

Coderec的专栏

个人资料



coderec

恒

访问：747次
积分：74分
排名：千里之外

原创：7篇 转载：0篇
译文：0篇 评论：0条

文章搜索

文章分类

linux (3)

java (3)

c/c++ (4)

文章存档

2013年12月 (1)

2013年08月 (1)

2013年06月 (1)

2013年05月 (4)

阅读排行

简记c、c++、java异常处 (231)

c++与java在嵌套类方面的 (123)

java定义在局部的内部类 (123)

Linux学习笔记之开机过程 (89)

简单配置自己的Bash Sh (86)

sed命令学习笔记 (66)

QuickPanel开发总结（附 (26)

评论排行

Linux学习笔记之开机过程 (0)

简记c、c++、java异常处 (0)

投票赢好礼，周周有惊喜！
2014年4月微软MVP申请开始了！
消灭0回答，赢下载分
“我的2013”年度征文活动火爆进行中！
专访Kinect手语翻译系统团队

c++与java在嵌套类方面的比较

分类：java c/c++

C++Java嵌套类

2013-05-30 12:29124人阅读评论(0)收藏举报

c++与java都允许类的嵌套，不过在机制和使用方法上还是有较大不同的

一、C++的嵌套类

C++的嵌套类可以在封装类的私有、公有、保护部分定义，当然这些访问控制也做用于嵌套类上。

虽然嵌套类在封装类内部定义，但它是一个独立的类，基本上与封装类不相关。它的成员不属于封装类，同样，封装类的成员也不属于该嵌套类。嵌套类的出现只是告诉封装类有一个这样的类型成员供其使用。并且，封装类对嵌套类成员的访问没有任何特权，嵌套类对封装类成员的访问也同样如此，它们都遵循普通类所具有的标号访问控制。

不过也有特例： 嵌套类可以直接访问封装类的静态成员、类型名（ typedef ）和枚举值（ enum ）。

[cpp]

01. #include <iostream>
02. using namespace std;
03.
04. class X{
05. class Y{
06. int m;
07. public:
08. Y(int t){ m=t; }
09. }
10.
11. public:
12. X(){}
13. }
14.
15. int main () {
16. X::Y y(5);
17. return 0;
18. }
19.

//使用“::”调用

如上代码所示是将成员函数都定义为内联函数了，若不想定义为内联，在外部定义函数是应该使用作用域操作符 ::

例如：

[cpp]

01. X::Y::Y() {
02.

blog.csdn.net/a450828540/article/details/89931601/5

简单配置自己的Bash Sh	(0)
c++与java在嵌套类方面	(0)
java定义在局部的内部类	(0)
sed命令学习笔记	(0)
QuickPanel开发总结（附	(0)

推荐文章

- * Android源码分析—带你认识不一样的AsyncTask
- * 互联网开发之路（一）：痛苦但去，快乐自来
- * android网络开源框架volley之二
- * 小强的HTML5移动开发之路（2）——HTML5的新特性
- * Real-Rime Rendering (10) - 图形硬件及学习小结（Graphics Hardware）
- * Android动画进阶—使用开源动画库nineoldandroids

```
03.     your code :
04.
05. }
```

嵌套类的定义也可位于封装类之外，定义时也要如上使用作用域操作符

注意事项：

- 1. 封装类无法直接访问嵌套类私有成员，若想直接访问，必须在嵌套类中使封装类成为其友元。
- 2. 嵌套类无法直接访问封装类私有成员，若想直接访问，必须在封装类中使嵌套类成为其友元。
- 3. 嵌套类可以直接访问封装类的静态成员、类型名（typedef）和枚举值（enum）。
- 4. 若嵌套类想访问封装类的非静态成员，必须通过封装类的一个对象作为载体才行。

