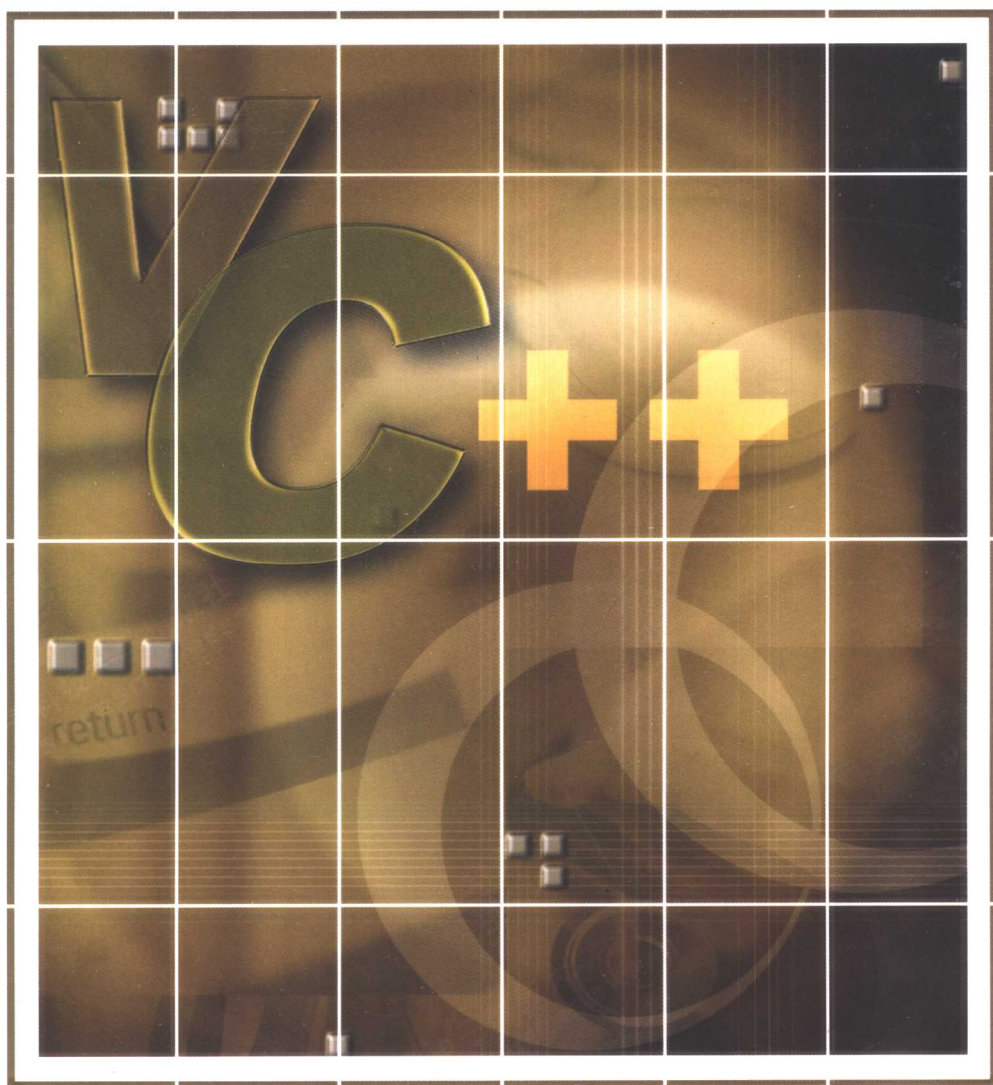


新世纪计算机类本科系列教材



面向对象程序设计 与VC++实践

揣锦华 编著

西安电子科技大学出版社
<http://www.xduph.com>



新世纪计算机类本科系列教材

面向对象程序设计 与 VC++ 实践

揣锦华 编著

西安电子科技大学出版社

2005

内 容 简 介

本书由三个层次构成。第一个层次为第 1~5 章,以 C++ 语言为基础,介绍面向对象程序设计的基础知识。第二个层次为第 6、7 章,介绍 Windows 编程的基础知识和 Windows 程序的基本特征,并介绍了应用程序编程接口(API)及其在 Windows 绘图程序设计中的应用。第三个层次为第 8~13 章,介绍使用 MFC 进行 Windows 应用程序设计的方法,并给出了 MFC 的一些高级应用。全书以 Visual C++ 6.0 开发环境为主线,语言通俗易懂,实用性强,实例丰富,每章均配有习题。

本书既可作为高等学校计算机及相关专业本科生和研究生的教学用书,也可供相关人员自学使用。

★本书配有电子教案,需要者可与出版社联系,免费提供。

图书在版编目(CIP)数据

面向对象程序设计 with VC++ 实践 / 揣锦华编著. — 西安: 西安电子科技大学出版社, 2005.2

(新世纪计算机类本科系列教材)

