

Visual C++ 6.0

编程案例精解

张宏军

党留群 等编著

赵天巨



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Visual C++6.0 编程案例精解

张宏军 党留群 赵天巨 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

全书共分为 12 章, 全面介绍了 Visual Studio 中 Visual C++ 的编程方法与技巧。第 1 章主要介绍了 Windows 编程机制, MFC 类库, Windows 应用程序框架以及 Visual C++ 6.0 集成开发环境的使用。从第 2 章到第 12 章, 采用实例解析的方式, 向读者介绍了文档视图结构、对话框应用程序框架、TCP 和 UDP 的 Intranet 编程、Internet 编程、数据库编程、图形图像编程、多线程编程、磁盘文件管理及多媒体应用等方面的编程方法。

本书几乎涵盖了 Windows 应用程序界面相关的所有元素。这些内容涉及到对话框及其中绝大部分的对话框控件、菜单栏、快捷菜单、工具栏、工具对话框条、滚动条和状态栏等界面元素的使用; 并且实现了启动屏效果、全屏编辑效果、抽屉效果、高级按钮控件和高级列表框控件等十分美观实用的界面效果。

本书实例新颖, 内容丰富, 可操作性强, 精选案例解析清楚。本书既适合于有一定的 C、C++ 语言基础, 并对 Windows 编程有一定了解的初学者快速提高, 又适合于已经对 Visual C++ 有一定学习和了解的中级读者深入学习。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

Visual C++6.0 编程案例精解 / 张宏军等编著. —北京: 电子工业出版社, 2005.1

ISBN 7-121-00743-6

I. V... II. 张... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 138161 号

责任编辑: 祁玉芹

印 刷: 北京市天竺颖华印刷厂

出版发行: 电子工业出版社出版

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 28 字数: 672 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 6000 册 定价: 39.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。

联系电话: (010)68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

Visual C++ 6.0 是迄今为止,功能最为强大的 C++集成开发环境之一。本书从实例开发的角度深入介绍了使用 Visual C++及其 MFC 开发 Windows 应用程序的方法与技巧,并将 C++语言的面向对象的特征融入到具体的实例中,使读者对面向对象的概念及面向对象的程序设计思想有更深入的理解。

本书采用图文并茂的方法,对例程的制作过程进行了详尽的描述,而且附以完整的实现代码及其代码注释。全书共分为 12 章,除了第 1 章概述外,其余 11 章分别是 11 个精选案例的精解。各章内容如下。

