

Windows

程序员使用指南(二)

——Microsoft 基本类库

[美] Namir Clement Shammas

张新宇 杨 明 译
沈金发 审校

Windows Programmer's Guide to Microsoft
Foundation Class Library



清华大学出版社

Windows

程序员使用指南(二)

——Microsoft 基本类库

[美] Namir Clement Shammas

张新宇 杨 明 译

沈金发 审核

清华大学出版社

(京)新登字 158 号

Windows 程序员使用指南(二)——Microsoft 基本类库

Windows Programmer's Guide to Microsoft Foundation Class Library Namir Clement Shammas

Authorized translation from the English language edition published by Sams, a Division of Prentice Hall Computer Publishing Inc.

Copyright © 1992 by Sams publishing

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission in writing from the Publisher.

Chinese language edition published by Tsinghua University Press.

Copyright © 1994 by Tsinghua University Press

本书英文版由 Prentice Hall 出版社属下的 Sams 计算机图书出版公司于 1992 年出版。版权为 Sams 所有。Sams 将本书的中文版专有出版权授予清华大学出版社。未经出版者书面允许,不得以任何方式复制或抄袭本书的内容。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标志,无标志者不得销售。

图书在版编目(CIP)数据

Windows 程序员使用指南(二): Microsoft 基本类库/(美)萨马斯(Shammas, N. C.)编;张新宇,杨明译. —北京:清华大学出版社,1994

书名原文:Windows Programmer's Guide to Microsoft Foundation Class Library

ISBN 7-302-01582-1

I. W… II. ①萨… ②张… ③杨… III. 窗口软件-软件工具-基本知识 IV. TP316

中国版本图书馆 CIP 数据核字(94)第 07026 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

责任编辑:杨靖

印刷者:清华大学印刷厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:31.5 字数:744 千字

版 次:1994 年 10 月第 1 版 1994 年 10 月第 1 次印刷

书 号:ISBN 7-302-01582-1/TP·665

印 数:0001—5000

定 价:56.00 元

致 谢

感谢参与此书编制、出版的各位人士。感谢出版商 Richard Swadley 和经理 Joseph Wikert 对这套程序员系列图书的远见卓识。非常感谢产品开发专家 Jennifer Flynn、产品编辑 Andy Saff 和复制编辑 Anne Clarke 的耐心工作。同时,感谢技术编辑 Robert McEwin 提出的有价值的意见和更正。感谢 Microsoft 公司迅速地寄来了 beta 版。最后,向 Sams 出版社中与此书出版有关的所有人员表示感谢。

目 录

致谢	VI
引言	1
第1章 Microsoft 基本类库的基础知识	3
1.1 常用的 Windows 数据类型	3
1.2 Microsoft 基本类的层次体系	4
1.2.1 类 CObject	5
1.2.2 类 CWinApp	7
1.2.3 类 CMenu	9
1.2.4 类 CWnd	11
1.2.5 类 CFrameWnd	19
1.2.6 类 CMDIFrameWnd	20
1.2.7 类 CMDIChildWnd	22
1.2.8 类 CDialog	23
1.2.9 类 CModalDialog	24
1.2.10 类 CStatic	25
1.2.11 类 CButton	26
1.2.12 类 CEdit	27
1.2.13 类 CListBox	28
1.2.14 类 CComboBox	30
1.2.15 类 CScrollBar	31
1.2.16 类 CGDIOBJECT	32
1.2.17 类 CPen	33
1.2.18 类 CBrush	33
1.2.19 类 CFont	34
1.2.20 类 CBitmap	34
1.2.21 类 CPalette	35
1.2.22 类 CRgn	36
1.2.23 类 CDC	37
1.2.24 类 CClientDC	42
1.2.25 类 CWindowDC	43
1.2.26 类 CPaintDC	43
1.2.27 类 CMetaFileDC	44
1.3 Windows API 函数	45
1.3.1 Windows 管理器接口函数	45
1.3.2 图形设备接口函数(GDI)	46
1.3.3 系统服务接口函数	47
1.4 调用 Windows API 函数	48
1.5 Windows 消息	50

