

C 语言 程序设计实践教程

C YUYAN CHENGXU SHEJI SHIJIAN JIAOCHENG

主编 夏欣



四川大学出版社

A large, bold, white letter 'C' is centered within a black rounded rectangular frame. The 'C' is slightly offset to the right, creating a sense of depth and shadow.

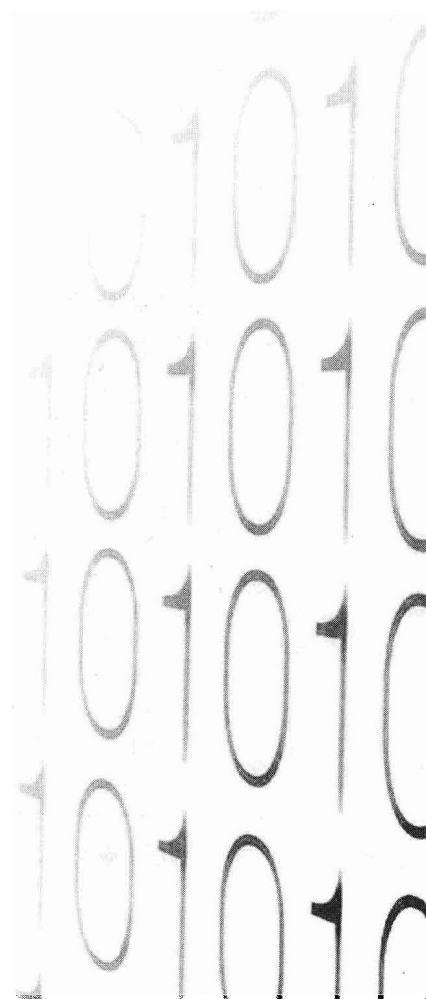
语言 程序设计实践教程

C YUYAN CHENGXU SHEJI SHIJIAN JIAOCHENG

主 编 夏 欣
副主编 郭新明
编 委 葛 龙 孙亚飞
王 舒 张 宇



四川大学出版社



责任编辑:毕 潜
责任校对:傅 奕
封面设计:墨创文化
责任印制:李 平

图书在版编目(CIP)数据

C 语言程序设计实践教程 / 夏欣主编. —成都: 四川大学出版社, 2010. 6

ISBN 978-7-5614-4868-7

I. ①C… II. ①夏… III. ①C 语言—程序设计—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 091330 号

书名 C 语言程序设计实践教程

主 编 夏 欣
出 版 四川大学出版社
地 址 成都市一环路南一段 24 号 (610065)
发 行 四川大学出版社
书 号 ISBN 978-7-5614-4868-7
印 刷 郫县犀浦印刷厂
成品尺寸 185 mm×260 mm
印 张 17.75
字 数 424 千字
版 次 2010 年 6 月第 1 版
印 次 2010 年 6 月第 1 次印刷
印 数 0 001~4 000 册
定 价 30.00 元

◆读者邮购本书,请与本社发行科
联系。电话:85408408/85401670/
85408023 邮政编码:610065

◆本社图书如有印装质量问题,请
寄回出版社调换。

◆网址:www.scupress.com.cn

版权所有◆侵权必究

前 言

程序设计是学习计算机应用与软件开发的基础，如果只会简单的计算机操作，不了解软件开发的实质，就无法从根本上了解计算机的工作原理，也很难应对信息技术日新月异的飞速发展。C语言作为一种通用的程序设计语言，结构简单，数据类型丰富，运算灵活方便。C语言是一种理想的结构化程序设计语言，因此，可用于编写高效简洁、风格优美的应用程序以及计算机系统程序。用C语言编写的程序，具有运算速度快、效率高、目标代码紧凑、可移植性好等特点。

本书由四川大学长期从事C语言程序设计的教学组编写，其目的是为了帮助程序设计初学者或大学非计算机专业的学生在短时间内快速入门，掌握C语言编程的特点与技巧。同时，非计算机专业的学生绝大多数会参加计算机等级考试，对于上机考试普遍感觉有一定的难度，本书在编写时也考虑了这个问题。

