

MFC编程基础

Windows应用程序的基本结构

- **Microsoft Windows**是一种基于图形用户界面的多任务操作系统，系统中可以同时运行多个应用程序。
- 每个**Windows**应用程序通过各自的“窗口”与用户进行交互。

应用程序的用户界面

- **Windows**应用程序的用户界面是通过多窗口形式来实现的，它可分为：
 - 单文档
 - 多文档
 - 对话框

单文档用户界面

■ 单文档用户界面

- ❑ 只能对一个文档的数据进行操作的应用程序而设计的。
- ❑ 在这样的应用程序中，必须先结束当前文档的操作，才能切换到下一个文档。
- ❑ 例如：记事本程序

多文档用户界面

■ 多文档用户界面

- ❑ 可以同时对多个文档的数据进行操作的应用程序而设计的。
- ❑ 不同文档的操作在不同的子窗口中进行。
- ❑ 不需要结束某个文档的操作就能在文档之间进行切换。
- ❑ 例如：IE, office

对话框用户界面

- 以对话框的形式对一个文档的数据进行操作的应用程序而设计的。
- 文档数据的操作以各种“控制”来实现，程序以按<确定><取消>按钮来结束。
- 例如：桌面属性对话框

消息驱动的程序结构

- **Windows应用程序的结构属于一种事件（消息）驱动的计算模型：**
 - 程序的任何一个动作都是在接收到一条消息后发生的。
 - 例如，用户从键盘输入，程序会收到WM_KEYDOWN/WM_KEYUP消息；
使用鼠标左键时，会收到WM_LBUTTONDOWN/WM_LBUTTONUP消息
选择某个菜单项时，会收到WM_COMMAND消息
窗口刷新时，会收到WM_PAINT消息
- 大部分的消息都关联到某个窗口：
 - 每个窗口都有一个消息处理过程（函数）。
 - 属于某个窗口的消息将由相应窗口的消息处理过程来进行处理。
 - 每个Windows应用程序都有一个消息队列。
 - windows系统将属于各个应用程序的消息放入其消息队列中，应用程序不断的从该队列中读取消息，并调用相应的处理函数来处理。
 - 这个“取消息-处理消息”的循环过程一直到程序运行结束时，才终止。

Windows应用程序的基本框架

- 主函数 WinMain
- 窗口消息处理函数

主函数 WinMain

- 每个Windows应用程序都必须提供一个主函数：**WinMain**。
 - 程序的执行从WinMain开始。
 - WinMain的主要功能是：注册窗口类、创建应用的主窗口、进入消息循环。
 - 注册窗口类：
 - 向windows报告本应用程序将创建那些种类的窗口，如窗口的基本风格、消息处理函数、图标、菜单等等。
 - 创建应用的主窗口：
 - 根据相应的窗口类创建并显示程序的主窗口，其他窗口在需要时创建。
 - 进入消息循环：
 - 实现不断的从消息队列中取消息，发现发送到相应的窗口（调用相应的窗口消息处理函数）

窗口消息处理函数

- 负责处理发送到相应窗口的消息，典型框架为

```
LRESULT CALLBACK WindowProc ( HWND hWnd, //窗口标识
                               UINT message, //消息标识
                               WPARAM wParam, //消息的参数1
                               LPARAM lParam //消息的参数2
                             )
{
    switch(message)
    {
        case WM_CHAR: //字符键消息
            ...
        case WM_COMMAND: //菜单消息
            ...
        case WM_PAINT: //窗口刷新消息
            ...
        case WM_DESTROY: //窗口关闭消息
            PostQuitMessage(0);
            break;
        default: //系统默认消息处理
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}
```

MFC类库

- MFC是微软提供的支持windows应用程序开发的一个类库。库中大部分类用于实现面向对象的方式进行windows应用程序的开发。
- 封装了windows应用程序中的一些基本元素（如窗口）的基本功能。
- 提供一种基于“文档-视”结构的面向对象的应用程序框架。
- MFC6.0提供了200多个类，初学者很难弄清这些类之间的复杂关系。
 - 初学者应该首先初步了解MFC类库的层级结构。
 - 熟悉常用类的功能。

