

# 第3章 窗口、文档与视图

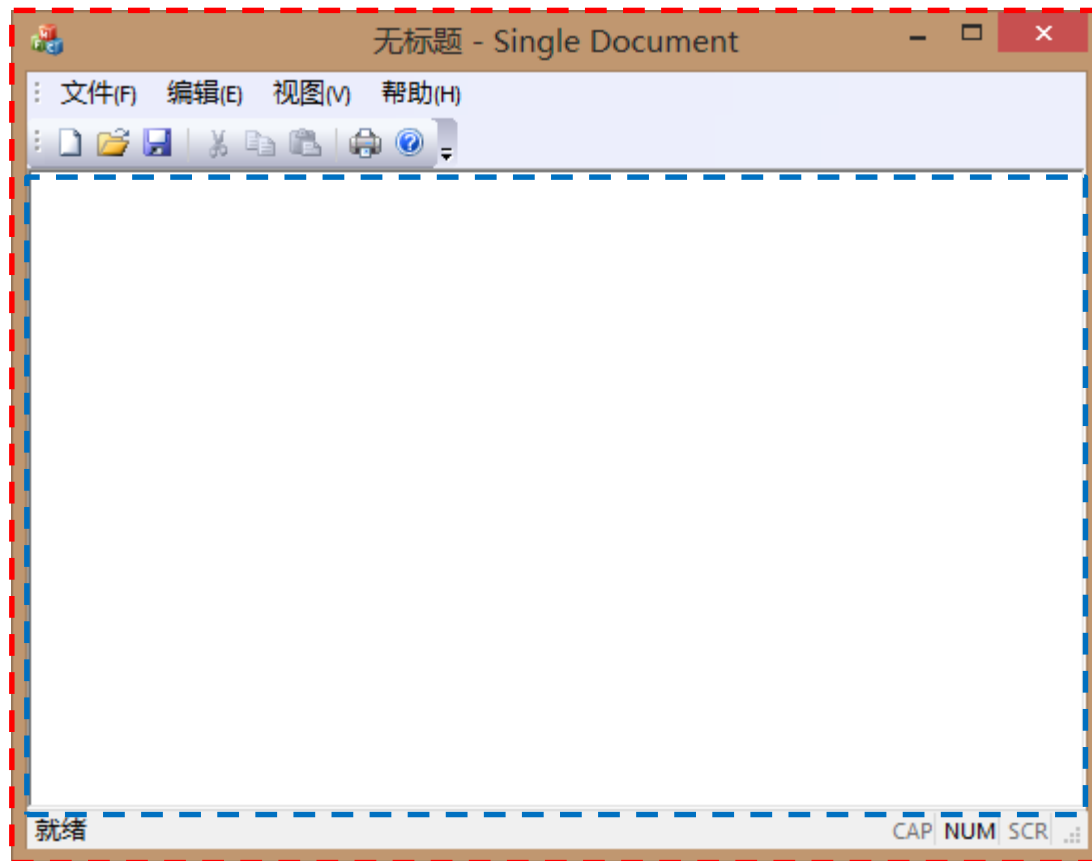
- 单文档应用程序与窗口
- 文档与视图
- 主菜单与快捷菜单
- 滚动与缩放显示
- 窗口分割
- 多文档应用程序的例子

# 边框窗口(1)

- 单文档界面(SDI)程序
  - ✓ CFrameWnd
- 多文档界面(MDI)程序
  - ✓ CMDIFrameWnd与CMDIChildWnd
- 对话框(Dialog)程序
  - ✓ CDialog

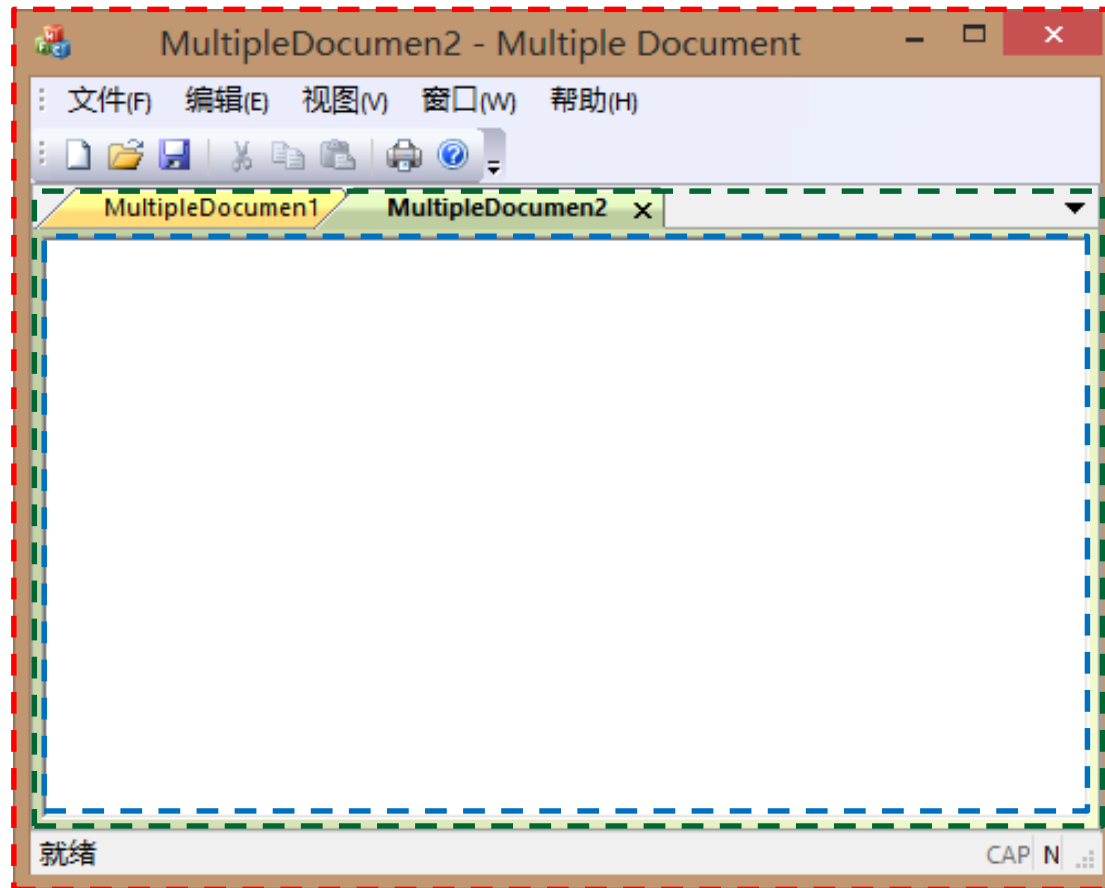
# 边框窗口 (2)

- 主边框窗口 → 视图窗口



# 边框窗口 (3)

- 主边框窗口 → 子边框窗口 → 视图窗口



# 边框窗口 (4)

例3-1

## ■ 主边框窗口例子

✓ GetParent ()、BS\_RADIOBUTTON

## ■ 在CMainFrame类中

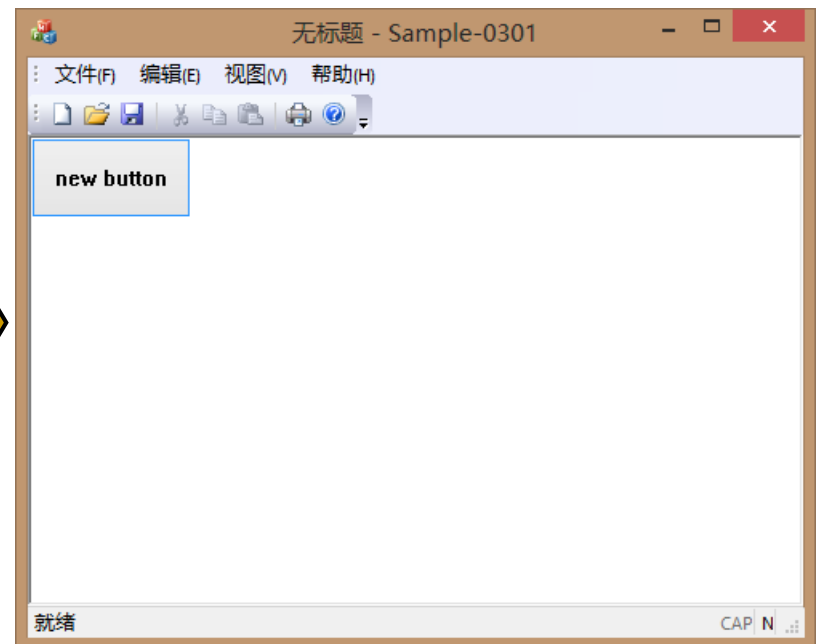
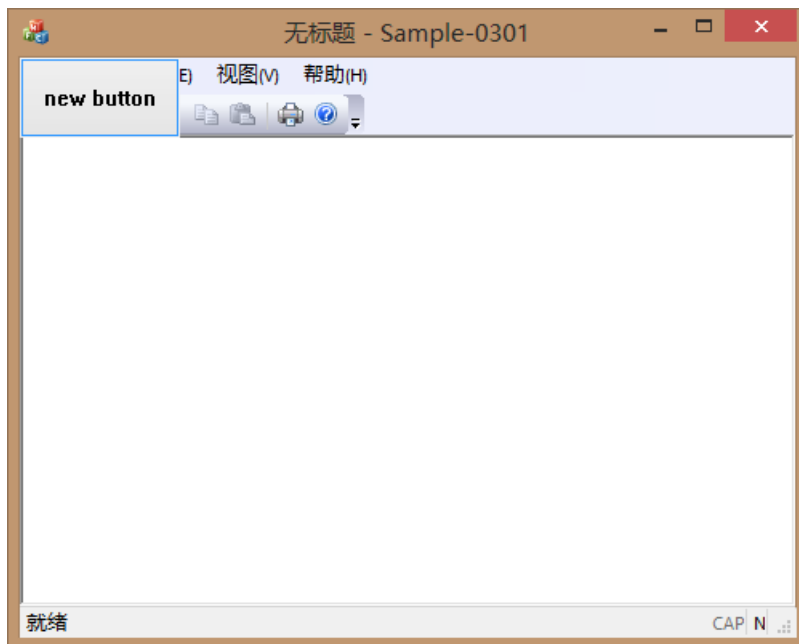
```
CButton m_button;
```

