



附光盘

举一反三



Visual C++

程序设计实战训练

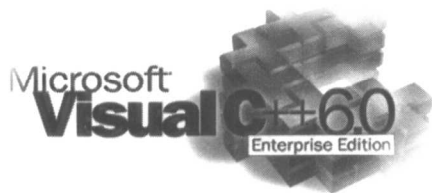
老虎工作室
张卫华
刘 征
赵志刚 编著

人民邮电出版社
POSTS & TELECOM PRESS

举一反三——

Visual C++程序设计实战训练

老虎工作室 张卫华 刘征 编著
赵志刚



This product is licensed to:

人民邮电出版社

图书在版编目 (CIP) 数据

Visual C++程序设计实战训练 / 老虎工作室 张卫华, 刘征, 赵志刚编著.

—北京: 人民邮电出版社, 2004.10

(举一反三)

ISBN 7-115-12643-7

I. V... II. ①老...②张...③刘...④赵... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 103665 号

内 容 提 要

本书以 Visual C++ 6.0 (简称 VC++) 为平台, 对最常用的 VC++ 知识要点进行了分类, 每一类都精心编写了若干实例, 深入分析了面向对象的编程思想。在此基础上, 依次介绍了控件使用、简单绘图、应用绘图、文件操作、数据库访问技术、多媒体应用、文本输出、设计菜单和工具栏以及浏览器的设计等内容。

为了方便读者的学习, 本书的配套光盘中收录了书中实例程序的源文件, 读者可以参考使用。

本书特别适合对 C/C++ 以及面向对象编程思想有一定了解, 希望进一步提高程序设计能力的读者学习, 也可作为 VC++ 的培训教材。

举一反三——Visual C++程序设计实战训练

◆ 编 著 老虎工作室 张卫华 刘 征 赵志刚
责任编辑 李永涛

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132692

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 19.25

字数: 468 千字

2004 年 10 月第 1 版

印数: 1-8 000 册

2004 年 10 月北京第 1 次印刷

ISBN 7-115-12643-7/TP·4201

定价: 38.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223



老虎工作室

主 编：沈精虎

编 委：	许曰滨	黄业清	杜俭业	姜 勇	宋一兵
	谭雪松	向先波	毕丽蕴	高长铎	田博文
	郭万军	詹 翔	宋雪岩	周 锦	冯 辉
	王海英	李 仲	马 震	蔡汉明	张 琴
	赵 晶	张 伟	朱 凯	彭 智	张艳花
	孙海侠	姜继红	李晓武	姚育成	杨平辉

内容和特点

目前,越来越多的软件开发采用面向对象的编程技术,而 Microsoft Visual C++ 6.0 (简称 VC++) 这一可视化的面向对象编程工具,以其强大的功能,在 PC 机上得到了广泛应用。

虽然 VC++ 功能强大,但是与其他编程语言相比,VC++ 却不易学习,让很多初学者望而却步。其实,学习 VC++ 和学习其他新技术一样,需要循序渐进。VC++ 的功能包罗万象,如果一开始就全面掌握它是不现实的。

学习编程语言的最终目的是编写程序,学习 VC++ 的一种行之有效的方式就是从例子入手,先绕开那些复杂抽象的概念、理论,直接从具体的例子开始分析,掌握此类程序所用到的知识,从而学习 VC++ 的编程方法,最后模仿例子编写出自己的程序。这样一来,无需太多知识,即可自己动手编写程序,并且通过编程可以进一步理解概念、理论,逐步扩展自己对 VC++ 的理解,慢慢进入 VC++ 编程世界。

本书采用“举一反三”的风格写作,对最常用的 VC++ 知识要点进行分类,每一类都精心编写了若干实例,通过这些实例详细介绍相关知识点在程序中的应用。本书实例都给出了详细的步骤,以及知识点的使用技巧,在叙述上力求深入浅出、通俗易懂,内容简明清晰、重点突出。读者通过对实例的学习,可以逐步掌握 VC++ 程序的编写。相信本书会为读者学习 VC++ 带来一定的帮助。

当然,本书不是一本 VC++ 大全,没有涉及到 VC++ 的全部内容,感兴趣的读者在学习本书的同时,可以参考其他 VC++ 教材,进一步拓展自己对 VC++ 的了解。

