

Visual C++ 程序设计

北京科海 总策划

刘荷花 陈信明 主 编

- 从实际应用的角度出发，帮助读者以最快的速度进入 Visual C++ 的世界，提高程序开发技术水平
- 资深讲师精心编写，融入作者多年的教学经验
- 结构清晰、实例丰富，具有很强的操作性和实用性

1CD 大型多媒体教学系统



- 全程语音讲解的多媒体学习环境
- 超大容量，播放时间长达168分钟



吉林电子出版社

北京科海技术图书

www.khp.com.cn

高等院校规划教材·计算机系列

Visual C++程序设计

刘荷花 陈信明 主 编

吉林电子出版社

北京科海培中技术有限责任公司

内 容 简 介

本书详细讲述了使用 Visual C++ 6.0 进行程序开发的过程。每章所讲知识点均配有大量的实例，实例相互关联，逐层递进。

全书分为 3 个部分共 9 章，第 1 部分为第 1~3 章，讲述 Visual C++ 基础知识，包括 Visual C++ 集成开发环境简介、C++ 语言基础知识和 Windows 程序设计基础知识；第 2 部分为第 4~8 章，讲述 MFC 程序设计，包括 MFC 基础知识、MFC 文档/视图结构、MFC 中的对话框和控件以及 MFC 的图形操作等；第 3 部分为第 9 章，着重讲述了 ActiveX 控件的知识，包括控件的制作和使用等。

此外，为了读者能更好的学习并掌握 Visual C++ 程序设计的方法与开发过程，我们在光盘中补充了大量基于本书内容之上的拓展演示视频，以期使读者能够在本书内容基础上学习到更多的知识，且提升实际应用技能。

本书结构严谨、通俗易懂，兼有普及与提高的双重功能，即可作为高等院校、计算机培训学校相关课程的教材，也可作为程序设计开发人员的参考用书。

编 者：刘荷花 陈信明

出 版 人：王保华

责任编辑：孟迎红 汤 洁 李晶璞

封面设计：林 陶

出版发行：吉林电子出版社

地 址：长春市人民大街 4646 号（邮编：130021）

印 刷：北京市艺辉印刷有限公司

开 本：787×1092 1/16

字 数：432 千字

印 张：17.75

版 次：2009 年 6 月第 1 版

印 次：2015 年 1 月第 2 次印刷

版 号：ISBN 978-7-89454-050-8

定 价：28.50 元（1 多媒体教学 CD+1 配套手册）

前言

作为一种计算机语言，C++语言有着无可比拟的优势。而 Visual C++ 6.0 是迄今为止功能最强大的 C++语言集成开发环境之一。为此，掌握 Visual C++的程序设计方法和内容，对提高学习者的计算机应用软件开发能力有着极其重要的意义。

本书简明易懂，对重要知识点采用实例进行解析，使读者快速入门，特别适合程序设计初学者使用。全书共分 9 章，前 3 章主要介绍基础知识，第 4~8 章讲解 MFC 程序设计的知识，第 9 章为 ActiveX 编程技术。各章具体内容如下：

第 1 章主要介绍了 Visual C++ 6.0 集成开发环境的界面及应用程序框架的创建过程。

第 2 章主要介绍了 C++语言和面向对象的基本知识。

第 3 章主要介绍了 Windows 编程基础及其系统中变量的命令规则。

第 4 章主要介绍了 MFC 的类库及使用该类库创建 Windows 应用程序的一般过程。

第 5 章主要介绍了 MFC 的文档/视图结构，并使用该结构创建了一个鼠标画线程序。

第 6 章主要介绍了 MFC 的菜单、快捷键、工具条和状态栏等资源，并为鼠标画线程序添加了这些资源。

第 7 章主要介绍了 MFC 的对话框与通用控件的知识，进一步完善鼠标画线程序，使其可以通过在对话框内输入两点来画线。

第 8 章主要介绍了 MFC 的图形和文本绘制的知识，最终完成了鼠标画线程序，使其可以绘制各种线型、各种颜色的线段。

第 9 章主要介绍了 ActiveX 控件的知识，制作并使用了一个 ActiveX 控件。

本书对每章的知识点都详细讲解，并配有大量的图片和实例，便于读者理解。本书各章实例相互联系，逐层递进，最终完成了一个较完善的鼠标画线程序。通过该实例的实现可以轻松掌握各知识点，并把这些知识点综合应用。

此外，为了读者能更好的学习并掌握 Visual C++程序设计的方法与开发过程，我们在光盘中补充了大量基于本书内容之上的拓展演示视频，以期使读者能够在本书内容基础上学习到更多的知识，且提升实际应用技能。

