



通用程序设计案例丛书

Visual C++ 程序开发 案例解析

杨富国 主编

王浩 唐巧琪 王健 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

通用程序设计案例丛书

Visual C ++

程序开发案例解析

杨富国 主编

王 浩 唐巧琪 王 健 编著

清华大学出版社
北京交通大学出版社

• 北京 •

内 容 简 介

本书通过系统而完整的程序设计案例详细介绍了使用 Visual C++ 6.0 开发各种应用系统的思路、方法、技巧和经验,案例范围涉及信息管理系统、网络应用系统、图形图像系统、多媒体系统、分布式组件对象模型及 ActiveX 控件等常用领域,并提供了全部案例的完整代码以供引用和借鉴。

本书具有一定的理论性和较强的实用性,可以帮助读者进一步掌握软件工程规范,增加系统开发经验。本书适用于计算机及其相关专业的本科高年级学生和研究生,以及具有一定程序设计基础的程序设计人员。

本书所有实例均可在北京交通大学出版社的网站 (<http://press.bjtu.edu.cn>) 下载。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

Visual C++ 程序开发案例解析/杨富国主编. —北京:清华大学出版社;北京交通大学出版社, 2006.3

(通用程序设计案例丛书)

ISBN 7-81082-690-5

I. V… II. 杨… III. C++ 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 007172 号

责任编辑:刘洵

出 版 者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印 刷 者:北京瑞达方舟印务有限公司

发 行 者:新华书店总店北京发行所

开 本:185×230 印张:22.75 字数:565 千字

版 次:2006 年 3 月第 1 版 2006 年 3 月第 1 次印刷

书 号:ISBN 7-81082-690-5/TP·263

印 数:1~5 000 册 定价:32.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@center.bjtu.edu.cn。

编委会成员名单

主 编：杨富国

编 委：（按姓氏笔画排序）

王 浩 王 倩 王 健

吕江英 邢 霞 陈立俊

杨茂龙 唐巧琪

前言

本书首先概略地介绍了 Visual C++ 6.0 的编程基础和软件系统建模与实现的基本理论,重点讲述应用程序案例的分析、设计与实现,所设计的案例涉及数据库信息管理、网络数据通信、图形图像处理、多媒体应用、ATL 应用等诸多方面,难度和规模与计算机专业本科毕业设计的软件系统相当,具有一定的系统性和较宽的覆盖面。

在案例的分析与实现中,运用面向对象原则和软件工程的规范流程,力求系统地展现从系统分析到程序实现的开发全过程,尽量避开了编程过程中的旁枝末节问题,使入门级的程序开发人员迅速熟悉和掌握程序开发思路和方法,找出解决实际问题的快捷途径。

本书具有以下主要特色

(1) 较强的实用性:本书所列举的案例应用范围较广,涵盖了 Visual C++ 6.0 系统应用的大部分领域,对于正在进行课题设计或项目开发的读者来说,很多案例稍加修改就可顺利应用。读者可以从北京交通大学出版社网站上下载随书所附案例的完整代码和可执行程序。请访问北京交通大学出版社网站: <http://press.bjtu.edu.cn>, 在“FTP 下载专区”单击“下载图书辅助资料”,即可找到本书的实例。

(2) 较深的理论性:本书不是仅介绍如何应用 Visual C++ 6.0 开发应用系统,而是把软件系统工程系统分析贯穿全书,可以帮助读者建立起系统分析的观念和规范性的设计思维,帮助读者从入门走向深入。

(3) 广泛的读者群:本书适用于计算机及其相关专业的本科高年级学生和研究生、具有一定程序设计基础且缺少系统开发经验的程序设计人员、学习软件工程理论的程序设计人员,也可供 Visual C++ 6.0 开发人员和计算机编程爱好者参考。

