

把每天看过的，有意义的文章发表出来。总结自己！

Email:wenhao@leochin.com



秋水LeoChin 北京 海淀区

加关注

所有的东西，比如西红柿和鱼，我想一个不剩全都喜欢！

今天 18:07

转发 | 评论

无限的受教育的机会是一个社会能提供给人民最有价值的服务！

2月3日 23:13

转发 | 评论(1)

[哈哈]我发现了一篇关于【新西兰】的#蚂蜂窝游记#----《【新西兰自驾游】现在出

昵称：Leo Chin

园龄：4年11个月

粉丝：707

关注：10

+加关注

< 2011年5月 >

日	一	二	三	四	五	六
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	31				

C++ this指针的用法

this指针的含义及其用法:

1. this指针是一个隐含于每一个成员函数中的特殊指针。它指向正在被该成员函数操作的那个对象。
2. 当对一个对象调用成员函数时，编译程序先将对象的地址赋给this指针，然后调用成员函数，每次成员函数存取数据成员时，由隐含使用this指针。
3. 当一个成员函数被调用时，自动向它传递一个隐含的参数，该参数是一个指向这个成员函数所在的对象的指针。
4. 在C++中，this指针被隐含地声明为: X *const this,这意味着不能给this 指针赋值；
在X类的const成员函数中，this指针的类型为: const X* const, 这说明this指针所指向的这种对象是不可修改的（即不能对这种对象的数据成员进行赋值操作）；
5. 由于this并不是一个常规变量，所以，不能取得this的地址。
6. 在以下场景中，经常需要显式引用this指针
 - （1）为实现对象的链式引用(如例1);
 - （2）为避免对同一对象进行赋值操作(如例2);
 - （3）在实现一些数据结构时，如list.

7. 举例:

//例1:

```
/* 编辑编译环境：Dev-C++ 4.9.9.2 */
```

```
/* file: person.cpp */
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
class Person {  
    public:  
        typedef enum {  
            BOY = 0,  
            GIRL = !BOY  
        } SexType;  
    public:
```

24	25	26	27	28	29	30
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	31	1	2	3	4

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

随笔分类

[Android\(220\)](#)
[ARM9\(113\)](#)
[BootLoader\(4\)](#)
[C\(98\)](#)
[C++\(42\)](#)
[FreeBSD\(2\)](#)
[IT\(13\)](#)
[Java\(94\)](#)
[Linux\(228\)](#)
[Linux 根文件系统\(10\)](#)
[Linux 内核移植\(11\)](#)
[Linux 驱动开发\(10\)](#)
[Linux 网络编程\(140\)](#)
[MCU\(54\)](#)
[PHP\(1\)](#)
[PPT\(2\)](#)
[Python\(9\)](#)
[QT\(35\)](#)

```
Person(char *n, int a, SexType s)
{
    name = new char[strlen(n)+1]; //这里的 name 等价于this->name
    strcpy(name,n);              //这里的 name 等价于this->name
    age = a;                      //这里的 age 等价于this->age
    sex = s;                      //这里的 sex 等价于this->sex
}

int get_age(void) const
{
    //age++; //compile error, 因为 age等价于this->age,而 get_age又是一个const成员函数,
    //不能对 this指针所指向的这种对象进行修改, 这也是const的一个作用。
    //这样做的好处是, 增加了代码的健壮性。
    return age;
}

Person& add_age(int a)
{
    age +=a;
    return *this; // 返回本对象的引用
}

private:
    char *name;
    int age;
    SexType sex;
};

void TestPerson(void)
{
    Person ZhangSan("ZhangSan", 20, Person::BOY);
    printf("ZhangSan.age = %d\n", ZhangSan.get_age());
    printf("ZhangSan.add_age = %d\n", ZhangSan.add_age(1).get_age()); //增加1岁的同时, 可以对新的年
    龄直接输出;

    return;
}

int main(void)
{
```

[Ruby\(2\)](#)
[STM32\(50\)](#)
[UCOS-II\(6\)](#)
[Web\(8\)](#)
[关于职场\(4\)](#)
[路由\(38\)](#)
[数据库\(6\)](#)
[网络文摘\(254\)](#)
[心理学\(2\)](#)

随笔档案

[2015年2月 \(1\)](#)
[2015年1月 \(2\)](#)
[2014年12月 \(2\)](#)
[2014年11月 \(2\)](#)
[2014年7月 \(1\)](#)
[2014年5月 \(3\)](#)
[2014年4月 \(3\)](#)
[2014年3月 \(9\)](#)
[2014年2月 \(1\)](#)
[2014年1月 \(7\)](#)
[2013年12月 \(12\)](#)
[2013年11月 \(8\)](#)
[2013年10月 \(1\)](#)
[2013年9月 \(1\)](#)
[2013年7月 \(5\)](#)
[2013年6月 \(4\)](#)
[2013年5月 \(10\)](#)
[2013年4月 \(14\)](#)
[2013年3月 \(17\)](#)
[2013年2月 \(4\)](#)
[2013年1月 \(8\)](#)
[2012年12月 \(40\)](#)
[2012年11月 \(31\)](#)
[2012年10月 \(17\)](#)
[2012年9月 \(22\)](#)
[2012年8月 \(8\)](#)

```
TestPerson();
while(1);

}

/* result:
   ZhangSan.age = 20
   ZhangSan.add_age = 21
*/

//例2:

/* 编辑编译环境: Dev-C++ 4.9.9.2 */
/* file: location.cpp */

#include <stdio.h>

class Location {
    int X,Y;//默认为私有的
public:
    void init(int x,int y) { X =x; Y = y;};
    void assign(Location& pointer);
    int GetX(){ return X; }
    int GetY(){ return Y; }
};

void Location::assign(Location& pointer)
{
    if(&pointer!=this) //同一对象之间的赋值没有意义，所以要保证pointer不等于this
    {
        X=pointer.X;
        Y=pointer.Y;
    }
}

int main()
{
    Location x;
    x.init(5,4);
```