## ■ 在CMainFrame::OnCreate () 中

```
m_button.Create(L"new button", WS_VISIBLE |  
BS_DEFPUSHBUTTON, CRect (0, 0, 100, 50), this,  
0);
```

# 边框窗口 (5)

## ■ 主边框窗口的例子



## 边框窗口 (6)

```
typedef struct tagCREATESTRUCT
{ LPVOID lpCreateParams;
  HINSTANCE hInstance;
  HMENU hMenu;
  HWND hwndParent;
  int cy, cx;
  int y, x;
  LONG style;
  LPCTSTR lpszName;
  LPCTSTR lpszClass;
  DWORD dwExStyle;
} CREATESTRUCT;
```

# 边框窗口 (6)

- 定制边框窗口的例子

- ✓ 窗口样式 (WS\_SYSMENU、WS\_MAXIMIZEBOX、WS\_MINIMIZEBOX)

- 在CMainFrame::PreCreateWindow() 中

```
cs. cx=500;  
cs. cy=400;  
cs. x=0;  
cs. y=0;  
cs. style&=~WS_MAXIMIZEBOX;
```



单文档程序中的CMainFrame类派生自哪个类？

- ☐ A CMDIChildWnd
- ☐ B CDocument
- ☐ C CDialog
- ☒ D CFrameWnd

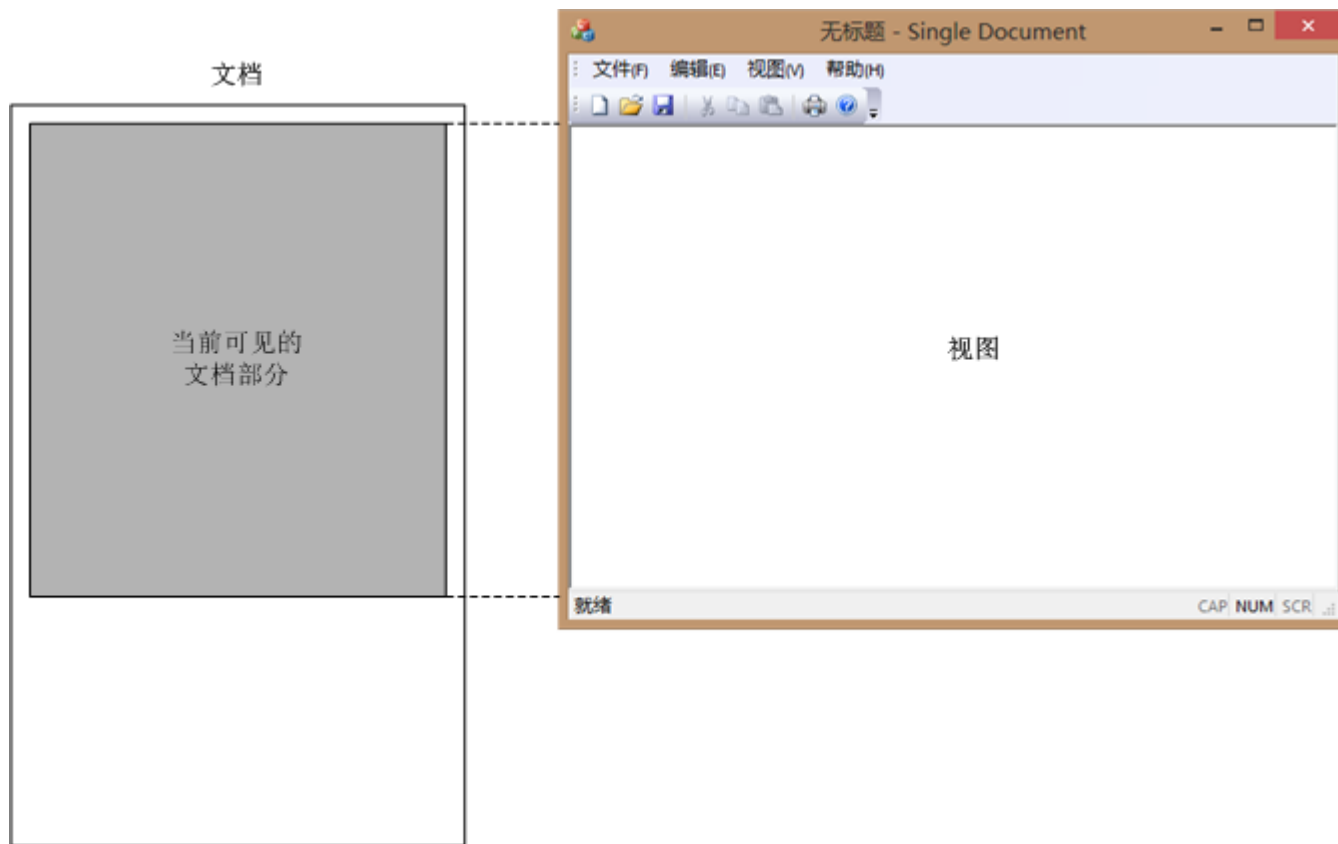
提交

# 文档与视图(1)

- 文档/视图是MFC中最常用的类
- 文档类负责数据管理，包括数据的读取、修改与保存
- 视图类负责与用户交互，将文档类数据以某种方式显示

# 文档与视图(2)

- 文档类基类CDocument，视图类基类CView

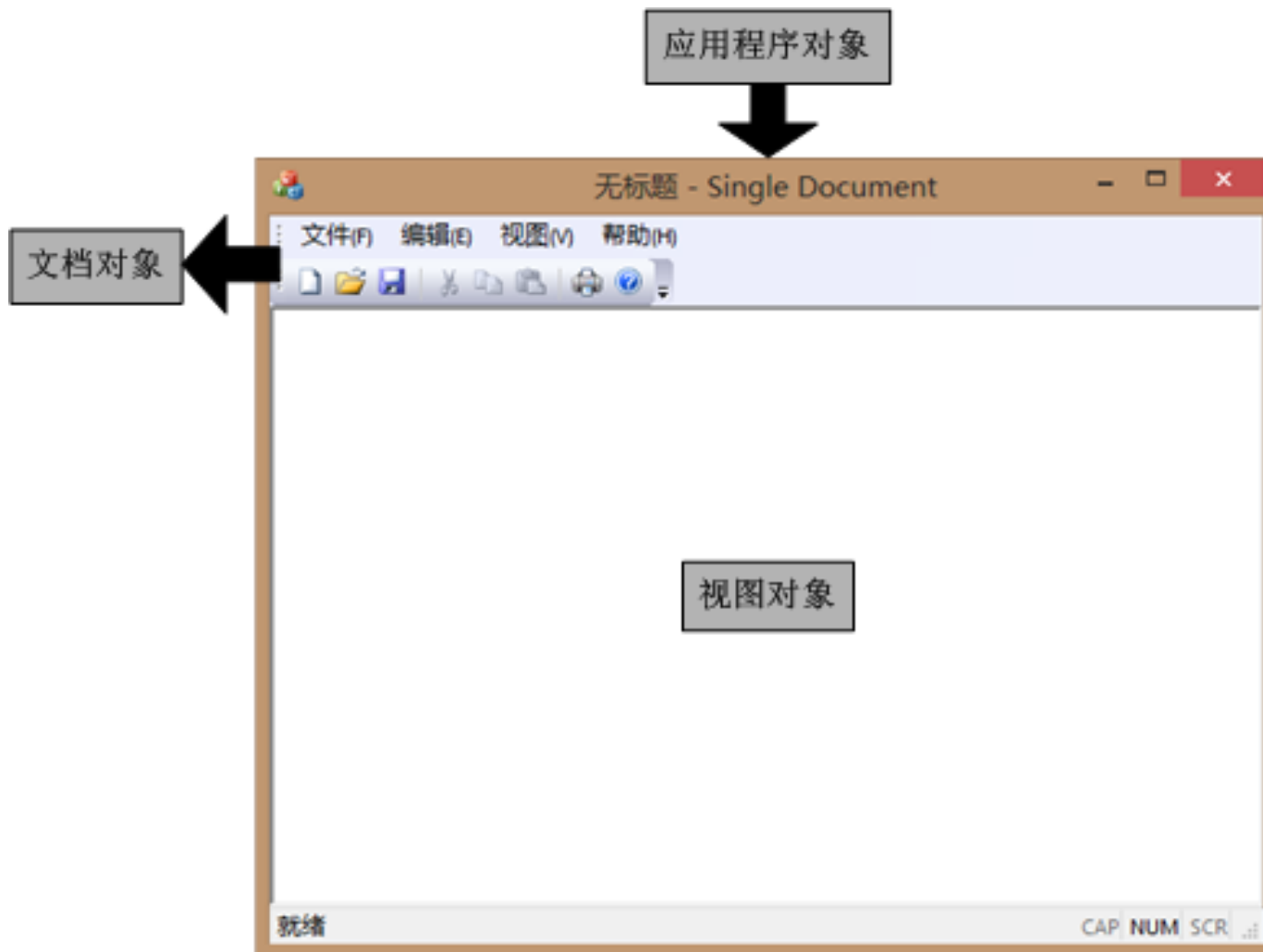


# 文档与视图(3)

- 文档对象管理来自数据源的数据，例如磁盘、串口、网卡等
- 串行化(serialize)从对象写入字节流，或从字节流读到对象

函数类型	说明	函数类型	说明
operator<<	写字节流	operator>>	读字节流
IsStoring	写状态	IsLoading	读状态

# 文档与视图(4)



# 文档与视图 (6)

- CView通过GetDocument() 获得CDocument对象指针，可访问CDocument数据
- CDocument通过UpdateAllViews() 以消息通知视图数据变化，需要重新显示