编 者

2006 年 3 月

目 录

第 1 章	VC++ 6.0 编程基础	1
1.1	VC++ 6.0 编程概述	1
1.1.1	Windows 程序模型	1
1.1.2	MFC 应用程序框架	2
1.1.3	MFC 消息映射	4
1.1.4	MFC 类库概念和组成	5
1.1.5	活动模板库 ATL	7
1.1.6	MFC 的局限性	8
1.2	VC++ 6.0 程序开发概述	9
1.2.1	VC++ 6.0 集成开发环境	9
1.2.2	应用程序开发向导 AppWizard	11
1.2.3	类向导 ClassWizard	11
1.2.4	编译、调试与诊断	13
1.2.5	MSDN	16
1.3	VC++ 6.0 应用程序类型	16
1.3.1	Win32 项目	16
1.3.2	MFC 项目	18
1.3.3	ATL COM 项目	21
1.4	VC++ 6.0 应用程序开发与部署示例	22
1.4.1	创建 MFC 应用程序框架	22
1.4.2	为程序编写源代码	25
1.4.3	打包并创建安装程序	28
1.5	小结	33
第 2 章	软件开发过程	35
2.1	概述	35
2.1.1	瀑布模型	35
2.1.2	原型模型	36
2.1.3	螺旋模型	36
2.2	软件定义及计划	37
2.2.1	问题定义	38
2.2.2	可行性分析	38

2.2.3	制定开发计划	38
2.3	需求分析	39
2.3.1	需求分析任务	39
2.3.2	需求分析过程	39
2.3.3	常用需求分析方法	40
2.3.4	需求文档	40
2.4	软件设计	41
2.4.1	设计目标	41
2.4.2	设计任务	41
2.4.3	软件设计过程	42
2.4.4	软件设计方法	44
2.5	编码	44
2.5.1	概述	44
2.5.2	程序设计风格	44
2.5.3	编码标准	45
2.6	软件测试	46
2.6.1	概述	46
2.6.2	测试目的和原则	46
2.6.3	测试方法	47
2.6.4	软件测试过程	48
2.7	软件维护	49
2.7.1	概述	49
2.7.2	影响维护工作量的因素	49
2.7.3	软件的可维护性	50
2.8	UML 简介	51
2.8.1	UML 概述	51
2.8.2	视图	53
2.8.3	图	54
2.8.4	UML 的应用领域	57
2.9	小结	58

第3章 ◀ 数据库信息管理案例

3.1	管理信息系统基础	59
3.1.1	管理信息系统的概述	59
3.1.2	管理信息系统的特点	59
3.1.3	管理信息系统的开发	60
3.2	SQL 语言	60
3.2.1	SQL 语法简介	61
3.2.2	SQL 命令	64

3.3	ADO 概述	66
3.3.1	ADO 对象模型	67
3.3.2	使用 ADO 编程	69
3.4	用 SQL Server 2000 创建数据库	72
3.4.1	相关概念简介	72
3.4.2	创建数据库	73
3.5	MIS 案例——小商店管理助手 ShopAssistant	78
3.5.1	需求分析	78
3.5.2	系统分析	79
3.5.3	数据库设计	81
3.5.4	界面设计	83
3.5.5	系统实现	83
3.5.6	系统运行与测试	109
3.6	小结	112
第 4 章	网络数据通信案例	113
4.1	网络信息系统编程基础	113
4.1.1	网络信息系统简介	113
4.1.2	网络信息系统的特点	113
4.2	网络编程基础	114
4.2.1	TCP/IP 协议	114
4.2.2	WinSock 基本定义	115
4.2.3	WinSock 基本函数	117
4.2.4	WinSock 扩展	119
4.2.5	WinSock 编程	120
4.2.6	MFC Socket	122
4.3	XML DOM	124
4.3.1	XML DOM 简介	124
4.3.2	DOM 使用	125
4.4	网络数据通信编程实例	126
4.4.1	需求分析	126
4.4.2	系统建模	127
4.4.3	界面设计	129
4.4.4	系统实现	130
4.4.5	系统运行与测试	155
4.5	小结	156
第 5 章	计算机多媒体应用案例	157
5.1	计算机多媒体编程基础	157
5.1.1	计算机多媒体的主要类型	157