由于时间仓促与编者水平有限，不足与欠妥之处在所难免，恳请广大读者不吝指正。

编者

2009 年 5 月

目 录

第1章 初识Visual C++	1
1.1 Visual C++ 6.0概述	1
1.1.1 C语言发展史简介	1
1.1.2 C++语言发展史简介	2
1.1.3 Visual C++的诞生	2
1.1.4 Visual C++的功能与特点	2
1.2 Visual C++ 6.0开发环境	3
1.2.1 启动Visual C++ 6.0	3
1.2.2 Visual C++ 6.0开发环境简介	4
1.3 Visual C++ 6.0入门实例	13
1.3.1 创建一个Win32 Console Application程序	14
1.3.2 创建一个MFC多文档程序	18
1.4 小结与提高	23
1.5 思考与练习	23
第2章 C++语言基础知识	25
2.1 一个简单的C++程序实例	25
2.2 C++的C语言基础	26
2.2.1 C++语言的数据类型	26
2.2.2 C++语言的标识符与关键字	27
2.2.3 C++语言的变量和常量	28
2.2.4 C++语言的运算符	30
2.2.5 C++语言的3种程序结构	33
2.2.6 C++语言的函数	36
2.2.7 C++语言的数组、指针和引用	40
2.3 类与对象	44
2.3.1 类	44
2.3.2 对象	45
2.3.3 构造函数和析构函数	47
2.3.4 函数重载	49
2.3.5 this指针	49
2.4 类的继承与派生	50
2.4.1 派生类	50

2.4.2 多重继承	54
2.5 多态性与虚函数	57
2.6 小结与提高	58
2.7 思考与练习	59
第3章 Windows编程基础	60
3.1 从DOS程序到Windows程序	60
3.1.1 DOS程序实例	60
3.1.2 Windows程序实例	61
3.1.3 初步解析Windows程序实例	62
3.2 窗口和消息	63
3.2.1 窗口实例	64
3.2.2 注册窗口类	66
3.2.3 创建窗口	68
3.2.4 显示窗口	69
3.2.5 消息循环	69
3.2.6 窗口消息处理函数	71
3.2.7 消息处理	71
3.3 Windows系统中常用的消息	74
3.3.1 消息分类	75
3.3.2 常用消息简介	75
3.4 Windows系统中变量的命名规则	76
3.4.1 Windows数据类型	76
3.4.2 匈牙利法则	77
3.5 小结与提高	78
3.6 思考与练习	78
第4章 MFC基础知识	80
4.1 MFC概述	80
4.1.1 MFC的历史沿革	80
4.1.2 MFC的优点	82
4.2 MFC基础类库简介	82
4.2.1 MFC类库概念	82
4.2.2 MFC基础类库层次结构	83
4.3 使用MFC类库创建框架窗口	86
4.3.1 MFC方法创建一个HelloWnd程序	86
4.3.2 HelloWnd程序解析	89
4.4 MFC中的消息映射简介	94
4.4.1 不使用消息映射的程序实例	94
4.4.2 创建消息映射程序实例	96

4.4.3 使用消息映射改变框架窗口大小	98
4.4.4 消息映射与虚函数	103
4.5 小结与提高	104
4.6 思考与练习	104
第5章 MFC文档/视图结构	106
5.1 文档/视图结构概述	106
5.1.1 文档/视图结构的概念	106
5.1.2 文档/视图结构程序的分类	107
5.1.3 文档/视图结构的意义	108
5.2 简析文档/视图结构运行机制	109
5.2.1 创建SDI应用程序实例	109
5.2.2 创建SDI框架窗口	111
5.2.3 文档类和视图类	118
5.3 文档/视图结构程序实例	119
5.3.1 屏幕画线程序实例	119
5.3.2 鼠标画线程序实例	121
5.3.3 进一步完善鼠标画线程序实例	128
5.3.4 文档类程序实例	135
5.4 小结与提高	141
5.5 思考与练习	141
第6章 MFC中的资源	143
6.1 资源简介	143
6.1.1 资源符号	143
6.1.2 菜单资源	144
6.1.3 快捷键资源	148
6.1.4 字符串表资源	149
6.1.5 工具条资源	150
6.1.6 状态栏资源	153
6.2 资源应用实例	156
6.2.1 添加菜单资源	156
6.2.2 修改菜单项状态	162
6.2.3 添加工具条	164
6.2.4 添加快捷键	167
6.2.5 修改状态栏	168
6.3 小结与提高	171
6.4 思考与练习	171
第7章 MFC中的对话框与控件	173
7.1 对话框与控件概述	173

