

2021/04/14_MFC原理_第3课_MFC框架的模拟实现、RTTI

笔记本: MFC原理
创建时间: 2021/4/14 星期三 9:59
作者: ileemi
标签: RTTI

- [CRuntimeClass 结构体](#)
- [类型识别相关宏](#)

CRuntimeClass 结构体

```
class CMyObject;  
  
struct CRuntimeClass {  
    const char m_szClassName[MAXBYTE]; // 保存当前类的名称  
    const CRuntimeClass* m_pBase;      // 指向基类  
    CMyObject* (*m_pfnCreateObject)(); // 保存派生类的函数指针  
  
    // 判断一个类是否是一个类的派生类  
    bool IsDerivedFrom(const CRuntimeClass* pRuntime) const {  
        const CRuntimeClass* pThis = this;  
        while (pThis != NULL) {  
            if (pThis == pRuntime) {  
                return TRUE;  
            }  
            pThis = pThis->m_pBase;  
        }  
        return FALSE;  
    }  
};
```

类型识别相关宏

在基类 CMyObject 类中进行定义:

```
// 用于类型识别  
/*  
有动态需求 — 声明
```

```

    * 为当前类添加静态CRuntimeClass类对象成员
    * 为当前类添加GetRuntimeClass虚函数
*/
#define DECLARE_DYNAMIC(class_name) \
public:\
virtual const CRuntimeClass* GetRuntimeClass() const;\
static const CRuntimeClass m_class##class_name;

/*
有动态需求 — 实现
    * 初始化静态CRuntimeClass类对象
    * 实现GetRuntimeClass虚函数
*/
// #class_name -- "class_name", base_class -- 指向基类
&base_class::m_class##base_class
#define _IMPLEMENT_RUNTIMECLASS(class_name, base_class, pfnNew) \
const CRuntimeClass class_name::m_class##class_name = { #class_name, \
RUNTIME_CLASS(base_class), pfnNew};\
const CRuntimeClass* class_name::GetRuntimeClass() const {\
    return RUNTIME_CLASS(class_name);\
}

#define IMPLEMENT_RUNTIMECLASS(class_name, base_class) \
_IMPLEMENT_RUNTIMECLASS(class_name, base_class, NULL)

// 产生对象
//static CMyObject* CreateObject();
// 基类不需要new对象
//CMyObject* CMyObject::CreateObject()
//{
//    return new CMyObject();
//}
//&CMyObject::CreateObject

/*
有动态需求 — 声明
    * 为当前类添加静态CRuntimeClass类对象成员
    * 为当前类添加GetRuntimeClass虚函数
    * 为当前类添加一个创建对象的静态函数
*/
#define DECLARE_DYNCREATE(class_name) \
DECLARE_DYNAMIC(class_name)\
static CMyObject* CreateObject();

/*
有动态需求 — 实现

```

```
    * 初始化静态CRuntimeClass类对象
    * 实现GetRuntimeClass虚函数
    * 实现创建对象的静态函数
*/
#define IMPLEMENT_DYNCREATE(class_name, base_class) \
    CMyObject* class_name::CreateObject() {\
        return new class_name;\
    }\
    _IMPLEMENT_RUNTIMECLASS(class_name, base_class,\
class_name::CreateObject)
```