# 单文档窗口实例1(1)

例3-2

- 在CTestDoc类定义中

```
public:  CString m_str;
```

- 在CTestDoc构造函数中

```
m_str=L"SDI Test Sample!";
```

- 在CTestView::OnDraw() 中

```
CRect rect;  
GetClientRect(&rect);  
pDC->TextOutW(rect.right/2, rect.bottom/2,  
pDoc->m_str);
```

# 单文档窗口实例1(2)

- 在CTestDoc::Serialize()中

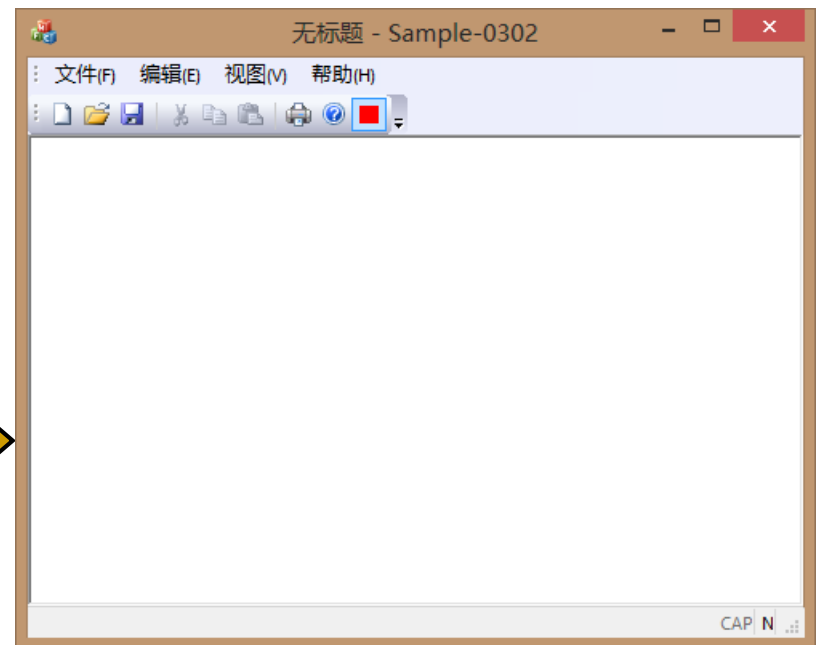
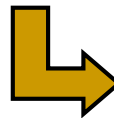
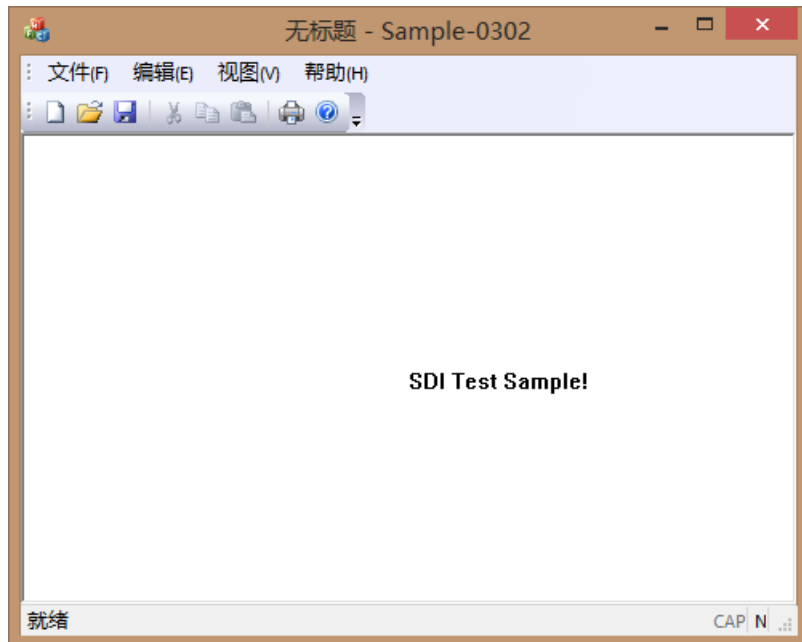
```
if(ar.IsStoring())  
    ar<<m_str;  
else  
    ar>>m_str;
```

- 在工具栏中添加按钮，清空客户区显示

```
m_str="";  
UpdateAllViews(NULL);
```



# 单文档窗口实例1 (3)



# 单文档窗口实例2(1)

例3-3

## ■ 在CTestDoc类定义中

```
private:
    CPoint m_point;
    CString m_str;
public:
    void Set(CPoint& point, CString& str);
    void Get(CPoint& point, CString& str);
```

# 单文档窗口实例2 (2)

- 在CTestDoc::Set() 中

```
m_point=point;  
m_str=str;
```

- 在CTestDoc::Get() 中

```
point=m_point;  
str=m_str;
```

# 单文档窗口实例2(3)

- 在CTestView::OnLButtonDown() 中

```
CTestDoc* pDoc=GetDocument();  
CString str;  
str.Format(L"Mouse Clicked at (%d,%d)",  
point.x, point.y);  
pDoc->Set(point, str);  
Invalidate(true);
```

# 单文档窗口实例2(4)

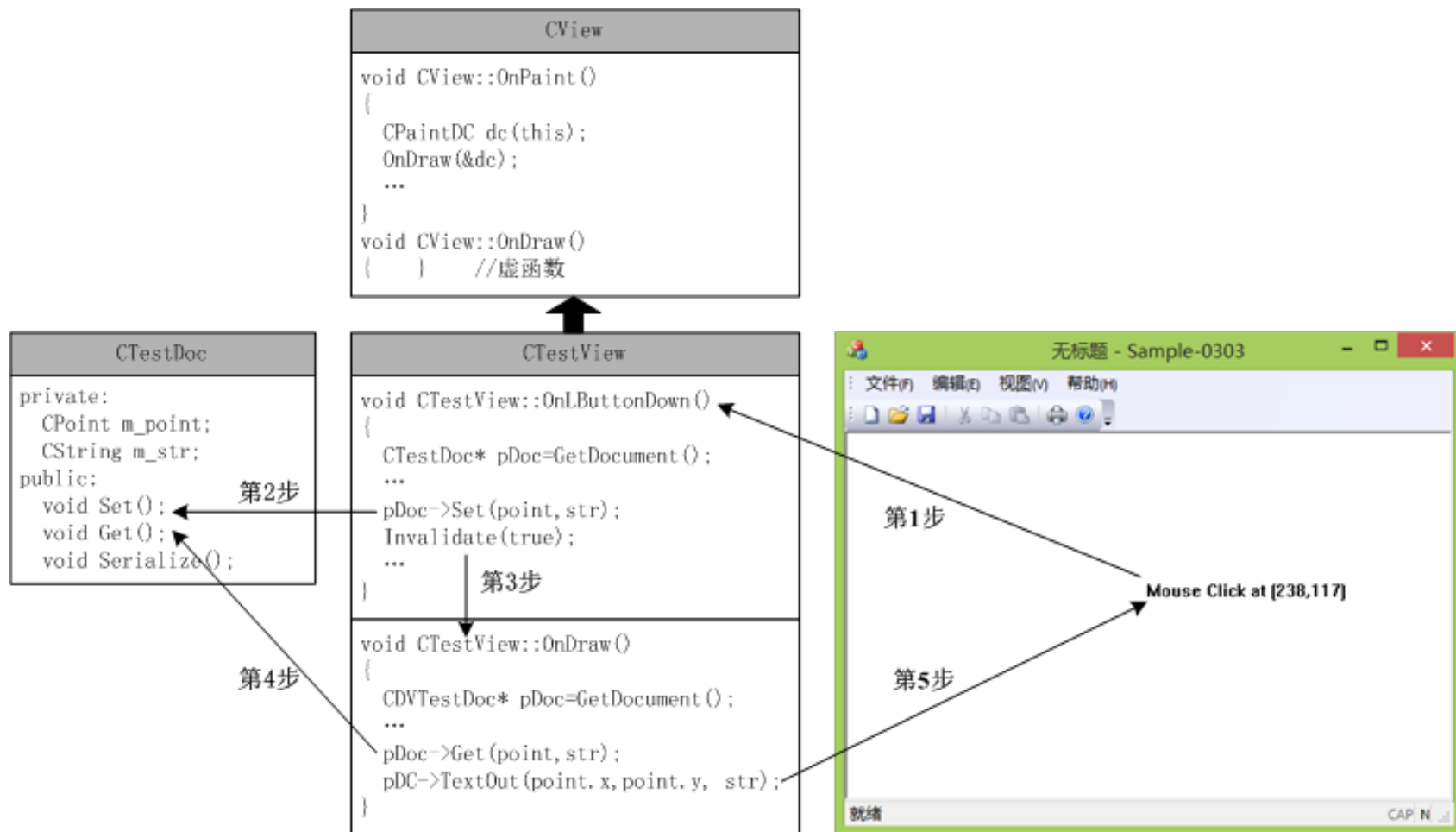
- 在CTestView::OnDraw() 中

```
CPoint point;  
CString str;  
pDoc->Get(point, str);  
pDC->TextOutW(point.x, point.y, str);
```