第 1 章主要介绍了 Windows 编程机制、MFC 类库和 Windows 应用程序框架,以及 Visual C++ 6.0 集成开发环境的使用。

第 2 章通过写字板应用程序的实现,介绍视图文档应用程序框架结构,其中实现了全屏编辑效果。

第 3 章通过计算器应用程序的实现,介绍对话框应用程序框架结构,其中实现了高级按钮和高级编辑框的效果。

第 4 章通过聊天系统——服务器应用程序的实现,介绍面向连接的可靠的 TCP 网络通信。

第 5 章通过聊天系统——客户端应用程序的实现,介绍面向连接的可靠的 TCP 网络通信和面向无连接的不可靠的 UDP 网络通信,其中实现了“抽屉”效果。

第 6 章通过 FTP 搜索工具应用程序的实现,介绍 Internet 的 FTP 编程。其中实现了基于 IMCP 协议的 Ping 命令和高级属性页控件。

第 7 章通过个人账务管理系统应用程序的实现,介绍数据库编程。

第 8 章通过绘图工具应用程序的实现,介绍图形相关编程。其中大量运用了 C++语言的面向对象的特征。

第 9 章通过麻将游戏应用程序的实现,介绍图形图像编程和非对等网络游戏的建立,其中实现了虚拟的三维坐标。

第 10 章通过代理服务器应用程序的实现,介绍多线程编程及多线程的互斥与同步。

第 11 章通过资源管理器应用程序的实现,介绍 Windows 文件系统的访问与操作。

第 12 章通过媒体播放器应用程序的实现,介绍了 DirectX 的简单应用、音量的调节、CD 和 WAV 文件的播放。

本书内容丰富,实例新颖,可操作性强,内容涉及 TCP 和 UDP 的 Intranet 编程,Internet 编程,数据库编程,图形图像编程,多线程编程,磁盘文件管理及多媒体应用等方面。本书几乎涵盖了 Windows 应用程序界面相关的所有元素,并且实现了启动屏效果、全屏编辑效果、抽屉效果、高级按钮控件、高级列表控件等十分美观实用的界面效果。

本书所选例程的综合性强，涉及知识广泛，可操作性强，均具有实际的使用价值。既适合于有一定的 C、C++ 语言基础，并对 Windows 编程有一定了解的初学者快速提高，又适合于已经对 Visual C++ 有一定学习和了解的中级读者深入学习。

本书由张宏军、党留群和赵天巨主持编写。其中张宏军编写本书的第 1 章~第 2 章，党留群编写第 3 章~第 4 章，赵天巨编写第 5 章~第 6 章，此外，参加编写的人员还有马强、喻波、黄霆、钟华、高海峰、王建胜、张浩和邵蕴秋等。由于编者水平有限，且编写时间仓卒，书中难免有疏漏和不足之处，恳请广大读者批评指正，我们的 E-mail 地址：qiyuqin@phei.com.cn。

为了最大限度地降低读者的购书成本，方便读者学习，本书中的素材可免费从网上下载。

网址：<http://www.firstarcicl.com.cn>

编 者
2004 年 12 月

目 录

第 1 章 Visual C++概述	1
1.1 Visual C++概述	1
1.2 Windows 编程模式	1
1.2.1 面向过程的 MS-DOS 编程	2
1.2.2 面向用户的事件驱动编程	2
1.2.3 Windows 消息机制	2
1.2.4 Windows 系统消息类型	3
1.3 MFC 类库	4
1.3.1 MFC 类库的优越性	4
1.3.2 MFC 类库说明	4
1.3.3 MFC 框架与消息映射	10
1.4 Windows 应用程序框架	11
1.4.1 文档/视图结构概述	12
1.4.2 文档类	12
1.4.3 视图类	12
1.4.4 框架窗口	13
1.5 Visual C++ 6.0 集成开发环境	13
1.5.1 菜单栏	14
1.5.2 工具栏	19
1.5.3 工作区窗口	20
1.5.4 正文窗口	21
1.5.5 输出窗口	21
1.6 本章小结	22
第 2 章 写字板	23
2.1 窗体设计	23
2.2 实现启动屏功能	26
2.2.1 添加启动屏 CSplashWnd 类	26
2.2.2 创建启动屏窗口	27
2.2.3 显示启动屏	28
2.2.4 绘制启动屏窗口区域	29
2.2.5 关闭启动屏	29
2.3 实现快捷工具栏	32
2.3.1 添加工具栏资源	32
2.3.2 工具栏的创建与显示	32
2.3.3 去除浮动工具栏中的系统菜单	34