ISBN 7-5606-1490-6

I. 面… II. 揣… III. ① 面向对象语言 - 程序设计 - 高等学校 - 教材

② C 语言 - 程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 002521 号

责任编辑 潘恩祥 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 20.75

字 数 491 千字

印 数 1~4 000 册

定 价 22.00 元

ISBN 7-5606-1490-6/TP·0792(课)

XDUP 1761001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

Visual C++ 6.0 是目前国内最为流行的新一代面向对象的程序开发环境,它是美国微软公司推出的 Visual Studio 6.0 开发工具套件之一。由于 Visual C++ 6.0 具有优越的性能和强大的功能,因此它成为该开发工具套件中最为完善的软件开发系统之一。Visual C++ 6.0 提供了一个高效的 Windows 编程环境,将程序和资源的编辑、编译、调试和运行融为一体,并且提供了大量的程序开发工具,如应用程序向导 AppWizard、类向导 ClassWizard 等。这些工具都高度集成于一个名为 Developer Studio 的开发平台中,利用该开发平台,用户可以自动生成应用程序框架,编写出各种各样的软件。而且,从最简单的基于对话的程序到复杂的多用户程序,都可以直接设计软件的 Windows 用户界面。除此之外,微软基础类库 MFC 使得应用程序框架、类框架及函数框架的自动生成更加快捷方便,大大提高了程序的质量。

本书以 C++面向对象程序设计语言为基础,以 Visual C++ 6.0 开发环境为主线,结合应用实例,强调“实用”,以通俗易懂的语言,将面向对象程序设计的思想和 Visual C++ 6.0 的具体应用展示给读者。全书共分为三个层次。第 1~5 章为第一个层次,该层次以 C++语言为基础介绍面向对象程序设计的基础知识,强化面向对象程序设计的思想。第 6、7 章为第二个层次,介绍 Windows 编程的基础知识和 Windows 程序的基本特征,并介绍应用程序编程接口(API)及其在 Windows 绘图程序设计中的应用。第 8~13 章为第三个层次,介绍使用 MFC 进行 Windows 应用程序设计的方法,具体包括 MFC 应用程序框架, MFC 应用程序的消息映射,常用基本控件,如编辑框、按钮、菜单、滚动条、列表框等的使用,以及文档/视图结构、加速键、工具条、状态条等用户界面的设计,最后介绍了 MFC 的一些高级应用,如动态链接库、多任务、数据库等。为了使读者能更深入地了解 Visual C++ 6.0,在每章中都给出了应用实例,以便读者参考,同时书中各章均配有习题,以方便读者自我考核。

本书注重培养读者分析问题和解决问题的能力,强化动手实践,引导理性思维和理性实践,内容由浅入深,循序渐进,即可作为大专院校的教材和教学参考书,也可供广大计算机工作者和软件开发者自学。

鉴于作者水平有限,书中难免存在错误和不妥之处,敬请读者批评指正。

作 者
2004 年 12 月

目 录

第 1 章 Visual C++开发环境	1	2.3.4 内联函数	24
1.1 VC++用户界面	1	2.3.5 系统函数的使用	25
1.1.1 菜单栏	2	2.4 数组	25
1.1.2 工具栏	7	2.4.1 一维数组	25
1.2 AppWizard 工具	8	2.4.2 多维数组	26
1.2.1 AppWizard 简介	8	2.4.3 数组应用	26
1.2.2 使用 AppWizard 创建 C++源程序	9	2.4.4 数组作为函数的参数	29
1.3 项目和项目工作区	9	2.4.5 数组与字符串	31
1.3.1 FileView	10	2.5 指针	32
1.3.2 ClassView	11	2.5.1 指针型变量	32
1.3.3 ResourceView	12	2.5.2 指针与字符串	34
1.3.4 项目开发步骤	13	2.6 构造数据类型	35
1.4 资源管理器的使用	13	2.6.1 结构体	35
1.5 ClassWizard 工具	14	2.6.2 共用体	36
1.6 使用联机帮助	14	2.6.3 枚举类型	37
习题	15	习题	37
第 2 章 C++编程基础	16	第 3 章 面向对象程序设计	39
2.1 数据类型与表达式	16	3.1 面向对象程序设计思想	39
2.1.1 常量	17	3.1.1 面向对象程序设计的特点	40
2.1.2 变量	18	3.1.2 面向对象的标记	42
2.1.3 引用	18	3.2 类和对象	43
2.1.4 表达式	19	3.2.1 类的声明	43
2.2 程序的控制结构	21	3.2.2 对象	44
2.2.1 选择结构	21	3.2.3 构造函数和析构函数	45
2.2.2 switch 语句	21	3.2.4 类的组合	48
2.2.3 循环结构	22	3.2.5 类模板	51
2.2.4 其他控制语句	23	3.2.6 string 类	53
2.3 函数	23	3.3 面向对象程序的结构	55
2.3.1 函数定义	23	3.3.1 作用域和生存期	55
2.3.2 调用函数	23	3.3.2 静态成员	60
2.3.3 默认参数的函数	24	3.3.3 友元	62
		3.3.4 常类型	64