5.1.2	多媒体程序特点	158
5.1.3	Windows 多媒体体系结构	159
5.2	媒体控制接口编程	161
5.2.1	MCI 设备与时间格式	161
5.2.2	MCI 函数与命令	162
5.2.3	MCI 函数与命令使用举例	163
5.3	混音器编程	166
5.4	多媒体应用案例——迷你播放器	170
5.4.1	需求分析	170
5.4.2	系统分析	171
5.4.3	界面设计	172
5.4.4	系统实现	173
5.4.5	系统运行与测试	189
5.5	小结	190
第 6 章	图形处理案例	192
6.1	图形处理系统基础	192
6.1.1	图形处理系统概述	192
6.1.2	图形开发技术	193
6.1.3	图形处理系统的开发	194
6.2	GDI 图形编程基础	195
6.2.1	GDI 与 DC	195
6.2.2	CDC 类的使用	197
6.2.3	图形元素的绘制	199
6.3	图形处理程序实例	203
6.3.1	需求分析	203
6.3.2	系统分析	203
6.3.3	界面设计	204
6.3.4	系统实现	204
6.3.5	系统运行与测试	227
6.4	小结	227
第 7 章	图像处理案例	229
7.1	数字图像处理概述	229
7.1.1	数字图像的基本概念	229
7.1.2	数字图像处理概述	230
7.2	VC++ 数字图像编程基础	231
7.2.1	位图和调色板	232
7.2.2	BMP 图像文件及其基本操作	233
7.3	数字图像处理实例	246

7.3.1	需求分析	246
7.3.2	系统分析	246
7.3.3	界面设计	246
7.3.4	系统实现	247
7.3.5	系统运行与测试	284
7.4	小结	285
第 8 章	DCOM 应用程序实例	287
8.1	组件对象模型基础	287
8.1.1	COM 与 DCOM	287
8.1.2	分布式应用	288
8.1.3	COM 应用的开发	288
8.2	COM 基本原理	289
8.2.1	COM 规则	290
8.2.2	COM 组件与接口详解	293
8.2.3	使用 COM	298
8.2.4	使用 DCOM	299
8.3	COM 实例开发——ATM 机	302
8.3.1	需求分析	302
8.3.2	系统分析	302
8.3.3	数据库设计	304
8.3.4	界面设计	304
8.3.5	系统实现	304
8.3.6	系统运行与测试	324
8.4	小结	325
第 9 章	ActiveX 控件编程	326
9.1	ActiveX 技术	326
9.1.1	ActiveX 技术简介	326
9.1.2	ActiveX 控件原理	327
9.1.3	ActiveX 控件的创建方法	328
9.1.4	ActiveX 控件的使用	333
9.2	ATL 窗口	336
9.2.1	ATL 窗口类	336
9.2.2	ATL 窗口应用程序的消息映射	336
9.2.3	创建 ATL 窗口	340
9.3	ActiveX 控件应用案例——时钟控件	342
9.3.1	需求分析	342
9.3.2	系统分析	343
9.3.3	界面设计	343

9.3.4 系统实现	344
9.3.5 系统运行与测试	351
9.4 小结	352
参考文献	353

第1章

VC++ 6.0 编程基础

Visual C++ 6.0 是一个全面的应用程序开发环境，它为程序开发人员提供了使用面向对象的 C++ 来开发 Windows 应用程序的强大手段。学习和掌握 Visual C++ 6.0 有一定的难度，程序开发人员需理解 C++ 语言，进而掌握 Visual C++ 6.0 的基本编程方法——Microsoft 基本类库（Microsoft Foundation Class, MFC）及活动模板库（Active Template Library, ATL）的基本原理及运用。因此有必要在本书的开头介绍一下 Visual C++ 6.0 的一些基础知识，对于不具备深厚编程经验的程序开发人员来说，浏览本章有助于在进入具体编程之前先通晓全局，快速建立对 Visual C++ 6.0 程序开发的整体认识。

1.1 VC++ 6.0 编程概述

1.1.1 Windows 程序模型

Windows 程序最显著的特点是提供了一个可供用户操作的图形用户界面（Graphical User Interfaces, GUI）。在 GUI 中有许多可供操作的可视对象，用户对这些对象所做的操作会产生一些特定事件，这些事件向应用程序发送消息，应用程序调用相应的消息处理函数，从而实现对用户操作的处理，如图 1-1 所示。因此，Windows 程序本质上是面向对象的、以事件驱动的程序模式，程序的运行过程就是用户操作不断产生事件、程序对这