本书共分5章。第1章介绍了Visual C++ 6.0的使用，第2章由浅入深安排了8个基础的上机实验，介绍了常见的程序算法，如排序、查找、删除等。在熟练掌握了8个实验的基础上，再进行综合的上机练习，完成第3章计算机等级考试的上机实验。配合《C语言程序设计》教材，本书在第4章给出了教材各章课后习题答案，并在第5章附上了5套全国计算机等级考试笔试模拟题。书中列举的例题及程序均在Visual C++ 6.0下编译调试通过。

本书由夏欣主编，其中第1章和第5章由郭新明编写，第2章、第3章、第4章由夏欣编写。参与本书编写及程序调试的人员还有孙亚飞、吴宏瑜、马义玲、王舒（女）、葛龙、张宇、戴丽娟、王忠平、孟宏源、贾鹏、王舒（男）等。感谢葛龙、孙亚飞、王舒（女）、张宇为本书提供了《C语言程序设计》课后习题的部分参考答案。

由于时间仓促，编者水平有限，加之计算机科学技术发展迅速，书中难免有不少缺点和错误，恳请专家和读者批评指正。联系方式：xiaxin@scu.edu.cn。

编者

2010年5月

目 录

第 1 章 Visual C++ 6.0 的使用	(1)
1.1 主窗口简介	(1)
1.1.1 主窗口	(1)
1.1.2 工具栏	(2)
1.1.3 菜单栏	(4)
1.1.4 工作区	(9)
1.2 在 Visual C++ 中编辑并运行一个 C 程序	(10)
第 2 章 上机实验——基础练习篇	(14)
2.1 实验一 最简单的 C 程序设计	(14)
2.1.1 实验目的	(14)
2.1.2 实验内容	(14)
2.2 实验二 选择结构程序设计	(18)
2.2.1 实验目的	(18)
2.2.2 实验内容	(18)
2.3 实验三 循环结构程序设计 (一)	(21)
2.3.1 实验目的	(21)
2.3.2 实验内容	(21)
2.4 实验四 循环结构程序设计 (二)	(23)
2.4.1 实验目的	(23)
2.4.2 实验内容	(23)
2.5 实验五 数组 (一)	(26)
2.5.1 实验目的	(26)
2.5.2 实验内容	(26)
2.6 实验六 数组 (二)	(29)
2.6.1 实验目的	(29)
2.6.2 实验内容	(30)
2.7 实验七 函数	(33)
2.7.1 实验目的	(33)
2.7.2 实验内容	(34)
2.8 实验八 指针	(38)
2.8.1 实验目的	(38)

2.8.2 实验内容	(38)
第3章 上机实验——综合练习篇	(43)
3.1 程序改错题分析	(44)
3.1.1 简单语法错误	(44)
3.1.2 逻辑错误	(47)
3.1.3 数组处理错误	(50)
3.1.4 字符串处理错误	(51)
3.1.5 函数处理错误	(54)
3.1.6 指针错误	(58)
3.1.7 链表处理错误	(61)
3.2 程序填空和程序设计题分析	(65)
3.2.1 整数各位的分离	(65)
3.2.2 素数问题	(68)
3.2.3 使用循环累加、累乘、累除处理一些简单数学公式	(70)
3.2.4 简单数学问题	(73)
3.2.5 找最大元或最小元及其下标值	(77)
3.2.6 排序算法	(82)
3.2.7 查找算法	(85)
3.2.8 数据的移动和交换	(90)
3.2.9 数据的删除	(93)
3.2.10 二维数组的处理	(97)
3.2.11 字符串的相关处理	(100)
3.2.12 链表的处理	(103)
3.2.13 结构体的处理	(108)
3.2.14 文件的相关操作	(111)
第4章 教材课后习题参考答案	(115)
4.1 第1章参考答案	(115)
4.2 第2章参考答案	(115)
4.3 第3章参考答案	(116)
4.4 第4章参考答案	(118)
4.5 第5章参考答案	(128)
4.6 第6章参考答案	(135)
4.7 第7章参考答案	(146)
4.8 第8章参考答案	(147)
4.9 第9章参考答案	(152)
4.10 第10章参考答案	(160)
4.11 第四套综合练习题参考答案	(164)
4.12 第五套综合练习题参考答案	(165)
4.13 第六套综合练习题参考答案	(167)