3.3.5 多文件结构.....	66	5.3.2 设置断点.....	126
3.4 动态内存分配.....	67	5.3.3 运行调试器.....	129
3.4.1 new 运算符.....	67	5.3.4 调试器窗口.....	130
3.4.2 delete 运算符.....	68	5.3.5 使用调试器.....	132
3.4.3 动态内存分配的应用实例.....	68	5.4 高级调试技巧.....	134
习题.....	74	5.4.1 调试过程中的异常处理.....	134
第4章 继承性与多态性.....	80	5.4.2 调试线程.....	135
4.1 继承与派生.....	80	5.4.3 调试动态链接库.....	136
4.1.1 派生类的声明.....	80	习题.....	136
4.1.2 派生类生成过程.....	81	第6章 Windows 编程基础.....	137
4.1.3 多继承.....	82	6.1 Windows 程序的特点.....	137
4.1.4 类的继承方式.....	84	6.2 基本概念.....	139
4.1.5 派生类的构造函数和析构函数.....	87	6.2.1 事件及事件驱动.....	139
4.2 派生中成员的标识与访问.....	90	6.2.2 消息.....	140
4.2.1 作用域分辨.....	90	6.2.3 对象与句柄.....	142
4.2.2 虚基类.....	92	6.2.4 API 函数.....	142
4.2.3 对象指针.....	94	6.3 Win32 程序的基本结构.....	143
4.3 运算符重载.....	97	6.3.1 Win32 源程序的组成.....	143
4.3.1 运算符重载的规则.....	97	6.3.2 Windows 数据类型.....	145
4.3.2 运算符重载为成员函数.....	98	6.4 利用 API 开发 Win32 程序实例.....	145
4.3.3 运算符重载为友元函数.....	100	6.5 使用 AppWizard 生成 Win32 程序.....	150
4.3.4 其他运算符重载.....	101	习题.....	156
4.4 虚函数.....	107	第7章 绘图与文本输出.....	157
4.4.1 虚函数的定义.....	107	7.1 设备环境.....	157
4.4.2 虚函数的限制.....	109	7.1.1 设备环境的属性.....	158
4.4.3 纯虚函数.....	109	7.1.2 获取设备环境.....	160
4.4.4 抽象类.....	110	7.1.3 映像模式.....	160
习题.....	111	7.2 绘图.....	162
第5章 VC++调试器.....	121	7.2.1 图形刷新.....	162
5.1 概述.....	121	7.2.2 绘图工具的应用.....	163
5.1.1 调试版本与发行版本.....	121	7.2.3 常用绘图函数.....	164
5.1.2 调试功能.....	122	7.3 绘图编程实例.....	166
5.2 VC++ 调试器界面.....	122	7.4 文本输出.....	173
5.2.1 Build 菜单.....	122	7.4.1 设置文本的设备环境.....	173
5.2.2 Debug 菜单.....	124	7.4.2 文本的输出过程.....	175
5.3 VC++ 调试器的应用.....	125	7.5 文本输出实例.....	177
5.3.1 建立应用程序的调试版本.....	125	习题.....	180