2.3.4	排列多个工具栏.....	35
2.4	实现状态栏滚动信息.....	36
2.4.1	使用标准状态栏.....	36
2.4.2	在状态栏中显示滚动文本.....	37
2.4.3	在状态栏中显示日期时间.....	38
2.5	实现全屏编辑功能.....	39
2.5.1	添加全屏控制工具栏.....	39
2.5.2	编辑菜单资源.....	39
2.5.3	添加菜单和按钮响应函数.....	40
2.5.4	打开全屏编辑功能.....	41
2.5.5	关闭全屏编辑功能.....	42
2.6	本章小结.....	43
第3章	计算器.....	45
3.1	窗体设计.....	45
3.2	编辑资源.....	47
3.2.1	编辑对话框及控件资源.....	47
3.2.2	编辑菜单资源.....	48
3.3	实现高级按钮控件类.....	49
3.3.1	添加成员变量和成员函数.....	49
3.3.2	添加创建按钮控件代码.....	50
3.3.3	设置按钮形状、外观、颜色、标题.....	51
3.4	实现高级编辑控件类.....	55
3.4.1	添加成员变量和成员函数.....	56
3.4.2	添加编辑控件代码.....	56
3.5	对话框界面设计.....	57
3.5.1	加载菜单资源.....	58
3.5.2	关联控件与对象.....	59
3.5.3	实现对话框扩展功能.....	60
3.5.4	屏蔽 Enter 键.....	62
3.6	数值计算功能的实现.....	62
3.7	本章小结.....	66
第4章	聊天系统——服务器端.....	67
4.1	窗体设计.....	67
4.1.1	建立应用程序框架.....	68
4.1.2	添加对话框控件.....	69
4.2	建立侦听套接字类.....	72
4.2.1	添加成员变量.....	72
4.2.2	添加成员函数.....	72

4.3	建立连接套接字类.....	74
4.3.1	添加成员变量.....	74
4.3.2	添加成员函数.....	74
4.4	实现网络服务功能.....	76
4.4.1	公共数据结构与通信协议.....	76
4.4.2	初始化主程序对话框.....	79
4.4.3	初始化服务器.....	81
4.4.4	建立侦听套接字.....	84
4.4.5	接受客户端连接.....	84
4.4.6	处理各种客户端请求.....	85
4.4.7	发送服务器系统消息.....	96
4.4.8	关闭服务器.....	98
4.5	实现其他本地功能.....	100
4.5.1	屏蔽 Enter 键功能.....	100
4.5.2	用户信息的加载与存档功能.....	101
4.5.3	查看用户信息.....	103
4.5.4	删除无用套接字.....	104
4.5.5	查看用户是否超时.....	106
4.6	本章小结.....	108

第 5 章 聊天系统——客户端 109

5.1	窗体设计.....	109
5.1.1	建立应用程序框架.....	110
5.1.2	编辑主对话框资源.....	111
5.1.3	编辑其他资源.....	112
5.2	实现高级按钮控件类.....	118
5.2.1	设置按钮的自绘制属性.....	118
5.2.2	响应鼠标的单击与移动操作.....	118
5.2.3	绘制对话框客户区域.....	121
5.3	实现高级列表控件类.....	124
5.3.1	设置列表控件元素序列.....	125
5.3.2	设置列表控件元素高度.....	125
5.3.3	设置父窗口.....	125
5.3.4	响应鼠标左键单击操作.....	126
5.3.5	响应鼠标移动操作.....	126
5.3.6	响应计时器消息.....	127
5.3.7	绘制列表框客户区.....	127
5.4	建立用户 Dialog 对话框类.....	130
5.4.1	添加成员变量.....	130
5.4.2	初始化对话框.....	131
5.4.3	绘制对话框背景.....	131

5.4.4	实现交互功能.....	132
5.5	建立用户注册对话框类.....	133
5.5.1	添加成员变量.....	134
5.5.2	初始化对话框.....	134
5.5.3	绘制对话框背景.....	135
5.5.4	实现交互功能.....	136
5.6	实现服务器设置对话框类.....	137
5.6.1	添加成员变量.....	137
5.6.2	初始化对话框.....	138
5.6.3	绘制对话框背景.....	139
5.6.4	实现交互功能.....	139
5.7	实现收发信息对话框类.....	140
5.7.1	添加成员变量.....	141
5.7.2	初始化对话框.....	141
5.7.3	绘制对话框背景.....	142
5.7.4	实现交互功能.....	143
5.8	建立套接字类.....	143
5.8.1	建立 CClientSockets 类.....	143
5.8.2	建立 CChatSockets 类.....	144
5.9	实现客户端功能.....	146
5.9.1	初始化应用程序.....	146
5.9.2	初始化主对话框窗口.....	148
5.9.3	连接服务器.....	149
5.9.4	接收和处理服务器信息.....	150
5.9.5	实现用户登陆功能.....	151
5.9.6	实现用户注册功能.....	152
5.9.7	实现下载与显示好友名单功能.....	153
5.9.8	实现发送信息功能.....	153
5.9.9	实现接收信息功能.....	154
5.9.10	实现用户资料修改功能.....	155
5.9.11	实现下线功能.....	156
5.9.12	实现服务器设置功能.....	156
5.9.13	实现快捷菜单.....	156
5.9.14	实现主对话框抽屉功能.....	157
5.10	本章小结.....	159

