

个人资料



touzani

访问： 227584次

积分： 2671分

排名： 第4073名

原创： 56篇

转载： 42篇

译文： 1篇

评论： 70条

文章搜索

文章分类

C/ C++ (44)

Java (2)

Windows编程技术 (12)

数学相关 (6)

数据结构与算法 (28)

杂文收藏 (12)

脚本语言 (1)

计算机图形学 (2)

文章存档

2008年02月 (1)

2007年12月 (4)

2007年11月 (4)

2007年10月 (3)

2007年09月 (5)

展开

阅读排行

c/c++ 数字转成字符串, 字 (36766)

C++ 命名空间namespace (14360)

MFC工具条与状态条设计 (8971)

平摊分析 (amortized ana (8574)

组件(component)技术介 (7392)

MFC编程之三: 绘图-1(画 (6147)

C++ BigInteger (beta ve (5322)

画出wav文件声音数据的 (4740)

有奖征资源, 博文分享有内涵

6月推荐文章汇总

微软Azure•英雄会编程大赛题

关注CSDN社区微信, 福利多多

社区问

答: 叶劲峰 游戏引擎架构

C++ 关键字 explicit, export, mutable

分类： C/ C++

2007-06-04 15:17

3670人阅读

评论(0)

收藏

举报

C++

class

iostream

编译器

include

struct

目录(?)

[+]

explicit

如果A类有某个构造函数的单个输入参数，是B类（包括基本数据类型）的对象或引用，则C++的编译器会在需要A类形参的函数调用中，自动调用该构造函数，将B类实参隐式地转换为A类实参。

这种自动类型转换的好处是，避免了定义函数的两个重载版本，代价是隐式调用构造函数会增加调用开销，并可能会带来其他问题。

在标准C++中，新增加了关键字explicit（显式/清楚/明确），（只能）用在带单个参数的构造函数前，告诉编译器，不能使用该构造函数进行隐式类型转换。如果确实想转换，则必须采用显式的类型转换方式来进行。从而避免了可能出现的问题和额外的调用开销。

例如：

```
class One {.....};

class Two {

public:

    explicit Two (const One& one) {.....}

};

void f (Two two) {.....}

int main ( ) {

    One one;

    f (one); // 错误（前面代码中无explicit时正确——调用Two(one)进行隐式类型转换）

    f (Two(one)); // 正确——采用了显式类型转换

}
```

export

为了访问其他编译单元（如另一代码文件）中的变量或对象，对普通类型（包括基本数据类、结构和类），可以利用关键字extern，来使用这些变量或对象时；但是对模板类型，则必须在定义这些模板类对象和模板函数时，使用标准C++新增加的关键字export（导出/出口/输出）。例如：

```
extern int n;

extern struct Point p;
```

使用Windows Media Pla (4328)

STL迭代器 (4255)

评论排行

C++ 命名空间namespace (13)

画出wav文件声音数据的 (9)

图形学笔记1——直线段 (6)

c/c++ 数字转成字符串, 字 (5)

MFC 鼠标拖动画圆 (4)

字符串匹配(string match (2)

给MM修电脑 (2)

组件(component)技术介 (2)

随机数 (2)

STL迭代器 (2)

推荐文章

最新评论

关于编译器与解释器的区别
liu_yujie2011com: 不错学习啦！深动形象！

画出wav文件声音数据的波形曲线
lwkobe: 你好 wave波形图画得时候需要滤波吗

组件(component)技术介绍
lidanger: 学习了。。多谢分享

C++ 命名空间namespace
jluhongfeng: 学习，领教，膜拜啦

C++ BigInteger (beta version)
crazy_martian: BigInteger x = 0;cout << -x << endl;BigInteger tt(...

STL迭代器
a529900438: 受益匪浅

c/c++ 数字转成字符串, 字符串转, yzg822: 学习了

C++ STL中用vector 改进内存的
A8572785: 。。。。。。。。

c/c++ 数字转成字符串, 字符串转, momaek2: 如果一个字符串中既有字母又有数字呢，这个又该怎么转换呢？

String matching using Rabin-Ka
lanmalong: 不错

```
extern class A a;

export template<class T> class Stack<int> s;

export template<class T> void f (T& t) {...}.
```

一般是在头文件中给出类的定义或全局函数的声明信息，而在代码文件中给出具体的（类成员函数或全局函数的）函数定义。然后在多个用户代码文件中包含该头文件后，就可以使用其中定义或声明的类和函数。头文件中一般不包含变量、结构和类对象的定义，因为这样可能会导致重复定义的编译错误。解决办法是，在某个代码文件中进行定义，在其他用户代码文件中用extern来引用它们。

但是对模板类型，则可以在头文件中，声明模板类和模板函数；在代码文件中，使用关键字export来定义具体的模板类对象和模板函数；然后在其他用户代码文件中，包含声明头文件后，就可以使用这些对象和函数了。例如：

```
// out.h: ( 声明头文件——只包含out函数的声明信息 )

template<class T> void out (const T& t);

// out.cpp: ( 定义代码文件——包含out函数的声明[通过include]和定义等全部信息 )

#include <iostream>

#include "out.h"

export template<class T> void out (const T& t) {std::cerr << t;}

//user.cpp: ( 用户代码文件——包含函数的声明头文件后就可以使用该函数 )

#include "out.h"

// 使用out()
```

说明：VC05目前还不支持export关键字（的编译）。

mutable

在类的常型（const）成员函数中，一般是不让改变类中数据成员的。如果想在常型成员函数中改变类的数据成员，在传统C++中，为达到此目，可采用一种奇怪的方式——先将this指针强制转换成一个本类的指针，然后就可以利用该指针来对类的数据成员进行任意的修改。但是，这种修改是隐藏在成员函数内部的，在类定义（头文件）中根本看不出来，而且它也破坏了设置常型成员函数的本意。

标准C++中新增加了一个关键字mutable（易变/可变/不定/无常的），用在类的数据成员前，明确表示该成员变量可以在常型成员函数中被修改。例如：

```
class A {

    int i;

    mutable int j;

public:

    void f ( ) const;

};

void A::f ( ) const {

    i++; // 错误——常型成员函数不允许改变数据成员的值

    ((A*)this)->i++; // 可以——已经过时，不被提倡

    j++; // 正确——mutable型成员变量

}
```



更多 0

上一篇 C++类型转换关键字 (*_cast)
下一篇 C++ 命名空间namespace

主题推荐 c++ 编译器 指针 标准 对象

猜你在找

- IT行业经典电子书和教程下载地址 ~
- Learn Python 3 笔记
- 第七章：无模式对话框 和 Windows通用对话框类
- MFC中实现自绘菜单
- HWND and HDC 以及 DC 释放问题
- ATOM全局原子
- VC线程学习之----CreateThread创建线程传递结构体参数
- Java读取资源文件的那些事
- MFC之取得屏幕大小，设置对话框大小
- B-Tree B+Tree mysql索引(MyISAM,InnoDB)

优优云

www.uuwise.com

验证码识别平台行业领导者

图片识别

代答题

平台

0秒

极速识别90%图片,要速度找优优！!!!单机200图片/S线程随便跑

极速、稳定、大开发支持

30台

集群服务器5千名工人7x24小时不间断工作

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 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

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2014, CSDN.NET, All Rights Reserved

