

鸡啄米

聚焦互联网、数码、软件开发和编程入门的IT休闲吧

[首页](#) [IT互联网](#) [数码生活](#) [软件开发](#) [职场人生](#) [娱乐休闲](#) [编程课堂](#) [安卓开发](#) [留言簿](#)

[首页](#) » [软件开发](#) » MFC六大核心机制之二：运行时类型识别（RTTI）

MFC六大核心机制之二：运行时类型识别（RTTI）

分类标签: [MFC](#) [VC++](#)

上一节讲的是[MFC六大核心机制之一：MFC程序的初始化](#)，本节继续讲解[MFC六大核心机制之二：运行时类型识别（RTTI）](#)。

typeid运算符

运行时类型识别（RTTI）即是程序执行过程中知道某个对象属于某个类，我们平时用C++编程接触的RTTI一般是编译器的RTTI，即是在新版本的VC++编译器里面选用“使能RTTI”，然后载入typeid.o.h文件，就可以使用一个叫typeid()的运算符，它的地位与在C++编程中的sizeof()运算符类似的地方（包含一个头文件，然后就有一个熟悉好用的函数）。typeid()关键的地方是可以接受两个类型的参数：一个是类名称，一个是对象指针。所以我们判别一个对象是否属于某个类就可以象下面那样：

C++代码

```
1.  if (typeid (ClassName) == typeid (*ObjectName) ) {
2.      ((ClassName*)ObjectName)->Fun ();
3.  }
```

像上面所说的那样，一个typeid()运算符就可以轻松地识别一个对象是否属于某一个类，但MFC并不是用typeid()的运算符来进行动态类型识别，而是用一大堆令人费解的宏。很多学员在这里很疑惑，好象MFC在大部分地方都是故作神秘。使们大家编程时很迷惘，只知道在这里加入一组宏，又在那儿加入一个映射，而不知道我们为什么要加入这些东东。

其实，早期的MFC并没有typeid()运算符，所以只能沿用一個老办法。我们甚至可以想象一下，如果MFC早期就有[template（模板）](#)的概念，可能更容易解决RTTI问题。

所以，我们要回到“古老”的年代，想象一下，要完成RTTI要做些什么事情。就好像我们在新一个（新型到我们还不认识）电器公司里面，我们要识别哪个是电饭锅，哪个是电磁炉等等，我们要查看登记的各电器一系列的信息，我们才可以比较、鉴别，那个东西是什么！

CRuntimeClass链表的设计

要登记一系列的消息并不是一件简单的事情，大家可能首先想到用数组登记对象。但如果用数组，我们要定义多大的数组才好呢，大了浪费空间，小了更加不行。所以我们要用另一种数据结构——链表。因为链表理论上可大可小，可以无限扩展。

链表是一种常用的数据结构，简单地說，它是在一个对象里面保存了指向下一个同类型对象的指针。我们大体可以这样设计我们的类：

C++代码

```
1.  struct CRuntimeClass
2.  {
3.      .....类的名称等一切信息.....
4.      CRuntimeClass * m_pNextClass; //指向链表中下一CRuntimeClass对象的指针
5.  };
```

链表还应该有一个表头和一个表尾，这样我们在查链表中各对象元素的信息的时候才知道从哪里查起，到哪儿结束。我们还要注明本身是由哪能个类派生。所以我们的链表类要这样设计：

订阅鸡啄米

[RSS](#) [+ 订阅到QQ邮箱](#)

分享

站内搜索

请输入搜索内容..



分类标签

[编程入门 \(135\)](#)
[C++ \(96\)](#)
[VC++ \(77\)](#)
[MFC \(67\)](#)
[VS2010 \(63\)](#)
[程序员 \(55\)](#)
[Android \(51\)](#)
[Java \(51\)](#)
[苹果 \(49\)](#)
[智能手机 \(47\)](#)
[腾讯 \(41\)](#)
[百度 \(37\)](#)
[阿里巴巴 \(33\)](#)
[谷歌 \(32\)](#)
[平板电脑 \(31\)](#)
[TCP/IP \(29\)](#)
[iPhone \(26\)](#)
[PHP \(26\)](#)
[Javascript \(25\)](#)
[奇虎360 \(24\)](#)
[Mysql \(24\)](#)
[Windows \(22\)](#)
[软件架构 \(20\)](#)
[小米 \(20\)](#)
[设计模式 \(19\)](#)
[iPad \(18\)](#)
[Web \(18\)](#)
[职场攻略 \(18\)](#)
[三星 \(16\)](#)
[创业 \(16\)](#)
[微软 \(13\)](#)
[iOS \(13\)](#)
[微信 \(13\)](#)
[HTML \(13\)](#)
[应用程序 \(12\)](#)