全书共分 10 章,各章内容简要介绍如下。

- 第 1 章: Visual C++ 与 Windows 编程。
- 第 2 章: 控件使用——计算器设计。
- 第 3 章: 简单绘图编程。
- 第 4 章: 应用绘图编程。
- 第 5 章: 文件操作——留言板。
- 第 6 章: 数据库访问技术。
- 第 7 章: 多媒体应用。
- 第 8 章: 文本输出。
- 第 9 章: 菜单和工具栏。
- 第 10 章: 设计一个浏览器。

读者对象

本书特别适合对 C/C++ 以及面向对象编程思想有一定了解,希望进一步提高程序设计能

力的读者学习，也可作为 VC++ 的培训教材。

光盘内容及使用方法

1. 光盘内容

为了方便读者的学习，我们将书中实例程序的源文件收录在本书的配套光盘中。下面是本书配套光盘内容的详细说明。

- 本书中的所有例子以章为单位存放，例如第 5 章例子存放在“第 5 章”目录下。
- 每一章中的例子目录名就是对应工程名字，目录是建立工程时自动生成的。
- 工程目录下除了相关文件外，还有两个子目录：**Release** 和 **res**，分别存放可执行文件和资源文件，这些都是系统自动生成的。每个例子都已经生成可执行文件，存放在该工程目录下的 **Release** 子目录中，图标为蓝色背景的 3 个立方体。
- 第 6 章的例子因为需要建立数据库连接，而读者机器上的连接可能与书中介绍的有区别，所以例子有可能不能直接运行。但该章所有程序都经作者编译运行过，读者只要按照书中给出的步骤建立工程，就一定可以得到一个可执行文件。
- 书中有的例子用到的位图文件，读者可以根据自己的喜好，选择任意其他的位图文件使用，譬如在 Windows 2000 的 WINNT 目录下面，就有很多位图文件（后缀名为*.bmp）可以选择，没必要一定使用书中的位图文件（为方便起见，光盘上第 3 章例子中用到的位图采用了 Windows 2000 的 WINNT 目录下面的 **Gone Fishing.bmp** 文件）。
- 第 9 章的例子中，在给菜单和工具栏添加图标时，需要读者自己手工绘制图标，所以读者按照例子建立工程时，由于手工绘制的差异，不可能和书中的一模一样。
- 书中很多例子中的长语句有换行，读者在照搬书中例子时，由于图书排版的原因，要注意换行处是否有回车符，有的话，要把回车符删掉。读者可参考光盘上的源程序。

2. 运行环境

- 硬件环境：奔腾 350MHz 以上多媒体计算机。
- 软件环境：Windows 2000; Microsoft Visual C++ 6.0。

感谢您选择了本书，希望我们的努力对您的工作和学习有所帮助，也希望您把对本书的意见和建议告诉我们。

老虎工作室网站 <http://www.laohu.net>，电子函件 postmaster@laohu.net。

老虎工作室

2004 年 8 月

第 1 章 Visual C++与 Windows 编程	1
1.1 Windows 编程基础	1
1.2 Visual C++6.0 简介	4
1.3 Windows 程序设计过程	6
1.3.1 两种不同的 Windows 编程方法	6
1.3.2 使用 MFC AppWizard	7
1.3.3 使用 MFC AppWizard 开发程序的流程	8
1.3.4 关于应用程序框架	9
1.4 小结	10
第 2 章 控件使用——计算器设计	11
2.1 相关知识	11
2.2 使用控件设计计算器	12
2.2.1 典型实例——用编辑框、静态文本框及按钮设计计算器	12
2.2.2 起步——用数字键盘和列表框设计计算器	22
2.2.3 进阶——单选按钮在计算器中的应用	28
2.2.4 提高——复选框在计算器中的应用	35
2.3 小结	38
第 3 章 简单绘图编程	39
3.1 相关知识	39
3.2 直线	40
3.2.1 典型实例——编制用默认画笔画一条简单直线的程序	40
3.2.2 起步——编制画 5 条蓝色、线型各异的水平直线的程序	44
3.2.3 进阶——编制自定端点画线段的程序	46
3.2.4 提高——编制随手画直线的程序	48
3.3 椭圆	51
3.3.1 典型实例——编制随手画简单椭圆的程序	51
3.3.2 起步——编制用单色实画刷画椭圆的程序	54
3.3.3 进阶——编制用库存画刷画椭圆的程序	55
3.3.4 提高——编制用位图画刷填充椭圆的程序	58