第 6 章 FTP 搜索工具..... 161

6.1	窗体设计.....	161
6.1.1	建立应用程序框架.....	161
6.1.2	添加资源.....	162
6.2	实现属性页控件类.....	164

6.2.1	添加数据据构.....	164
6.2.2	添加属性页类.....	165
6.3	实现 FTP 站点搜索功能.....	166
6.3.1	添加站点搜索类.....	167
6.3.2	实现 Ping 命令.....	168
6.3.3	扫描主机端口.....	173
6.3.4	限制地址控件输入.....	173
6.3.5	添加 FTP 站点数据集.....	175
6.5.6	实现站点搜索功能.....	176
6.3.7	查看历史数据.....	181
6.4	实现 FTP 文件列表下载功能.....	183
6.4.1	添加文件列表下载类.....	183
6.4.2	实现连接 FTP 站点功能.....	184
6.4.3	添加文件列表数据集.....	185
6.4.4	实现连接数据库功能.....	187
6.4.5	实现文件列表查看功能.....	189
6.4.6	实现文件列表下载功能.....	192
6.5	实现文件搜索功能.....	196
6.5.1	添加文件搜索类.....	196
6.5.2	实现数据库连接功能.....	197
6.5.3	实现文件搜索功能.....	199
6.6	实现主对话框界面功能.....	200
6.7	本章小结.....	202
第 7 章	个人事务助理	203
7.1	窗体设计.....	203
7.1.1	建立应用程序框架.....	204
7.1.2	添加对话框条资源.....	205
7.1.3	建立 ODBC 数据源.....	206
7.2	实现高级按钮控件类.....	208
7.2.1	添加成员变量和成员函数.....	208
7.2.2	添加创建按钮控件代码.....	210
7.2.3	设置按钮形状、外观、颜色和标题.....	210
7.3	实现高级标签控件类.....	215
7.3.1	添加数据据构.....	215
7.3.2	添加属性页类.....	215
7.4	实现个人账务管理功能.....	217
7.4.1	添加账务数据集类.....	217
7.4.2	实现账务录入功能.....	218
7.4.3	实现账务查询功能.....	223

7.5 实现个人通信簿功能	230
7.5.1 添加通信数据集类	230
7.5.2 实现通信簿功能	231
7.6 实现个人事务助理工具	238
7.6.1 初始化主框架类	238
7.6.2 初始化视图类	239
7.6.3 操作界面的转换	242
7.7 本章小结	243
第8章 绘图工具	245
8.1 窗体设计	245
8.1.1 建立应用程序框架	246
8.1.2 添加资源	248
8.2 初始化应用程序框架	249
8.3 实现逻辑坐标类	251
8.4 实现命令基类及各种命令类	255
8.4.1 实现命令基类	255
8.4.2 实现直线绘制命令类	256
8.4.3 实现矩形绘制命令类	261
8.4.4 实现圆弧绘制命令类	265
8.4.5 实现平移操作命令类	270
8.4.6 实现旋转操作命令类	276
8.5 实现图元基类及各种图元类	281
8.5.1 实现图元基类	281
8.5.2 实现直线图元类	283
8.5.3 实现矩形图元类	285
8.5.4 实现圆弧图元类	289
8.6 实现图元绘制和操作功能	292
8.7 实现图元选取功能	296
8.8 窗口的重绘及文件的存取	299
8.8.1 窗口的重绘	299
8.8.2 文件的存取	300
8.9 本章小结	300
第9章 麻将游戏	301
9.1 窗体设计	302
9.1.1 建立应用程序框架	302
9.1.2 添加对话框资源	303
9.1.3 添加位图资源	304
9.2 界面相关功能的实现	304