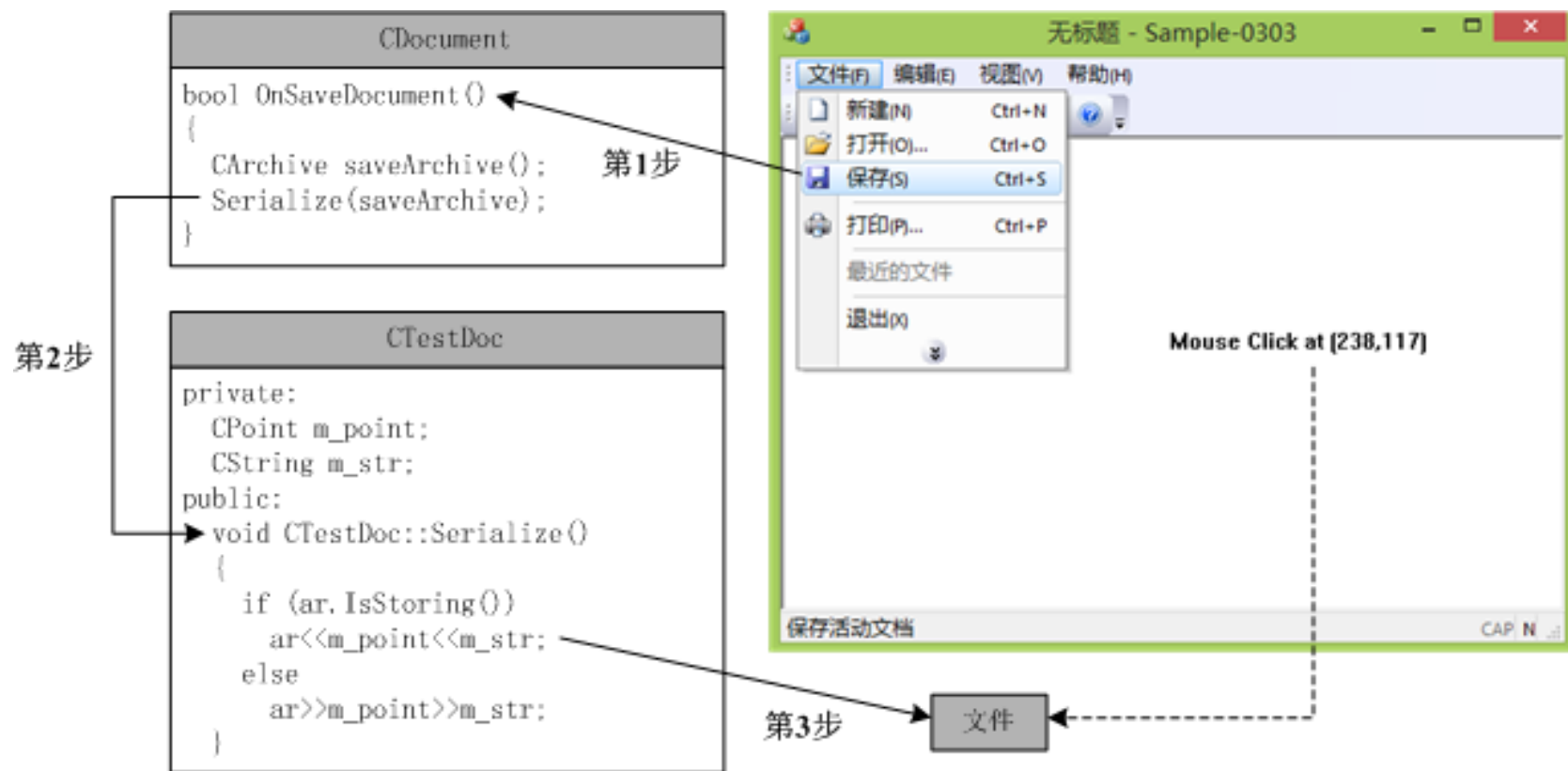
- 在CTestDoc::Serialize() 中

```
if(ar.IsStoring())  
    ar<<m_point<<m_str;  
else  
    ar>>m_point>>m_str;
```

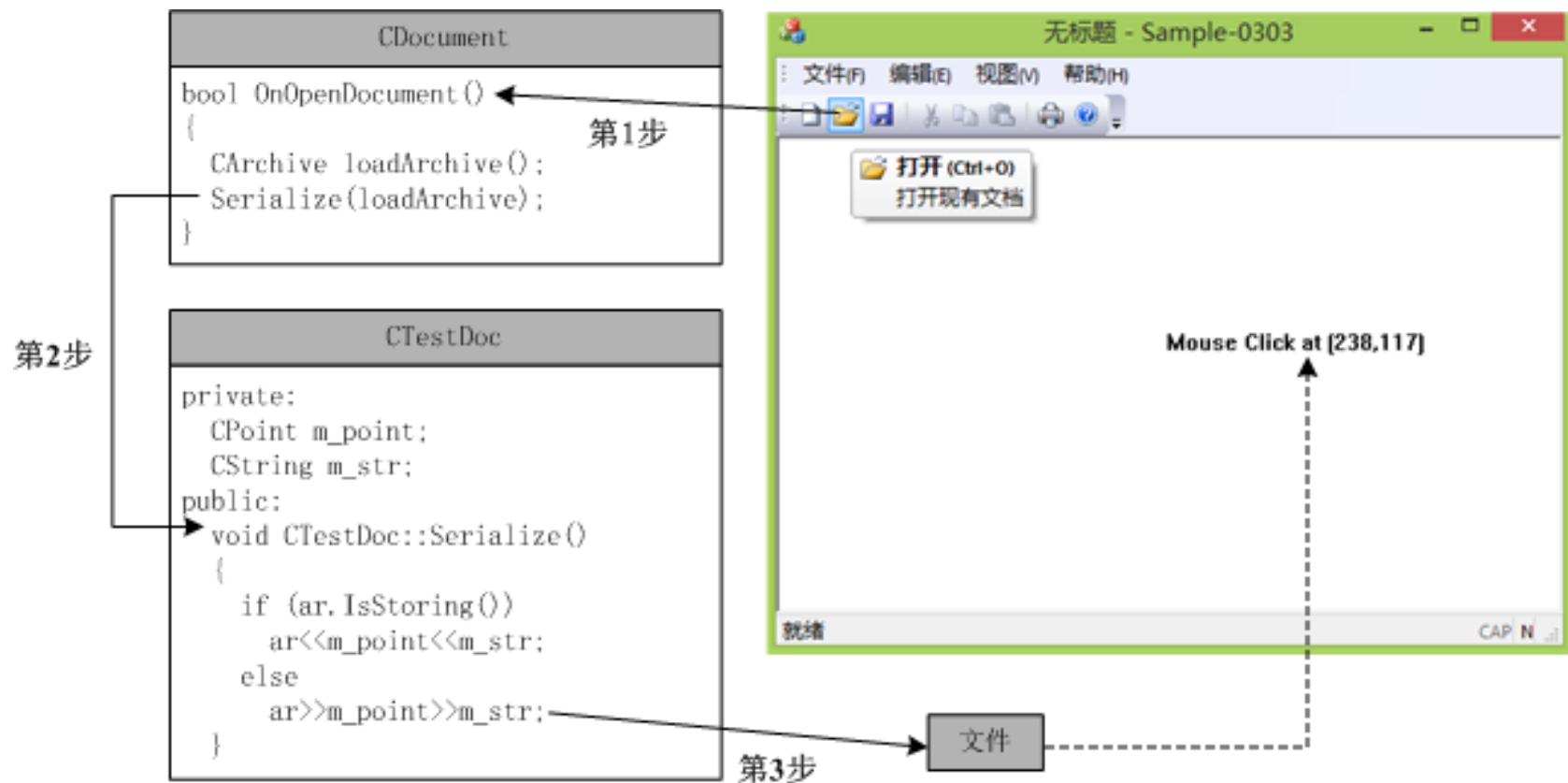
# 单文档窗口实例2(5)



# 单文档窗口实例2(6)



# 单文档窗口实例2(7)





# 菜单操作(1)

- `GetMenu()`：访问菜单
- `GetSubMenu()`：访问子菜单
- `CheckMenuItem()`：标记菜单
- `SetDefaultItem()`：设置默认项
- `SetMenuItemBitmaps()`：设置位图
- `EnableMenuItem()`：菜单生效

