



计算机基础教育系列教材

Visual C++程序设计教程

杨明 李明才 刘莉 余亮 编著

安徽大学出版社

内 容 简 介

本教程以大、中专及新高职非计算机专业学生为对象,以当前流行的中文 Windows 98 为背景,立足于社会上流行的中文 Office 97 办公软件中的 Word 97、Excel 97 和 PowerPoint 97 及网络的基本知识,力求图文并茂,文字通俗易懂,内容普遍实用,尤其注重计算机操作技能的培养。全书通过大量的生活实例作引导,循序渐进地将最主要的内容介绍给读者。为方便读者自学和检测,各章均附有大量的测试题、操作题(含参考答案),可作为各专业计算机基础知识与操作技术的入门教材、各类培训班的培训教材、公务员计算机应用能力考核与计算机爱好者的自学教材,对微软的 ATC 认证考试、教育部的 NIT 考试也有很大的参考价值,是一本掌握计算机最新操作技术的实用教材。

计算机基础教育系列教材 编 委 会 名 单

主 任:孙家启

委 员:	王忠仁	王志雁	王永国	方潜生
	尹荣章	石 竹	冯年荣	冯崇岭
	孙家启	孙道德	仲 红	朱 武
	朱学勤	齐学梅	吴国凤	李 雪
	何 明	张伟林	张国平	陈桂林
	郑尚志	周鸣争	周恒忠	姚合生
	赵林玲	聂会星	徐奇观	黄海生
	程承士	谢荣传	蔡之让	潘 瑜

(按姓氏笔划为序)

秘书长:聂会星

编写说明

为了支持计算机基础教育改革与建设,促进计算机基础课程教学与水平考试向纵深发展,我们按照计算机文化基础教育、技术基础教育和应用基础教育三个层次,组织编写了计算机基础教育系列教材。这套教材囊括了计算机文化基础、计算机文化基础上机操作教程、高级语言(QBasic, Visual Basic, C, Visual C++, PaSCAL, FORTRAN77, FORTRAN90, Fox-Pro For Windows, Visual FoxPro 等)程序设计、软件技术基础、微型计算机原理、计算机网络、微型机组装与维护、CAI 课件制作及应用等方面内容,涵盖计算机水平考试的一、二、四级(全国等级考试的一、二、三级),因而具有广泛的适应性。这套教材所具有的突出特点是:紧扣计算机基础教育大纲(即计算机水平考试大纲),兼具普通教材与考试辅导材料的双重功能;立意创新,内容简练,其大量针对性极强的习题和典型例题分析为其他教材所少见;编写人员都是教学、科研第一线有着丰富教学与实践经验的教师,他们深谙相关知识的张弛取舍。我们还聘请了三位知名专家担任高级顾问,以确保本系列教材的编写质量。

本系列教材的先期版本现已问世,第一辑各册已于 2000 年底全部出齐。由于计算机技术的发展比人们想象的还要快,所以本系列教材又增加了不少新内容,我们今后还将不断调整教材内容、平台和版本,使之与届时发展相适应,以便教材以更新更好的面目呈现在读者面前。

本系列教材编写目的明确,它特别适合于作为普通高校非计算机专业的本、专科教学用教材或成教、夜大、函大计算机专业的教材,也可供各地计算机水平考试考点使用,还可供广大计算机自学者、工程技术人员参考。

计算机基础教育系列教材编委会

2001 年 2 月

前 言

Visual C++ 6.0 程序设计是 Microsoft 公司的 Developer Studio 6.0 工具集的重要组成部分,是一种用于 Windows 98 或 Windows NT 应用程序的可视化开发工具。它改善了传统的编程手段,使得程序员可以直接在用户界面良好的可视化开发环境中进行工作。Visual C++ 6.0 还集成了多种有用的工具与功能,从而大大提高了应用程序的开发效率。Visual C++ 6.0 是理工类高年级学生和计算机研究人员应用开发的一种工具,是面向对象程序设计语言的典型代表。为了读者学习和掌握面向对象的程序设计思想,特此奉献本书给广大读者。