9.2.1	实现盒子 CBox 类	304
9.2.2	实现坐标 CPosition 类	306
9.2.3	实现游戏 CPlay 类	308
9.2.4	实现界面 CInterface 类	313
9.3	网络相关功能的实现	326
9.3.1	实现消息 CMsg 类	326
9.3.2	实现连接套接字 CClientSockets 类	328
9.3.3	实现侦听套接字 CServerSockets 类	330
9.4	麻将游戏功能的实现	332
9.4.1	响应鼠标左键单击消息	332
9.4.2	响应鼠标移动消息	340
9.5	本章小结	343
第 10 章	代理服务器	345
10.1	窗体设计	345
10.1.1	应用程序框架的建立	346
10.1.2	添加控件资源	347
10.2	实现共享内存池	348
10.2.1	数据结构	349
10.2.2	初始化内存池	349
10.2.3	实现内存分配功能	350
10.2.4	实现内存释放功能	351
10.2.5	关闭内存池	353
10.3	实现代理服务功能	354
10.3.1	数据结构	354
10.3.2	启动服务器	355
10.3.3	分析用户请求	356
10.3.4	转发用户请求	358
10.3.5	获取目的数据	361
10.4	实现代理服务器	365
10.4.1	初始化服务器	365
10.4.2	启动服务器	365
10.4.3	关闭服务器	366
10.4.4	显示日志	367
10.5	本章小结	367
第 11 章	资源管理器	369
11.1	窗体设计	369
11.1.1	建立应用程序框架	370
11.1.2	添加控件资源	371
11.1.3	添加菜单资源	372

11.2	实现系统图标列表类.....	373
11.2.1	初始化图标链表.....	374
11.2.2	关闭图标链表.....	375
11.2.3	获取图标.....	376
11.3	实现高级树形控件类.....	377
11.3.1	初始化树形控件.....	378
11.3.2	节点的打开与关闭.....	380
11.3.3	获取指定节点的路径.....	383
11.3.4	获取指定路径的节点.....	384
11.3.5	打开指定路径节点.....	386
11.4	实现高级列表控件类.....	387
11.4.1	获取指定路径文件列表.....	387
11.4.2	创建快捷菜单.....	388
11.4.3	实现文件操作功能.....	389
11.5	实现资源管理器功能.....	393
11.5.1	初始化资源管理器.....	393
11.5.2	实现文件列表浏览功能.....	394
11.5.3	实现文件操作功能.....	397
11.5.4	实现地址转到功能.....	400
11.6	本章小结.....	402

第 12 章 媒体播放器.....403

12.1	窗体设计.....	403
12.1.1	建立应用程序框架.....	403
12.1.2	添加控件资源.....	405
12.2	实现音量控制类.....	405
12.3	实现 Wav 文件播放类.....	411
12.4	实现 CD 播放类.....	415
12.4.1	音频设备的打开和关闭.....	417
12.4.2	CD 的播放, 停止和暂停.....	419
12.4.3	CD 曲目信息的获取.....	421
12.4.4	CD 曲目的前进、后退和跳转.....	425
12.5	媒体播放器的实现.....	430
12.5.1	媒体播放器的初始化.....	430
12.5.2	实现 CD 播放功能.....	431
12.5.3	实现停止和暂停功能.....	433
12.5.4	实现后退和前进功能.....	434
12.5.5	播放 Wav 音频文件.....	435
12.5.6	实现音量的调节.....	436
12.6	本章小结.....	436

