# 从这一节课开始，我们来学习MFC编程

MFC应用程序表面上看是找不到WinMain函数的,也看不见wind32常见的注册窗口,创建窗口,显示窗口,更新窗口和消息循环的代码,但是它实际上是有的,保存在mfc源码文件夹里面,它的代码已经写好,不需要我们写,只是在编译的时候链接到我们的项目,我们只需要写自己感兴趣的消息处理,这些代码被分散到不同的文件中,有写代码做了一点处理.目的是使得我们编程更加容易.我们必须要理解mfc程序的运行机制.

## 下面就是学习演练.我们先使用vs2012来创建项目,后来改为用vs2019继续编写程序,因为是MFC第一课,代码比较简单,不过对于我们理解mfc很有好处

## 1.首先我们使用vs2012来新建一个MFC项目

|  |
| --- |
|  |
|  |
|  |
|  |

### 可以看出vs2012的mfc工程比vc6的要复杂很多

|  |
| --- |
|  |

## 2.这个mfc2012帮我们创建了一些额外的窗口：输出窗口、文件视图窗口、类视图窗口、属性窗口、树状视图窗口等等，如果不需要，可以把他们取消勾选，我们另外新建一个mfc项目，做了上面那几步后，继续点击next按钮直到出现这个界面，初始状态是这样子的，

|  |
| --- |
|  |

## 把不需要的取消勾选，如图

|  |
| --- |
|  |

## 点击完成，项目的最终效果如下

|  |
| --- |
|  |

## 3.我们先用这种设置来学习，等以后学会了，有需要才使用高级选项。虽然外观和vc6一样，但是功能还是满丰富的。

|  |
| --- |
|  |
|  |

## 4，点击类视图可以看到它有5个类

|  |
| --- |
|  |

## 5.这几个类的基类有些已经更新了，如CAboutDlg现在继承自CDialogEx，CDialogEx是CDialog的派生类，Cappmfc1App现在继承自CWinAppEx，他是CWinApp的子类，CMainFrame现在继承自CFrameWndEx他是CFrameWnd的子类

|  |
| --- |
|  |
|  |
|  |

## 6.vs2012MFC的类的方法和vc6也有一些不一样，多了一些方法

|  |  |
| --- | --- |
| VC++6.0 | Vs2012 |

## 7.有一个很奇怪的现象，在mfc应用程序中根本找不到WinMain函数，那是因为他被隐藏起来了，由连接器在链接的时候把它链接进来。我们只需要处理自己感兴趣的消息，添加处理代码即可。那么怎么找到这个WinMain函数呢？在MFC的源码了里面，进入vs2012的安装目录，人家点击VC-》altmfc->src->mfc,就可以进入MFC的源码目录。也可以使用vs2019，他的源码路径：C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\VC\Tools\MSVC\14.28.29910\atlmfc\src\mfc

|  |
| --- |
|  |

## 8.然后在地址栏旁边的搜索框里面输入：WinMain，找到一个叫做appmodule.cpp结果如下

|  |
| --- |
|  |

## 9.用记事本打开这个源文件，可以看到他里面有一个\_tWinMain他的参数和win32程序的WinMain函数的参数是一模一样的，其实它确实就是代替win32的WinMain函数的函数。这个WinMain是在应用程序类初始化的时候被调用的。它在里面会调用一个接着AfxWinMain的函数，参数是一样的，这个函数在一个叫做winmain.cpp的源文件里面。

## 10.那么这些mfc类是如何和WinMain函数关联起来的呢？注意：c++有一个特性，如果你在main函数之前定义了全局变量或者全局对象，这个变量或者对象会在main函数最新之前完成初始化。在mfc的原理也是一样的，他在应用程序类里面定义了一个全局对象变量theApp。所以当程序运行的时候，会先初始化这个全局对象，然后就会进入应用程序类的构造函数中，我们知道，子类初始化的时候会先调用父类做了是CWinAppEx的构造函数初始化父类对象，在这个构造函数里面会调用WinMain函数。这样子他们就关联起来了。

## 11.那么，它为什么要定义一个全局对象呢？因为它是代表应用程序的顶级对象，每一个mfc应用程序有且只有一个theApp全局对象。mfc用它来唯一表示我们的应用程序，就相当于win32应用程序的实例句柄的作用。

## 12.我们在我们的项目在找不到win32开发时候的代码,ns2因为微软帮我们写了,把他们放在mfc的源码在,其中appmodule.cpp有一个\_tWinMain宏,它其实即使WinMain,在他里面调用一个叫做AfxWinMain的函数,然后定义窗口类和注册窗口类的代码又在另外一个文件,我们需要在源码里面搜索.如果你有兴趣可以新建一个mfc项目,然后用vs打开mfc对应的源码,设置断点来慢慢跟踪调试.不过这个非常耗时.而且你必须找对文件.反正原来WinMain里面的所有代码你能够找到.