3.4 位图	61
3.4.1 典型实例——编制直接在窗口中显示位图的程序	61
3.4.2 起步——编制用平铺方式显示位图的程序	64
3.4.3 进阶——编制用伸缩方式显示位图的程序	65
3.4.4 提高——编制运行时选择位图文件的程序	67
3.5 小结	72
第4章 应用绘图编程	73
4.1 直方图	73
4.1.1 典型实例——编制画简单直方图的程序	73
4.1.2 起步——编制画三维直方图的程序	75
4.1.3 进阶——编制有坐标轴与数据标注的直方图程序	77
4.1.4 提高——编制从文件中获取数据的直方图程序	81
4.2 圆饼图	84
4.2.1 典型实例——编制画一个扇形的程序	85
4.2.2 起步——编制由多个扇形组成的圆饼图程序	87
4.2.3 进阶——编制突出显示一个扇形的程序	90
4.2.4 提高——编制给扇形标注数据的程序	92
4.3 函数曲线	95
4.3.1 典型实例——编制简单的多项式函数曲线程序	96
4.3.2 起步——编制翻转坐标系 y 方向的程序	101
4.3.3 进阶——编制窗口变换与映射方式的程序	103
4.3.4 提高——编制给窗口加上滚动条的程序	107
4.4 其他函数曲线	112
4.4.1 典型实例——编制参数函数曲线程序	112
4.4.2 起步——编制极坐标函数曲线程序	117
4.4.3 进阶——编制分数维图形程序	118
4.4.4 提高——编制曲线轨迹动画程序	123
4.5 小结	128
第5章 文件操作——留言板	129
5.1 相关知识	129
5.2 编制写留言板程序	130
5.2.1 典型实例——编制简单的写留言并保存的程序	131
5.2.2 起步——编制有滚动条的留言板程序	133
5.2.3 进阶——编制能选择留言保存路径的程序	134
5.2.4 提高——编制能签署留言日期的程序	136
5.3 读留言板	138

5.3.1 典型实例——编制简单的阅读留言板程序	138
5.3.2 起步——编制给留言板窗口添加图片的程序	140
5.3.3 进阶——编制选择留言打开路径的程序	141
5.3.4 提高——编制留言内容转存的程序	143
5.4 小结	145
第 6 章 数据库访问技术	147
6.1 相关知识	147
6.1.1 注册数据源	147
6.1.2 ODBC 两个类的运用	149
6.1.3 ADO(Active Data Object, 活动数据对象)	149
6.2 基于 ODBC 技术的数据库访问	149
6.2.1 典型实例——显示数据库表中的所有记录	149
6.2.2 起步——查询数据库表中的记录	156
6.2.3 进阶——删除数据库表中的记录	161
6.2.4 提高——在数据库表中添加记录	162
6.3 基于 ADO 技术的数据库访问	164
6.3.1 典型实例——显示数据表中所有记录	165
6.3.2 起步——创建新表	169
6.3.3 进阶——添加并显示记录	175
6.3.4 提高——查询、修改与删除记录	180
6.4 小结	188
第 7 章 多媒体应用	189
7.1 相关知识	189
7.1.1 多媒体控制接口 MCI	189
7.1.2 ActiveMovie 控件	192
7.1.3 MMControl 控件	192
7.2 CD 播放器	193
7.2.1 典型实例——简单的 CD 播放器	193
7.2.2 起步——美化 CD 播放器	202
7.2.3 进阶——增加播放下一首的功能	204
7.2.4 提高——增加播放上一首的功能	206
7.3 VCD 播放器	208
7.3.1 典型实例——简单的 VCD 播放器	208
7.3.2 起步——给播放器增加音量控制功能	210
7.3.3 进阶——用 MMControl 控件制作播放器	211
7.3.4 提高——给播放器添加进度条	214