本书以高等学校非计算机专业和计算机专业学生为教学对象,系统介绍了 Visual C++ 6.0 的语言基础、面向对象的程序设计方法、可视化的开发环境及实用工具与功能。兼顾全国高等学校计算机考试的需要,全书内容覆盖了《Visual C++ 6.0 程序设计》考试范围的内容,所以它又是一本考试指导书——计算机等级(水平)考试系列教材之一。

这是一门实践性很强的课程,除了课堂教学外,必须做一定量的实验。作为教材,建议课程总学时为 60 学时,其中含上机实验 28 学时。

本书共分 13 章。第 1 章、第 6 章至第 8 章及附录由杨明编写,第 2 章至第 5 章由李明才编写,第 9 章至第 11 章由刘莉编写,第 12 和第 13 章由余亮编写。编写过程中,得到了本系列教材高级顾问们的悉心指导,尤其是孙家启教授的指导,并参阅了国内外有关著作,谨在此一并表示衷心的感谢。

由于水平有限,加之时间仓促,错误与疏漏之处难免,恳请读者批评指正。

编者

2001 年 2 月

目 录

第 1 章 Visual C++ 6.0 开发环境	1
1.1 Visual C++ 6.0 主窗口	1
1.2 Visual C++ 6.0 工具栏	1
1.3 Visual C++ 6.0 菜单栏	5
1.4 项目与项目工作区	12
1.5 资源与资源编辑器	13
第 2 章 C++ 导论	21
2.1 面向对象技术的形成和发展	21
2.2 面向对象技术的基本概念	22
2.3 面向对象分析基本原理与方法	24
2.4 面向对象设计方法与步骤	24
2.5 面向对象程序设计	25
习题	27
第 3 章 C++ 基本知识	28
3.1 基本输入/输出	28
3.2 变量	29
3.3 存储类	31
3.4 用户自定义数据类型	33
3.5 指针	38
3.6 语句	41
3.7 函数	52
习题	55
第 4 章 类与数据封装	58
4.1 类定义	58
4.2 数据封装	59
4.3 成员函数	59
4.4 构造函数与析构函数	61
4.5 类的静态成员	65
4.6 友元	67
4.7 对象组织	68
4.8 函数重载	73
4.9 运算符重载	75
4.10 类与结构	81

4.11 对象类型转换	81
习题	83
第 5 章 类继承	85
5.1 基础类与派生类	85
5.2 派生类的构造函数与析构函数	87
5.3 派生类的指针	90
5.4 多重继承	91
5.5 二义性及其支配规则	93
5.6 虚基类	96
习题	97
第 6 章 虚拟函数与多态性	99
6.1 静态联编与动态联编	99
6.2 虚拟函数	100
6.3 多重继承中虚拟函数的二义性	108
6.4 纯虚拟函数与抽象类	109
6.5 运算符虚拟函数	112
习题	113
第 7 章 模板、异常错误处理	115
7.1 模板的定义	115
7.2 函数模板	115
7.3 类模板	117
7.4 异常错误处理	120
习题	127
第 8 章 I/O 流与文件操作	128
8.1 VC++ 流类结构	128
8.2 操纵符与格式控制	128
8.3 ios 类成员的应用	131
8.4 ostream 类与输出	138
8.5 istream 类与输入	139
8.6 文件管理	140
8.7 输出运算符和输入运算符的重载	146
习题	149
第 9 章 MFC 基本类库	150
9.1 MFC 的特点	150
9.2 MFC 的类层次	150
9.3 MFC 的通用集合类	159
习题	161
第 10 章 Visual C++ 界面设计	162

