

2020/12/16_COM_第2课_C++版接口设计以及通用插件的设计

笔记本： COM

创建时间： 2020/12/16 星期三 14:55

作者： ileemi

- [C++版接口设计](#)
- [C++ typeid 的使用](#)
- [通用的插件设计](#)

C++版接口设计

纯虚函数替换 dll 中的数组，编译实现纯虚函数的时候内部有一个虚表。

使用抽象类设计接口，使用抽象类指针获取表（接口）（`ICRInterface* GetInterface`）。

使用这个虚表的时候，在 dll 中重定义一个类来继承这个类（虚函类）（实现虚函数等价与填表）。

给表的时候，返回子类的类名（`return new CLine()`）。

接口设计的注意事项：

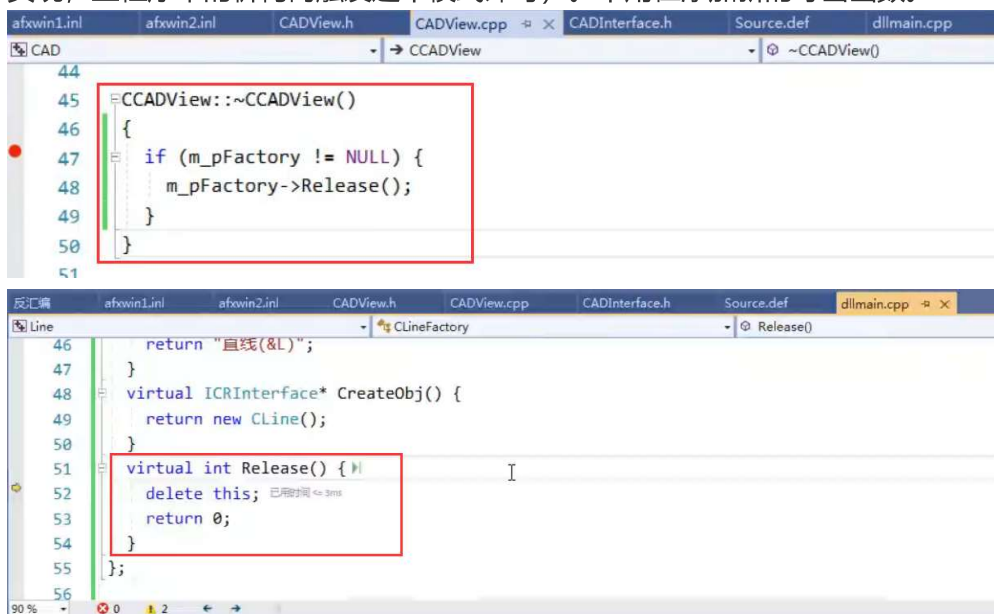
- 接口不变原则（存在兼容性问题）
- 抽象类一般不加成员
- 抽象类成员函数的位置一般也不要改变（改变后虚表错位）
- 主程序更新，所有插件都需要更新（插件不更新，使用主程序新功能的插件会导致程序崩溃），解决办法就是重新定义一个纯虚函数继承原来的纯虚函数，新定义的纯虚函数中添加新的功能。派生类使用的新功能的时候进行判断（可以判断类名），是新定义的纯虚函数就使用新功能。

```

7  class ICRInterface {
8  public:
9      virtual const char* GetMenuName() = 0;
10     virtual int OnDraw(HDC hdc) = 0;
11     virtual int SetBeginPoint(POINT pt) = 0;
12     virtual int SetEndPoint(POINT pt) = 0;
13 };
14
15 class ICRInterface2 :public ICRInterface {
16 public:
17     virtual int New1(POINT pt) = 0;
18 };
19
20 ICRInterface* pObj = new CLine();
21 if (pObj->GetClassName() == "ICRInterface2") {
22     New1();
23 }
24
25 class ICRInterface2 :public ICRInterface {
26 public:
27     virtual const char* GetClassName() {
28         return "ICRInterface2";
29     }
30     virtual int New1(POINT pt) = 0;
31 };

```

- dll 中 new 的对象在主模块中进行释放会导致程序崩溃。解决办法：对象在 dll 中 new 的就必须在 dll 中进行释放（可以在类工厂中定义一个方法，dll 中实现，主程序中的析构内触发这个模式即可）。不用在添加新的导出函数。



```

44
45 ~CCADView()
46 {
47     if (m_pFactory != NULL) {
48         m_pFactory->Release();
49     }
50 }
51
52
53
54
55
56

```

```

46     return "直线(&L)";
47 }
48 virtual ICRInterface* CreateObj() {
49     return new CLine();
50 }
51 virtual int Release() {
52     delete this;
53     return 0;
54 }
55 };
56

```

C++ typeid 的使用

使用时包含头文件 `typeinfo`

返回定义时的类型，不具备多态性。

代码示例:



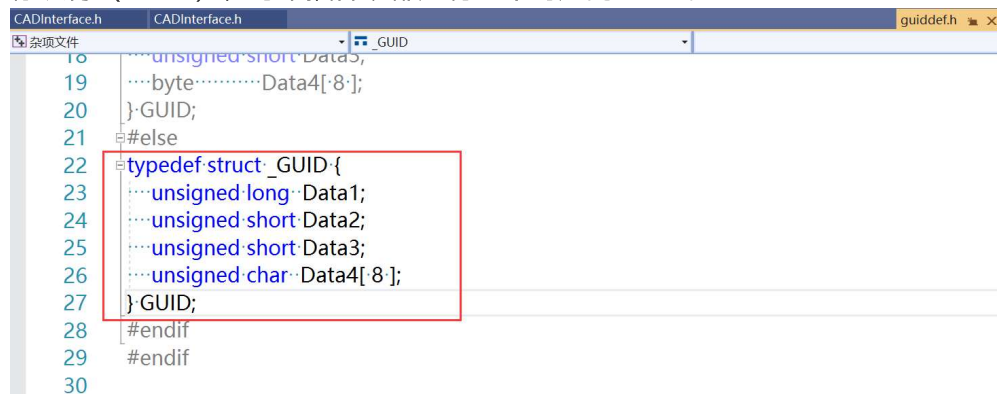
通用的插件设计

通用的插件 (插件1 + 插件2 + 插件3 + UI插件) --> 搭积木

插件 (任何语言可以使用) --> 远程电脑 (插件在服务器)

通用的插件的前提:

- 通用的接口 (接口未知、数量未知)
`void** GetInterface()` 或者 `void*[] GetInterface()`
微软做法: `void* QueryInterface(int type); return ary[typr];`
- 插件进行唯一区分 (避免同一个程序使用同名不同功能的插件), 使用全局唯一标识符 (GUID), 每个插件以都应有一个对应的GUID。



之前: `void* QueryInterface(const char* name); return ary[typr];`

升级: `void* QueryInterface(const GUID& id); return ary[typr];`

```
// {BAB705C7-3A29-4174-A1B7-44B66D4FA275}
static const GUID id = {
    0xbab705c7,
    0x3a29,
    0x4174,
    { 0xa1, 0xb7, 0x44, 0xb6, 0x6d, 0x4f, 0xa2, 0x75 }
};
```

- 返回值统一 (不好设计, 错误内容不好定义, 使用微软定义的错误码, 对应的头文件 -- **winerror.h** (**E_OUTOFMEMORY**))
`PRESULT QueryInterface(const GUID& id, void** ppObject);` --> 返回值通过参数进行返回。