## 13下面我们一起来学习一下模拟CWnd类的封装过程,新建一个工程,取名WinMain,是一个window应用程序项目.选择空项目,然后新建一个cpp文件叫WinMain.cpp,然后我们在这里编写代码,先定义一个CWnd类

|  |
| --- |
|  |

## 14,然后我们来实现这个CWnd类的成员方法,先是CreateEx,他就是就是调用win32 sdk平台的全局函数CreateWindowEx函数

|  |
| --- |
|  |

## 15,然后是CWnd的ShowWindow函数

|  |
| --- |
|  |

## 16.然后是UpdateWindow函数

|  |
| --- |
|  |

## 17.写WinMain函数,注意这是一个模拟代码

|  |
| --- |
|  |

## 18,把显示窗口和更新窗口代码和win32平台sdk的比较一下,发现他们少了一个窗口句柄参数,因为它已经被封装为类成员函数

|  |
| --- |
|  |

### 需要注意的是这个wnd实例对象它其实不是一个窗口,它是一个窗口类对象,他持有一个窗口句柄的成员变量,他是通过这个成员变量来和一个具体的窗口关联起来.窗口的销毁和这个类没有一点关系,但是为了防止内存泄漏,需要在这个类对象的生命周期结束之前先把窗口销毁以释放窗口资源所占用的内存,否则如果窗口类被销毁了但是窗口没有销毁,可能会发生内存泄漏.所以上面的代码还不完善,需要在析构函数里面添加销毁窗口的代码.也就是说其实MFC项目里面的CView类和CMainFrame类他们的对象并不是真正意义上的窗口,他们只是持有一个窗口句柄成员变量,他们通过这个句柄成员变量和特点的窗口关联起来.

## 19.MFC从2008开始在原来vc++6的基础上添加了很多新类，我们需要使用msdn2008慢慢学习，下面我们来学习一下CButton类。这个类只有一个无参构造函数，构造完成对象后可以调用Create方法来初始化。注意：创建按钮需要在父窗口创建之后创建，否则它没有地方放置。也就是按钮需要在父窗口的OnCreate函数里面创建。因为这个函数就是响应WM\_CREATE消息的。一个按钮类它也是一个窗口类，它除了有按钮的样式以外，还可以有窗口样式。可以用|操作符组合这两个样式。

|  |
| --- |
|  |

## 20.可是当我们运行程序的时候，并没有看到按钮,有2个原因，1是因为我们这样子创建的是一个在栈中的对象，是一个局部变量，当函数执行完成，它就被销毁了，所以这里不能定义局部变量，我们把按钮的定义代码剪切然后粘贴到CMainFrame.h中，可以声明为私有成员变量。

|  |
| --- |
|  |

## 21.我们还可以在类名上面点击右键-》添加成员变量，注意在vs2019中，添加成员变量是在类向导中完成的，而且你不能够在类名上面点击右键，你需要选择一个类，然后点击项目-》类向导，然后点击里面的成员变量选项卡，如图

|  |
| --- |
|  |
|  |

### 然后点击添加自定义按钮，出现下面的界面

|  |
| --- |
|  |

### 然后我们这里我们这么填写

|  |
| --- |
|  |

### 点击确定，然后把类向导窗口调小一点，点击应用，然后点击确定

|  |
| --- |
|  |

### 其实还挺麻烦的，比如直接在类的头文件里面写

## 22.第二个问题就是按钮创建后需要显示出来

|  |
| --- |
|  |

### 然后它就可以显示了

|  |
| --- |
|  |

## 23.可以把按钮创建在我们的view类里面，不过需要注意：view类没人没有OnCreate函数，需要我们给他添加WM\_CREATE消息的响应函数，在vs2019中需要选择这个view类，然后点击项目-》类向导-》消息映射，然后选择WM\_CREATE消息

|  |
| --- |
|  |

### 然后点击添加处理程序按钮，发现函数添加进去了

|  |
| --- |
|  |

### 然后点击应用，然后确定退出类向导，然后我们在子类文件创建按钮的代码

|  |
| --- |
|  |

### 运行程序，效果如下：

|  |
| --- |
|  |

## 24.其实，也可以在创建按钮的时候，给他的样式添加一个WS\_VISIBLE指明它是可见的，

|  |
| --- |
|  |

### 效果是一样的：

|  |
| --- |
|  |

## 25.如果我们想在view的OnCreate函数里面添加代码，但是把按钮的父窗口各位空间类MainFrame，怎么办，其实还是满简单的，使用view类的GetParent方法(其实是CWnd类的方法)就可以获取到CMainFrame类的指针

|  |
| --- |
|  |

### 效果如下：

|  |
| --- |
|  |

### 按钮又跑到框架上面去了

## 26.我们可以通过修改按钮的样式把按钮改为单选按钮

|  |
| --- |
|  |

### 效果如下：

|  |
| --- |
|  |

## 27.还可以各位CheckBox

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

## 28.还可以创建3态按钮

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

### 3态按钮和复选框很像，不过连续点击2次会变灰，再点击一次就又回到未选中状态

### 可见，几乎所有的按钮种类都是用CButton来创建的。