7.1.1	对话框的基本概念	173
7.1.2	控件的基本概念	174
7.2	对话框和控件运行机制	175
7.2.1	生成基于对话框的Windows应用程序	175
7.2.2	使用对话框编辑器和ClassWizard	178
7.2.3	基于对话框的应用程序的运行机制	183
7.2.4	对话框中控件的消息映射机制	185
7.3	常用控件	186
7.3.1	按钮控件	186
7.3.2	编辑框控件	187
7.3.3	静态控件	188
7.3.4	列表框控件	192
7.3.5	组合框控件	195
7.3.6	进度条控件	199
7.3.7	滑块控件	199
7.4	在SDI中使用对话框	203
7.5	小结与提高	211
7.6	思考与练习	211
第8章	MFC图形操作	213
8.1	设备上下文	213
8.1.1	图形设备接口	213
8.1.2	设备上下文概述	214
8.1.3	CDC类	215
8.1.4	CDC类的常用函数	216
8.2	图形绘制	218
8.2.1	绘图工具	218
8.2.2	与绘图相关的类	221
8.2.3	常用的绘图函数	223
8.2.4	绘图实例	225
8.3	字体	230
8.3.1	字体类	230
8.3.2	绘制文本	230
8.3.3	字体的应用实例	232
8.4	使用画笔修改鼠标画线实例	233
8.4.1	新建对话框	233
8.4.2	创建对话框类	234
8.4.3	修改视图类	238
8.4.4	运行程序	240

8.5	小结与提高	240
8.6	思考与练习	241
第9章	ActiveX控件	242
9.1	ActiveX简介	242
9.1.1	基于文档	242
9.1.2	链接	243
9.1.3	嵌入	243
9.1.4	容器与服务器	244
9.1.5	ActiveX对象	244
9.2	COM模型	244
9.2.1	COM通信	244
9.2.2	IUnknown接口	245
9.3	ActiveX控件	245
9.3.1	ActiveX自动化	245
9.3.2	ActiveX控件简介	247
9.4	ActiveX应用程序实例	247
9.4.1	ActiveX容器应用程序实例	247
9.4.2	使用已注册的ActiveX控件创建应用程序实例	250
9.4.3	创建ActiveX控件	255
9.4.4	使用自己创建的ActiveX控件	266
9.5	小结与提高	268
9.6	思考与练习	268
主要参考文献		270

第 1 章

初识 Visual C++

可视化与面向对象程序设计为当今公认的主流编程技术，而在众多面向对象程序设计语言中，C++语言得到了最为广泛的应用。本书将要讲解的 Visual C++是由 Microsoft 公司开发的，基于 Windows 操作系统平台的 C++语言可视化编程工具。

Visual C++在同为 Microsoft 公司产品的 Windows 操作系统平台强有力的支持下，其执行速度和对操作系统的访问权限之高，是许多其他编程工具无法比拟的。由此，Visual C++成为众多程序设计初学者与程序员在 Windows 操作系统平台下编程的首选工具。



本章主要内容

- Visual C++ 6.0 概述
- Visual C++ 6.0 开发环境简介
- 利用 Visual C++ 6.0 创建一个 Win32 Console Application 程序
- 利用 Visual C++ 6.0 创建一个 MFC 多文档程序

1.1 Visual C++ 6.0 概述

Visual C++ 6.0 是运行于 Windows 操作系统平台的交互式可视化集成开发环境，它将程序代码的编辑、编译、连接和调试等功能集于一体，为编程人员提供了一个完整而便捷的开发环境，并提供了许多有效的协助开发工具。无论是程序设计的初学者，还是有经验的程序员，都可以使用 Visual C++ 6.0 迅速地开发出功能强大、令人满意的应用程序。

1.1.1 C 语言发展史简介

C++语言的前身 C 语言问世于 20 世纪 70 年代初。1978 年美国电话电报公司 (AT&T) 贝尔实验室正式发布了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《The C Programming Language》一书，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》一书中并没有定义一个完整的 C 语言标准，后来由美国国家标准学会在此基础上制定了一个 C 语言标准，并于 1983 年发表，通常称之为 ANSI C。

早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人

们认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的应用,成为当代最优秀的程序设计语言之一。

C 语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于 C 语言实现了对硬件的编程操作,因此 C 语言集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也可用于应用软件的开发。此外,C 语言还具有效率高、可知性强等特点,因此被广泛地移植到了各种类型的计算机上,从而形成了多种版本的 C 语言。