二、java的嵌套类

1、静态嵌套类：

C++的嵌套类与java中的静态嵌套类“类似”。

```
[cpp]
01. //java
02.
03. class X{
04.     static class Y{                                //非
05.     public
06.         private int a;
07.         public Y(int m){ a=m; }
08.         public void printY(){
09.             system.out.println( "xxxxxxx");
10.         }
11.     }
12.     private Y yref;
13.     public X(){ }
14.     Y get_yref(){ return yref; }
15. }
16.
17. class test{
18.     public static void main( String[] args) {
19.         X x=new X();
20.         x.get_yref().printY();                        //因为非
public所以要先获取对象
```

```
[cpp]
01. //java
02.
03. class X{
04.     public static class Y{                            //1
05.     private int a;
06.     public Y(int m){ a=m; }
07.     public void printY(){
08.         system.out.println( "xxxxxxx");
09.     }
10. }
11.
12. private Y yref;
13. public X(){ }
14. Y get_yref(){ return yref; }
15. }
16.
17. class test{
18.     public static void main( String[] args) {
19.         X.Y x=new X.Y();                               /pi
直接使用成员操作符“.”来调用
20.         x.printY();
```

静态嵌套类与C++嵌套类类似，可以直接访问封装类的**static**成员，但是，如要访问非静态成员就要使用一个封装类的对象作为载体。此外**java**中嵌套类可以直接访问封装类中的私有成员。

2.非静态嵌套类：内部类

内部类允许我们直接访问封装类的所有成员，不论静态成员还是非静态，不论**private**还是**protected**更别说**public**。

请看如下代码：

```
[cpp]
01. //java
02.
03. class X{
04.     private int a;
05.     private static int b=30;
06.
07.     public class Y{                                //内部类，非静态
08.         private int m;
09.         private int n;
10.         public Y(){
11.             m = a;                                //访问private成员
12.             n = b;                                //访问静态私有成员
13.         }
14.     }
15.
16.     public X(int n) { a = n; }
17. }
18.
19. class test {
20.     public static void main ( String[] args) {
21.         X x=new X.Y();
22.         // X.Y y = new X.Y()                      //错的!!!
23.         X.Y y= x.new Y();                          //这样才对
24.     }
25. }
```

接下来说一下名字冲突的情况下以前缀避免冲突：

```
[cpp]
01. //java
02.
03. class X{
04.     private int m;
05.     private static int n=300;
06.
07.     public class Y{
08.         private int m;
09.         private int n;
10.         public Y(){
11.             this.m = X.this.m;                    //此处留意
12.             this.n = X.this.n;
13.         }
14.     }
15.
16.     public X(){}
17. }
```

java的接口也可以包含嵌套的接口或者类，当接口包含了一个类时，这个类会被隐式的声明为**public**和**static**

最后，如果有多重嵌套例如：**x (y (z()))**可以在**z**中使用**x**.前缀访问**x**成员，还有，内部类不允许有静态成员。

java还可以有匿名内部类，另开一篇再写把。

更多 0

上一篇：简单配置自己的Bash Shell
下一篇：java定义在局部的内部类、匿名内部类（接《c++与java嵌套类比较》）

管理软件开发平台 第1名
工作流+报表+权限+表单+移动应用

免费下载 
www.justep.com

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

专区推荐内容

超极本触摸屏一种全新的动手方式
[Intel CPU中的安全指令...
是时候结束数据中心冲突了？
什么是微服务器
优化、并行化、及向量化
如何从数据库调出数据显示到页面

<< >>

更多招聘职位

我公司职位也要出现在这里

【碧桂园控股集团】asp.net开发工程师
【风奇咨询】JAVA 高级工程师/架构师
【上海帖易文化传播有限公司】维护工程师（C#）
【口袋巴士】Android开发工程师
【酷趣】前端开发工程师
【南京海震智能科技有限公司】ios工程师

核心技术类目

全部主题 Java VPN Android iOS ERP IE10 Eclipse CRM JavaScript Ubuntu NFC
WAP jQuery 数据库 BI HTML5 Spring Apache Hadoop .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	Spark	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](#)
[QQ客服](#) [微博客服](#) [论坛反馈](#) 联系邮箱：webmaster@csdn.net 服务热线：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
世纪乐知(北京)网络技术有限公司 提供技术支持
江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2012, CSDN.NET, All Rights Reserved 