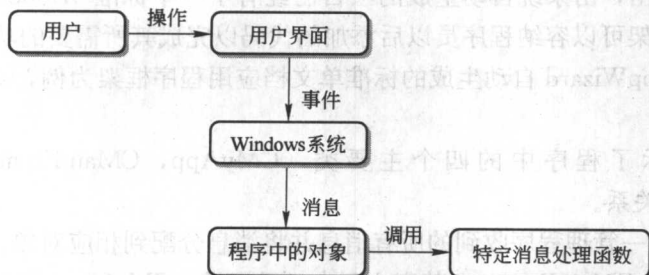


图 1-1 Windows 程序模型

些操作进行处理的过程。因此, Windows 程序没有固定流程, 而是由处理各种事件的特定子流程所构成的集合。

Visual C++ 将一个 Windows 程序的各种源代码文件、资源文件及开发过程中产生的各种中间文件组织为一个项目 (Project)。在 Visual C++ 中, 应用程序是以 Project 的形式存在的。为了进行协调和管理, Visual C++ 用 Workspace 来组织多个项目, Workspace 文件扩展名为 dsw, 一个 Workspace 中可以包含多个 Project。与 Workspace 文件相配合的是 Workspace 在本地计算机的配置信息文件, 该文件以 opt 为扩展名。

Project 文件 (.dsp) 存放与项目的有关信息, .clw 文件用来存放项目中的类和资源的信息。对应每个 Project 有一个 readme.txt 文件, 列出该应用程序中用到的所有文件的信息。

项目中的各种类由头文件 (.h) 和实现文件 (.cpp) 组成, 头文件包含类的定义, 实现文件包含类成员函数的实现代码。项目中包含的位图、菜单之类的资源由资源文件 (.RC) 描述, 以 RC2 为扩展名的文件也是资源文件, 但其中的资源不能在 VC 的 IDE 中直接编辑, 而只能由手工编辑。项目的 Resource 文件夹中还有一些以 ico, bmp 等为扩展名的文件是具体的资源, 使用 RC 资源文件对这些资源进行统一的管理。

1.1.2 MFC 应用程序框架

传统 Windows 应用程序编程方法需要程序员对 Windows 编程原理有深刻认识, 手工编写代码量很大, 调试非常困难, 这就迫切需要快捷的、可重用的编程方法, MFC 就是其中出现最早、应用最广泛的一种。

MFC 的本质就是一个包含微软公司定义好的大量对象的类库, 包括了编写 Windows 应用程序的常用对象, 如用户界面设计、数据库访问、文件操作、多媒体使用、网络编程等, 在进行程序设计的时候, 程序员只需调用 MFC 中已有对象, 使用其方法。如果 MFC 类不完全满足需求, 还可以在 MFC 类基础上加上所需的属性和方法派生出所需的对象。

MFC 编程方法使得程序员编程时无需关心对象方法的实现细节, 同时类库中的各种对象的强大功能足以完成程序中的绝大部分所需功能, 这使得程序员编写的代码量大为减少。

若使用 AppWizard 建立一个项目, 该项目可以直接编译运行, 不需要程序员写一行代码。由此可以看出, 由系统自动生成的项目已经有了一个标准 Windows 应用程序所需的基本框架, 该框架可以容纳程序员以后添加的代码以完成其所需要的功能。下面以一个由 Visual C++ AppWizard 自动生成的标准单文档应用程序框架为例, 对 MFC 应用程序框架做详细介绍。

图 1-2 显示了程序中的四个主要类 (CMyApp, CMainFrame, CMyView 和 CMyDoc) 之间的关系。

CMyApp 类统一管理程序收到的所有消息并将消息分配到相应对象。CMainFrame 是 CMyView 的父类, CMainFrame 在其客户区中显示视类。CMyView 类显示数据, 而数据来源于 CMyDoc 类。在 MFC 程序中, 程序的数据是放在文档中的, 而显示数据则是利用窗口方式, 文档与窗口分离, 因此一个文档可以同时具有多个窗口, 或者一个程序中