第 8 章 MFC 编程基础.....	181	10.2.3 创建和初始化列表框对象	247
8.1 MFC 类库简介	181	10.3 按钮类与列表框类的应用实例	247
8.2 使用 AppWizard 开发 MFC 应用程序... 185		习题	256
8.2.1 生成 MFC 应用程序框架	185	第 11 章 文档/视图结构	257
8.2.2 查看 AppWizard 生成的信息	190	11.1 概述	257
8.2.3 浏览项目信息与添加代码	193	11.1.1 文档/视图结构	257
8.3 MFC 应用程序框架与 Win32 程序		11.1.2 文档和视图的相互作用	257
的关联	195	11.1.3 SDI 和 MDI	258
8.3.1 MFC 应用程序框架对 Win32 程序		11.1.4 文档/视图结构的特点	259
的封装	196	11.2 文档类	259
8.3.2 MFC 应用程序的启动流程	200	11.2.1 文档类的结构及其方法	259
8.4 MFC 应用程序的消息映射	201	11.2.2 文档类的持续性	263
8.4.1 消息类别	201	11.3 视图类	267
8.4.2 消息映射原理	203	11.3.1 视图类的结构及其方法	267
8.4.3 消息处理过程	203	11.3.2 视图类的派生类	270
习题	204	11.4 文档/视图结构应用实例	271
第 9 章 基本控件使用(一).....	206	习题	276
9.1 编辑框类	206	第 12 章 设计用户界面	277
9.1.1 编辑框类的结构及其方法	206	12.1 菜单和加速键	277
9.1.2 编辑框类的应用实例	210	12.1.1 使用菜单	278
9.2 菜单类	216	12.1.2 定义加速键	283
9.2.1 菜单	217	12.2 工具栏	284
9.2.2 菜单类的结构及其方法	217	12.2.1 工具栏类的结构及其方法	284
9.2.3 菜单类的应用实例	220	12.2.2 创建工具栏类实例	290
9.3 滚动条类	226	12.3 状态栏	294
9.3.1 滚动条类的结构及其方法	227	12.3.1 状态栏的结构及其方法	294
9.3.2 创建与初始化滚动条	228	12.3.2 利用 AppWizard 自动创建	
9.3.3 滚动条类的应用实例	229	状态栏	296
习题	236	12.3.3 自定义状态栏实例	297
第 10 章 基本控件使用(二)	238	12.4 对话框栏	301
10.1 按钮类	238	12.4.1 对话框栏的结构及其方法	301
10.1.1 按钮类的结构	238	12.4.2 创建对话框栏实例	302
10.1.2 按钮类的方法	240	12.5 ReBar 栏	307
10.2 列表框类	240	12.5.1 ReBar 栏的结构及其方法	307
10.2.1 列表框类的结构	241	12.5.2 创建 ReBar 栏实例	308
10.2.2 列表框类的方法	244	12.6 通用对话框	309
		12.6.1 CFontDialog	309

12.6.2 CColorDialog	310	13.2.1 创建进程	317
12.6.3 CFileDialog	311	13.2.2 创建多线程	318
习题	311	13.3 数据库	319
第 13 章 高级应用	313	13.3.1 ODBC	319
13.1 动态链接库	313	13.3.2 DAO	320
13.1.1 静态链接库和动态链接库	313	习题	322
13.1.2 MFC 动态链接库	315	参考文献	323
13.2 多任务	317		

第1章

Visual C++开发环境

Visual C++简称 VC++。它拥有友好的界面和强大的功能，为用户提供了一个良好的 Windows 系统下的可视化开发环境。在该环境下，用户可以开发任何用 C++或 C 语言编写的程序，包括程序的建立、编辑、浏览、保存、打开、编译、链接、调试和优化等。更重要的是，使用 VC++提供的工具、向导以及 MFC 类库，可以在一个很短的时间里创建一个完整的 Windows 应用程序，并且获得大量的在线帮助。

本章主要介绍 VC++开发环境的组成和功能，目的是使读者认识、熟悉这一开发环境，以便充分利用它强大的功能，快速开发各种 Windows 应用程序。

本章将引入一些新的概念和知识，对于初学者来说，所介绍的部分内容可能暂时用不到，理解起来也会困难一些，开始阅读时可先跳过这些内容，等学到了后续的有关章节后，再回过头来阅读本章，可能会有更多的收获。