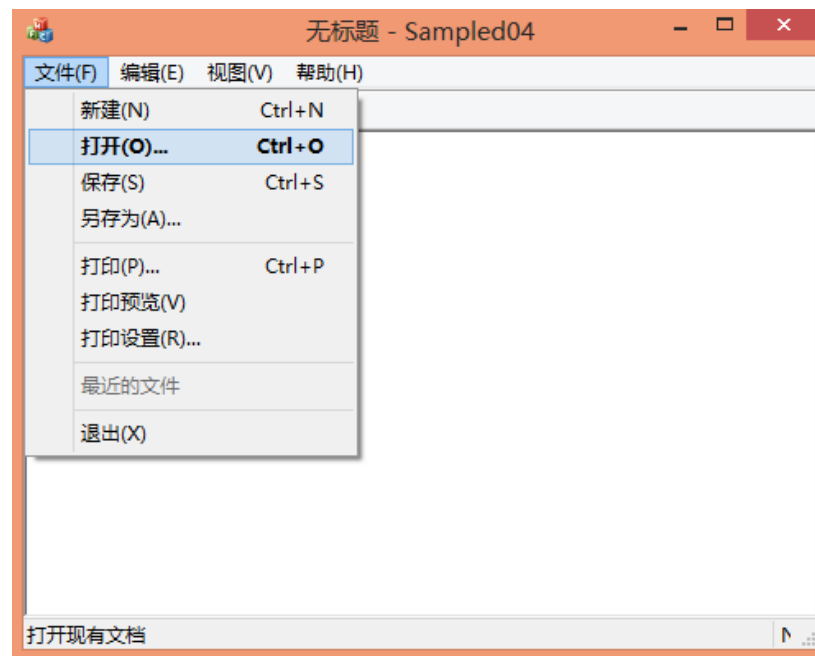
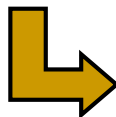
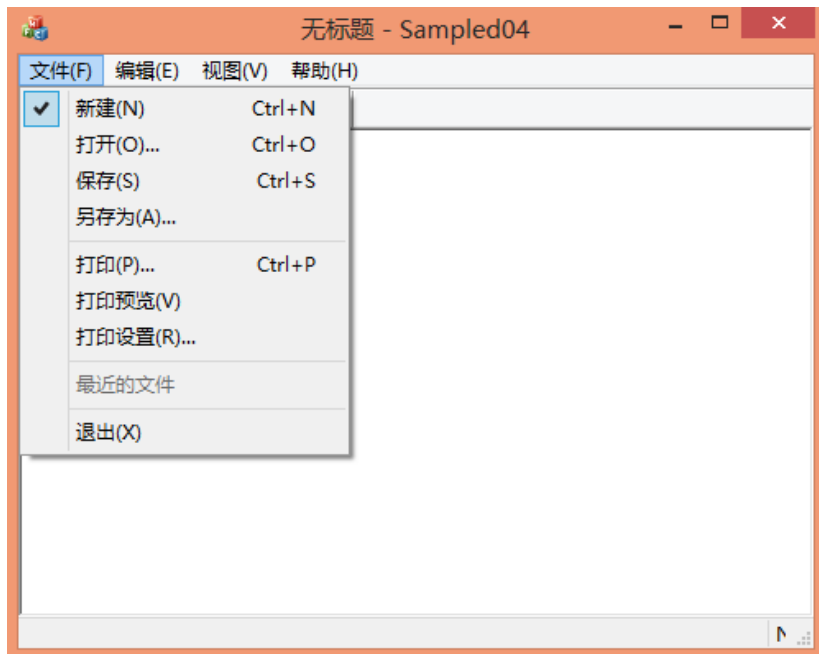
# 菜单操作 (2)

例3-4

## ■ 标记菜单与设置默认项

```
GetMenu () -> GetSubMenu (0) -> CheckMenuItem  
(0, MF_BYPOSITION | MF_CHECKED) ;  
GetMenu () -> GetSubMenu (0) -> CheckMenuItem  
(ID_FILE_NEW, MF_BYCOMMAND | MF_CHECKED) ;  
GetMenu () -> GetSubMenu (0) -> SetDefaultItem  
(1, true) ;  
GetMenu () -> GetSubMenu (0) -> SetDefaultItem  
(ID_FILE_OPEN) ;
```

# 菜单操作 (3)

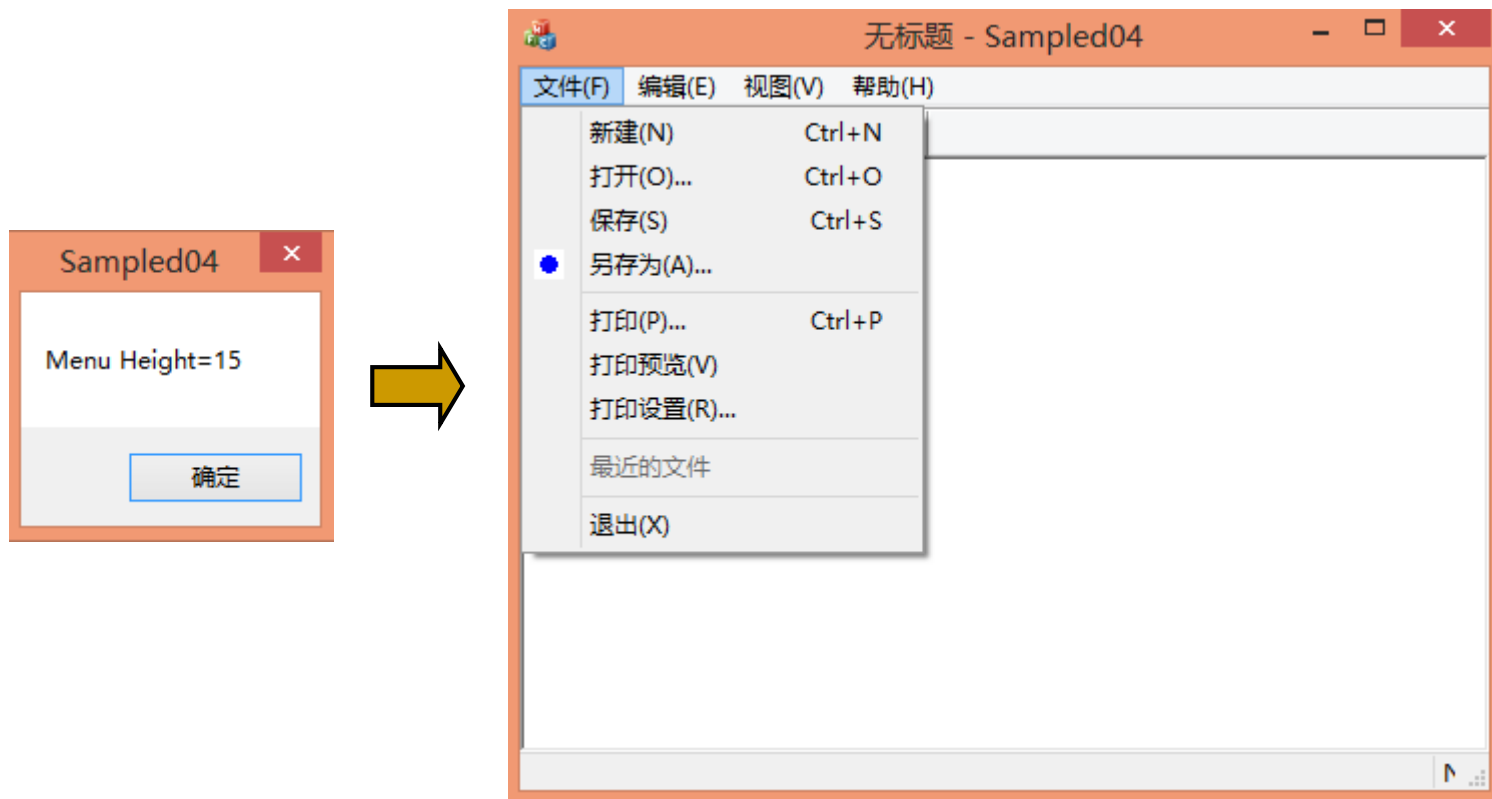


# 菜单操作(4)

## ■ 设置位图与检测高度

```
m_bitmap.LoadBitmapW(IDB_MYBITMAP);  
GetMenu()->GetSubMenu(0)->SetMenuItemBit  
maps(3, MF_BYPOSITION, &m_bitmap, &m_bitmap);  
CString str;  
str.Format(L"Menu Height=%d",  
GetSystemMetrics(SM_CYMENUCHECK));  
MessageBox(str);
```

# 菜单操作 (5)



# 菜单操作 (6)

## ■ 禁用菜单项

- ✓ 在CMainFrame构造函数中

```
m_bAutoMenuEnable=false;
```

- ✓ 在CMainFrame::OnCreate() 中

```
GetMenu()->GetSubMenu(0)->EnableMenuItem(2,  
MF_BYPOSITION|MF_DISABLED);
```

```
GetMenu()->GetSubMenu(0)->EnableMenuItem  
(ID_FILE_SAVE_AS, MF_BYCOMMAND|MF_DISABLED);
```

