2020/12/17_COM_第3课_插件的设计以及跨语言注意事项

笔记本： COM
创建时间： 2020/12/17 星期四 14:59
作者： ileemi

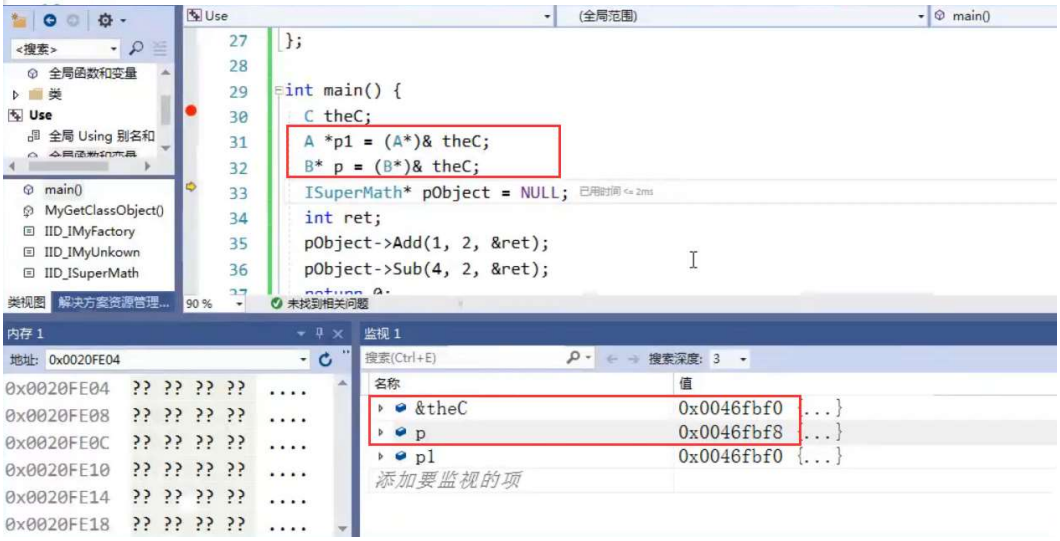
超级算法库

> C++官网：https://isocpp.org/

每一个接口（每个抽象类）都应该定义一个GUID

- 定义一个对应的接口并继承 IMyUnkown，同时创建GUID
- 接口在C++中使用抽象类进行实现
- 公开基类，不公开派生类（派生类内部更改后，基类不受影响）
- 公开的基类，内部添加新功能后，新功能不使用不能将其删除。
- 注意引用计数问题

```
4
5   HRESULT CSuperMath::QueryInterface(const GUID& iid, void** ppObject)
6   {
7     if (memcmp(&iid, &IID_IMyUnkown, sizeof(GUID)) == 0) {
8       *ppObject = (IMyUnkown*)this;
9     }
10    else if (memcmp(&iid, &IID_ISuperMath, sizeof(GUID)) == 0) {
11      *ppObject = (ISuperMath*)this;
12    }
13
14    return E_NOINTERFACE;
15  }
```



基类的虚表不代表和派生类一样，通过成员函数（QueryInterface）转换虚表

软件升级，插件查询是否有新功能可以使用：

```
36
37    INewSuperMath* pNewMath = NULL;
38    pSuperMath->QueryInterface(IID_INewSuperMath, (void**)&pNewMath);
39    if (FAILED(hr)) {
40      printf("请升级插件版本");
41      return 0;
42    }
43    pNewMath->Mul(4, 2, &ret);
44    printf("4*2=%d\n", ret);
45    return 0;
46  }
```

定义同一个对象：

CSuperMath.cpp | CSuperMathFactory.cpp | CSuperMathFactory.h | CSuperMath.h | dllmain.cpp | Use.cpp

SuperMath → CSuperMathFactory → CreateObj(void ** ppObj)

```
34    return m_nRefCount;
35  }
36
37  HRESULT CSuperMathFactory::CreateObj(void** ppObj)
38  {
39    static CSuperMath *pObject =  new CSuperMath();
40    *ppObj = pObject;
41    return S_OK;
42  }
43
```

插件跨语言

注意事项：

- 不同语言的调用约定不一样
- 强制接口中的方法使用统一的调用约定（该约定所有语言都支持），使用栈来传递参数，推荐使用 `__stdcall`
- 更改框架的调用约定为标准调用约定
- 接口头文件要做到C语言C++兼容

由于C++编译器较多，编译器生成的虚表结构有各不相同。C++没有二进制标准（编译后的二进制代码没有标准）。
后来发明了C++二进制标准（application binary interface）

接口的设计需要满足下面的五个条件（下面的几点可以会导致虚表结构不同）：

- 位置：对象首地址处存储虚表
- 虚表数量：有且只有一个基类允许有虚函数（多重继承会产生多个虚表，两个基类都有虚函数的情况）
- 虚表内部函数的数量：虚析构可能会生成两个虚表项，所以不允许写虚析构
- 虚表函数的顺序：避免函数重载（函数重载会影响虚表函数的定义顺序），C语言没有函数重载
- 不允许使用虚继承：不同编译器的做法都不一样