第 1 章 Visual C++概述

本章主要向读者介绍 Windows 应用程序设计模式、消息机制、MFC(Microsoft Foundation Class)类库、Windows 应用程序框架结构和 Visual C++ 6.0 集成开发环境。现在由于集成开发环境的功能越来越强大,越来越完善,使得用户不用过多关心编程细节。这样,一方面确实方便了用户的程序设计,而另一方面造成用户对 Windows 编程内在机制的了解却越来越少。事实上,只有充分理解 Windows 应用程序的结构和运行机制,才能更好地掌握 Windows 编程。

本章的重点如下。

- (1) Visual C++概述。
- (2) Windows 编程机制。
- (3) MFC 类库。
- (4) Windows 应用程序框架。
- (5) Visual C++ 6.0 集成开发环境。

1.1 Visual C++概述

Microsoft Visual C++是运行于 Windows 平台上的交互式可视化集成开发环境。它是 Microsoft Visual Studio 家族的成员之一。

Visual C++之所以称为集成开发环境,是因为它集程序的代码编辑、编译、连接和调试等功能于一体,给编程人员提供了一个完整而方便的开发环境,并提供许多有效的协助开发工具,如 SourceSafe 等。

Visual C++的 AppWizard 可以为大部分类型的应用程序提供框架代码,用户无需书写代码,只需要简单地单击几个按钮就可以生成一个完整的可以运行的程序框架。

与其他的可视化集成开发环境比较,除了共同的优点之外,Visual C++还具有许多优势。虽然使用 Visual C++设计一些非常简单的程序时,可能体现不出其优势,甚至还很麻烦,需要书写很多的代码,但当用户需要完成更加灵活的、特定的界面时,使用 Visual C++则更加方便,它可以让用户“随心所欲”。

由于 Visual C++基于 C++语言,并且是来自 Windows 操作系统本身的开发者 Microsoft 公司,因此,在众多的可视化集成开发环境中,它是开发 Windows 应用程序的最佳选择。

1.2 Windows 编程模式

在 Windows 操作系统没有问世之前,面向过程的编程模式占有主导地位。在 Windows 操作系统出现之后,产生了事件驱动的编程模式。事件驱动的编程是一种非面向过程,而是面向用户的程序设计方法,它是围绕着消息的产生与处理而进行的。

1.2.1 面向过程的 MS-DOS 编程

传统的 DOS 程序都有而且只有一个 `main` 函数，可以在 `main` 函数中调用其他的函数来完成各种各样的功能和过程。所有函数的调用以及执行都是按照用户预定好的顺序进行的，这种最初的编程模式称之为面向过程的编程方式。

在面向过程的程序中，整个程序按照一定顺序进行；它是一系列预先定义好的操作序列的组合，且该程序完全占用着 CPU，控制整个程序执行的过程。

1.2.2 面向用户的事件驱动编程

当 Windows 操作系统出现之后，面向过程的编程方式已经不能满足用户的要求。一方面，随着程序代码量的逐步增大，面向过程编程的开发与维护工作也越来越困难；另一方面，面向过程的程序设计方式只能适合于在单任务的操作系统上运行。因此，必须有一种新的程序设计模式，即事件驱动编程方式。

Windows 是一种多任务的操作系统，可以同时运行多个程序。每一个程序都不能独占系统资源，而是共享各种系统资源，比如 CPU 和存储器等。事件驱动程序恰好适合于 Windows 的多任务的特点。

事件驱动编程方式完全不同于面向过程的程序设计方式，它是由事件的产生来驱动的。事件的产生是随机的、不确定的，没有预定顺序的。因此，各种事件可以以各种不同的、合理的顺序出现，那么，依赖于事件驱动的程序也可以按各种不同的、合理的流程来执行。也正是这一点，使得多个程序共享系统资源成为可能。