1.5.1	Windows 消息类型	50
1.5.2	对消息的响应	52
1.5.3	发送消息	53
1.5.4	用户定义的消息	54
1.6	小结	55
第 2 章	建立基本 MFC 应用程序	57
2.1	建立最小的 MFC 应用程序	57
2.2	扩展窗口操作	61
2.3	增加一个菜单	63
2.3.1	创建菜单资源	64
2.3.2	建立样本菜单	66
2.4	响应菜单选择	69
2.5	建立多个实例	76
2.6	关闭窗口	82
2.7	小结	85
第 3 章	基本 Windows 对象	86
3.1	建立只读文字窗口	86
3.2	在窗口中写文字	90
3.3	建立窗口	96
3.4	注册窗口类	101
3.5	改变光标	111
3.6	建立子窗口	117
3.7	小结	125
第 4 章	滚动窗口	126
4.1	滚动文字	126
4.1.1	成员函数 SetScrollRange	126
4.1.2	成员函数 SetScrollPos	127
4.1.3	成员函数 ScrollWindow	127
4.1.4	成员函数 OnVScroll	128
4.1.5	成员函数 OnHScroll	128
4.1.6	成员函数 OnSize	129
4.1.7	成员函数 OnPaint	129
4.2	一个简单的滚动窗口	129
4.3	一个一般化的滚动类	138
4.4	改进的一般滚动窗口	151
4.5	小结	165
第 5 章	Microsoft 基本类控制	166
5.1	静态文字控制	166
5.1.1	CStatic 类	166
5.1.2	静态文字的范例	168
5.2	编辑控制	183
5.2.1	CEdit 类	184

5.2.2	有关剪辑板的编辑函数	184
5.2.3	文字查询函数	185
5.2.4	改变编辑控制内容的函数	186
5.2.5	面向命令的计算器应用程序	187
5.3	按钮控制	201
5.3.1	CButton 类	201
5.3.2	处理按钮消息	202
5.3.3	按钮的操作	202
5.3.4	修改后的计算器应用程序	203
5.4	在窗口控制之间移动	216
	• 按钮操作测试程序	233
5.5	小结	243
第 6 章	分组控制	244
6.1	复选框控制	244
6.1.1	CButton 类与复选框	244
6.1.2	响应复选框消息	245
6.2	无线按钮控制	245
	• CButton 类和无线按钮	245
6.3	组框控制	246
	• CButton 类和成组框	246
6.4	改进的计算器 COCA(版本 3)程序	246
6.4.1	使用应用程序	248
6.4.2	COCA(版本 3)程序代码	248
6.5	小结	269
第 7 章	滚动条、列表框和组合框	270
7.1	滚动条控制	270
7.1.1	CScrollBar 类	270
7.1.2	响应滚动条通知消息	272
7.1.3	倒数计时器程序	272
7.1.4	倒数计时器程序代码	273
7.2	列表框控制	283
7.2.1	CListBox 类	283
7.2.2	响应列表框通知消息	287
7.2.3	一个列表测试应用程序	287
7.2.4	列表测试应用程序代码	288
7.3	同步列表滚动	298
7.3.1	同步列表滚动应用程序	298
7.3.2	同步列表滚动应用程序代码	300
7.4	处理多重选择列表	308
7.4.1	一个多重选择列表测试应用程序	308
7.4.2	多重选择列表测试应用程序代码	310
7.5	文件与目录列表框	322

7.6	组合框控制	330
7.6.1	CComboBox 类	330
7.6.2	响应组合框通知消息	333
7.6.3	组合框用作历史列表框	333
7.6.4	COCA 版本 4 程序	334
7.6.5	COCA 版本 4 程序代码	335
7.7	小结	350
第 8 章	对话框	351
8.1	构造对话框	351
8.2	实现模式对话框	352
8.3	实现非模式对话框	357
8.4	对话框作窗口	370
8.5	传送控制数据的基本知识	380
8.5.1	数据传送缓冲区类型	380
8.5.2	数据传送例示程序	382
8.6	小结	415
第 9 章	标准对话框	417
9.1	软件需求	417
9.2	CFileDialog 类	417
9.2.1	支持类与结构	418
9.2.2	调用文件对话框	421
9.2.3	辅助函数	421
9.2.4	改进的文件统计工具应用程序	422
9.3	CFontDialog 类	425
9.3.1	支持类与结构	426
9.3.2	辅助函数	429
9.3.3	一个简单的字体对话框测试程序	429
9.4	CColorDialog 类	432
9.4.1	支持类与结构	432
9.4.2	辅助函数	434
9.4.3	一个简单的颜色对话框测试程序	435
9.5	CPrintDialog 类	437
9.5.1	支持类与结构	439
9.5.2	辅助函数	441
9.5.3	一个简单的打印对话框测试程序	442
9.6	CFindReplaceDialog 类	446
9.6.1	支持类与结构	447
9.6.2	通知父窗口	449
9.6.3	辅助函数	450
9.6.4	一个简单的查找/替换对话框测试程序	451
9.7	小结	455
第 10 章	MDI 窗口	457