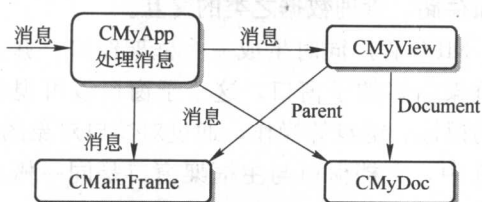


图 1-2 标准 Windows 程序中四个主要类之间的关系

处理多个文档。

1. CMyApp 类

CMyApp 类是应用程序运行的基础，其父类是 MFC 中的 CWinApp。CMyApp 类除了构造函数，另一个重要的成员函数就是 InitInstance。一个 CMyApp 类型的对象实例生成时，系统会调用 InitInstance 函数完成生成该新实例的一些初始化工作。

InitInstance 函数定义了一个文档模板对象指针 pDocTemplate，通过 new 操作符，系统动态生成这个文档模板对象，然后使用 AddDocTemplate 函数将其加入到应用程序文档模板链表 pDocTemplate 当中。这个文档模板的作用就是把程序用到的框架窗口 CMainFrame、文档 CMyDoc、窗口 CMyView 与应用对象 CMyApp 联系起来。

CMyApp 类提供了用户与 Windows 应用程序之间的交互界面。这个类的对象被生成后自动地与 Windows 系统建立联系，接收 Windows 传送的消息，并交给程序中相应的对象去处理，使得开发 C++ 的 Windows 程序变得简单方便。

2. CMainFrame 类

CMainFrame 类也是一个框架窗口，是由 MFC 中的 CFrameWnd 派生而来。CMainFrame 从 CFrameWnd 类那里继承了处理一般窗口事件的方法，比如改变窗口的大小，窗口最小化等的成员函数，因此不需程序员再关心此类消息的处理。在 MFC 程序中并不需要经常对 CMainFrame 类进行操作，而更多的是对窗口类进行操作，以编辑和修改程序中的数据。

在 MFC 程序中，当程序的一个实例被运行的时候，系统根据前面在 CMyApp 类中介绍的文档模板对象 pDocTemplate 自动生成类 CMainFrame，CMyView，CMyDoc 的对象，而不需程序员创建这些类的对象。

3. CMyView 与 CMyDoc 类

CMyView 类和 CMyDoc 类密切相关，图 1-3 可以说明窗口类（CMyView）与文档类（CMyDoc）的关系。

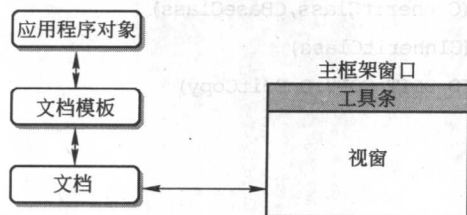


图 1-3 窗口与文档的对应关系

文档由文档模板对象生成，并由应用程序对象管理，用户通过与文档相联系的窗口对

象来实现与文档之间诸如存储、管理数据之类的交互。

生成一个新文档时, MFC 程序同时生成一个框架窗口, 并且在框架窗口的客户区中生成一个窗口对象作为框架窗口的子窗口, 这个子窗口以可视化的方式表现文档中的内容。窗口负责处理用户的鼠标、键盘等操作, 通过对窗口对象的处理达到处理文档对象的目的。在单文档界面 SDI 中, 文档窗口与主框架窗口是同一概念, 这时的窗口对象显示在文档窗口的客户区当中。

1.1.3 MFC 消息映射

Windows 应用程序基于消息驱动, MFC 通过消息映射机制处理消息。最直接的封装方式是采用虚成员函数, 如果要响应某个消息就重载相应的虚函数。MFC 没有采取虚函数, 而是采用宏 (Macro) 来建立各种消息与其处理函数的对应表, 这是为了避免为多达上百个消息建立一个很大的虚函数分发表 (vtable), 而且不用对 C++ 语言做扩展, 以确保其兼容性。