1.1.2 C++语言发展史简介

在 C 语言的使用过程中,人们逐渐发现了 C 语言的一些缺陷,如在大规模软件开发过程中的程序结构复杂性的控制、代码的重用以及与日益广泛使用的面向对象分析方法的一致性等。为了解决这些难题,1983 年又由贝尔实验室的 Bjarne Stroustrup 进一步扩充和完善了 C 语言,使其包含了“类”的结构,特别是它提供了面向对象的程序设计功能,这就形成了 C++语言,最初称为“带类的 C”。

C++语言的开发,使软件开发领域发生了巨大的变革,使软件开发的周期缩短了,程序的正确性提高了,开发成本降低了,为此,C++语言得到了推广使用。Bjarne Stroustrup 在其所著的《The C++ Programming Language》一书中,对 C++语言做了全面的介绍,特别是在该书的第三版,作者对如何提出 C++语言中的一些思想做了介绍,对深刻了解 C++语言的内涵有极大的帮助。

1.1.3 Visual C++的诞生

在 Windows 3.0 之前,最好的 C/C++编程工具是 Borland 公司的 Turbo C/C++系列,但是随着 Windows 3.0 的推广,Microsoft 公司推出了 Microsoft C/C++ 7.0,首次采用了微软基础类库(Microsoft Foundation Class Library, MFC)。正是由于 MFC 的出现,让 C++程序员的工作变得更加高效,也使得今天的 Visual C++得以成为优秀的主流编程工具。此时,由于 Visual Basic 在 Windows 平台上的成功,Microsoft 公司决意把 Microsoft C/C++的下一个版本 Microsoft C/C++ 8.0 改名为 Visual C++,这就是 Visual C++ 1.0。其后的各个 Visual C++的版本,如同 Microsoft 公司的其他工具软件,随 Windows 平台的变化而又有不同的改变。随着 Windows 98/2000 操作系统功能的增强,相对于前一版本,Visual C++ 6.0 增加了支持 IE 4.0 的控件和类,增强了对 OLE DB 的支持,完善了 DHTML,提供了独立的 MSDN 帮助系统等,同时,集成环境内部工具的功能也得到了相应的提升。

1.1.4 Visual C++的功能与特点

与和 Microsoft 公司的另一编程工具 Visual Basic 不同,Visual C++并不是一个完全可视化的编程工具。虽然 Visual C++的集成开发环境提供了一个快速编程的框架,极大地提

高了编程效率，但是要真正掌握 Visual C++，还必须对 C/C++ 语言编程有深入的了解，理解 MFC 库和 Windows 下的编程方法。

在 Windows 下编程，通常是通过调用 Windows 应用程序接口（Application Program Interface, API）得以实现的。Visual C++ 6.0 将大量的 Windows API 以 C++ 格式进行封装，通过 MFC 方式提供给开发人员，大大简化了开发人员的编程工作，提高了工作效率。因此，要学好 Visual C++，掌握 MFC 是关键。

在 Visual C++ 6.0 中，可以使用 MFC 完成大多数的工作，也可以直接调用 Windows API 完成一些底层的开发。用 Visual C++ 6.0 开发的程序代码简练，运行速度快。

在 Visual C++ 6.0 中，提供了两个功能强大的编程工具：AppWizard（应用程序向导）和 ClassWizard（类向导）。

利用 Visual C++ 6.0 的 AppWizard 可以在很短的时间内创建出 Windows 应用程序的框架；而 ClassWizard，则可以在应用程序框架上迅速增加新的类、成员变量与函数，使 Windows 编程得以快速实现，两者结合，让 Windows 下的编程变得非常简单。

1.2 Visual C++ 6.0 开发环境

开发环境是程序员同 Visual C++ 交互的界面，程序员通过该界面可以进行创建工程文件，创建或访问 C++ 源代码、资源文件等操作。本节简要介绍 Visual C++ 6.0 的用户界面，并对各种常用的窗口、菜单、按钮的意义和功能有重点地进行讲解。

1.2.1 启动 Visual C++ 6.0

在 Visual C++ 6.0 软件安装完成之后，Visual C++ 6.0 会被自动添加到 Windows “开始”菜单“所有程序”组中的 Microsoft Visual C++ 6.0 文件夹之下，如图 1-1 所示。执行“开始”→“所有程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令，启动 Visual C++ 6.0，可以进入 Visual C++ 6.0 的开发环境。