2012年7月 (27)
2012年6月 (19)
2012年5月 (43)
2012年4月 (22)
2012年3月 (32)
2012年2月 (21)
2012年1月 (50)
2011年12月 (39)
2011年11月 (86)
2011年10月 (118)
2011年9月 (76)
2011年8月 (69)
2011年7月 (54)
2011年6月 (141)
2011年5月 (148)
2011年4月 (63)
2011年3月 (4)
2011年2月 (12)
2011年1月 (59)
2010年12月 (66)
2010年11月 (19)
2010年2月 (1)

Blog

积分与排名

积分 - 887889

排名 - 57

最新评论

1. Re:JAVA帮助文档全系列 JDK1.5 JD
K1.6 JDK1.7 官方中英完整版下载
你又多了一位粉丝。

--透明的月儿

2. Re:Linux CGI编程

好心的博主，请问如何才能让我自己写的
web server调用php-fastCgi接口？

```
Location y;  
y.assign(x);  
  
printf("x.X = %d, x.Y = %d \n", x.GetX(), x.GetY());  
printf("y.X = %d, y.Y = %d ", y.GetX(), y.GetY());  
  
while(1);  
return 0;  
}  
/* result:  
  x.X = 5, x.Y = 4  
  y.X = 5, y.Y = 4  
*/
```

作者: [Leo Chin](#)

出处: <http://www.cnblogs.com/hnrainll/>

本博客文章,大多系网络中收集,转载请注明出处

相关标签: [嵌入式培训](#)、[嵌入式开发](#)、[嵌入式学习](#)

分类: [C++](#)

绿色通道:

好文要顶

关注我

收藏该文

与我联系



[Leo Chin](#)

[关注 - 10](#)

[粉丝 - 707](#)

[+加关注](#)

1

0

(请您对文章做出评价)

« 上一篇: [QT槽和信号机制介绍](#)

--冯煜博

3. Re:TCP三次握手及四次挥手详细图解
wireshark抓包分析TCP三次握手

--seanyxie

4. Re:c语言中逗号运算符和逗号表达式
多谢，原来如此。

--KW洗革拉

5. Re:JAVA帮助文档全系列 JDK1.5 JD
K1.6 JDK1.7 官方中英完整版下载
好像下载不了呀

--小峰Coconut

阅读排行榜

1. JAVA帮助文档全系列 JDK1.5 JDK1.6
JDK1.7 官方中英完整版下载(226517)
2. linux FTP配置详解(83210)
3. TCP三次握手及四次挥手详细图解(676
17)
4. Linux常用的网络命令(34155)
5. java常用设计模式(31645)
6. linux stat函数讲解(29210)
7. Linux man命令的使用方法(26885)
8. make clean与make distclean的区别(2
6853)
9. KEILC51编译问题ERROR L104: MUL
TIPLE PUBLIC DEFINITIONS重复定义(
24970)
10. C标准库的setlocale()用法笔记(23244
)

» 下一篇: [QApplication类详细描述](#)

posted @ 2011-05-20 14:03 Leo Chin 阅读(5705) 评论(1) 编辑 收藏

评论列表

#1楼 2014-05-07 13:58 TFS

既然编译器把对象的地址赋给this指针，为什么不能取得this的地址？
我个人认为是可以取到this指向的地址，因为我在程序中可以这样使用this指针。如下：
if(this==&objModel)
return *this;

这样对吗？

支持(0) 反对(0)

[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。

【免费课程】系列: [JavaScript入门篇](#)

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
融云，免费为你的App加入IM功能——让你的App“聊”起来！！

【活动】百度开放云限量500台，抓紧时间申请啦！



最新IT新闻:

- [报告称2019年中国电商规模将突破1万亿美元](#)
 - [新浪百度腾讯负责人谈如何治理账号乱象](#)
 - [甲骨文收购营销公司Datalogix 价格逾12亿美元](#)
 - [SEC与四大中国合作所接近达成和解](#)
 - [中国电信去年终端销量8500万部 今年宽带提速至50M](#)
- » [更多新闻...](#)



最新知识库文章:

- [通俗解释「为什么数据库难以拓展」](#)
 - [手机淘宝高质量持续交付探索之路](#)
 - [高效运维最佳实践（01）：七字诀，不再憋屈的运维](#)
 - [什么是工程师文化？](#)
 - [大数据架构和模式（五）对大数据问题应用解决方案模式并选择实现它的产品](#)
- » [更多知识库文章...](#)