10.1 AppWizard 和应用框架	162
10.2 Windows 标准控制	167
10.3 对话框	174
10.4 文档与视图	177
10.5 菜单	179
10.6 工具条	180
10.7 状态条	184
10.8 对话条	186
习题	187
第 11 章 Visual C++ 6.0 通用控制	188
11.1 动画控制和进展控制	188
11.2 标签控制和属性对话框	189
11.3 图像列表和列表控制	191
11.4 其他通用控制	194
习题	198
第 12 章 Visual C++ 6.0 绘图技术	199
12.1 理解 GDI	199
12.2 基本的图形对象	201
12.3 画笔对象	206
12.4 刷子对象	212
12.5 字体对象	215
12.6 映射模式	221
12.7 点位图及其操作	225
12.8 图标和光标的管理和编程	228
12.9 打印和打印预览	229
第 13 章 Visual C++ 6.0 高级应用	232
13.1 线程和进程	232
13.2 生成和管理辅助线程	235
13.3 同步化线程	241
13.4 多进程之间互相访问	245
13.5 COM 技术介绍	255
参考书目	269

第 1 章 Visual C++ 6.0 开发环境

Visual C++ 6.0 提供了良好的可视化编程环境,使得在 Visual C++ 应用程序中可进行各种操作,包括建立、打开、浏览、编辑、保存、编译、链接和调试等,点击鼠标即可进行,因此非常方便。

1.1 Visual C++ 6.0 主窗口

当正确安装 Visual C++ 6.0 到 Windows 系统之后,单击“开始”按钮,指向“程序”(programs)“Microsoft Visual Studio 6.0”,然后单击“Microsoft Visual C++ 6.0”,即可启动 Visual C++ 6.0,进入集成开发环境 Developer Studio,参见图 1-1。

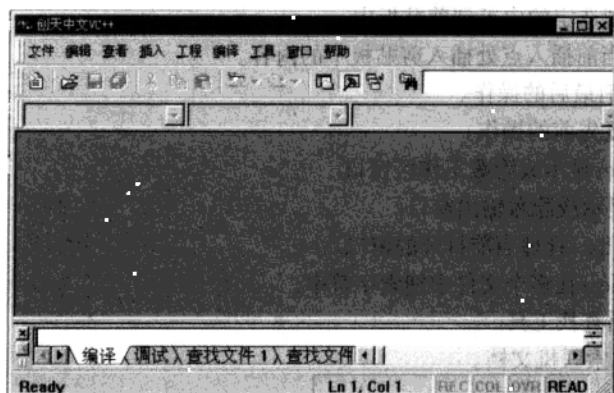


图 1-1 Visual C++ 6.0 开发环境

集成开发环境的主窗口包括标题栏、菜单栏、工作区窗口(缺省时处于船坞化状态)、源代码编辑窗口、输出窗口(缺省处于船坞化状态)和状态条。

从上图可以看出,构成 Visual C++ 集成开发环境的主窗口由 3 个子窗口组成,它们分别是:编辑窗口和船坞化的工作窗口及输出窗口。如果试着用鼠标拖动这两个船坞化窗口的边框也可以使它们成为普通的窗口。

下面分别介绍 Visual C++ 6.0 集成开发环境的界面元素。

1.2 Visual C++ 6.0 工具栏

为了使用工具栏,必须使工具栏出现在 C++ 的主窗口中,缺省时,C++ 已经使船坞化的 Standard 和 Build 两个工具栏显示在主窗口中。要使用其他工具栏,只需移动鼠标指向工具栏的位置,然后单击右键,屏幕上弹出一快捷菜单,单击要使用的工具栏,使其前面出现√,则相应的工具栏就会出现在 C++ 主窗口中。

下面逐一介绍每一工具栏所含的各工具的作用与用法。

1.2.1 标准工具栏

标准工具栏含有有 15 个工具,主要用于建立项目工作区,如图 1-2 所示。



图 1-2 标准工具栏

各按钮的具体功能为:

- New Text File :创建新的文本文件。
- Open:打开已有的文档。
- Save:保存文档。
- Save All:保存所有打开的文件。
- Cut:剪切选定的内容到剪贴板中。
- Copy:复制选定的内容到剪贴板中。
- Paste:在当前插入点处插入剪贴板中的内容。
- Undo:取消最后的操作。
- Redo:重复先前的操作。
- Workspace:显示或隐藏工作区窗口。
- Output:显示或隐藏输出窗口。
- Windows list:管理当前打开的窗口。
- Find in files:在多个文件中搜索字符串。
- Find:激活查找工具。
- Search:搜索联机文档。