7.4 小结	218
第 8 章 文本输出	219
8.1 相关知识	219
8.2 文本输出	219
8.2.1 典型实例——在指定位置输出文字	219
8.2.2 进阶——设置输出文本的字体和颜色	221
8.2.3 进阶——动态输出文本字体并设置颜色	224
8.2.4 提高——格式化输出文本并设置背景色	227
8.3 小结	229
第 9 章 菜单和工具栏	231
9.1 相关知识	231
9.2 菜单	232
9.2.1 典型实例——添加自己的菜单项	232
9.2.2 起步——菜单项的变灰和复原	246
9.2.3 进阶——给菜单项加上检查标记	250
9.2.4 提高——右键单击弹出快捷菜单	255
9.2.5 再提高——在快捷菜单中加入图标	258
9.3 工具栏	262
9.3.1 典型实例——在系统缺省工具栏上添加按钮	262
9.3.2 起步——在工具栏中加入组合框	265
9.3.3 进阶——在工具栏中加入编辑框	277
9.4 状态栏	282
9.4.1 典型实例——在状态栏中显示信息	283
9.4.2 起步——在状态栏中开辟新的分区	286
9.4.3 进阶——精确定义分区的宽度	287
9.5 小结	290
第 10 章 设计一个浏览器	291
10.1 相关知识	291
10.2 制作浏览器	291
10.2.1 典型实例——单一 URL 地址连接程序的实现	291
10.2.2 起步——带地址栏的浏览器	294
10.2.3 进阶——添加前进、后退功能的浏览器	296
10.2.4 提高——添加前进、后退及刷新功能的浏览器	298
10.3 小结	298

第1章 Visual C++与 Windows 编程

随着 Visual C++功能的不断完善和 Windows 操作系统在 PC 上的广泛应用,掌握 Visual C++环境下的程序设计方法已成为很多人的心愿。不过,Visual C++程序设计方法与传统编程方法存在着很大差异,以致一些对 C 语言熟悉的人也颇感陌生。正是因为如此,Microsoft 在 Visual C++中提供了功能强大的 MFC AppWizard,大大降低了编程难度。

本书通过大量的实例,循序渐进地介绍如何使用 Visual C++ 6.0 的 MFC AppWizard 来编写各种类型的应用程序,帮助读者顺利进入 Visual C++ 6.0 编程世界。书中所有例子的开发环境均为 Visual C++ 6.0,操作系统为 Windows 2000。

本章主要介绍了如下内容:

- 类和对象
- 窗口、句柄和 ID
- Windows 程序的运行机制
- Windows 程序的设计过程
- 使用 MFC AppWizard 开发程序的步骤
- 应用程序框架的组成

1.1 Windows 编程基础

Windows 是大家所熟知的图形用户界面的操作系统,其特点就是为各种类型的应用程序提供标准化的窗口和菜单。Windows 环境下的程序设计思想是面向对象的,程序设计平台大多是可视化(Visual)的,面向对象的编程思想和可视化是紧密联系在一起的。Visual C++ 6.0 就是可视化的、面向对象的编程工具。现在,我们先来了解一下用 Visual C++ 6.0 进行 Windows 编程所涉及的几个概念。

一、类和对象

1) 类

类(Class)是实体的模型,是对同类实体的抽象,它把数据(称做类的属性)和函数(称做类的行为或方法)封装在一起作为一种类型。数据和函数称为类的成员。对类的成员的访问有 3 种控制方式: private、protected 和 public。

类实现了对数据的封装,一般用户对类中数据成员的操作是通过该类的成员函数进行的,所以类的成员函数也称为类的接口。

类还实现了类型定义,用来声明对象。

(2) 对象

对象(Object)是类的实例,即用类定义的变量。一般把系统内部已定义的实例称做变



量，把用户自定义类型的实例称做对象。不过，在 C++ 中，对象和变量可以理解为一回事。要使用类的函数，必须定义一个该类的对象。

在 Windows 中，对象可以是所有的规范部件，如窗口、菜单、工具栏、按钮和编辑框等，当然，读者自己编写的程序也是对象。

二、窗口

窗口是 Windows 风格应用程序的基本部件，所有的程序都需要窗口，程序的运行及结果的显示等都是在窗口中进行的，程序和用户之间信息的交换也是通过窗口来实现的。

三、句柄

句柄 (Handle) 是 Windows 编程中的一个重要概念。句柄是一个 4 字节长的整数值，是应用程序中不同对象的编号，每个对象的句柄是惟一的。应用程序通过句柄来访问相应的对象。