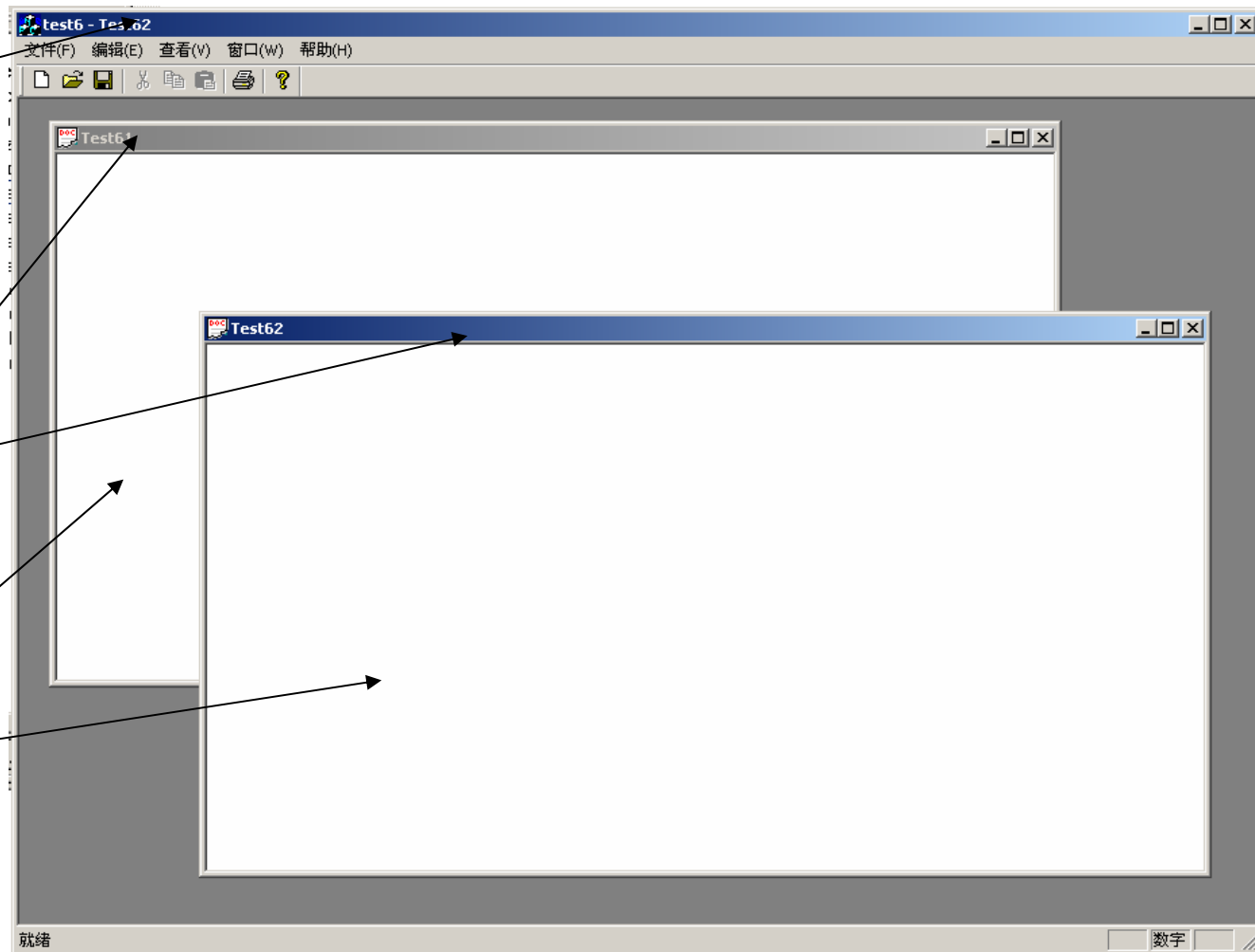
面向对象的Windows应用程序结构

- Windows应用程序由以下对象构成
- 窗口对象
 - 每个窗口都是一个对象，能处理windows的窗口消息
- 文档对象
 - 每个窗口处理的数据也是对象，称为文档对象
- 应用程序对象
 - 每个应用程序都是一个对象，负责管理属于它的窗口对象和文档对象以及实现消息循环

CMDIFrameWnd

CMDIChildWnd

CView



MFC类库分类

- 应用程序类
- 文档类和文档模板类
- 窗口类
- 图形和打印类
- 集合类
- 文件和数据库类
- OLE支持类
- Internet和网络类
- 调试和异常类
- 各种辅助类

应用程序类 CWinApp

- CWinApp提供了Windows应用程序的管理功能，包括对属于本应用的各个窗口的管理以及实现消息循环。
- 每个应用只能有一个CWinApp或其派生类的对象，在应用运行中这个对象协调其他对象的动作。
- CWinApp封装了基于MFC的windows应用的初始化、运行和终止等功能。
 - 注册应用的窗口类，然后创建显示主框架窗口，之后向应用的消息处理函数发送消息。