事件驱动的程序设计是一种面向用户的程序设计方式。相对于面向过程的程序设计方式来说，事件驱动的程序设计方式是一种被动的程序设计方式。面向过程的程序需要主动去查询用户操作，并根据用户操作调用相应的处理函数。而事件驱动的程序设计方式是一种被动的程序设计方式。程序总是处于等待用户输入事件状态，然后被动地等待用户操作；用户的各种操作都称之为事件，事件驱动的程序设计方式需要事先为各种需要处理的事件编写事件响应函数；当某个用户操作产生，即某个事件发生时，相应的响应函数就会被调用。程序取得事件并做出反应，处理完毕并返回后，又处于等待事件状态。

1.2.3 Windows 消息机制

要更好地使用 Visual C++ 进行 Windows 编程，就需要进一步了解其消息机制。在 Windows 应用程序中，事件驱动是围绕着消息的产生与处理展开的，消息是对发生的事件的描述信息。消息通知程序有关事件的发生。一条消息包含有消息的名字、标识、消息发生时的一些参数，以及处理这条消息的函数入口指针。

每当用户进行某种操作，比如鼠标单击或键盘按键，就会触发相应的事件。而事件是以消息的方式通知 Windows 应用程序的。一旦应用程序获得某条消息，就根据消息映射表查找相应消息的响应函数的入口地址，调用该函数处理消息，完成用户预期的功能。

在 Windows 操作系统中，应用程序主要以窗口的形式存在。窗口是一个可视的人机交互界面，用来接收各种事件，如用户键盘/鼠标事件、外设的请求事件、定时器的请求事件、信号量的请求事件等。因此，它也就成为应用程序的控制消息发送端和接收端。即 Windows 应用程序是围绕窗口进行的，窗口不仅提供了可视化的应用程序的界面，也是 Windows 消息

的产生和响应的地方。

消息的产生是由于相应的事件被触发；消息的发送以队列形式进行；消息响应遵循一定的顺序。MFC 类库为这种消息响应机制提供了完整的处理功能。MFC 类库中的很多类都具有处理相应消息的功能。在面向过程的程序设计方式中，对外设，比如鼠标、键盘等的控制是通过轮询方式进行，即分别定时查询这些设备的输入请求来完成的。而在 Windows 环境中，这些控制是通过消息机制完成的。因此，Windows 也被称为“基于事件驱动的、消息机制的”操作系统。消息机制是 Windows 能进行多任务并发处理的基础，它保证了 Windows 下同时运行的程序能够协同作业。

在 Windows 中，应用程序都包含一个消息循环。该消息循环持续反复检测消息队列，查看是否有用户事件消息，这些用户事件消息包括鼠标移动、单击、双击、键盘操作和计时器到达等。

事实上，这些事件首先被 Windows 系统接收到。当 Windows 接收到这些事件后，会产生一些相应的描述事件的消息，并且将这些消息分发到相应的应用程序。应用程序接到这些消息后，根据不同的消息查询消息映射，调用其相应的消息响应函数，完成一定的功能与过程。这一系列动作称之为消息响应。

1.2.4 Windows 系统消息类型

Windows 系统具有 3 种不同类型的消息，即标准 Windows 消息、控件通知消息和命令消息。

1. 标准 Windows 消息

标准 Windows 消息是指 Windows 定义的一组特定的消息。这些消息通常使用 WM 前缀，比如 WM_TIMER, WM_PAINT, WM_INITDIALOG 等。这类消息会在特定事件发生时被触发，它们具有特定的用于确定如何对消息进行处理的一些参数。标准 Windows 消息可以用来告诉程序它正在启动或关闭，也可以用来告诉窗口它正在重画或移动等。标准 Windows 消息一般由窗口和视图处理。