每个对象除了句柄外，另外还有一个识别号码 (ID)，它也是一个 4 字节长的整数值，可以看做是对象的名字。

句柄和 ID 的区别主要有以下 3 个方面。

- 句柄是系统内给定的，而 ID 是用户自行设定的；
- 句柄是创建窗口时得到的，而 ID 是程序编写时确定的；
- 句柄具有惟一性，而 ID 可以重复。

四、消息及事件驱动

消息 (Message) 是由事件产生的，任何事件的发生都会伴随着相关消息的产生，消息用来描述事件发生的信息。

利用 Visual C++ 6.0 开发平台设计应用程序，与传统的设计方法存在很大差异。过去人们习惯使用顺序的、过程驱动的方法来设计程序，程序的运行次序是明确的，程序中的事件或过程的顺序是预知的、确定的。而现在 Windows 程序设计的情况就大不相同了。虽然程序也是按照一定的流程编写，程序之间也存在调用关系，但是程序的运行次序却不是设计者预定的。

Windows 程序运行的特点就是围绕事件的发生，也就是消息的产生来运行消息处理函数。换句话说，程序是由事件驱动的。因此，程序运行的次序不是明确的，而是按照事件发生的次序，即消息产生的次序来运行的。所以编写 Windows 程序时，不需要考虑程序的运行次序，只要编写针对各个消息的处理函数就可以了，当某个消息产生时，会自动驱动相应的函数运行。

Windows 中一些常用的消息见表 1-1。

表 1-1 常用消息一览表

常用消息	消息含义
WM_COMMAND	由菜单或者控件命令产生的消息
WM_PAINT	窗口尺寸改变、滚动条移动、显示请求等



续表

常用消息	消息含义
WM_QUIT	退出应用程序的消息
WM_LBUTTONDOWN	鼠标左键按下的消息
WM_LBUTTONUP	鼠标左键抬起的消息
WM_LBUTTONDOWNBLCLK	双击鼠标左键的消息
WM_MOUSEMOVE	鼠标移动消息
WM_MOVE	改变窗口位置的消息
WM_SIZE	改变窗口尺寸的消息

在应用程序运行中，消息是最重要的通信手段。Windows 中的一个消息由一个结构类型 MSG 来表示。MSG 定义如下：

```
typedef struct tagMSG
{
    HWND hwnd;           // 获取消息的窗口句柄
    UINT message;         // 消息编号
    WPARAM wParam;        // 记录消息附加信息的字参数
    LPARAM lParam;        // 记录消息附加信息的长字参数
    DWORD time;           // 消息被发送到消息队列的时间
    POINT pt;             // 消息发送时的光标所在位置
} MSG;
```

其中，POINT 是一个坐标类型，POINT 本身也是一个结构类型，它的两个变量 x 和 y 分别表示屏幕上一个点的横坐标和纵坐标。

用户在输入设备上的任何一个操作，都将通过一条消息传送到相关的程序中。图 1-1 给出了鼠标在移动、按下和弹起等操作过程中的消息传送情况。

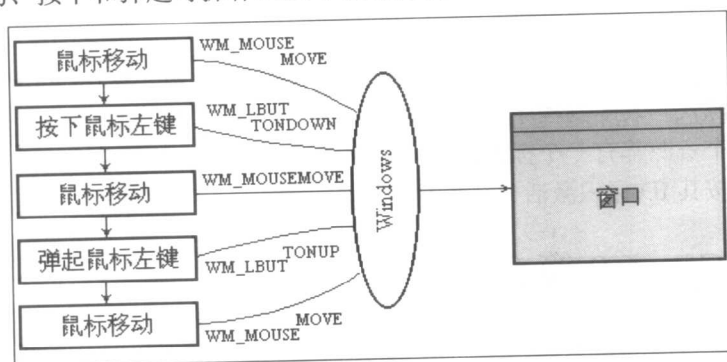


图1-1 消息传送示意图

如图 1-1 所示，用户操作鼠标的信息首先传送到 Windows 的消息队列上，按先后次序排队；然后由 Windows 从队首取出消息送给一个 CwinThread 类（这是由 MFC 封装的类）；后者再根据收到的先后次序逐一分发给接收者窗口。比如 WM_LBUTTONDOWN 消息产生，那么当收到消息后作出的反应可以是“在屏幕上描绘该按钮被按下去的图样”。