文档类 CDocument

- 文档类(Document Classes)封装了应用的数据管理，文档类对象有文档模板创建。
- CDocument用于存储和管理程序中的数据
- 与CView类一起构成“文档-视”软件体系结构
- “文档-视”软件体系结构的好处？
 - 在这种结构中，程序所处理的数据保存在“文档”对象中，数据的显示以及与用户的交互功能则由“视”对象来完成，一个“文档”对象可以对应多个“视”对象
 - 使得数据的内部存储形式和数据的外部表示形式相互独立，对同一个文档数据可以用不同的方式进行显示和操作。

文档模板类 CDocTemplate

- 文档模板类(Document Template Class)将文档、视图以及框架窗口相互联系起来。
 - 在创建新文档或视图时协调文档、视图和框架窗口对象的创建
- CDocTemplate用于支持“文档-视”结构
 - 用于创建由三个对象（CFrameWnd, CView, CDocument）所构成的对象组

窗口类

- 窗口类(Windows Class)包括
 - 框架窗口类(Frame Windows)
 - 对话框类(Dialog Boxes)
 - 视图类(Views)
 - 控件类(Controls)
 - 控件条(Control Bars)
 - 分割窗口支持类(CSplitterWnd)
 - 属性簿(Property Sheets)
 - 菜单类(Menus)
-

窗口支持类 CWnd

- MFC定义了窗口支持类CWnd来支持窗口类。
- CWnd作为所有窗口类的基类，包含了大量的成员函数，为窗口类提供基本操作。包括以下功能
 - 一般的消息处理
 - 键盘和鼠标消息处理
 - 创建初始化窗口、管理窗口的尺寸位置、管理窗口的滚动等
 - 菜单管理
 - 窗口状态和窗口间的关系管理等