2. 控件通知消息

控件通知消息是指由控件产生的通知消息，比如按钮的单击消息、编辑框的内容改变消息、列表框的列表项双击消息等都属于控件通知消息。同标准 Windows 消息一样，控件通知消息也由窗口和视图处理。例如，一个按钮被单击时，它会向父窗口发送 EN_CLICKED 消息，父窗口会根据消息映射，查找并调用其相应的响应函数；类似地，当一个编辑框被按下时，它就会向父窗口发送 EN_EXCHANGE 消息，其父窗口也会根据消息映射，查找并调用其相应的响应函数。

绝大多数控件通知消息与其他标准 Windows 消息一样在框架消息循环中被处理，但是有一个例外：EN_CLICKED 控件通知消息被当做命令消息处理，并与其他命令消息一起进行消息循环。

3. 命令消息

命令消息来自应用程序窗口界面对象，它包括菜单命令、快捷键和工具栏按钮等。当上述的窗口界面对象被点击时，都会产生一条 WM_COMMAND 通知消息。

命令消息不同于 Windows 标准消息和控件通知消息，它可以被通知给更多的对象来处理。如果某条命令消息直接影响到某个对象，则由该对象来处理这条命令消息；如果该对象没有处理这条命令消息的响应函数，那么这一命令消息就会按照一定的顺序移向其他窗口，包括文档、应用程序模块、主框架和视图等。

1.3 MFC 类库

对于任何一种操作系统，如果想在其上进行应用程序开发，那么就必须了解它所提供的 API(Application Program Interface, 应用编程接口)。如果说以前的标准 Windows API 函数库是用于编写 Windows 应用程序的 C 函数库，那么 MFC(Microsoft Foundation Class)则是用于编写 Windows 应用程序的 C++类库。

MFC 以层次结构组织，其中封装了大部分 Windows 的 API 函数，所包含的功能涉及到整个 Windows 操作系统。MFC 不仅为用户提供了 Windows 图形环境下应用程序的框架，而且提供了创建应用程序的框架。使用 MFC 类库和 Visual C++ 提供的高度可视化的应用程序集成开发环境，可以极大地简化应用程序的开发过程，缩短开发周期，并且在代码的可靠性和可重用性上可以得到很大的提高。

在面向对象的程序设计中，MFC 提供一个功能完整的应用程序框架，使程序员可以将其方便地扩展为一个完整的 Windows 应用程序。这使得程序员不必重新从头设计，从而节省了大量的开发时间。而且 MFC 类库所提供的大量的以类为基础的代码框架，非常有利于编程时某些良好架构的形成和具体功能的实现。

1.3.1 MFC 类库的优越性

MFC 完整地封装了 Windows API 函数，不仅包括最常使用的与窗口、控件、菜单和对话框等相关的 API 函数，而且包括具有共性的应用程序操作，如打印、状态条和工具栏等。

MFC 库具有与以 C 语言为基础的 Windows API 开发的 Windows 应用程序共存的能力。在同一程序中，程序员可以同时使用 MFC 中的类和 Windows API 函数。

MFC 提供了自动的消息处理功能。MFC 库将自动处理每一条 Windows 消息，且每一条 Windows 消息被直接映射到一个进行相应处理的成员函数。

MFC 提供了面向对象的 Windows 应用程序接口，与传统的 Windows API 相比较，它具有以下一些明显的优势。

- (1) 有效简化了编写 Windows 应用程序的难度。
- (2) 与 C 语言 API 具有相当的执行速度，使经常使用的代码量最小化。
- (3) 能够直接调用任何一个 API 函数的能力。
- (4) 比 C 语言的 Windows API 更易使用。
- (5) 更容易使用一些强大的复杂功能的抽象概念，如 ActiveX、打印、工具栏和状态栏等。

1.3.2 MFC 类库说明

MFC 类库对 Windows API 函数进行了全面的封装，其封装过程利用面向对象的派生与继承的特点，形成了一个完整的对象层次清晰的面向对象的 Windows API 体系。MFC 类库的层次结构见表 1-1。