1.2.2 编译工具栏

编译工具栏含有 8 个工具,如图 1-3 所示。当用户建立好应用程序项目后,可利用这些工具对项目进行编译、连接以及调试和运行。

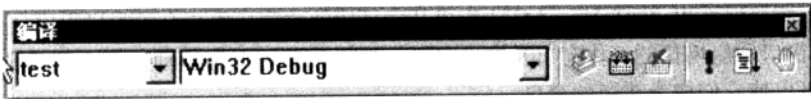


图 1-3 编译工具栏

各工具按钮的具体功能为:

- Select Active Project:选择活动项目。
- Select Active Configuration:选择活动的配置。Visual C++ 提供了两种活动配置,它们分别是 Win Release (基于 Win32 平台的发行版)和 Win32 Debug(基于 Win 32 平台的调试版)。
- Compile:编译文件。要编译的文件必须是构成项目的一部分。
- Build:建立项目。

- Stop Build:停止建立项目。
- Execute Program:执行程序。
- Go:启动或继续程序的执行。
- Insert/Remove Breakpoint:插入或删除断点。

1.2.3 编译微型条工具栏

该工具栏包括 6 个工具按钮,如图 1-4 所示。其功能分别相当于编译工具栏中的后 6 个工具按钮,主要用于建立较小的项目。

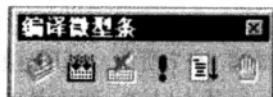


图 1-4 编译微型条工具栏

1.2.4 资源工具栏

资源工具栏含有 9 个创建资源的工具和 1 个显示资源状态的工具,如图 1-5 所示。这些工具用来帮助用户创建应用程序所需的各种资源。



图 1-5 资源工具栏

下面列出每个工具的名称及快捷键,具体功能将在本章的 2.5 节中详细介绍。

- New Dialog:创建新对话框。
- New Menu:创建新菜单。
- New Cursor:创建新光标。
- New Icon:创建新图标。
- New Bitmap:创建新位图。
- New Toolbar:创建新工具栏。
- New Accelerator:创建新快捷键。
- New String Table:创建新串表。
- New Version:创建新版本信息。
- Resource Symbols:显示资源符号。

1.2.5 向导条工具栏

向导条工具栏含有 4 个工具,如图 1-6 所示。其中前 3 个实际上是列表框,最后一个是带缺省动作的列表框。利用该组工具可以方便地建立和修改应用程序中涉及的所有类。下面只列出各工具的名称及简单应用,详细功能描述见本章的 1-5 节。

- WizardBar C++ Class:在此列表框中选择要修改的类。
- WizardBar C++ Filter:在此列表框中选择要过滤的事件。
- WizardBar C++ Member:选择类成员。