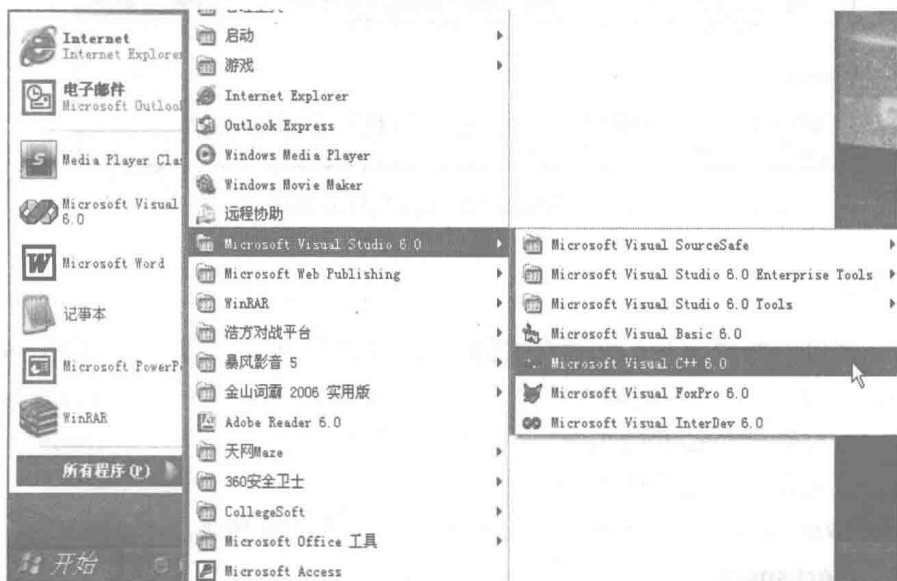


图 1-1 启动 Microsoft Visual C++ 6.0

此外,还可以在“开始”菜单中 Microsoft Visual C++ 6.0 上右击,执行“发送到”→“桌面快捷方式”命令,如图 1-2 所示,在桌面上建立 Microsoft Visual C++ 6.0 的快捷方式,然后通过双击 Microsoft Visual C++ 6.0 桌面快捷方式打开 Visual C++ 6.0。

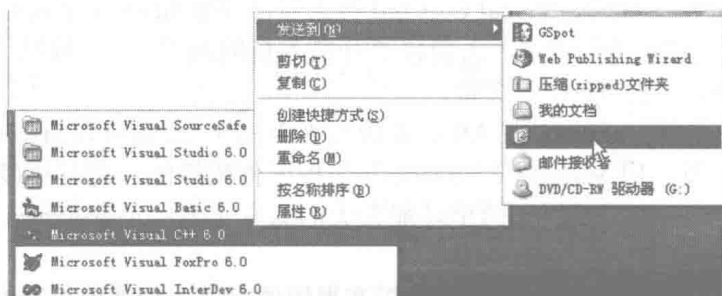


图 1-2 创建 Visual C++ 6.0 快捷方式

1.2.2 Visual C++ 6.0 开发环境简介

Visual C++ 6.0 的开发环境如图 1-3 所示,由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态栏组成,下面将分别对它们进行重点介绍。