# 浮动菜单(1)

- 用户单击鼠标右键时，弹出浮动菜单(弹出菜单、快捷菜单)
- 浮动菜单创建方式
  - ✓ 隐式创建
  - ✓ 非隐式创建
- 弹出菜单消息WM\_CONTEXTMENU
  - ✓ 消息中的坐标是屏幕坐标

# 浮动菜单 (2)

例3-5

## ■ 创建菜单IDR\_POPUPMENU

一级菜单: Popup

二级菜单: Line(&L)      ID\_POPUP\_LINE

          Ellipse(&C) ID\_POPUP\_ELLIPSE

## ■ 在CTestView::OnContextMenu() 中

```
CMenu menu;  
menu.LoadMenu(IDR_POPUPMENU);  
CMenu* pMenu=menu.GetSubMenu(0);  
pMenu->TrackPopupMenu(TPM_LEFTALIGN,  
point.x, point.y, this);
```



## 浮动菜单(3)

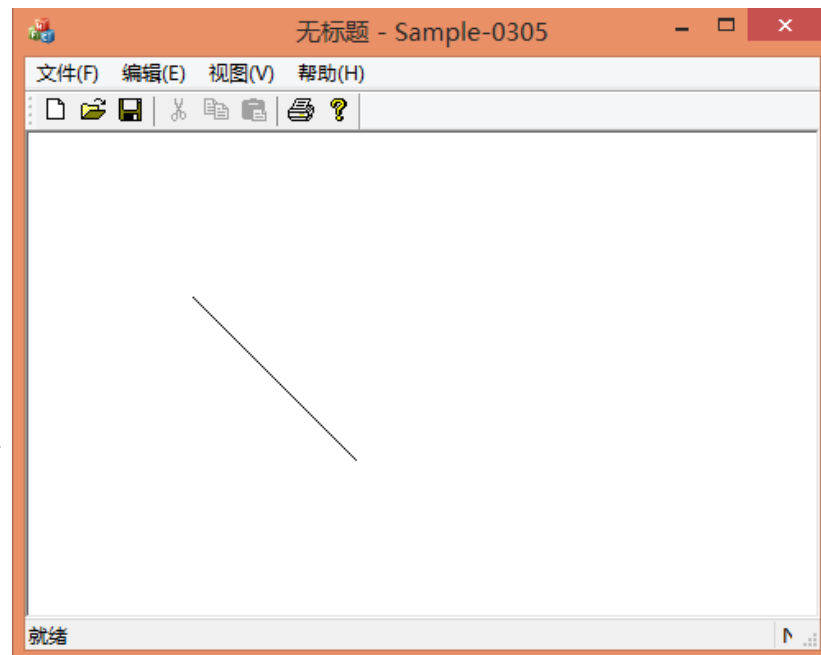
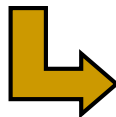
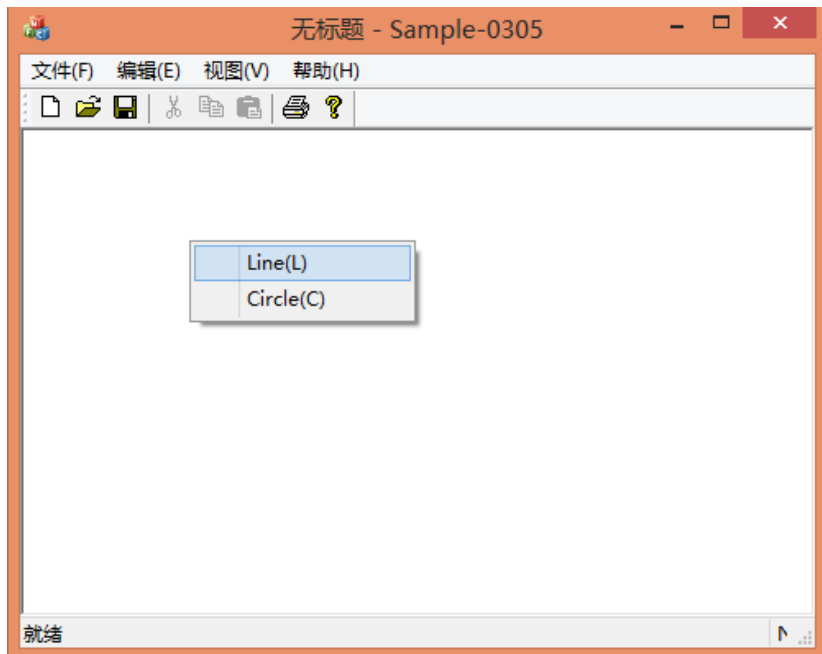
- 在CTestView::OnPopupMenu() 中

```
CDC* pDC=GetDC();  
pDC->MoveTo(100, 100);  
pDC->LineTo(200, 200);  
ReleaseDC(pDC);
```

- 在CTestView::OnPopupMenuEllipse() 中

```
CDC* pDC=GetDC();  
pDC->Ellipse(100, 100, 200, 200);  
ReleaseDC(pDC);
```

# 浮动菜单(4)



# 浮动菜单(5)

例3-6

## ■ 非隐式浮动菜单(鼠标右键消息)

```
CMenu menu;  
menu.CreatePopupMenu();  
menu.AppendMenu(MF_ENABLED, ID_POPUP_LINE,  
L"Line");  
menu.AppendMenu(MF_ENABLED, ID_POPUP_ELLIP  
SE, L"Ellipse");  
CPoint pt; GetCursorPos(&pt);  
menu.TrackPopupMenu(TPM_LEFTALIGN, pt.x,  
pt.y, this);
```

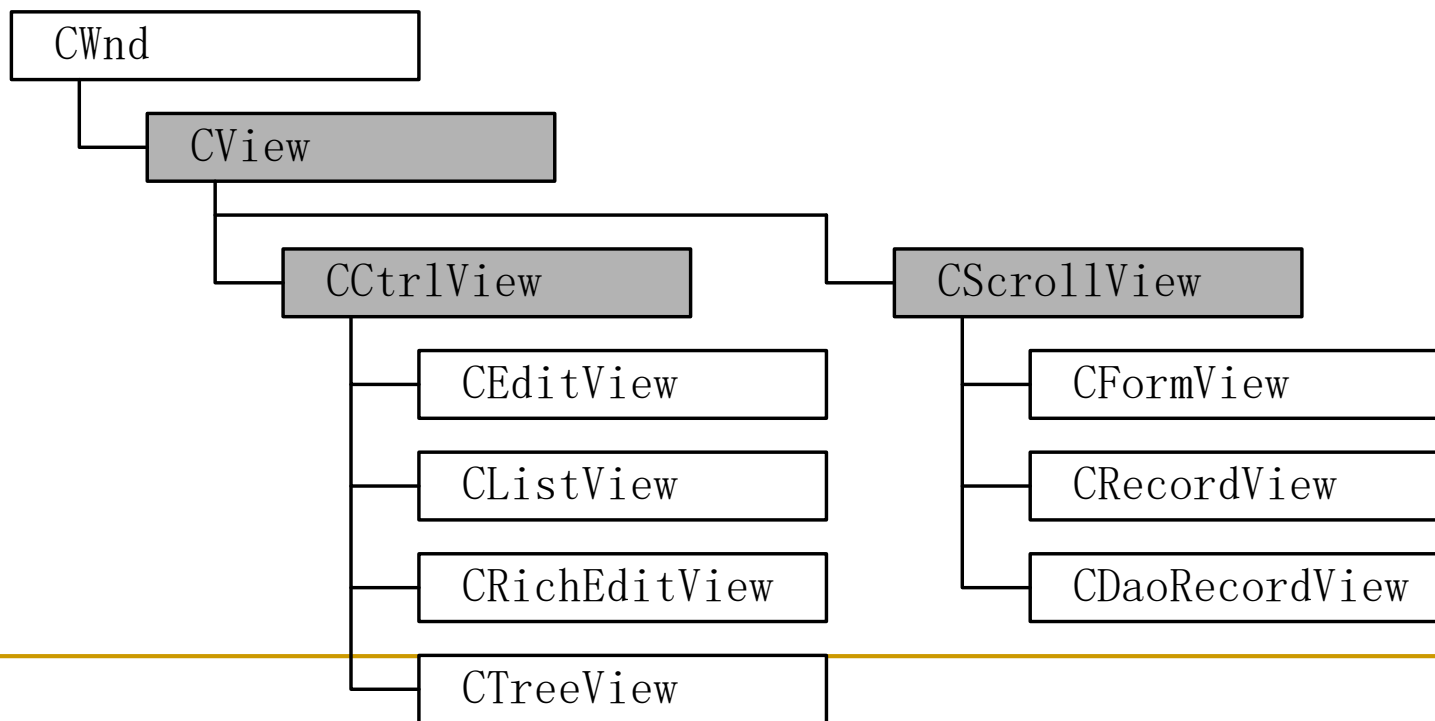
哪个消息专用于弹出浮动菜单？

- ☐ A WM\_KEYDOWN
- ☐ B WM\_LBUTTONDOWN
- ☒ C WM\_CONTEXTMENU
- ☐ D WM\_VSCROLL

提交

# 派生视图类

- MFC提供的视图派生类
  - ✓ **CCtrlView**: 可编辑、列表框和树状控件
  - ✓ **CScrollView**: 视图滚动和缩放显示



# 编辑视图(1)

例3-7

- CTestView的基类设为CEditView
- 在菜单中增加菜单项(ID\_TEST)，或工具栏中增加按钮
- 在CTestView::OnTest()中

