因为枚举创建的单例在JVM层面上也能保证实例是thread-safe的。

人们可能会争论有更好的方式去写单例用来替换double checked locking 方法，但是每种方法有他自己的优点和缺点，象我很多时候更愿初始化通过类加载静态字段，如下所示，但是记住他不是lazy loaded形式的单例。

### 静态工厂实现法:

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

docker(7)  
Dubbo(2)  
echart(1)  
Hibernate(9)  
Java(12)  
Javaweb(1)  
JPA(3)  
Jquery(12)  
Js(5)  
Linux(9)  
Maven(4)  
Mq  
Mybatis(7)  
Mysql(4)  
Nginx(2)  
Oracle(2)  
Redis(1)  
Solr  
Spring(3)  
SpringBoot(1)  
Springmvc(16)  
sql(1)  
SSH  
SSM  
vue学习(1)

这是我最喜欢的一种方式来实现单例模式，因为单例是静态的final变量，当类第一次加载到内存中的时候就初始化了，所以创建的实例固然是thread-safe。

```
1  /**
2   * Singleton pattern example with static factory method
3   */
4
5  public class Singleton{
6      //initailzed during class loading
7      private static final Singleton INSTANCE = new Singleton();
8
9      //to prevent creating another instance of Singleton
10     private Singleton(){}
11
12     public static Singleton getSingleton(){
13         return INSTANCE;
14     }
15 }
```

你可以调用Singleton.getSingleton()获取实例。

## 2. 枚举自己处理序列化

传统单例存在的另外一个问题是一旦你实现了序列化接口，那么它们不再保持单例了，因为readObject()方法一直返回一个新的对象就像java的构造方法一样，你可以通过使用readResolve()方法来避免此事发生，看下面的例子：

```
1  //readResolve to prevent another instance of Singleton
2  private Object readResolve(){
3      return INSTANCE;
4  }
```

这样甚至还可以更复杂，如果你的单例类维持了其他对象的状态的话，因此你需要使他们成为transient的对象。但是枚举单例，JVM对序列化有保证。

## 3. 枚举实例创建是thread-safe

webUI(1)  
Zookeeper(2)  
分布式架构(6)  
计算机网络(1)  
深入理解Java  
算法(4)

### 随笔档案

2018年7月 (1)  
2018年6月 (1)  
2018年5月 (1)  
2018年1月 (2)  
2017年12月 (4)  
2017年11月 (5)  
2017年10月 (3)  
2017年9月 (10)  
2017年8月 (6)  
2017年7月 (3)  
2017年6月 (3)  
2017年5月 (4)  
2017年4月 (2)  
2017年3月 (4)  
2017年2月 (5)  
2017年1月 (3)  
2016年12月 (11)  
2016年11月 (28)  
2016年10月 (13)  
2016年9月 (2)  
2016年8月 (1)  
2016年7月 (1)  
2016年6月 (6)  
2016年4月 (1)

正如在第一条中所说的，因为创建枚举默认就是线程安全的，你不需要担心double checked locking。

总结：枚举单例有序列化和线程安全的保证，而且只要几行代码就能实现是单例最好的的实现方式，不过你仍然可以使用其它的方式来实现单例，但是我仍然得不到一个更有信服力的原因不去使用枚举。如果你有的话，不妨告诉我。

原文链接：[Javarevisited](#) 翻译：[ImportNew.com](#) - 刘志军

译文链接：<http://www.importnew.com/6461.html>

[ 转载请保留原文出处、译者和译文链接。 ]

一辈子很短，努力的做好两件事就好；第一件事是热爱生活，好好的去爱身边的人；第二件事是努力学习，在工作中取得不一样的成绩，实现自己的价值，而不是仅仅为了赚钱；

分类： Java



林加欣  
关注 - 1  
粉丝 - 32  
[+加关注](#)

« 上一篇：[排序算法总结\(基于Java实现\)](#)

» 下一篇：[windows下搭建vue开发环境+IIS部署](#)

0

0

posted @ 2017-11-29 22:34 林加欣 阅读(1789) 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](#)。

## 最新评论

1. Re:Vmware虚拟机三种网络模式详解

点赞点赞!!!!!!

--虚无缥缈的云

2. Re:Vmware虚拟机三种网络模式详解

写得很好，有帮助！

--柜员\_徐无咎

3. Re:各种排序算法思想复杂度及其java程序实现

```
for(j=i-1;j>=0&&temp<num[j];j--) { num[j+1]=num[j]; //将前面较大的元素向后移动 .....
```

--白菜\_白菜

4. Re:Vmware虚拟机三种网络模式详解

非常喜欢,楼主大神

--popeye998877

5. Re:Vmware虚拟机三种网络模式详解

赞！很有帮助！

--此账号已注销

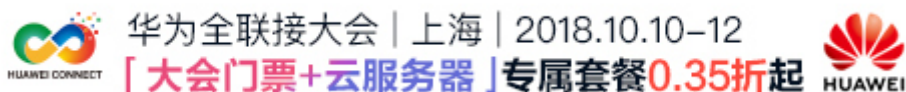
## 阅读排行榜

1. Vmware虚拟机三种网络模式详解(19522)
2. JAVA, JSP, Servlet获取当前工程路径-绝对路径(18314)
3. mybatis example使用 and和or联合查询 (转) (13291)
4. list集合为空或为null的区别(11735)



#### 最新IT新闻:

- 火星沙尘暴开始消散 NASA只剩45天来联络“机遇号”
  - 前端要凉? 微软开源Sketch2Code, 草图秒变代码
  - TIOBE 9月排行榜: C++式微 第3名被Python拿下
  - 腾讯港股盘中下跌4.43% 市值今日已蒸发1428亿港元
  - 创业者的社交梦该醒了
- » 更多新闻...



#### 最新知识库文章:

- 如何招到一个靠谱的程序员
  - 一个故事看懂“区块链”
  - 被踢出去的用户
  - 成为一个有目标的学习者
  - 历史转折中的“杭派工程师”
- » 更多知识库文章...

#### 历史上的今天:

- 2016-11-29 hibernate延迟加载(get和load的区别)
- 2016-11-29 spring持久类po或者javabean为什么常常实现序列化?
- 2016-11-29 Java过滤器与SpringMVC拦截器之间的关系与区别

5. spring容器注入一个接口的两个实现类(8120)

#### 评论排行榜

1. Vmware虚拟机三种网络模式详解(8)
2. 推荐60个jQuery插件(转)(2)
3. springmvc上传图片并显示图片--支持多图片上传(2)
4. 史上最全最强SpringMVC详细示例实战教程(1)
5. list集合为空或为null的区别(1)

#### 推荐排行榜

1. Vmware虚拟机三种网络模式详解(12)
2. 谈谈渲染, 玩玩nginx——前后端分离, 转发请求到Tomcat的尝试(3)
3. JAVA, JSP, Servlet获取当前工程路径-绝对路径(3)
4. Java中基本数据类型和包装类(2)
5. springmvc上传图片并显示图片--支持多图片上传(2)

Copyright ©2018 林加欣