c++代码

```
1.  struct CRuntimeClass
2.  {
3.      .....类的名称等一切信息.....
4.      CRuntimeClass * m_pBaseClass; //指向所属的基类。
5.      CRuntimeClass * m_pNextClass; //定义表尾的时候只要定义此指针为空就可以了。
6.
7.      static CRuntimeClass* pFirstClass; //这里表头指针属于静态变量，因为我们知道
      static变量在内存中只初始化一次，就可以为各对象所用！保证了各对象只有一个表头。
8.  };
```

有了CRuntimeClass结构后，我们就可以定义链表了：

c++代码

```
1.  static CRuntimeClass classCObject={NULL,NULL}; //这里定义了一个CRuntime
      Class对象，因为classCObject无基类，所以m_pBaseClass为NULL。因为目前只有一个元
      素（即目前没有下一元素），所以m_pNextClass为NULL（表尾）。
```

至于pFirstClass（表头），大家可能有点想不通，它到什么地方去了。因为我们这里并不想把classCObject作为链表表头，我们还要在前面插入很多的CRuntimeClass对象，并且因为pFirstClass为static指针，即是说它不是属于某个对象，所以我们在用它之前要先初始化：CRuntimeClass* CRuntimeClass::pFirstClass=NULL;。



现在我们可以前面插入一个CRuntimeClass对象，插入之前我得重要申明一下：如果单纯为了运行时类型识别，我们未必用到m_pNextClass指针（更多是在运行时创建时用），我们关心的是类本身和它的基类。这样，查找一个对象是否属于一个类时，主要关心的是类本身及它的基类。

c++代码

```
1.  CRuntimeClass classCCmdTarget={ &classCObject, NULL};
2.  CRuntimeClass classCWnd={ &classCCmdTarget ,NULL };
3.  CRuntimeClass classCView={ &classCWnd , NULL };
```

好了，上面只是仅仅为一个指针m_pBaseClass赋值（MFC中真正CRuntimeClass有多个成员变量和方法），就连接成了链表。假设我们现在已全部构造完成自己需要的CRuntimeClass对象，那么，这时候应该定义表头。即要用pFirstClass指针指向我们最后构造的CRuntimeClass对象——classCView。

c++代码

```
1.  CRuntimeClass::pFirstClass=&classCView;
```

现在链表有了，表头表尾都完善了，问题又出现了，我们应该怎样访问每一个CRuntimeClass对象？要判断一个对象属于某类，我们要从表头开始，一直向表尾查找到表尾，然后才能比较得出结果

[新浪 \(12\)](#)[微博 \(11\)](#)[软件工程师 \(10\)](#)[诺基亚 \(10\)](#)[京东商城 \(10\)](#)[比特币 \(10\)](#)[Facebook \(9\)](#)[周鸿祎 \(9\)](#)[操作系统 \(8\)](#)[Galaxy \(8\)](#)[社交网络 \(8\)](#)[搜索引擎 \(8\)](#)[移动互联网 \(8\)](#)[C \(8\)](#)[亚马逊 \(7\)](#)[更多标签](#)

分享

完全随机文章

VS2010/MFC编程入门之五（MFC消息映...
从Facebook和Twitter获取新闻的用户...
VS2010/MFC编程入门之二十五（常用控...
VS2010/MFC编程入门之三十五（菜单：...
VS2010/MFC编程入门之四十一（文档、...
VS2010/MFC编程入门教程之目录和总...
程序员的选择：技术vs管理
2015产品校招——阿里腾讯百度360小米...
C、C++、python、Java、php、C#六种流行...
App推广秘籍最全篇
程序员修炼指南——引导你成为真正的...
85后工作5年工资竟然涨了25倍——月薪...
TCP/UDP网络编程入门教程之十五：TC...
memcached使用场景和方法总结
从《奋斗》到《欢乐颂》看青年的价值观...
StackOverflow 创始人推荐程序员看...
魅族的掉队已成事实，生态链不是那么...
详解HTML5 LocalStorage本地存储
如何避免成为下一个雅虎
说说Javascript闭包这点事