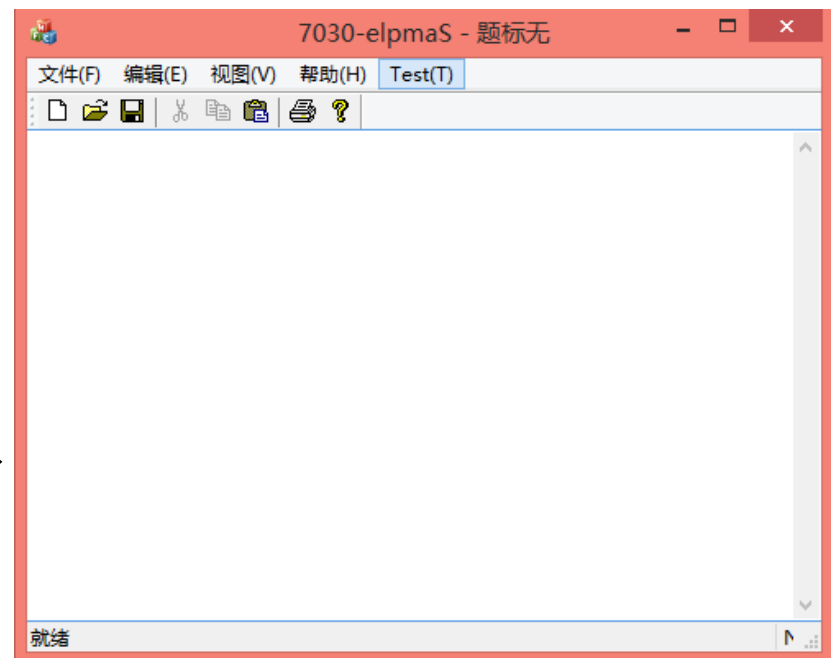
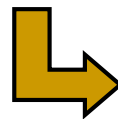
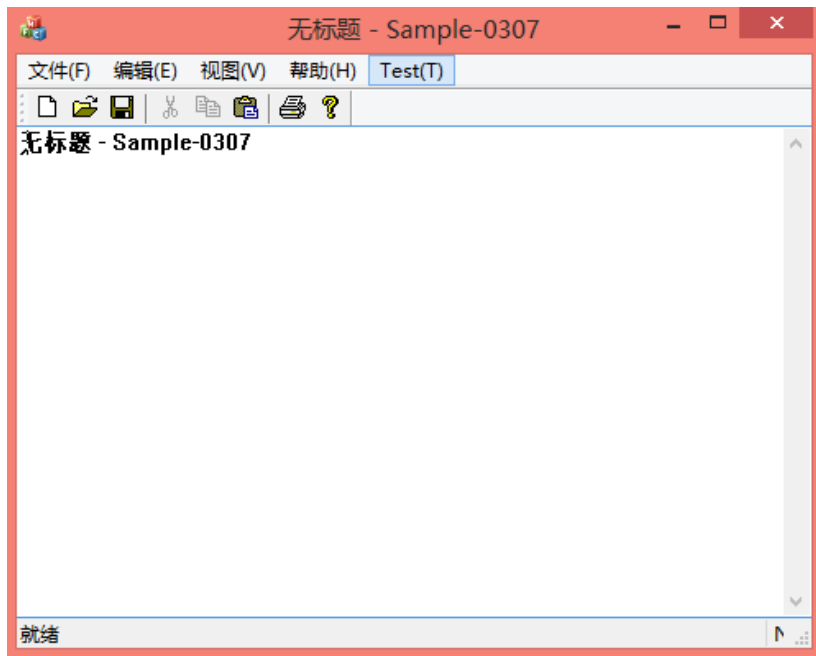
```
CDC* pDC=GetDC();  
CString str;  
GetParent()->GetWindowText(str);  
pDC->TextOutW(0, 0, str);
```

## 编辑视图(2)

- 在CTestView::OnTest()中

```
CString str1, str2;  
GetParent()->GetWindowText(str1);  
str2.Empty();  
for(int i=str1.GetLength()-1; i>=0; i--)  
    str2+=str1.GetAt(i);  
GetParent()->SetWindowText(str2);
```

# 编辑视图(3)





# 滚动显示(1)

- 滚动显示可浏览比视图窗口大的文档，需要从CScrollView类派生
- 在创建滚动视图时，需要重载CView成员函数OnInitialUpdate()，调用SetScrollSizes()计算视图大小
- OnInitialUpdate()是虚函数，用户选择File→New或Open后调用，初始化视图对象，调用OnUpdate()函数

# 滚动显示(2)

例3-8

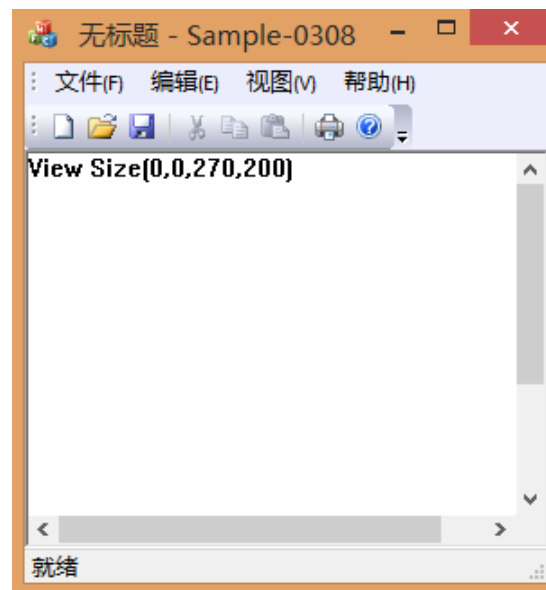
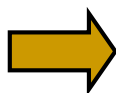
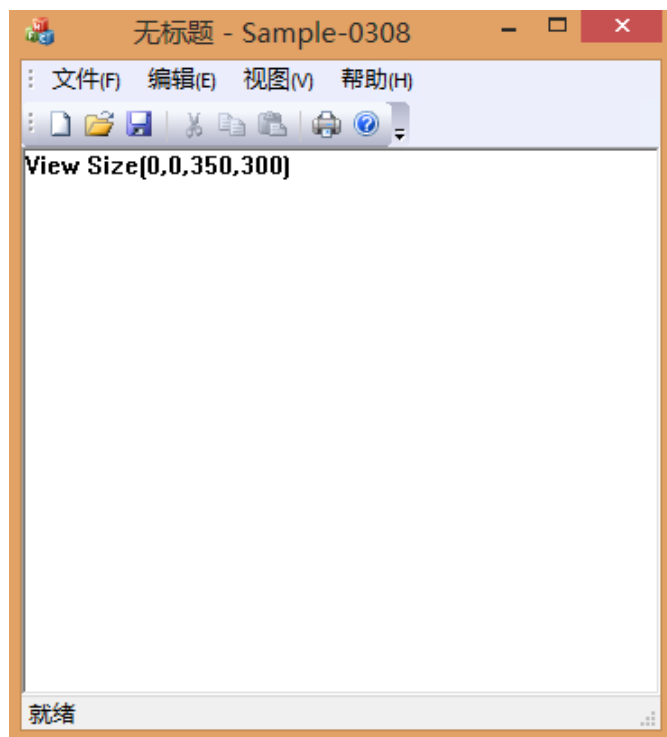
- CTestView的基类为CScrollView
- 在CTestView::OnDraw() 中

```
CRect rect;  
GetClientRect(&rect);  
CString str;  
str.Format(L"View Size (%d, %d, %d, %d)", rect.  
left, rect.top, rect.right, rect.bottom);  
pDC->TextOutW(0, 0, str);
```

# 滚动显示(3)

- 在CTestView::OnInitialUpdate()中

```
sizeTotal.cx=sizeTotal.cy=300;
```



# 缩放显示(1)

- 视图自动适应边框窗口大小
  - ✓ 在缩放显示时，视图没有滚动条
  - ✓ 重载 `OnInitialUpdate()` 函数
  - ✓ 调用 `SetScaleToFitSize()` 函数
- 缩放公式
  - ✓  $\text{显示}x = \text{实际}x * \text{ViewWidth} / \text{sizeTotal.cx}$
  - ✓  $\text{显示}y = \text{实际}y * \text{ViewLength} / \text{sizeTotal.cy}$