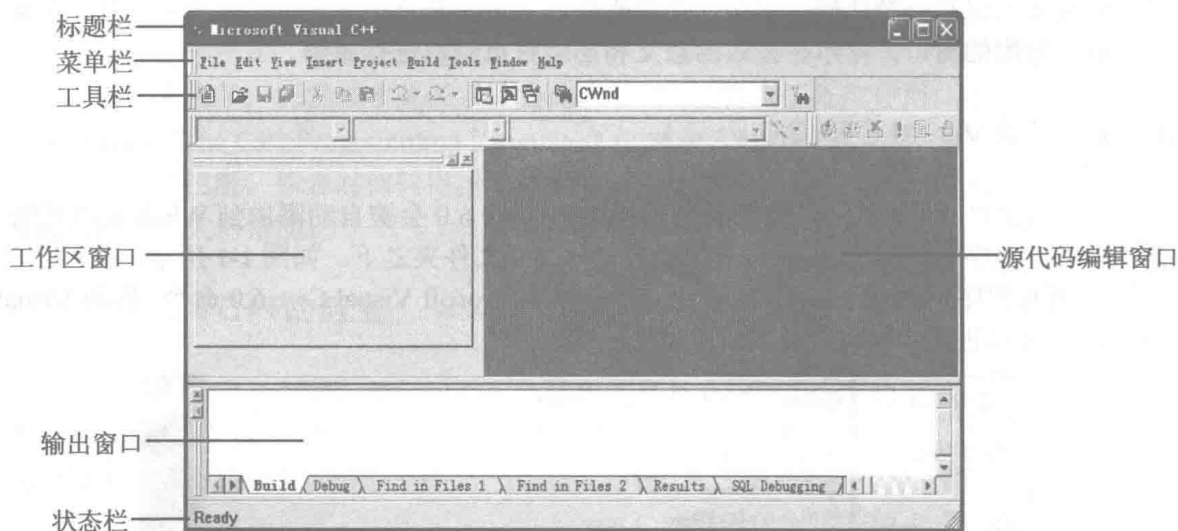


图 1-3 Visual C++ 6.0 集成开发环境

1. 菜单栏

(1) File 菜单中的内容是与文件有关的打开、关闭、打印等操作,如图 1-4 所示。

- **New:** 新建一个工程或文件。
- **Open:** 打开一个已有的文件。
- **Close:** 关闭一个已有的文件。
- **Open Workspace:** 打开一个已有的工作区(工程文件)。
- **Save Workspace:** 保存工作区。
- **Close Workspace:** 关闭工作区。

(2) Edit 菜单中包含了用户需要的大部分编辑操作，如剪切、复制、粘贴、查询等，如图 1-5 所示。其子菜单随着当前正在编辑的文件类型不同而变化。



图 1-4 File 菜单

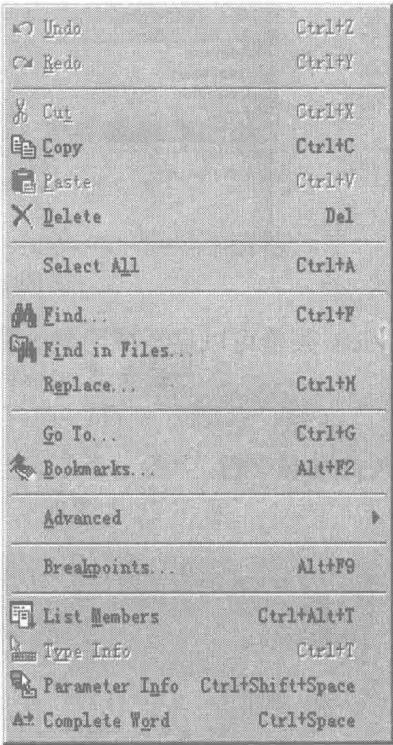


图 1-5 Edit 菜单

- **Find:** 用于打开 Find 对话框，如图 1-6 所示，该对话框为用户在众多工程文件中查找提供了方便。

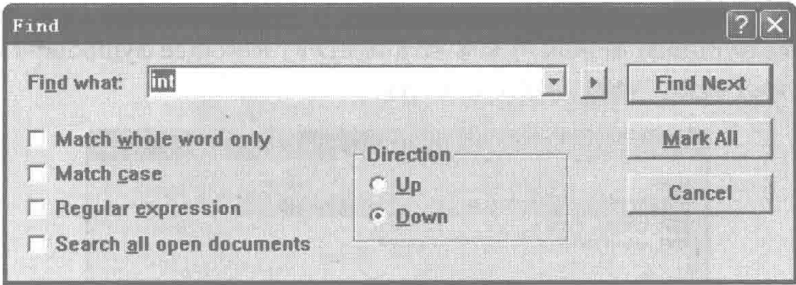


图 1-6 Find 对话框

- **Go To:** 打开 Go To 对话框，如图 1-7 所示，该对话框用于将当前编辑行跳到指定的行号、地址、引用、书签以及其他选项。
- **Bookmarks:** 用于管理文本文件中的书签（不包含临时书签）。
- **Advanced:** 此命令中包含许多子菜单，如图 1-8 所示，用来查找、编辑文本等。
 - Incremental Search 命令比 Find 命令查询得更快。
 - Format Selection 命令类似于格式刷，使用当前选中的代码以相同的规则自动缩进。
 - Tabify Selection/Untabify Selection 命令用于空格键和 Tab 键的转换。