- WizardBar Action(Default):根据前 3 个列表框中的选择执行不同的操作,缺省时的具体操作因前三者的不同而有所区别。

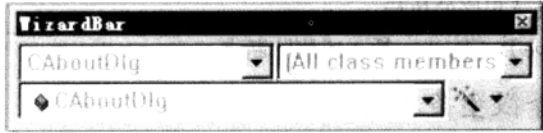


图 1-6 向导工具栏

1.2.6 调试工具栏

调试工具栏含有 15 个工具,如图 1-7 所示。利用这些工具可以调试已开发的项目,以便找到问题所在。只有处于调试运行状态时这一组工具才有效。

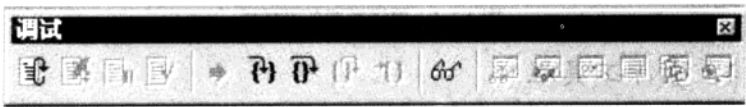


图 1-7 调试工具栏

- Restart:启动程序的执行,并处于调试状态。
- Stop Debugging:停止调试执行程序。
- Break Execution:中断程序的执行。
- Apply Code Change:认同程序代码的改变,即当用户在调试过程中改变程序代码后,使用该工具使代码有效。
- Show Next Statement:显示下一条要执行的语句。
- Step Into:进入被调函数内单步执行。
- Step Over:单步执行,但跳过被调用的函数。
- Step Out:从被调用的函数内跳出,继续执行调用语句的下一条语句。
- Run to Cursor:运行到当前光标处。
- Quick Watch:快速查看当前的调试状态。
- Watch:打开一独立窗口,在其内显示用户要查看的变量的值和类型。用户需输入变量名,调试程序自动显示变量类型及其值。
- Variables:打开一独立窗口,在该窗口内有 3 个选项卡,分别显示当前语句和上一条语句所用变量、正在执行函数的局部变量以及 this 指针所指对象的信息。
- Registers:打开一独立窗口,在该窗口内显示 CPU 各寄存器状态。
- Memory:打开一独立窗口,在该窗口内显示当前语句调用的所有函数,当前函数在栈顶。
- Disassembly:打开一独立窗口,在该窗口内显示反汇编代码。

1.2.7 编辑工具栏

编辑工具栏包括 8 个工具,如图 1-8 所示。这些工具用于源程序的编辑,提高用户查找和定位信息的效率。

- Toggle Bookmark:在当前光标位置设置或取消书签。

- Next Bookmark:光标定位到下一个书签。
- Previous Bookmark:光标定位到上一个书签。
- Clear All Bookmark:清除所有书签。
- Find:在当前编辑窗口内查找。
- Increase Indent:增加缩进量。
- Derease Indent:减少缩进量。
- Toggle Whitespace Display:锁定空白显示。



图 1-8 编辑工具栏

1.2.8 数据库工具栏和 Atl 工具栏

数据库工具栏有 3 个工具,而 Atl 工具栏只有一个工具,如图 1-9 所示。

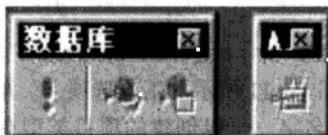


图 1-9 数据库和 Atl 工具栏

数据库工具栏中的工具用于连接远程数据库服务器的操作。Atl 工具栏中的工具用于向用户项目添加 Atl 对象。

- Run:运行所选择的数据库对象。
- Add Data Connect:向活动项目中添加数据库连接。
- Add New Database Item:向数据库中添加新项目。
- New Atl Object:向当前项目添加新的 Atl 对象。

1.3 Visual C++ 6.0 菜单栏

Visual C++ 6.0 菜单栏包括的菜单有文件、编辑、查看、插入、工程、编译、工具、窗口、帮助等 9 个菜单,这与 5.0 版的菜单相同。下面简要介绍一下各菜单的功能。

1.3.1 文件菜单

文件菜单的各个命令主要完成文件的建立、保存、打开、关闭以及打印等工作。常用的命令有如下几个:

1. 新建命令

选择该命令将打开一个对话框(如图 1-10 所示),通过该对话框用户可以创建新的文件、项目、工作区或者其他文档。

缺省时打开文件选项卡,在其中单击要创建的文件类型,然后在文件文本框中键入要创建的文件名字。可以创建的文件类型有:Active Server Page (服务器页文件),Binary file(二

进制文件),Bitmap File (位图文件),C/C++ Header File (C/C++ 头文件),C++ Source File (C++ 源文件),Cursor File(光标文件),HTML Page(HTML 文件),Icon File(图标文件),Macro file(宏文件),SQL Script File(SQL 脚本文件),Resource Script(资源脚本文件),Resource Template(资源模板文件),Text File(文本文件)等。