1.1 VC++用户界面

VC++用户界面通常被称为集成开发环境(IDE)，具有包括创建源码、编辑代码、编译、链接和调试等功能。图 1-1 是 VC++用户界面。

由图 1-1 可知，VC++用户界面包括：

- 菜单栏；
- 工具栏；
- 项目工作空间；
- 用于文件和资源编辑的主工作空间；
- 输出窗口；
- 状态栏。

另外，调试时，VC++还提供各种窗口，包括变量窗口、观察窗口、寄存器窗口、存储器窗口、调试堆栈窗口和反汇编窗口等。



图 1-1 VC++ 用户界面

1.1.1 菜单栏

VC++主菜单包括 File、Edit、View、Insert、Project、Debug、Tools、Windows 和 Help, 下面分别介绍。

1. File 菜单

在 VC++所提供的 File 菜单中, 大多数功能与 Windows 其他软件中的菜单项大同小异。所以, 这里我们重点介绍一些常用的、VC++特有的菜单命令。

1) 创建新文件

创建新文件由文件菜单中的 New 菜单项来完成。如图 1-2 所示是 New 对话框的 Files 选项卡。

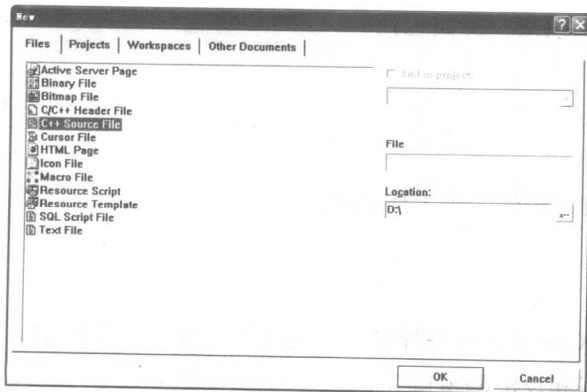


图 1-2 New 对话框的 Files 选项卡

其中,“C++ Source File”在编写 C++源程序时经常用到。

2) 创建新项目

创建新项目则由 New 对话框的 Projects 选项卡来实现,如图 1-3 所示。

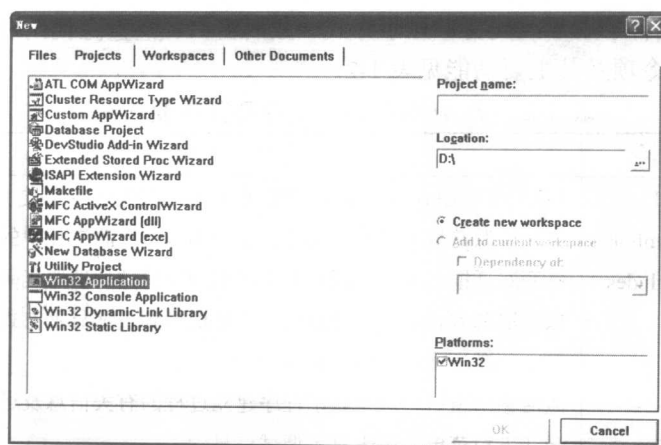


图 1-3 New 对话框的 Projects 选项卡

通常我们在编写 Win32 应用程序时就是选择 Projects 选项卡中的“Win32 Application”来实现的,而在使用 MFC 开发 Windows 应用程序时则要选择“MFC AppWizard(exe)”。

2. Edit 菜单

Edit 菜单中包含了文件编辑所需要的全部命令,各子命令项及其主要功能见表 1-1。

表 1-1 Edit 菜单的子命令及其主要功能

子 命 令	主 要 功 能
Undo	撤消上一次编辑操作
Redo	恢复被取消的编辑操作
Cut	将选定的文本从活动窗口剪切掉,移到剪贴板中
Copy	将活动窗口中被选定的文本复制到剪贴板中
Paste	将剪贴板中的内容粘贴到另一个指定文件或程序中
Delete	删除选定的对象或光标所在处的字符
Select All	一次性选定当前窗口中的全部内容
Find...	查找指定的字符串,可设置查找方向、区分英文大小写、整词查找等
Find in Files...	在多个文件中查找指定的字符串
Replace...	将文件中指定的字符串替换为另一指定的字符串
Go To	将光标转移到当前窗口的指定位置
Bookmarks...	给源文件设置和取消书签,书签的作用是在源文件中做标记
Advanced	用于文件编辑或修改的若干高级命令
Breakpoints	设置、删除和查看程序中的断点
List Members	列出当前所有的成员变量
Type Info	列出各种类型及其相关信息
Parameter Info	列出所使用的参数及其相关信息
CompleteWord	调入输入字体方式

其中, Breakpoints 命令用来在程序中设置、删除和查看断点。此后, 程序执行到指定位置就中断程序的执行。

3. View 菜单

View 菜单中包含了用于改变窗口显示方式、激活窗口、检查源代码和调试信息的各种命令选项。这些命令项及其主要功能见表 1-2。

表 1-2 View 菜单的子命令及其主要功能

子 命 令	主 要 功 能
Class Wizard	启动 MFC Class Wizard, 创建并编辑应用程序中的类
Resource Symbols	打开资源符号浏览器, 浏览、新增和编辑资源文件中的符号
Resource Includes	打开对话框, 修改资源符号文件名并编辑预处理器代码
Full Screen	以全屏方式显示当前窗口, 并可切换到正常显示方式
Workspace	显示并激活项目工作区窗口
Output	激活输出窗口, 用于显示程序建立过程的有关信息及运行结果
Debug Windows	激活下拉菜单, 用于显示调试信息
Refresh	刷新选定的内容
Properties	激活属性对话框, 用于设置或查看对象属性

下面是 View 菜单下一些子命令的有关说明。

1) Class Wizard 命令

Class Wizard 命令用于启动类向导 Class Wizard。类向导是一个适用于 MFC 应用程序的专用工具, 使用它可以完成以下功能:

(1) 创建新类。新类是从处理 Windows 消息和记录集(Recordset)的主框架基类中派生的。

(2) 建立消息映射函数。将消息映射给与窗口、对话框、控件、菜单选项和加速键有关的处理函数。

(3) 删除消息处理函数。

(4) 为控件引入或删除成员变量。该成员变量用于自动初始化、收集并验证输入到对话框、表单视图或记录视图中的数据。

(5) 添加自动化方法和属性。在创建新类时, 将自动化方法和属性添加到用户界面。

(6) 添加 ActiveX 事务处理。ActiveX 是一个经过打包的可重复使用的控件, 用 ClassWizard 可以添加 ActiveX 事务处理到应用程序中。

(7) 显示指定类的基本信息。

有关 ClassWizard 的具体使用在后面相关章节中将详细介绍。

2) Resource Symbols 命令

Resource Symbols 命令用于浏览、添加和编辑资源文件中的资源符号, 也可以删除未使用的资源符号, 还可以直接定位到使用资源符号的位置。选择 View 菜单下的 Resource Symbols 命令, 屏幕弹出 Resource Symbols 对话框, 在该对话框中可完成上述功能。

3) Resource Includes 命令

Resource Includes 命令用于打开 Resource 对话框, 完成修改资源符号文件名、包含的

头文件名以及编辑预处理器代码等功能。系统中的所有资源符号都将按默认方式保存在 resource.h 文件中。

4) Workspace 命令

Workspace 命令用来显示并激活项目工作区窗口，通常工作区中有 ClassView、ResourceView 和 FileView 三个选项卡。

5) Output 命令

Output 命令用于激活输出窗口，用于显示程序编译和链接过程中的有关信息及程序调试运行时的输出结果。

6) Debug Windows 命令

Debug Windows 命令可激活下一级子菜单，子菜单中包含了程序调试运行状态下可用的相关子命令，用于激活调试信息窗口，这些子菜单命令及其功能说明如下。

4. Insert 菜单

Insert 菜单中包含了用于插入类、表单、资源、文件、对象等命令，也可以复制资源。Insert 菜单的各命令项及其主要功能见表 1-3。

表 1-3 Insert 菜单的子命令及其主要功能

子 命 令	主 要 功 能
New Class	激活 New Class 对话框，创建新的类并添加到项目中
New Form	激活 New Form 对话框，创建新的表单并添加到项目中
Resource	激活 Insert Resource 对话框，创建新的资源或插入资源到资源文件中
Resource Copy	复制选定的资源
New ATL Object	激活 ATL Object Wizard 对话框，选定新的 ATL 对象并插入到当前项目中

5. Project 菜单

Project 菜单中包含了用于管理项目和项目工作区的所有命令。Project 菜单中的主要子命令项及其主要功能见表 1-4。

表 1-4 Project 菜单的子命令及其主要功能

子 命 令	主 要 功 能
Set Active Project	选择指定项目为工作区中的活动项目
Add to Project	激活下级菜单，完成添加文件、文件夹、数据链接及可重用部件到项目中
Source Control	用于编辑项目的从属关系
Settings	设置项目的各种属性
Export Makefile	以外部 Make 文件格式创建执行文件
Insert Project into Workspace	插入已有项目到当前工作区中

6. Build 菜单

Build 菜单中包含了用于编译、建立和运行应用程序所需要的所有命令。Build 菜单中的各命令项及其主要功能见表 1-5。

表 1-5 Build 菜单的子命令及其主要功能

子 命 令	主 要 功 能
Compile	编译当前项目中所有的 C、C++ 源代码文件或资源文件
Build	查看当前项目中的所有文件，并对最近修改过的文件进行编译和链接
Rebuild All	忽略以前的编译和链接工作，重新编译和链接整个项目文件
Batch Build...	成批编译和链接，即一次编译和链接多个项目文件
Clean	清除当前项目的中间文件和输出文件
Start Debug	启动程序调试器，用于跟踪程序的调试和执行
Debugger Remote Connection...	对远程调试链接设置进行编辑
Execute	运行程序
Set Active Configuration...	选择当前项目的配置，例如 Win32 Release 和 Win32 Debug
Configurations...	编辑项目配置
Profile...	启动剖析器，用于检查程序的运行行为

7. Tools 菜单

Tools 菜单命令主要用于选择或定制集成开发环境中的一些实用工具，包括浏览程序符号、激活常用工具、更改选项设置、定制菜单与工具栏等。另外，它还提供了定制 Tools 菜单本身的命令，主要用于在工具菜单中添加更多的命令项。表 1-6 列出了 Tools 菜单中的主要子命令及其主要功能。