10.1	MDI 应用程序的特征和成分	457
10.2	建立一个 MDI 应用程序的基础	458
10.3	类 CMDIFrameWnd	458
	• 建立 MDI 框架窗口	459
10.4	类 CMDIChildWnd	460
	• 建立 MDI 子窗口	461
10.5	一个简单的文字浏览器	461
10.6	一个改进的文字浏览器	472
10.7	小结	484
第 11 章	控制资源正本	485
11.1	对话框资源	485
11.2	DIALOG 选项语句	486
11.2.1	STYLE 语句	486
11.2.2	CAPTION 语句	486
11.2.3	MENU 语句	486
11.2.4	CLASS 语句	487
11.2.5	FONT 语句	487
11.3	对话框控制资源	488
11.3.1	一般控制资源	488
11.3.2	LTEXT 语句	489
11.3.3	RTEXT 语句	490
11.3.4	CTEXT 语句	490
11.3.5	CHECKBOX 语句	490
11.3.6	PUSHBUTTON 语句	491
11.3.7	DEFPUSHBUTTON 语句	491
11.3.8	LISTBOX 语句	491
11.3.9	GROUPBOX 语句	491
11.3.10	RADIOBUTTON 语句	492
11.3.11	EDITTEXT 语句	492
11.3.12	COMBOBOX 语句	492
11.3.13	SCROLLBAR 语句	493
11.4	小结	493
附录	磁盘安装	494

引 言

毫无疑问,编写 Windows 应用程序比编写 DOS 应用程序要复杂得多。当大家普遍关注 DOS 的时候,大多数程序员并没有遇到早期 Windows 程序员常常碰到的可怕经历(比如,最常见的是,在 Windows 程序中输出 Hello World 这句话往往需要大量语句行)。

Windows 3.0 和 Windows 3.1 的普遍推广使大量程序员接触到一种新的编程技术,即建立 Windows 应用程序。用于 Microsoft 编译器的 Microsoft SDK 的出现并不十分令人满意。值得庆幸的是,面向对象的 C++ 语言的出现,已使程序员有所解脱,而 Windows 界面的类层次体系(如 Microsoft 基本类(MFC)库),又极大地提高了 C++ 语言的能力。

1. 本书的使用者

本书指导读者怎样利用 MFC 类库编写 Windows 应用程序。读者必须具备以下知识:

- 熟悉 C++ 语言
- 熟悉 Windows 的用法
- 初次编写 Windows 应用程序

本书的主要目的,是用实用方便的方法,向用户提供 MFC 库中涉及用户界面的各种成分。本书没有全部介绍 MFC 库中的类,虽然大家希望如此,但是时间不允许,而且也受频繁发表的软件新版本的影响。