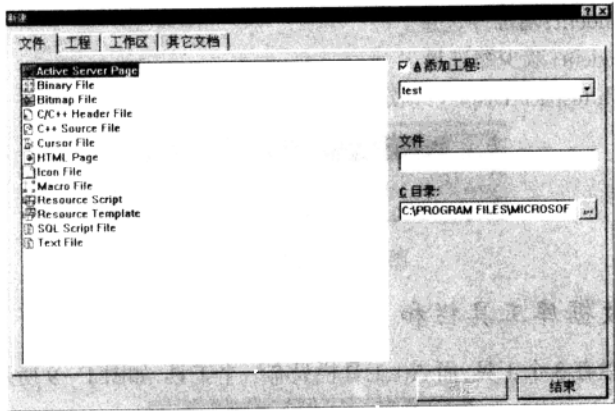


图 1-10 新建对话框的文件选项卡

选择工程标签,将打开如图 1-11 所示的工程选项卡。单击要创建的项目类型,然后在工程文件文本框中键入项目的名字。可以创建的项目类型有:ATL COM AppWizard(ATL 应用程序),Cluster Resource Type Wizard(簇资源类型),Custom Appwizard(自定义的应用程序向导),Database Project(数据库项目),DevStudio Add_in Wizard(自动化宏),Extended Stored Proc Wizard(基于 SQL 服务器的扩展存储进程),ISAPI Extension Wizard(Internet 服务器或过滤器),Makefile(旧版 C/C++ 生成文件),MFC ActiveX ControlWizard(ActiveX 控件程序),MFC AppWizard(exe)(MFC 可执行程序),MFC AppWizard(dll)(MFC 动态链接库),New Database Wizard(SQL 服务器数据库),Utility Project(单元项目),Win32 Application(Win32 应用程序),Win32 Console Application(Win32 控制台应用程序),Win32 Dynamic-link Library(Win32 动态链接库),Win32 Static Library(Win32 静态库)。

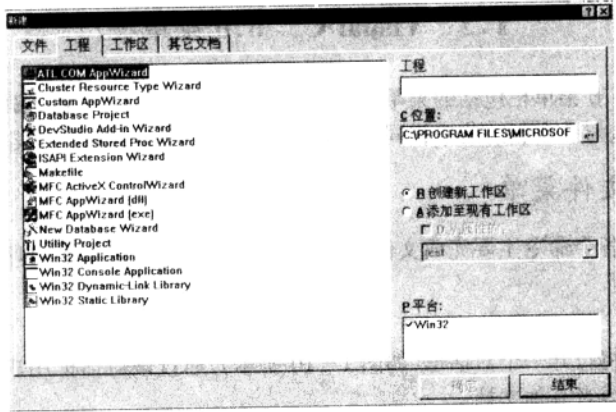


图 1-11 新建对话框的工程选项卡

注释:

在新建对话框的工作区和其他文档选项卡中,可以创建各种类型的工作区文件和文档。

2. 打开命令

选择打开命令将弹出“打开”对话框用来打开已经存在的文件,如图 1-12 所示。这些文件可以是 C++ 文件、Web 文件、宏文件、资源文件、定义文件、工作区文件和项目文件等。

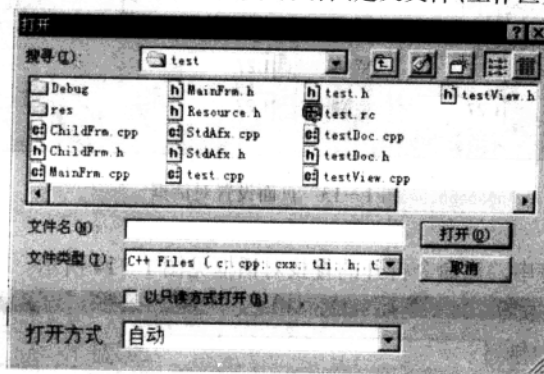


图 1-12 “打开”对话框

3. 关闭命令

关闭命令可关闭在活动窗口中打开的文件。要激活或选择某一窗口可以通过鼠标单击该窗口来实现。若文件未保存就关闭,则系统会提示用户保存该文件。

4. 打开工作区

选择打开工作区命令将弹出打开工作区对话框。它主要用于打开工作区文件,也可用于打开其他类型的文件。

5. 存盘命令

存盘命令用于保存活动的窗口或当前选定窗口中的文件内容。若所保存的文件是首次保存,则系统会提示另存对话框;如果当前文件是只读方式打开的,则保存选项是灰色,表示此项不能使用,即不能修改该文件。