- **Breakpoints:** 用于设置断点，结合 Go 命令使用，以观察代码的运行状态。

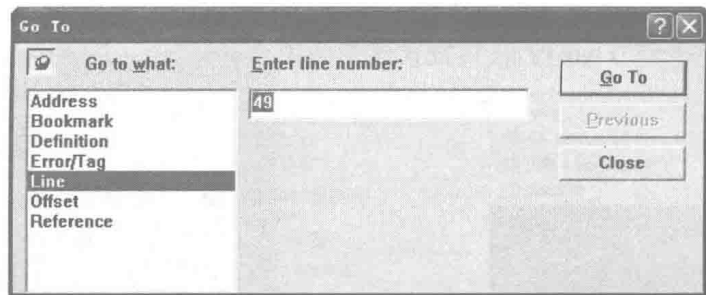


图 1-7 Go To 对话框

(3) View 菜单可以显示或隐藏一些窗口，如图 1-9 所示。

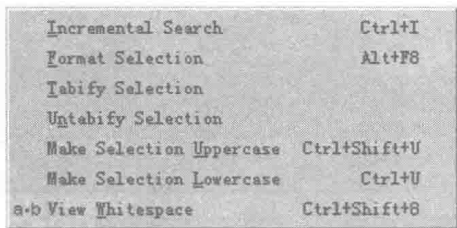


图 1-8 Advanced 菜单

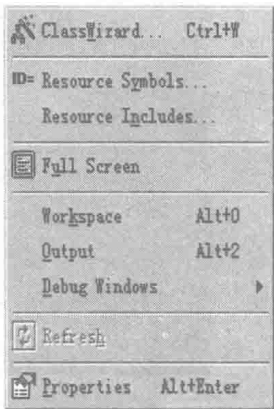


图 1-9 View 菜单

- **ClassWizard:** 可以调出 ClassWizard 对话框，本书后面章节将对该对话框的强大功能进行详细介绍。
- **Resource Symbols:** 将调出用来管理资源 ID 的 Resource Symbols 对话框，如图 1-10 所示，可以创建、修改或删除资源 ID。

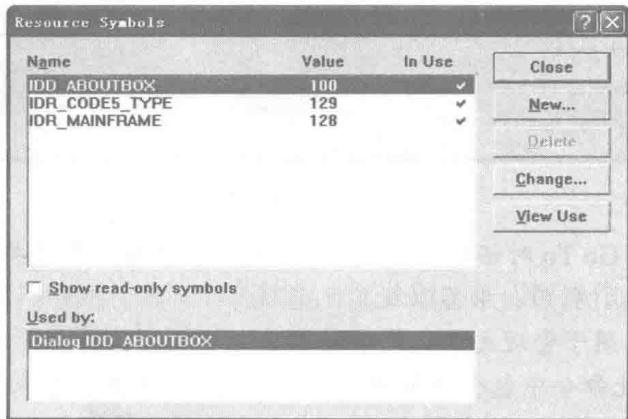


图 1-10 Resource Symbols 对话框

- **Properties:** 可以调出 Class Properties 对话框，如图 1-11 所示，该对话框显示了当前选中对应对象的详细信息。

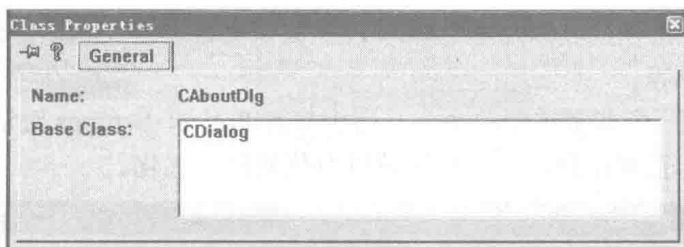


图 1-11 Class Properties 对话框

(4) Insert 菜单用于创建类、资源以及 ATL 对象等，如图 1-12 所示。

- **New Class:** 创建一个新的类并添加到工程中。
- **New Form:** 创建新的表单并添加到工程中。
- **Resource:** 插入快捷键、位图、鼠标以及对话框等资源。
- **File As Text:** 新建一个文本文件。
- **New ATL Object:** 启动 ATL Object Wizard 对话框，添加 ATL 对象到工程中。