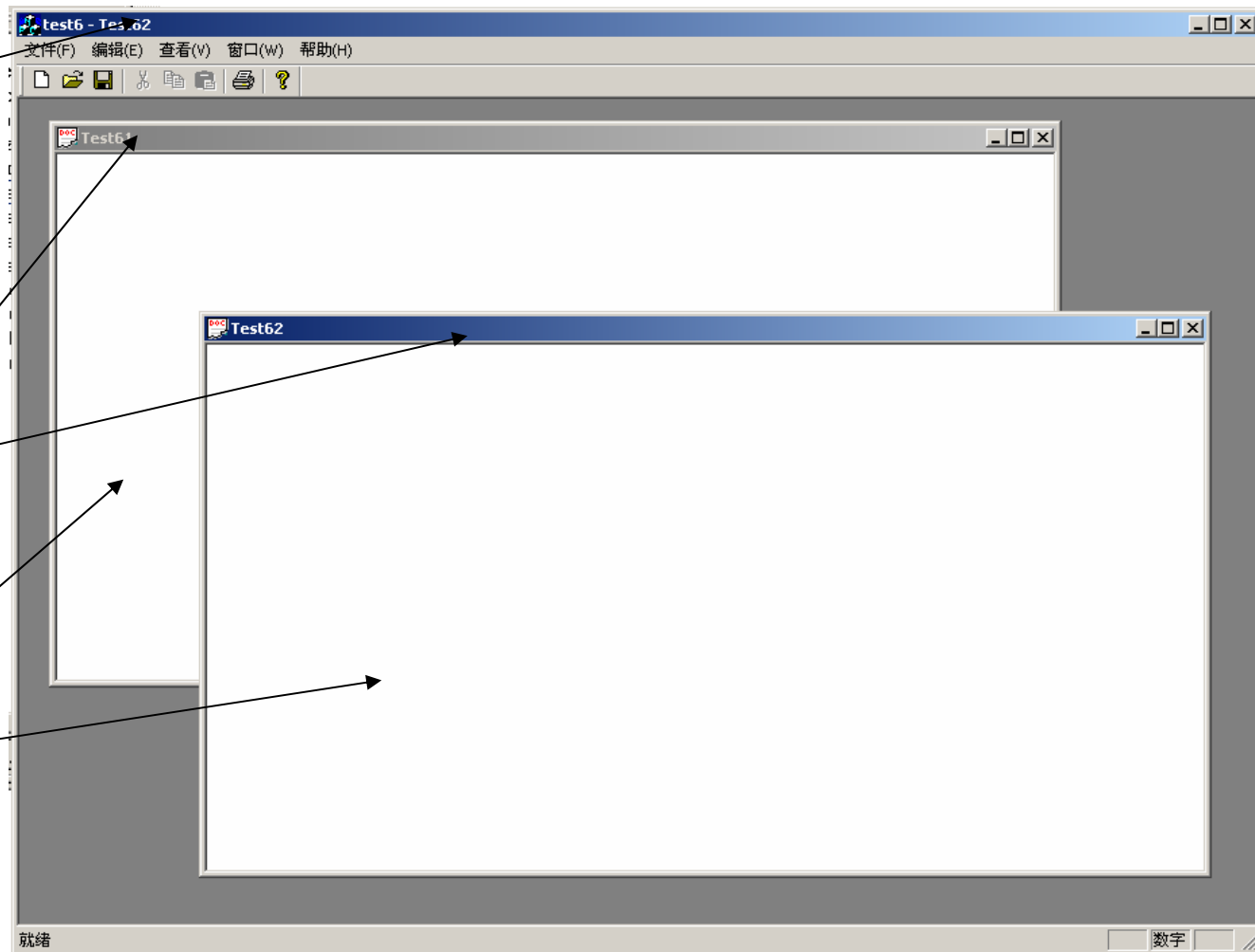
框架窗口类

- 用户通过框架窗口类与windows进行交互。
- 框架窗口通常包含其他窗口，如：视图、工具栏、状态栏等等。
- 常用框架窗口类：
 - CFrameWnd、CMDIFrameWnd、CMDIChildWnd

CMDIFrameWnd

CMDIChildWnd

CView



CFrameWnd

- 提供了框架窗口的基本功能，包括对工具条的管理以及对“视”的管理等。
- 创建框架窗口类的对象后，调用成员函数Create或LoadFrame来创建窗口。
 - Create函数的参数：类注册名、窗口名、风格、位置、尺寸等等。
 - LoadFrame需要的参数少，因为它可以从资源获得默认值，这些资源的ID号要一样，如：IDR_MAINFRAME

- CMDIFrameWnd 多文档主框架窗口类
 - 多文档应用中的主窗口
 - 具有管理子窗口的功能
- CMDIChildWnd 多文档子框架窗口类
 - 实现子窗口的基本功能

视图类CView

- 一类特殊的窗口
- 位于单文档应用主窗口和多文档应用子窗口的客户区（可显示区）。
- CView类管理着框架窗口的客户区；
- 为用户与windows之间提供可视接口；
- 该类接受键盘、鼠标的输入，还允许用户对数据预览和打印；
- 视图一般通过文档模板与文档关联；
- 提供了一些派生类，以方便数据的显示。如
 - ScrollView（带滚动功能的视）
 - CEditView(带编辑功能的视)
 - CFormView（带表格功能），
 - CHtmlView（带Web浏览功能）

对话框类

■ 通用对话框类

- ❑ CFileDialog类：打开文件(File Open)和另存为(Save As)，封装了打开和保存文件的功能。
- ❑ CFontDialog类：字体选择对话框；
- ❑ CColorDialog类：颜色选择对话框；
- ❑ CPrintDialog类：打印及打印设置对话框；

■ 属性簿支持类

- ❑ CPropertyPage类：属性簿的页

控件类

- 控件类封装了通用的windows可视控件；
- MFC还提供了一些新控件，如控制栏；这些新的控件类的名字以“Ctrl”结尾；

常用控件类

- CStatic: 静态文本控件
 - 成员函数SetWindowText改变显示的文本;
- CEdit: 编辑控件
 - 单行或多行编辑控件, 支持剪切、粘贴、文本选择等操作。
- CButton: 按钮控件
 - 命令按钮、复选框、单选按钮
- CListBox: 列表控件
 - 用户从列表框中选择一个选项;
- CComboBox: 组合框控件
 - 将编辑框(CEdit)和列表框(CListBox)组合起来。

图形类和打印类

- 在Windows环境下，所有图形输出都是在设备描述表(Device Context)上进行的。
- 图形和打印类封装了设备描述表和绘图工具。

常用输出类

- CDC类：
 - 输出类的基类，用来支持屏幕或打印机所有的图形输出和状态处理。
- CPaintDC类：
 - 支持CWnd成员函数OnPaint使用的设备描述表，只能响应WM_PAINT消息。
- CClientDC类：
 - 支持在窗口客户区使用的设备描述表。如：响应鼠标事件等。

图形工具类

- CBrush类封装了图形设备界面(GDI)刷子，可被选为当前刷用来填充所画对象的内部区域。
- CPen类封装了图形设备界面(GDI)笔，可被选为当前笔画对象的边界。

文件输入输出类

- 文件输入输出类为传统的磁盘文件提供界面。
- 所有的CFile类派生类都可被CArchive类使用进行串行化。
- CFile类封装了二进制文件的数据及操作。

简单应用示范

- 1. 创建一个单文档应用程序Hello World
- 2. 用MFC的向导创建一个多文档应用程序

小结

- Windows应用程序的基本结构
 - 单用户
 - 多用户
 - 消息驱动
- Windows应用程序的基本框架
 - 主函数
 - 窗口消息处理函数
- MFC类库
- “文档-视”结构是MFC支持的一种软件体系结构，强调了数据的内部表示形式和数据的外部表示形式的相互独立