表 1-6 Tools 菜单的子命令及其主要功能

子 命 令	主 要 功 能
Source Browser	浏览与程序中所有符号(类、函数、数据、宏和类型)有关的信息
Visual Component Manager	弹出 VCM、VBD 窗口
Register Control	将 OLE 控件注册到操作系统中
Error Lookup	检查大多数 Win32 API 函数返回的标准错误代码信息
ActiveX Control Test Container	为测试 ActiveX 控件提供一个环境
OLE/COM Object Viewer	提供了安装在系统中的所有 OLE 以及 ActiveX 对象的信息
Spy++	激活 Spy++，用于给出系统的进程、线程、窗口和窗口消息的图形表示
MFC Tracer	程序执行或调试时，用于激活各种级别的调试消息，并由 MFC 将这些消息发送到输出窗口
Customize	激活定制对话框，可对命令、工具栏、工具菜单和键盘加速键进行定制
Options	激活选择对话框，以便对完成环境设置(如窗口设置、调试器设置、兼容性设置、目录设置、工作区设置、宏设置、帮助系统设置、格式设置等)进行更改
Macro	创建和编辑宏文件
Record Quick Macro	开始进行宏的录制
Play Quick Macro	执行已经录制的宏

8. Windows 菜单

Windows 菜单命令主要用于控制集成开发环境中各种窗口的属性,包括排列窗口、打开或关闭窗口、使窗口分离或重组等操作。表 1-7 列出了 Windows 菜单命令中的子命令及其主要功能。

表 1-7 Windows 菜单的子命令及其主要功能

子命令	主要功能
New Windows	打开新窗口,以便在多个窗口中显示当前文件的内容
Split	将文档窗口拆分为多个面板,以便同时查看同一文档的不同内容
Docking View	打开或关闭窗口的船坞化特征(船坞化窗口总是附属于应用程序窗口的下边界,也可浮动于屏幕上的任何位置)
Close	关闭选定的活动窗口
Close All	关闭所有打开的窗口
Next	激活上一个文档窗口
Previous	激活下一个文档窗口
Cascade	将当前所有打开的窗口在屏幕上重叠摆放
Tile Horizontally	将当前所有打开的窗口在屏幕上横向平铺摆放
Tile Vertically	将当前所有打开的窗口在屏幕上纵向平铺摆放

9. Help 菜单

VC++ 提供了大量详细的帮助信息,通过 Help 菜单,用户可以得到所需要的帮助信息。而且 Microsoft 采用因特网 Web 浏览技术,将帮助文件作为超文本文档(HTML)在 WWW 上发布。

使用 VC++ 帮助系统时,单击 Help 菜单下的 Contents(帮助目录)、Search(搜索)或 Index(索引)子命令,将打开帮助窗口。按照帮助窗口的导航,即可得到帮助。

1.1.2 工具栏

VC++ 集成开发环境提供了十多种工具栏,每个工具栏由一组工具栏按钮组成,这些工具栏按钮分别对应一些菜单命令的功能,使用它们可大大方便用户的操作。需要说明的是,有些工具栏只在一定的状态下才被激活,对于这些工具栏的功能和使用将在相关的章节中介绍。这里仅对集成开发环境中几个常见的工具栏做简要介绍。

1. Standard 工具栏

图 1-4 所示的是 VC++ 集成开发环境提供的 Standard(标准)工具栏,其中包含了 15 个按钮,与其他环境下的标准工具栏大同小异,这里不再赘述。

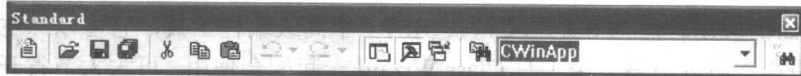


图 1-4 Standard 工具栏

2. Build 工具栏

图 1-5 所示的是 Build 工具栏,其中包含了六个按钮,它们的功能如图所示。

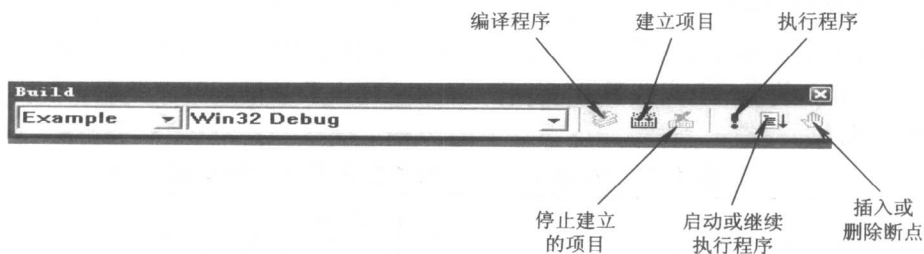


图 1-5 Build 工具栏

3. WizardBar 工具栏

VC++ 提供了两个功能强大的向导工具，一个是应用程序向导(AppWizard)，一个是类向导(ClassWizard)。这是两个重要的软件开发工具，它们的详细功能和使用将在相关章节中介绍，这里仅说明向导工具栏(WizardBar)。