# 缩放显示(2)

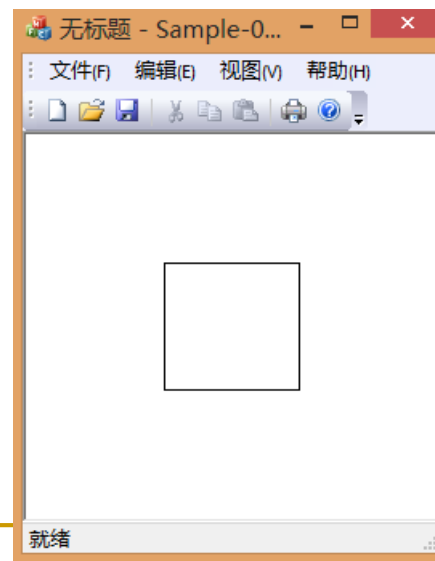
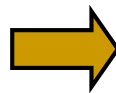
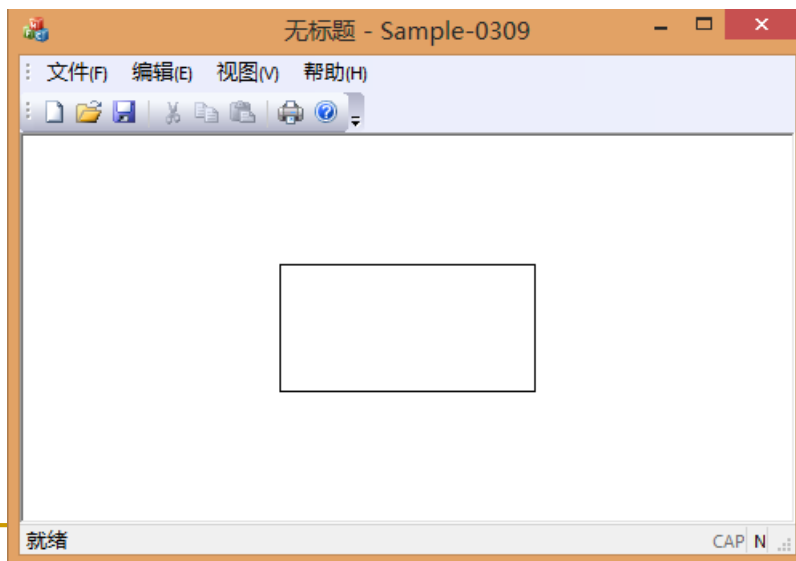
例3-9

- 在CTestView::OnDraw() 中

```
pDC->Rectangle(100, 100, 200, 200);
```

- 在CTestView::OnInitialUpdate() 中

```
SetScaleToFitSize(sizeTotal);
```



# 分割窗口 (1)

- 窗口可分割成多个面板，每个面板是同一文档的独立视图
- 动态分割：面板使用同一视图类，使用的是 `CSplitterWnd` 的 `Create()`
- 静态分割：面板使用不同视图类，使用的是 `CSplitterWnd` 的 `CreateStatic()`

# 分割窗口 (2)

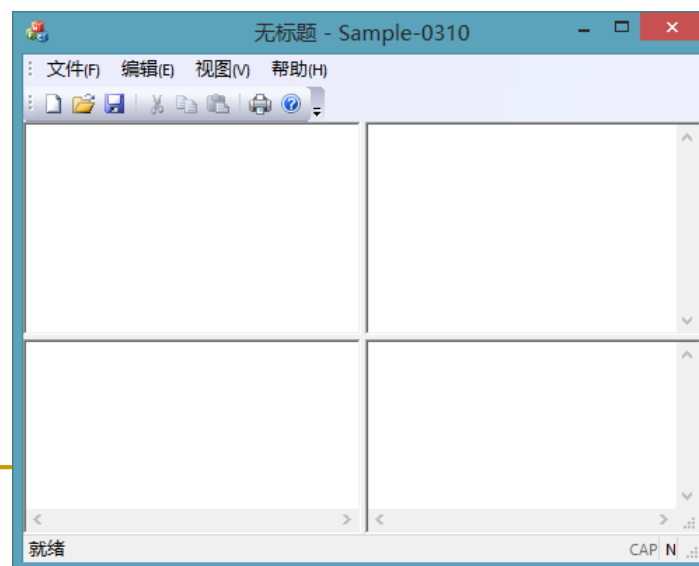
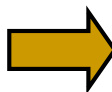
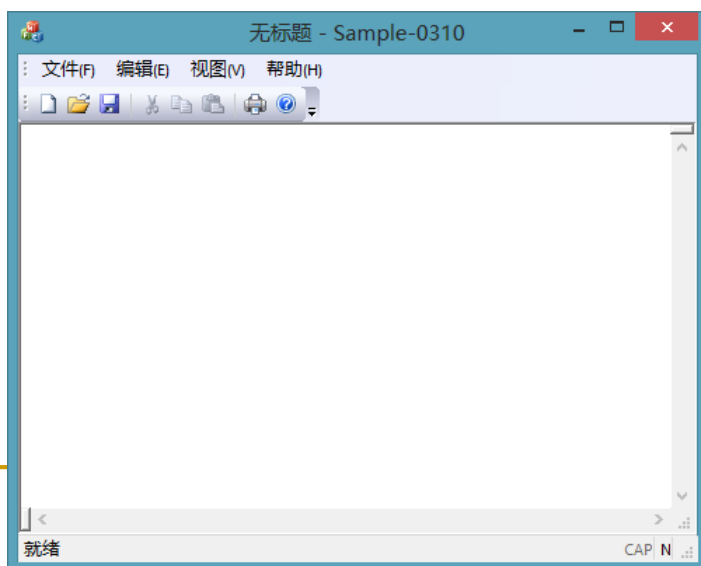
例3-10

- 在CMainFrame类定义中

```
CSplitterWnd m_splitter;
```

- 在CMainFrame::OnCreateClient()

```
return m_splitter.Create(this, 2, 2, CSize  
(20, 20), pContext);
```



# 多文档窗口实例(1)

例3-11

- 在CTestDoc类定义中

```
public:  CString m_str;
```

- 在CTestDoc构造函数中

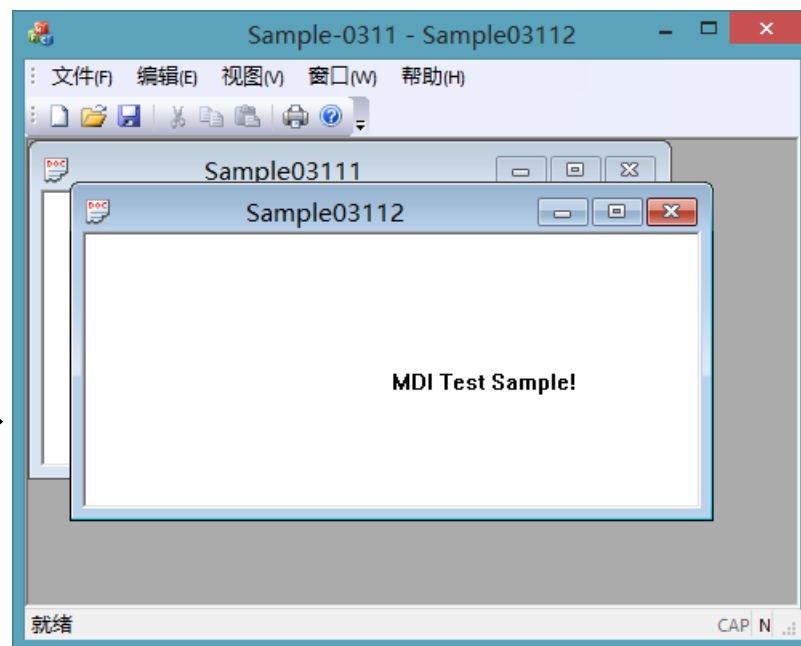
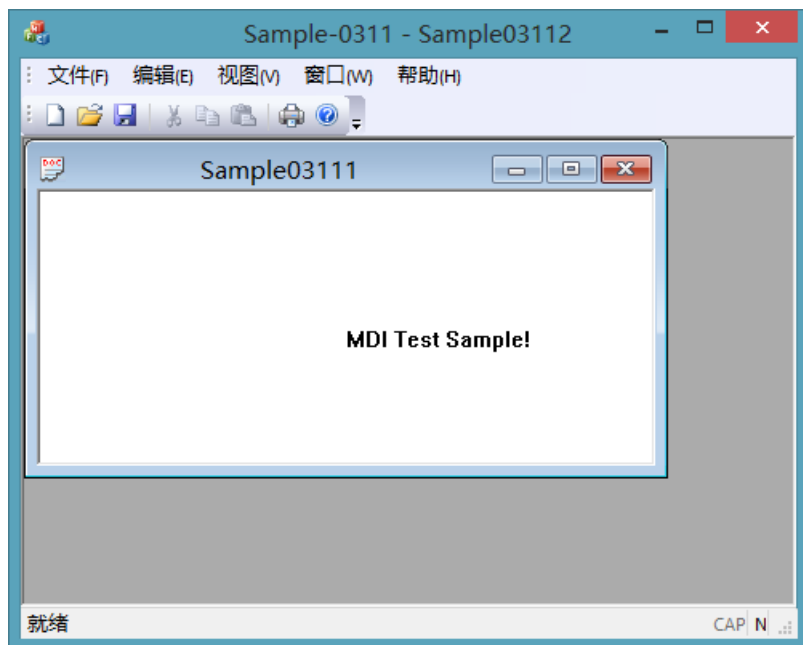
```
m_str=L"MDI Test Sample!";
```

- 在CTestView::OnDraw() 中显示文本

```
CRect rect;  
GetClientRect(&rect);  
pDC->TextOutW(rect.right/2, rect.bottom/2,  
pDoc->m_str);
```



# 多文档窗口实例 (2)



# 第3次作业

- 设计单文档窗口程序，通过菜单打开一个对话框，在编辑框控件中输入文本，单击“确定”按钮，在客户区输出相应文本
- 设计单文档窗口程序，带有浮动菜单(Line、Ellipse、Rectangle)，用户选择某个菜单项时，在客户区输出相应图形，并将图形类型与坐标保存到文档中



谢谢大家