(5) Project 菜单拥有与项目维护相关的菜单项，如图 1-13 所示。

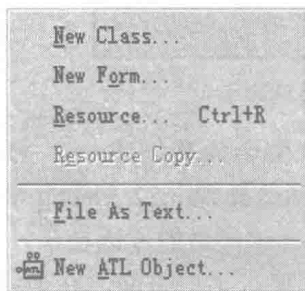


图 1-12 Insert 菜单

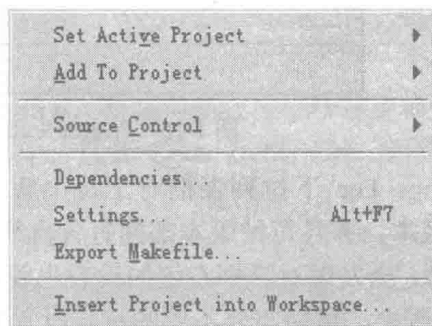


图 1-13 Project 菜单

① Add to Project: 该命令可以将以下内容添加到工程中，如图 1-14 所示。

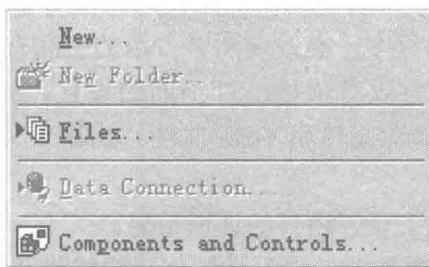


图 1-14 Add to Project 级联菜单

New 命令用于新建文档，其作用与执行 File→New 命令相同。

New Folder 命令用于添加一个新的文件夹。

Files 命令用于将文件添加到当前项目中。

Data Connection 命令用于将项目与数据源相连。

Components and Controls 命令用于向项目中添加组件。这是一个非常简单并且实用的编程渠道。使用数据库时，可以用其来添加 ADO 组件；使用 MapX 等软件做一个地理信息系统时，也可以用其在对话框中添加一个控件以导入地图；在读取或者创建一个 X 文件

来渲染 3D 模型时，可以用其添加 GUID 管理工具来创建模板；此外，还可以用其来添加启动动画、右键菜单等。

② Settings: 是一个非常重要的命令，用于打开 Project Settings 对话框，如图 1-15 所示，它对工程进行一系列设置，这些设置可以对代码进行优化。

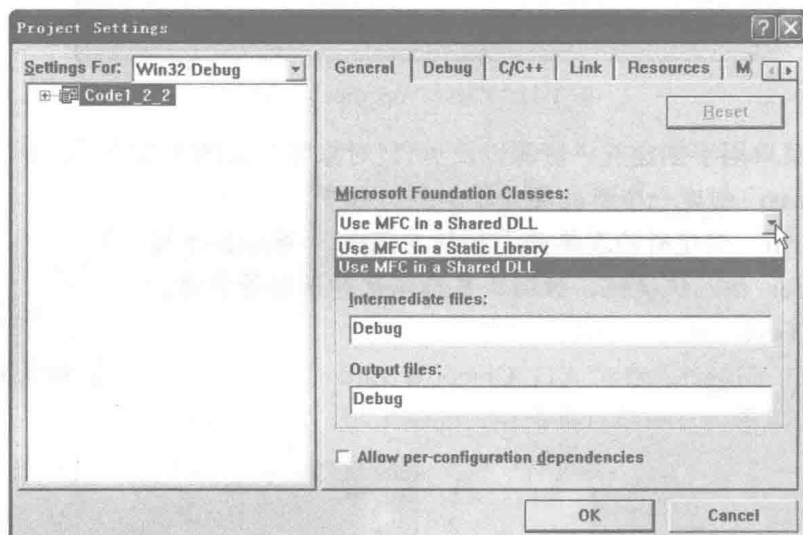


图 1-15 Project Settings 对话框的 General 选项卡

Settings For 下拉列表框对于每个新工程，Visual C++都会为其创建一个 Debug 和 Release 版本，并且每种版本都拥有各自的设置。当用户分别选择 Win32 Debug 和 Win32 Release 时，会发现对话框右侧的设置也跟着变化。Visual C++这样做的原因在于，一方面，在开发过程中，可能需要不同的特定设置来帮助调试应用程序；另一方面，在最终发布应用程序的时候，也会用到一些设置。这些设置决定了编译器将哪些选项设置为默认，以及编译器使用哪些库。一般情况下，在调试程序时，为了追踪文本格式的代码，使用 Win32 Debug；而在发布软件时，为了节省空间，优化代码，使用 Win32 Release。如果分别用两种方式编译、运行一个较大的程序，会发现两者占用的空间有明显的变化。

General 选项卡中的命令可以对项目进行不同的设置，如项目使用 MFC 类库的方式（静态和动态）、中间文件（如源文件和 OBJ 文件）或输出文件（EXE、DLL、XCX）所在目录。

Debug 选项卡中的命令用于设置项目的调试信息。

C/C++选项卡中的命令非常重要，用于设置代码的优化及调试信息。如图 1-16 所示，Warning Level 的级别默认为 Level 3，优秀的程序员应该能在 Level 1 中顺利地编译、运行自己的代码。Reset 按钮可以使工程重置为默认样式。

Project Options 选项根据需要有不同不同的设置。

设置默认 char 行为。Visual C++并没有把 unsigned char 设置为默认值，因为许多程序使用 unsigned char 来存储从 0~255 之间的值，当输入 char 变量时，编译器将变量声明解析成 unsigned char 变量。每次都输入 unsigned char 太浪费空间，为此用户可以通过以下方式使编译器将 unsigned char 作为默认使用。在 Category 下拉列表框中，选择 General 选项，并在 Project Options 文本框中，将“/J”添加到文本的最后。