通常，将根据消息启动响应函数的过程称做“消息映射”。图 1-2 描述了这一过程。

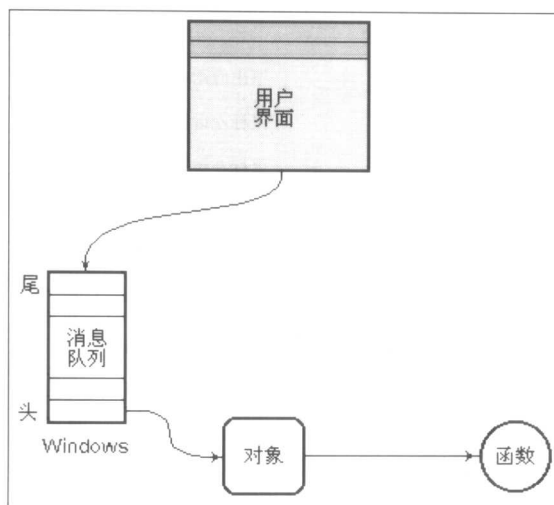


图1-2 事件驱动过程

五、工程及资源

应用程序通常被称为“工程”(Project)，有时也被翻译为“项目”。一个工程有一个工程名，其若干个类和资源被分放在多个文件中，这些文件集中存放在一个独立的文件夹中，该文件夹由 Visual C++ 6.0 在创建工程时自动建立。

每个工程有许多资源，每个资源都要由若干数据来描述。工程中包含的资源如下：

- 菜单 (Menu)
- 图标 (Icon)
- 对话框 (Dialog)
- 工具栏 (Toolbar)
- 位图 (Bitmap)

为了管理方便，Visual C++ 6.0 将工程的资源存储于“*.rc”及“*.bmp”文件中，待程序连接时再将资源代码装配到程序主体中。

工程中的每个资源都有一个标识，由“资源编辑器”实施管理。当用户对某个资源进行操作时，系统将按其 ID 标识激活一个关联的函数。

1.2 Visual C++ 6.0 简介

Visual C++ 6.0 是 Microsoft 公司推出的 Microsoft Visual Studio 6.0 系列中的拳头产品，是在 PC 上开发软件不可多得的有力工具，得到了广泛应用。

Visual C++ 6.0 是一种 C/C++ 编译程序，内含一个集成开发环境，简称 IDE (Integrated Development Environment)。IDE 包括的内容相当多，如程序编辑器、资源编辑器、编程向导、类向导和连接器等。齐全的功能使 Visual C++ 6.0 成为十分优秀的软件开发工具。