2. 怎样使用本书

本书包括十一章,分别代表了用 MFC 库编程的不同方面。

第 1 章介绍了 MFC 类、Windows API 函数和消息管理机制。本章主要讲述了利用 MFC 库建立 Windows 应用程序的背景知识。

第 2 章以简单的 MFC 程序来说明其基本成分。本章还讨论了使用菜单和菜单资源、建立 MFC 应用程序的多个实例、安全地关闭窗口的方法。

第 3 章更为具体地研究了操纵应用程序窗口的方法。本章给出了更多关于注册窗口类、建立非缺省窗口的细节,同时还讨论了向窗口写文字和保存此文字的方法。

第 4 章是第 3 章的收尾,描述了滚动窗口,并解释了激活水平和垂直滚动条的机制。

下面三章提供了各种窗口控制。第 5 章讨论了静态文字、编辑框和按钮控制。

第 6 章描述了组框、复选框和无线按钮控制。这些控制可以使读者能最好地调节 Windows 应用程序的动作。

第 7 章讨论了滚动条、列表框和组合框控制。本章还分析了处理多重选择列表框、建立同步滚动的列表框和合成诸如历史列表之类的组合框的控制。

第 8 章讨论了模式和非模式对话框。本章分析了建立和执行不同的对话框、将对话框

用作应用窗口和传送数据的方法。本章还提供了一些例子,来说明数据在应用程序和对话框之间的传送。

第 9 章描述了标准的 Windows 对话框。这些对话框支持文件选择、颜色、字体选择、打印、打印机设置、文字搜索和替换。

第 10 章描述了多文档界面(MDI)窗口。本章考察了符合 MDI 特点的应用程序的不同成分,提供了两个样本程序。这些例子是典型的、强化的 MDI 窗口实例。此类窗口还包括命令按钮和复选框控制。

第 11 章描述了建立各种控制的资源描述语法。本章提供的信息是建立对话框所必需的。

Microsoft 基本类库的基础知识

Microsoft 基本类库(MFC)提供了开发 Windows 应用程序的强有力的工具,没有它,编写 Windows 应用程序的过程会变得更困难、更易失败,而且代码量相当大。Microsoft 基本类库成功地将面向对象和事件驱动的编程概念联系了起来,并证明了这两种方法可以很好地配合使用。本章介绍了 Microsoft 基本类库和与 Windows 有关的一些基础知识,包括:

- 常用的 Windows 数据类型
- Microsoft 基本类库数据类型的划分约定
- Microsoft 基本类的层次体系
- Windows API 函数
- 如何响应 Windows 消息
- 怎样发送消息
- 用户自定义的消息

本章的信息提供了利用 Microsoft 基本类库建立 Windows 应用程序所需的各方面的知识。尽管 Microsoft 基本类库是开发 Windows 应用程序的良好工具,熟悉 Windows API 函数还是很必要的。

1.1 常用的 Windows 数据类型

Windows 编程包含了许多数据类型,本节重点介绍初次进行 Windows 编程时最可能涉及的数据类型,熟悉了它们有助于更好地理解 MFC 的类声明。Windows 数据类型包括一些简单类型和一组结构。简单数据类型包括许多 typedef 型的数据,它们声明的是含义更明显的别名,这使得函数和变量的声明变得很清楚。表 1.1 列出了最常用的 Windows 数据类型。

除了简单的数据类型(如表 1.1 中所列)之外,Windows 编程中还涉及了一些其他结构。目前,只需了解两个简单但十分重要的结构:POINT 和 RECT。

POINT 结构用于存储一个点的 x 和 y 坐标,声明如下:

```
typedef struct tagPOINT{  
    int x;  
    int y;  
} POINT;
```

RECT 结构定义了矩形的左上角和右下角坐标,其声明如下:

```
typedef struct tagRECT{  
    int left;  
    int top;
```

表 1.1 最常用的 Windows 数据类型

数据类型	意 义	数据类型	意 义
char	有符号的 8 位字符	FARPROC	指向函数的长型指针
int	有符号的 16 位整数	HANDLE	通用柄
long	有符号的 32 位整数	HDC	显示描述表标识柄
short	有符号的 16 位整数	HWND	窗口柄
void	空值	LONG	与 long 类型相同
BOOL	代表逻辑值的 16 位整数	LPSTR	指向字符串的长型指针
BYTE	无符号的 8 位整数	LPVOID	指向 void 类型的长型指针
UINT	无符号的 32 位整数	PWORD	指向 WORD 类型的指针
DWORD	无符号的 32 位整数或 段/偏移量地址	WORD	无符号的 16 位整数
FAR	建立远指针的数据类型属性		

```

int right;
int bottom;
} RECT;

```

其中, left 和 right 成员是矩形左上角和右下角的 x 坐标。top 和 bottom 域分别为矩形左上角和右下角的 y 坐标。

1.2 Microsoft 基本类的层次体系

Microsoft 基本类的层次体系包含几个子层次, 它们分别管理 Windows 应用程序的不同方面。这些子层次为:

- 窗口类
- GDI 类
- 杂项窗口类(即与 Windows 应用程序的可视界面无关, 可用于 DOS 应用程序的那些类。)
- 对象链接和嵌入(OLE)类
- 文件类
- 对象 I/O 类
- 异常情况类
- 收集类
- 其它支持类