MFC 消息映射机制确实比较复杂, 不易理解。简单说来, 消息映射机制实际上是各种消息及其处理函数的对应表, 以及分析处理这种关系的程序代码。

1. Windows 消息种类

Windows 中消息主要有三种。

① 标准 Windows 消息: 以 WM_ 为前缀, 如 WM_MOVE、WM_QUIT 等 (WM_COMMAND 例外)。

② 命令消息: 以 WM_COMMAND 为消息名, 消息中有命令标志 ID 作为区别, 由菜单、工具栏等命令接口对象产生。

③ 控件通知消息: 也以 WM_COMMAND 为消息名, 由编辑框、列表框、子窗口发送给父窗口, 包含控件通知码以区分控件。

标准 Windows 消息及控件通知消息主要由窗口类 (CWnd 类) 的派生类处理。而命令消息可以由窗口类 CWnd 处理、文档类 CMyDoc 及 CMyApp 类处理。

2. MFC 中消息映射机制

在 MFC 的框架结构下, “消息映射” 通过宏形成一张消息映射表格来实现。一旦消息发生, Framework 就可以根据消息映射表格来进行消息映射和命令传递。

在需要消息处理的类的头文件里, 都会含有宏 DECLARE_MESSAGE_MAP() 以声明该类拥有消息映射表格, 然后在类源文件的消息映射表中添加消息映射:

```
BEGIN_MESSAGE_MAP(CInheritClass,CBaseClass)
    ///{{AFX_MSG_MAP(CInheritClass)
        ON_COMMAND(ID_EDIT_COPY,OnEditCopy)
        .....
    ///}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

这里把消息映射和消息处理函数联系在一起。其中 BEGIN_MESSAGE_MAP 宏有两个参数, 分别是拥有消息表格的类及其父类。ON_COMMAND 宏指定命令消息的处理函

CObject 是 MFC 大部分类的基类, CObject 类描述了几乎所有的 MFC 中其他类的一些公共特性。CObject 类为其派生类提供了三个重要的特性支持: 序列化 (Serialization) 支持、运行时 (Run-time) 类信息支持和诊断 (Diagnostic) 调试支持等。其中, 序列化支持是以流的方式将某个类对象中的持久性数据输出或输入到外部存储介质如磁盘文件等的过程; 运行时类信息支持则可以重新获取一个对象的类名及其他一些有关对象在运行时的信息, 是 C++ 中除运行时类型信息 (Run-time Type Information, RTTI) 机制外的另一个重要工具; 诊断和调试支持作为 CObject 类的一个组成部分, 可以在实现 CObject 派生类时执行有效性检查并可向调试窗口输出状态信息。

MFC 类按照继承关系可以分成两大类: 大多数的 MFC 类是从 CObject 继承下来; 另外一些类则不是从 CObject 类继承下来, 这些类包括字符串类 CString、日期时间类 CTime、矩形类 CRect、点 CPoint 等, 它们提供程序辅助功能。

2. 宏和全局函数

若某个函数或变量不是某个类的一个成员, 那么它是一个全局函数或变量。

Microsoft 基本宏和全局函数提供以下功能:

- 数据类型;
- 消息映射;
- 运行时对象类型服务;
- 应用消息和管理;
- 诊断服务;
- 对象连接和嵌入 (OLE) 服务;
- 异常处理;
- 标准命令和 Windows IDs;
- CString 格式化及信息框显示。

宏全部大写, 不带任何特别前缀。全局变量以 `afx` 为前缀, 而全局函数以 `Afx` 为前缀。MFC 全局函数可以在任何时候的任何地方直接使用。表 1-2 列出了几个比较重要的全局函数。

表 1-2 部分重要的全局函数

函 数 名	函 数 说 明
AfxAbout	无条件终止一个应用程序, 通常在发生无法回复的错误时使用
AfxBeginThread	创建一个新的线程并开始执行
AfxEndThread	终止当前正在执行的线程
AfxMessageBox	显示一个 Windows 消息窗口
AfxGetApp	返回一个指向应用程序对象的指针
AfxGetAppName	返回应用程序名
AfxGetMainWnd	返回一个指向应用程序主窗口的指针
AfxGetInstanceHandle	返回一个标识当前应用程序实例的句柄
AfxRegisterWndClass	为一个 MFC 应用程序注册一个用户自定义的窗口类