最新评论及回复

CAddSheet(LPCTSTR ...
有没有XTP的教学？求教
就画个界面，搞这么负责，难怪MFC要被淘汰
讲的太好了，很全，很清楚！楼主你的Q...
楼主，请问如何动态给重写的CList...
普通人只有被剥削的份
已点广，，，，，告支持楼主
#include <afxco...
关掉王者荣耀。它就像鸦片，勾引小孩子...
关掉王者荣耀。它就像鸦片，勾引小孩子
楼主，请问CTabCtrl和CList...
一、初始化函数中在设置好子对话框位置...
创建两组Radio可以在Radio的...
一定要通过【类向导】添加类，【类向导...
写的太棒了。
谢谢楼主
加油
弱的问一声？符号常量的用法是否跟C语...
蛮实用的可是在8年后才看到[REV...
好的程序员一定是挣钱的

最近发表

鸡啄米开始承接项目啦

吗。肯定不是这样！

类中构造CRuntimeClass对象

大家可以这样想一下，我们构造这个链表的目的，就是构造完之后，能够按主观地拿一个CRuntimeClass对象和链表中的元素作比较，看看其中一个对象是否属于你指定的类。这样，我们需要有一个函数，一个能返回自身类型名的函数GetRuntimeClass()。

上面简单地说了链表的过程，但单纯有这个链表是没有任何意义。回到MFC中来，我们要实现的是在每个需要有RTTI能力的类中构造一个CRuntimeClass对象，比较一个类是否属于某个CRuntimeClass对象的时候，实际上只是比较CRuntimeClass对象。

如何在各个类之中插入CRuntimeClass对象，并且指定CRuntimeClass对象的内容及CRuntimeClass对象的链接，这里起码有十行的代码才能完成。在每个需要有RTTI能力的类设计中都要重复那十多行代码是一件乏味的事情，也容易出错，所以MFC用了两个宏代替这些工作，即DECLARE_DYNAMIC(类名)和IMPLEMENT_DYNAMIC(类名，基类名)。从这两个宏我们可以看出在MFC名类中的CRuntimeClass对象构造连接只有类名及基类名的不同！

到此，可能会有朋友问：为什么要用两个宏，用一个宏不可以代换CRuntimeClass对象构造连接吗？个人认为肯定可以，因为宏只是文字代换的游戏而已。但我们在编程之中，头文件与源文件是分开的，我们要在头文件头声明变量及方法，在源文件里实现具体实现。即是说我们要在头文件中声明：

c++代码

```
1. public:
2. static CRuntimeClass classXXX //XXX为类名
3. virtual CRuntime* GetRuntimeClass() const;
```

然后在源文件里实现：

c++代码

```
1. CRuntimeClass* XXX: : classXXX={.....};
2. CRuntime* GetRuntimeClass() const;
3. { return &XXX:: classXXX; } //这里不能直接返回&classXXX，因为static变量是类拥有而不是对象拥有。
```

我们一眼可以看出MFC中的DECLARE_DYNAMIC(类名)宏应该这样定义：

c++代码

```
1. #define DECLARE_DYNAMIC(class_name) public: static CRuntimeClass class_##class_name; virtual CRuntimeClass* GetRuntimeClass() const;
```

其中##为连接符，可以让我们传入的类名前面加上class，否则跟原类同名，大家会知道产生什么后果。

有了上面的DECLARE_DYNAMIC(类名)宏之后，我们在头文件里写上一句

DECLARE_DYNAMIC(XXX)

宏展开后就有了我们想要的：

```
public:
static CRuntimeClass classXXX //XXX为类名
virtual CRuntime* GetRuntimeClass() const;
```

对于IMPLEMENT_DYNAMIC(类名，基类名)，看来也不值得在这里代换文字了，大家知道它是知道回事，宏展开后为我们做了什么，再深究真是一点意义都没有！有了此链表之后，就像有了一张存放各类型的网，我们可以轻而易举地RTTI。