图 1.1 为 Microsoft 基本类层次体系中的窗口类和杂项窗口类。此图还包括了类 CArchive, 它是 Microsoft 基本类库支持的两个对象 I/O 类之一。本书重点描述图 1.1 中的类, 并用其他 MFC 类, 如 CString 类, 来支持例子程序。

下面将描述图 1.1 中的各个类。描述包括类声明和各个类主要功能的简要描述。

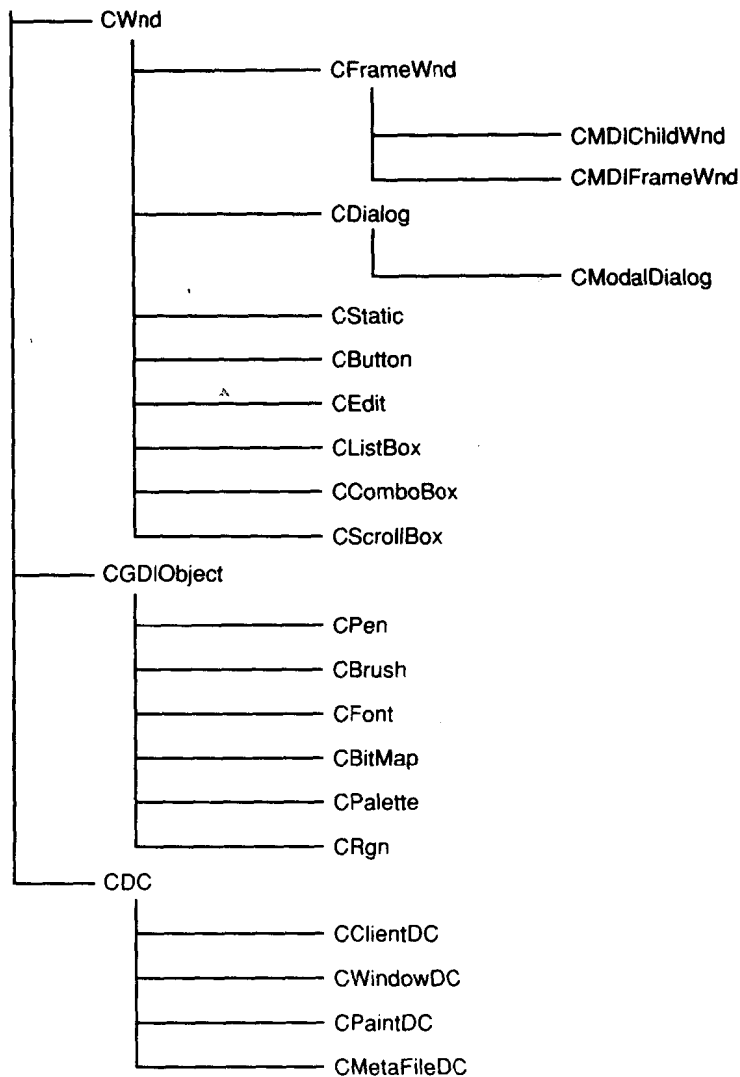


图 1.1 Microsoft 基本类层次体系中的窗口类和杂项窗口类

1.2.1 类 CObject

类 CObject 是 Microsoft 基本类库的抽象基类。其类声明为：

```

class CObject
{
public:

// Object model (types, destruction, allocation)
virtual CRuntimeClass* GetRuntimeClass() const;
virtual ~CObject(); // virtual destructors are necessary

```

```

        // Diagnostic allocations
        void* operator new(size_t, void* p);
        void* operator new(size_t nSize);
        void operator delete(void* p);

#ifdef _DEBUG
        // for file name/line number tracking using DEBUG_NEW
        void* operator new(size_t nSize, const char FAR* lpszFileName,
                           int nLine);
#endif

        // Disable the copy constructor and assignment by default
        // so you get compiler errors instead of unexpected
        // behavior if you pass objects by value or assign objects.
protected:
    CObject();
private:
    CObject(const CObject& objectSrc);
    void operator=(const CObject& objectSrc);

// Attributes
public:
    BOOL IsSerializable() const;
// Overridables
    virtual void Serialize(CArchive& ar);
    // Diagnostic Support
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;

// Implementation
public:
    // dynamic type checking/construction support
    BOOL IsKindOf(const CRuntimeClass* pClass) const;
    static void PASCAL Construct(void* pMemory);

    static CRuntimeClass classCObject;
};

