

个人资料



zjufirefly

访问：6420次

积分：464

等级：BLOG > 2

排名：千里之外

原创：42篇

转载：4篇

译文：0篇

评论：1条

文章搜索

文章分类

C++ (14)

其他 (5)

链接 (13)

linux (13)

shell (7)

操作 (11)

文章存档

2014年12月 (8)

2014年11月 (7)

2014年10月 (8)

2014年09月 (6)

2014年08月 (4)

展开

阅读排行

git操作命令 (327)

blktrace (287)

mysql常用命令 (280)

python (270)

cgroups (262)

linux下如何配置openvpn (260)

2015年4月微软MVP申请

《京东技术解密》有奖试读，礼品大放送

“我的2014”年度征文活动火爆开启

CSDN 2014博客之星

为何迭代器前置式递增比后置式递增效率要高

分类：C++2014-05-04 22:14140人阅读评论(0)收藏编辑删除

后置式需要一个额外的临时对象，因为它必须存放迭代器的原本位置并将它返回。所以一般使用++itr，而不要使用itr++。

前置及后置代码实现：

```
[cpp]
01. #include <iostream>
02. using namespace std;
03.
04.
05. class UPInt {
06. public:
07.     UPInt& operator++();           // 前置式声明
08.     const UPInt operator++(int);  // 后置式声明
09.     UPInt& operator+=(int);
10.     UPInt():value(0) {}
11. public:
12.     int value;
13. };
14.
15. UPInt& UPInt::operator+=(int i)
16. {
17.     value += i;
18.     return *this;
19. }
20.
21. // 前置式实现
22. UPInt& UPInt::operator ++()
23. {
24.     *this += 1;
25.     return *this;
26. }
27.
28. // 后置式实现
29. const UPInt UPInt::operator ++(int)
30. {
31.     UPInt oldValue = *this; // 对原值进行了复制，性能较差
32.     ++(*this);
33.     return oldValue;
34. }
35.
36. int _tmain(int argc, _TCHAR* argv[])
37. {
38.     UPInt up;
39.
40.     cout << up.value << endl;
41.     cout << (up++).value << endl;
42.     cout << (++up).value << endl;
43.
44.     return 0;
45. }
```

- vim-ctags-taglist-netrw

(249)
- 性能测试命令字段解释

(213)
- 正则表达式

(211)
- SystemTap

(195)
- 评论排行
- 为什么需要auto_ptr_ref

(1)
- valgrind内存检查

(0)
- cgroups

(0)
- linux下如何配置openvpn

(0)
- git操作命令

(0)
- thrift

(0)
- Apache开源软件

(0)
- vim-ctags-taglist-netrw

(0)
- vim常用配置

(0)
- shell按行读取字符串，并

(0)

- 推荐文章
- * 挣扎与彷徨——我的2014
- * 校招回忆录——小米篇
- * Android UI-自定义日历控件
- * 30岁程序员回顾人生、展望未来
- * 2014年终总结，我决定要实现的三个目标
- * Android 启动问题——黑屏 死机解决方法

- 最新评论
- 为什么需要auto_ptr_ref

zjufirefly: explicit auto_ptr_ref(_Tp1* __p): _M_ptr(__p){}...

上一篇

如何为排序，查找，set，map提供自定义比较功能

下一篇

list中如何查找两个元素间的某个元素

顶

0

踩

0

主题推荐

迭代器

namespace

iostream

对象

性能

猜你在找

- 遍历聚合对象中的元素迭代器模式五
- 遍历聚合对象中的元素迭代器模式四
- 遍历聚合对象中的元素迭代器模式三
- 遍历聚合对象中的元素迭代器模式一
- 遍历聚合对象中的元素迭代器模式二
- java中for循环和迭代器哪个效率高
- 前置迭代器 VS 后置迭代器
- 后置递增和递减
- 后置递增和递减
- JAVA学习第三十三课常用对象API 迭代器

amaze UI

最懂中文的HTML5
开源跨屏前端框架

Amaze UI v2.0.0

正式版发布

立即使用

查看评论

暂无评论

发表评论

用户名:

zjufirefly

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