6. 另存命令

该命令也是用于保存文件,只是把已打开的文件用一个新的文件加以保存而已。如果在保存文件时想要备份,则使用另存选项,将该文件用于另一个名字保存起来。

7. 全部保存

保存命令用于保存当前活动窗口或选定窗口中的文件,而全部保存命令用来保存所有窗口内的文件内容。若某个或某些文件内容未保存过,则系统会自动提示用户输入有效的文件名。

8. 页面设置

该命令用来设置和格式化打印结果。选择该命令后打开页面设置对话框,如图 1-13 所示。利用该对话框可以为要打印的文档页设置标题、脚注,以及上、下、左、右 4 个边距。

1.3.2 编辑命令

编辑菜单包含用于编辑和搜索的命令,这些命令与其他的 Windows 应用程序中同名命

令具有相似的功能。例如用于撤消操作的“重复/撤消”,用于剪贴操作的“剪切/拷贝/粘贴”命令。此外,编辑菜单中还有如下一些常用的命令。

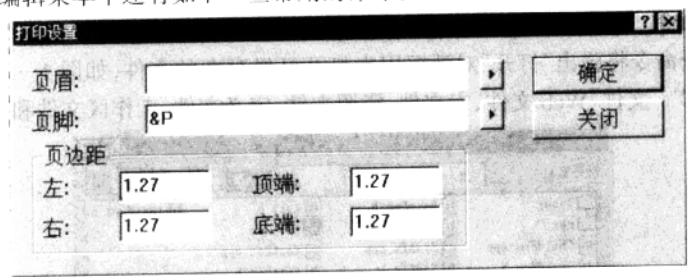


图 1-13 页面设置对话框

1. 查找命令

查找指定的字符串。该命令打开的查找对话框见图 1-14 所示。

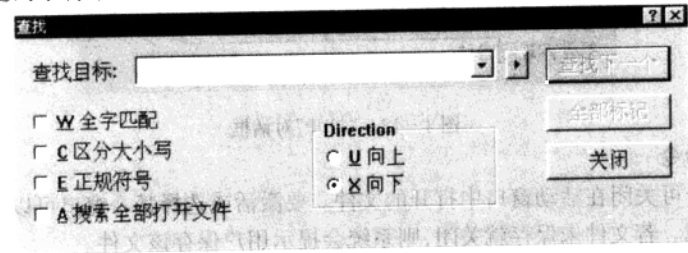


图 1-14 查找对话框

在查找对话框中,可以在查找目标文本框中输入欲查找的字符串,并设置查找的方向(向上/向下)。另外,可根据需要进行区分大小写字符串查找或进行整词查找。

正规符号复选框用来按照正则表达式查找文件中匹配的文本。正则表达式是指用特殊的字符序列去匹配文件中的某个文本模式。

表 1-1 正则表达式的查找模式

查找模式	含义
*	匹配任意多个字母,如 Data*1,其查找结果为 Data1,Data21,Datajkl1
.	匹配单个字符,如 Data.,其查找结果为 Data1,Data2,而不包括 Datajkl1
^	匹配以指定字符串开头的每一行,如 Data^,则所有以 Data 开头的行均是
+	匹配以指定字符串结束的字符串,如 +value,则结果是 ivalue,fvalue 等
\$	匹配以指定字符串结束的每一行,如//AFG_MSG\$,结果是以//AFG_MSG 结尾的所有行。
[]	匹配指定集合中的字符,如 Data[A..Z],结果可是 DataA,DataB,而 Data1 则不是
\	在搜索时准确匹配\前的字符,如 Data[A..Z\0..9],结果可是 DataA1,DataB5
\	括号间指定字符串可以任意重复 如 \ abc\ *result,其结果是:result,no result,mone result