类向导允许用户在源文件中添加自己的类，为类的函数成员映射消息。WizardBar 是配合类向导操作的工具栏，它包括了“WizardBar C++ Class”、“WizardBar C++ Filter”和“WizardBar C++ Member”。其在屏幕上的显示如图 1-6 所示。



图 1-6 WizardBar 工具栏

由图 1-6 可见，向导工具栏包含了三个下拉列表框和一个按钮，它们分别用于显示用户选择的类、消息标识和函数生成方式，一个按钮可激活一个下拉菜单，用来实现 ClassWizard 各文件之间的互联查询。通过 WizardBar 工具栏的使用，使类的设计和维护实现了自动化，方便了程序的编辑和调试。

WizardBar 的主要功能包括：

- (1) 增加一个新类；
- (2) 建立一个新的函数或方法；
- (3) 跳到一个已存在的函数或方法。

1.2 AppWizard 工具

1.2.1 AppWizard 简介

AppWizard 应用程序向导是 VC++ 提供的一个高级编程工具，它可以产生应用程序的 C++ 源代码框架。通过与另一个工具 Class Wizard 一起配合使用，可大大节省开发应用程序的时间和精力。

AppWizard 是一个标准的 C++ 源代码生成器。它通过一系列的对话框来提示用户输入所需创建的程序的信息，如它的名字和位置。用户还可以指定它是否具有某些特性，如是

否多文档界面, 是否带工具栏, 是否支持数据库、OLE 等。设置完成后, AppWizard 生成一些构成应用程序框架的文件。

由 AppWizard 生成的应用程序是一个基本的 Windows 应用程序, 用户可以对它进行编译并运行。但是, 这个程序并不能完成任何工作, 只是为用户继续增加代码而提供了一些框架性的代码, 目的是节省用户设计应用程序框架的时间和精力。接下来, 用户所要做的工作只是直接往框架中添加自己的处理代码。

1.2.2 使用 AppWizard 创建 C++源程序

使用 AppWizard 创建一个 C++源程序的步骤如下:

- (1) 在如图 1-2 所示的 New 对话框中选择 Files 选项卡。
- (2) 从文件列表中选择文件类型 C++ Source File。
- (3) 在文件框中, 输入新文件名 Myprg。
- (4) 在目录框中, 指定文件的存放目录, 可以直接输入路径名, 也可以单击旁边的“...”按钮, 浏览选择一个路径。
- (5) 单击确定按钮, 进入 C++源程序编辑状态, 如图 1-7 所示。在屏幕右边的编辑区域输入程序代码后, 即可编译、链接、运行源程序。

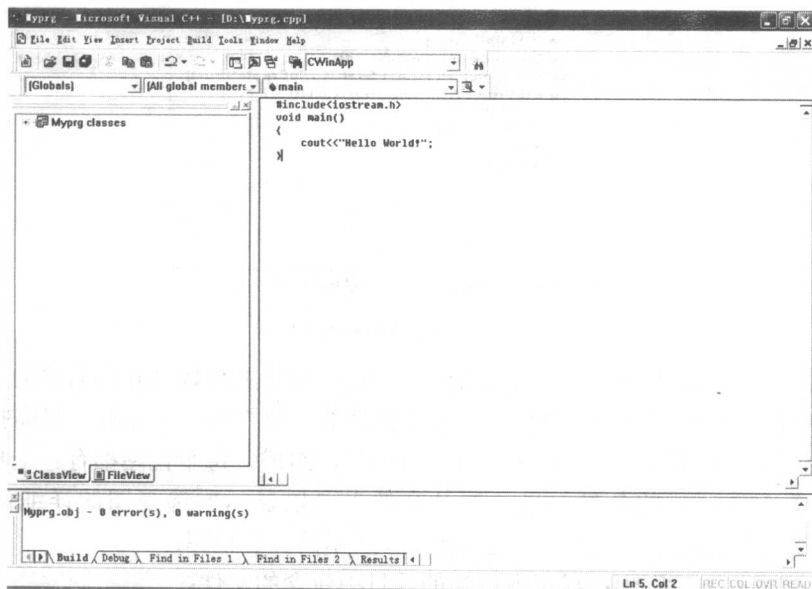


图 1-7 C++源程序编辑屏幕

1.3 项目和项目工作区

AppWizard 生成的应用程序框架是通过项目工作区来管理的。项目工作区是一个包含用户的所有相关项目和配置的实体。项目则是与应用程序相关的一组文件及其配置, 用以