```

类 CObject 是 Microsoft 基本类库中各类及其各种可能的派生类的根类。CObject 类并不是一个抽象的空类。相反,它向其所有派生类提供了大量操作。这些操作包括:

- 创建和删除对象
- 串行化的支持
- 诊断
- 与收集类的兼容性

类 CObject 声明了 new 操作符的三个版本和一个 delete 操作符。

Microsoft 基本类库还声明了一些支持 CObject 类及其所有派生类的宏。这些宏为

- 宏 RUNTIME_CLASS, 返回与所命名的类匹配的 CRunTimeClass 结构。

- 宏 DECLARE_DYNAMIC, 使程序能够存取运行时的类数据。该宏可用于各个类声明中。
- 宏 IMPLEMENT_DYNAMIC, 使程序能够存取运行时的类数据。该宏用于类的实现部分。
- 宏 DECLARE_SERIAL, 使得在 Windows 应用程序中可以串行化并存取运行时类数据。该宏可用于各个类声明中。
- 宏 IMPLEMENT_SERIAL, 使得在 Windows 应用程序中可以串行化并存取类数据。该宏用于类的实现部分。

1.2.2 类 CWinApp

类 CWinApp 创建 Windows 应用程序。用户编写的应用程序必须恰好含有类 CWinApp 的一个派生类。应用程序中类 CWinApp 派生类的实例拥有其他所有对象。类 CWinApp 的声明如下所示：

```
class CWinApp : public CObject
{
    DECLARE_DYNAMIC(CWinApp)
public:

    // Constructor
    CWinApp(const char* pszAppName = NULL);
    void SetCurrentHandles();

    // Attributes
    // Startup args (do not change)
    const char* m_pszAppName;          // from constructor
    HANDLE m_hInstance;
    HANDLE m_hPrevInstance;
    LPSTR m_lpCmdLine;
    int m_nCmdShow;

    // Running args
    CWnd* m_pMainWnd;                  // main window (optional)

    // Operations
    // Cursors
    HCURSOR LoadCursor(LPSTR lpCursorName);
    HCURSOR LoadCursor(UINT nIDCursor);
    // for IDC_ values
    HCURSOR LoadStandardCursor(LPSTR lpCursorName);
    // for IDC_ values
    HCURSOR LoadOEMCursor(UINT nIDCursor); // for OCR_ values

    // Icons
    HICON LoadIcon(LPSTR lpIconName);
    HICON LoadIcon(UINT nIDIcon);
    HICON LoadStandardIcon(LPSTR lpIconName); // for IDI_ values
    HICON LoadOEMIcon(UINT nIDIcon); // for OIC_ values
```

```

        BOOL PumpMessage();

// Overridables
    // hooks for your initialization code
    virtual BOOL InitApplication();
    virtual BOOL InitInstance();

    virtual int Run();

    // called by standard 'Run' implementation
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    virtual BOOL OnIdle(LONG lCount); // return TRUE if
                                        // more idle processing
    virtual int ExitInstance(); // return app exit code

// Implementation
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:
    MSG m_msgCur;

#ifdef _DEBUG
    // Diagnostic trap for when going back to
    // message pump is not permitted.
protected:
    int m_nDisablePumpCount;
public:
    void EnablePump(BOOL bEnable);
#endif
};

```

类 CWinApp 含有大量公用数据成员,这些数据成员存储 Windows 传递给预定义的 WinMain()函数的数据(WinMain()函数的作用类似于 C 程序中的 main()函数)。下面列出这些数据成员的功能:

- m_hInstance 成员,存储应用程序当前实例的标识柄。
- m_hPrevInstance 成员,存储上一个实例的标识柄。如果当前实例是目前唯一运行着的实例,m_hPrevInstance 具有值 NULL。
- m_lpCmdLine 成员,存储调用 Windows 应用程序时,用户可能键入的命令行参数。
- m_nCmdShow 成员,存储 Windows 应用程序的状态:正常、最小化、最大化或隐藏。

其他公用数据成员为:

- m_pMainWnd 成员,指向主程序窗口的指针。
- 8 •