启动 Visual C++ 6.0 后，其界面如图 1-3 所示 (图中打开的是单文档工程 EXAMPLE)。

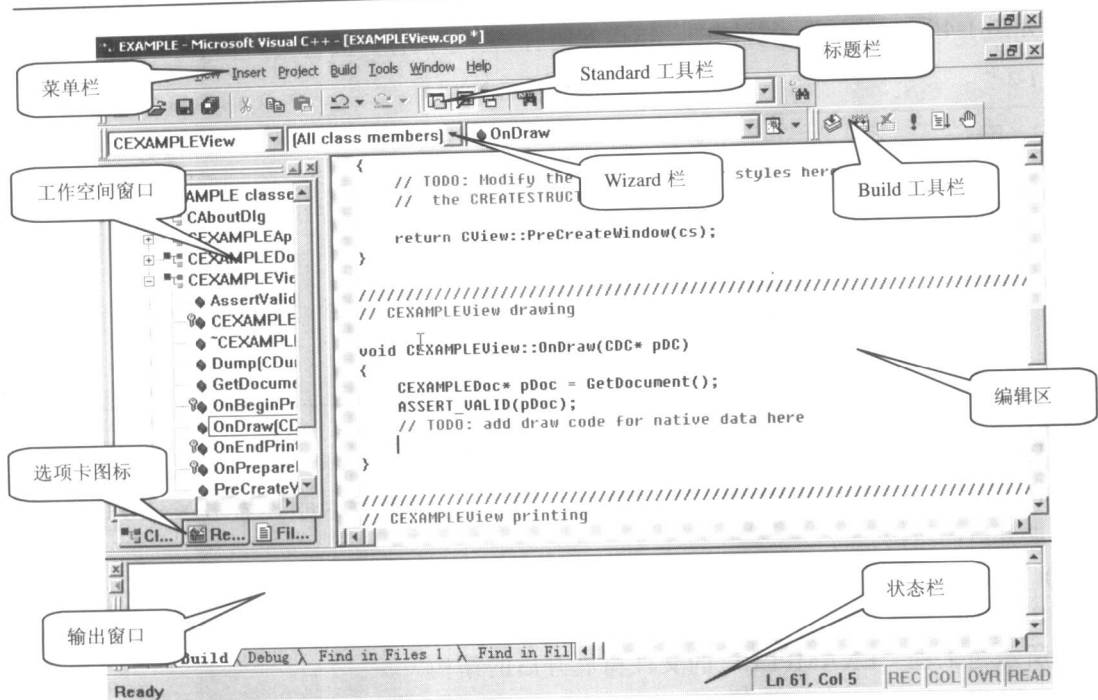


图1-3 Visual C++ 6.0 的软件开发平台

下面重点介绍工作空间窗口 (Workspaces)。

启动 Visual C++ 6.0 后, 选择菜单栏中的【File】/【Open】命令打开一个工程, 从工程文件夹中选择 “*.dsw” 文件。“*.dsw” 文件是记录文件, 它记录了开发环境的有关参数, 每当打开一个已有的工程时, 将其 “*.dsw” 文件打开, 该工程的工作空间窗口便被激活。

工作空间窗口有 3 个选项卡: 【ClassView】、【ResourceView】和【FileView】, 它们按树形结构分别记录着项目的 “类”、“资源” 和 “文件”。单击左侧的加号就可以将树形结构一层层地展开, 单击左侧的减号就可以将树形结构一层层地折叠起来。

(1) 【ClassView】选项卡

比如打开了一个工程 “EXAMPLE”, 在它的工作空间窗口选择【ClassView】选项卡, “EXAMPLE” 的类树形结构便显示出来。其中有工程 “EXAMPLE” 的所有类的名称, 各个类所包含的成员函数和成员变量。展开后的类结构如图 1-4 所示。

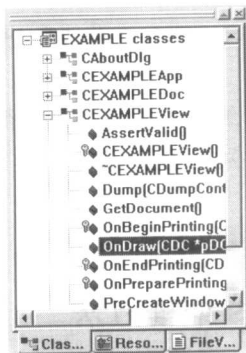


图1-4 【ClassView】选项卡



在如图 1-4 所示的树形类视图中，结点左侧有一些小图标，其含义如表 1-2 所示。

表 1-2 树形类视图中小图标的含义

图标	含义	图标	含义
	工程		公有成员函数
	类		私有成员变量
	私有成员函数		保护成员变量
	保护成员函数		公有成员变量

(2) 【ResourceView】选项卡

【ResourceView】选项卡中显示的是工程“EXAMPLE”的各种资源，有加速器、对话框、按钮、菜单、工具栏等，另外还有各资源的 ID。如图 1-5 所示是工程“EXAMPLE”的资源树展开后的结构。