IsKindOf函数

CObject有一个函数BOOL IsKindOf(const CRuntimeClass* pClass) const;，被它以下所有派生类继承。

小白照样读懂的VLAN原理讲解

SSH电商项目实战之十：商品类基本模块的搭建

SSH电商项目实战之九：添加和更新商品类别功能的实现

SSH电商项目实战之八：查询和删除商品类别功能的实现

SSH电商项目实战之七：Struts2和Json的整合

长文：内容产业的赢家与输家

SSH电商项目实战之六：基于DataGrid的数据显示

SSH电商项目实战之五：完成数据库的级联查询和分页

SSH电商项目实战之四：EasyUI菜单的实现

SSH电商项目实战之三：使用EasyUI搭建后台页面框架

SSH电商项目实战之二：基本增删查改、Service和Action的抽取以及使用注解替换xml

大妈：我们不懂ICO和X币，但知道比炒房厉害

SSH电商项目实战之一：整合Struts2、Hibernate和Spring

面临连续亏损，HTC出售手机还是VR业务？

分享

此函数实现如下：

C++代码

```
1.  BOOL CObject::IsKindOf(const CRuntimeClass* pClass) const
2.  {
3.      CRuntimeClass* pClassThis=GetRuntimeClass();//获得自己的CRuntimeClass
      对象指针。
4.      while(pClassThis!=NULL)
5.      {
6.          if(pClassThis==pClass) return TRUE;
7.          pClassThis=pClassThis->m_pBaseClass;//这句最关键，指向自己基类，再回头比
      较，一直到尽头m_pBaseClass为NULL结束。
8.      }
9.      return FALSE;
10. }
```

说到这里，运行时类型识别（RTTI）算是完成了。

来自：<http://blog.csdn.net/liyi268/article/details/304175>

除非特别说明，鸡啄米文章均为原创

转载请标明本文地址：<http://www.jizhuomi.com/software/269.html>

2012年11月26日

作者:鸡啄米 分类:软件开发 浏览:238515 评论:6

相关文章:

[MFC六大核心机制之一：MFC程序的初始化](#) (2012-11-22 22:20:43)

[VS2010中如何实现自定义MFC控件](#) (2012-11-18 22:56:16)

[VS2010/MFC编程入门教程之目录和总结](#) (2012-10-31 22:4:12)

[VS2010/MFC编程入门之五十四（Ribbon界面开发：使用更多控件并为控件添加消息处理函数）](#) (2012-10-27 21:56:47)

[VS2010/MFC编程入门之五十三（Ribbon界面开发：为Ribbon Bar添加控件）](#) (2012-10-21 21:53:16)

[VS2010/MFC编程入门之五十二（Ribbon界面开发：创建Ribbon样式的应用程序框架）](#) (2012-10-17 22:0:34)

[VS2010/MFC编程入门之五十一（图形图像：GDI对象之画刷CBrush）](#) (2012-10-12 22:19:2)

[VS2010/MFC编程入门之五十（图形图像：GDI对象之画笔CPen）](#) (2012-10-8 22:53:50)

[VS2010/MFC编程入门之四十九（图形图像：CDC类及其屏幕绘图函数）](#) (2012-9-26 21:22:28)

[VS2010/MFC编程入门之四十八（字体和文本输出：文本输出）](#) (2012-9-22 22:41:48)

1楼: 润初颜

代码看不懂，但是那个图片不错呀！嘻嘻！

2012/11/27 10:39:57 [回复该留言](#)

2楼: 李丹旋

RTTI 正在学习中，了解了不少知识！多谢博主

鸡啄米 于 2012-11-27 22:57:18 回复
不用谢，多多交流

2012/11/27 11:04:46 [回复该留言](#)

3楼: 篮球比分

依然在学习中

2012/11/27 13:23:17 [回复该留言](#)

4楼: 吴克难的博客

分享

很深奥，来学习下！

鸡啄米 于 2012-11-28 23:00:24 回复
比较底层

2012/11/28 13:41:00 [回复该留言](#)

5楼. [vgamed](#)

麻烦答疑两个问题：

1. m_pNextClass是做什么用的？
2. 什么时候赋值pFirstClass，MFC里对应的宏是？

2014/11/3 15:04:03 [回复该留言](#)

6楼. [wanghui](#)


大神能不能举例子说明一下呀？代码完全搞不懂呀？

2015/5/12 10:22:11 [回复该留言](#)

上一篇：[解密为何iOS设备较之Android设备更加流畅](#)

下一篇：[魅族叫阵小米 谷歌是胜者](#)

[发表评论](#):

<input type="text"/>	名称(*)
<input type="text"/>	邮箱(选填)
<input type="text"/>	网站链接(选填)
<input type="text"/>	验证(*) 

正文*)(留言最长字数:1000)

☐ 记住我,下次回复时不用重新输入个人信息

[\[URL\]](#) [\[URL2\]](#) [\[EMAIL\]](#) [\[EMAIL2\]](#) [\[B\]](#) [\[I\]](#) [\[U\]](#) [\[S\]](#) [\[QUOTE\]](#) [显示UBB表情>>](#)

◎欢迎参与讨论，请在这里发表您的看法、交流您的观点。

Copyright © 2011-2020 鸡啄米. 版权所有.

联系邮箱:jizhuomi@126.com Powered By Z-Blog 

[无觅相关文章插件](#)

分享