2. 定位命令

选择定位命令将弹出定位对话框,可以指定如何将光标移到当前活动窗口的指定位置。

3. 书签命令

该命令可以设置、命名、删除和读取书签。

4. 断点命令

该命令用来设置、删除和查看断点。断点告诉调试器应在何时何地中断程序的执行过程,以便检查程序代码、变量和寄存器值。断点分为位置断点、数据断点、消息断点和条件断点。

5. List Members 命令

列表光标处对象所属类的成员(包括其基类的成员)。可以通过双击相应的成员名,将该成员添加到光标处。

6. Parameter Info 命令

该命令用来显示光标处函数的参数信息,以供书写函数调用时参考,从而保证函数调用时的参数表与其声明的原型相一致,减少错误发生的机会。

1.3.3 查看菜单

查看菜单包含用于检查源代码和调试信息的命令项,它可以用来设置按全屏方式显示窗口,或者打开工作区窗口、output 窗口和各种调试窗口。另外,从该菜单中还可以访问 MFC ClassWizard。

1.3.4 插入菜单

插入菜单可以创建新的类、资源、窗体并将它们插入到文档中;可以将文件作为文本插入到文件中,也可以添加新的 ATL 对象到项目中。插入菜单还包括以下常用命令:

1. 资源拷贝

该命令用来创建选定资源的备份,即复制选定的资源。

2. File As Text

该命令用来选择插入到文档中的文件。

3. New ATL Object

该命令用来启动 ATL Object Wizard,以便添加新的 ATL 对象到项目中。

1.3.5 工程菜单

工程菜单中的命令用于管理项目和工作区。可以选择指定的项目为工作区中的活动项目,也可以把文件、文件夹、数据链接以及可再用部件添加到项目中,还可以编辑和修改项目间的依赖关系。

1.3.6 编译菜单

编译菜单包括用于编译、建立和执行应用程序的命令。较为重要和常用的有编译、建立、执行和配置等命令。

1. 编译命令

此命令用于编译显示在源代码的编辑窗口中的源文件并检查源文件中是否有语法错误。在编译过程中若有警告或错误,则将在输出窗口中显示错误信息。可以向前或向后浏

览输出窗口中的错误信息,然后按 F4 键在源代码编辑窗口中显示相应的代码行。

2. 建立命令

建立命令用来建立项目中的所有文件。对于代码或组件结构比较复杂的项目,建立都是很有力的工具。若建立过程中检测出某些语法错误,就将它们显示在编辑窗口中。

3. 执行命令

该命令用于运行程序。Visual C++ 将根据被运行程序的目标格式自动调用相应的环境(如 MSDOS, Windows 95 或 Windows NT 等)。

4. 配置命令

配置命令是用于设置应用程序的工作方式。选择配置命令将弹出配置对话框(如图 1-15)。

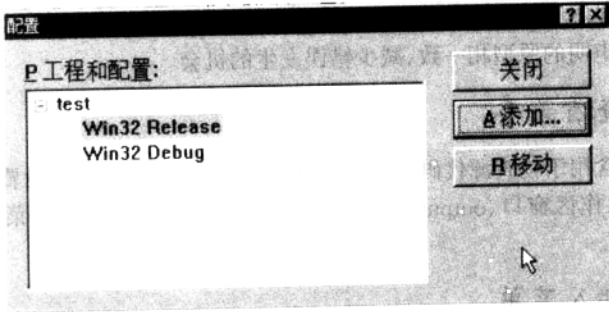


图 1-15 配置对话框

1.3.7 调试菜单

启动调试器后,调试菜单将取代建立菜单出现在菜单栏中。调试菜单中包括以下常用命令:

1. 去命令

此命令用于调试过程中从当前语句启动或继续运行,等价于建立工具栏的去按钮。

2. Restart 命令

系统重新装配程序到内存并放弃所有变量的当前值。

3. Stop Debugging 命令

用于中断当前的调试过程并返回正常的编辑状态。

4. Step into 命令

在调试过程中单步执行程序,而且当程序执行到某一函数调用语句时,进入该函数内部,从头单步执行。

5. Step Over 命令

在调试过程中单步执行程序,但当程序执行某一函数调用语句时,不进入该函数内部,而是直接执行该调用语句,接着再执行调用语句后面的语句。

6. Step Out 命令

该命令与 Step Into 命令配合使用。当执行 Step into 命令进入函数内部并开始单步执行时,若发现并需要对该函数的内部进行单步调试,那么就可以使用 Step Out 命令使程序