(3) 【FileView】选项卡

【FileView】选项卡中显示的是各类文件，有头文件、实现文件、资源文件等。如图 1-6 所示是工程“EXAMPLE”的文件树展开后的结构。

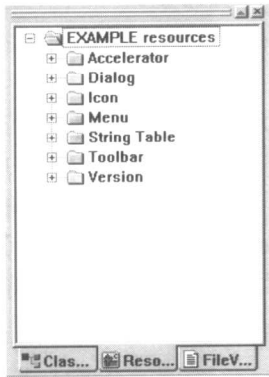


图1-5 【ResourceView】选项卡

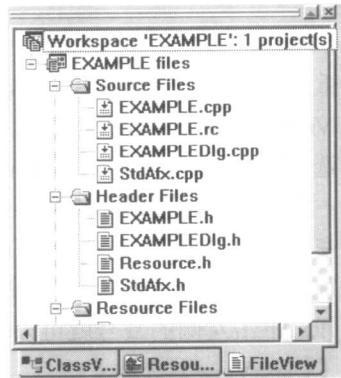


图1-6 【FileView】选项卡

1.3 Windows 程序设计过程

Windows 编程思想是面向对象的，那么，此类程序设计的一般过程是怎样的呢？我们简单地把整个过程分为 4 步。

- 对要解决的问题进行分析归纳，总结其一般特征，即抽象化。
- 根据抽象的结果建模，将其定义为类。
- 建立类的实例，即对象。
- 调用对象的各种方法解决问题。

1.3.1 两种不同的 Windows 编程方法

要建立一个 Windows 应用程序，应该如何下手？方法有如下两种。



(1) 基于 Windows API 函数的程序设计方法

API 就是应用程序编程接口, Windows API 由大量的 C 函数所组成, 这些函数在 Windows API 参考手册中都有详细介绍。基于 API 设计 Windows 应用程序时, 开发者需要直接调用 API 函数, 必须编写大量代码, 并且熟知 Windows 应用程序的框架, 这无疑增加了程序的编写难度, 当然这样编写的程序会比较灵活。

(2) 基于 MFC 的程序设计方法

MFC (Microsoft 基本类库) 是一个很大的 C++ 类层次结构, 其中封装了大量的类及其函数, 很多 Windows 程序所共有的标准内容可以由 MFC 的类来提供。如窗口、按钮等标准部件都有相应的类, 程序中用到这些标准部件时, 直接使用即可。因此, 使用 MFC 编程, 程序员可以把注意力集中于自己的程序所特有的部分, 不需要事事亲自动手。与直接调用 API 函数编程相比, 这将会大量减少程序员编写的代码数量, 使编程工作变得更容易; 同时, 程序标准部分由 MFC 的类来提供, 将使程序变得更规范, 更具可读性, 效率也更高。

MFC 的最大优点是它以高效简洁的方式为开发者做了所有标准部分的工作。MFC 中包含了成千上万行正确、经过优化和功能强大的 Windows 代码。只要定义了某个类的对象, 就可以调用该类的成员函数, 轻轻松松完成本来应该由你自己一行行书写的程序。从这点上讲, MFC 极大地降低了程序开发难度。

Visual C++ 6.0 应用程序开发程序环境特别适合于使用 MFC (也有其他开发环境使用 MFC)。本书的目的就是帮助读者如何利用 MFC 快速建立 Windows 应用程序。

1.3.2 使用 MFC AppWizard

Windows 程序远比 DOS 程序庞大, 其编写也复杂得多, 尽管 MFC 封装了大量的类及其函数, 然而如果直接使用 MFC, 仍然是比较困难的。既然 MFC 封装的类是关于 Windows 程序的标准部分, 那么有没有一种标准的、规范的方法可以轻松的使用 MFC 呢? 答案是肯定的, 那就是 MFC AppWizard!

AppWizard (应用程序向导) 是 Visual C++ 6.0 开发环境中强大的编程工具, 用它可以帮助用户创建各种不同类型的应用程序。

AppWizard 怎样引导用户呢? 它通过一系列对话框, 让用户对自己将要建立的工程进行必要的选择, 以此定制工程。譬如要创建工程 “EXAMPLE”, 需要若干个步骤, 每个步骤用户都可以根据实际情况做出自己的选择, 这个过程就定制了工程 “EXAMPLE”, 最后 AppWizard 根据用户的选择, 自动创建工程 “EXAMPLE” 的各种文件如工程文件、源代码文件、头文件和资源文件等, 从而创建新工程 “EXAMPLE”。

MFC AppWizard 对编程的工作量会有多大影响呢? 请看下面的比较。

(1) 不使用 MFC AppWizard

一个完整的工程一般由资源文件 (.rc)、源代码文件 (.cpp)、头文件 (.h)、模块定义文件 (.def) 和工程文件 (.mak) 等 5 种不同类型的文件组成, 如果程序所需要的一切内容都由用户自己编写, 仅图形用户界面程序的框架部分, 所要编写的代码数量也是很可观的 (千行左右), 即使用 MFC, 如果不借助 AppWizard, 也要编写大量代码。