4.14	第七套综合练习题参考答案	(168)
4.15	第八套综合练习题参考答案	(168)
4.16	第九套综合练习题参考答案	(169)
第5章	二级C语言笔试模拟试题	(171)
5.1	第一套模拟试题	(171)
5.2	第二套模拟试题	(184)
5.3	第三套模拟试题	(197)
5.4	第四套模拟试题	(212)
5.5	第五套模拟试题	(224)
5.6	试题参考答案与试题分析	(238)
5.6.1	第一套模拟试题参考答案与分析	(238)
5.6.2	第二套模拟试题参考答案与分析	(244)
5.6.3	第三套模拟试题参考答案与分析	(252)
5.6.4	第四套模拟试题参考答案与分析	(258)
5.6.5	第五套模拟试题参考答案与分析	(263)
附录	全国计算机等级考试二级C语言考试大纲	(271)
	第一部分 公共基础知识	(271)
	第二部分 C语言程序设计部分	(272)

第1章 Visual C++ 6.0 的使用

1.1 主窗口简介

1.1.1 主窗口

启动 Visual C++(简称 VC)6.0,进入 Developer Studio 编译环境,如图 1-1 所示。

主窗口由标题栏、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。

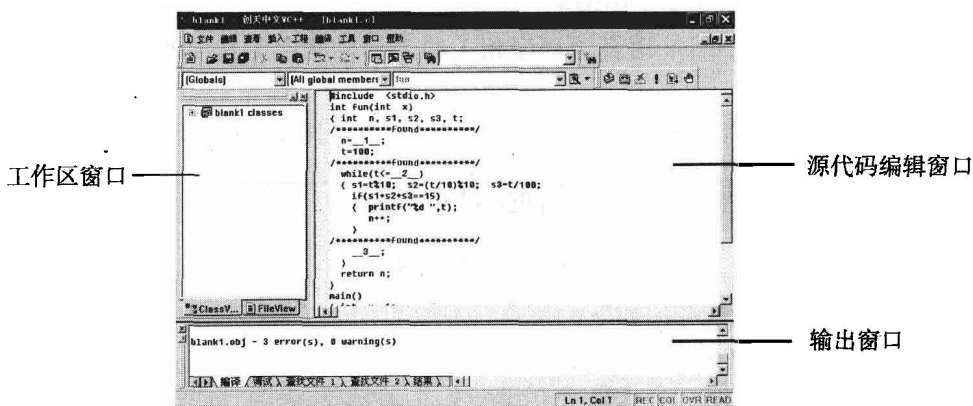


图 1-1 Visual C++ 6.0 编译环境窗口

屏幕窗口中最上方是标题栏,显示所打开的应用程序名。标题栏左端是控制菜单图标,单击后弹出窗口控制菜单。标题栏右端从左至右有 3 个控制按钮,分别为最小化、最大化(还原)和关闭按钮,可用它们快速设置窗口的大小。



标题栏下方是菜单栏,由 9 个菜单项组成。单击菜单项,弹出下拉式菜单,可使用这些菜单项实现集成环境的各种功能。

菜单栏下方是工具栏,它由若干个功能按钮组成,单击按钮可实现某种操作功能。

工具栏的下方有左右两个窗口,左窗口是工作区窗口,右窗口是源代码编辑窗口。

在工作区窗口和源代码编辑窗口的下方有一个输出窗口,在创建项目时,用来显示项目创建过程中的错误信息。

屏幕最底部是状态栏,它可给出当前操作或所选命令的提示信息。

在上述窗口组成部分中,工作区窗口可通过单击工具栏中的 Workspace 按钮  隐藏或显示,输出窗口可通过单击工具栏中的 Output 按钮  隐藏或显示。隐藏这些窗口可以扩

大源代码编辑区的大小。

1.1.2 工具栏

为了用户操作方便,该系统在主窗口中提供了多种工具栏,每种工具栏中有若干个按钮,每个工具栏中的按钮表示某种操作。在鼠标指向某个按钮时,将显示出该按钮的功能。

主窗口在默认情况下只显示两个常用的工具栏:Standard 和 Build 工具栏。如果要使用其他工具栏,可用鼠标指向工具栏的位置,单击右键,出现如图 1-2 所示的快捷菜单。该菜单中,提供了 10 个工具栏供选择。单击要选用的工具栏后,在该工具栏名前出现符号√,且该工具栏将出现在主窗口中。如果不想用某个工具栏,也可在工具栏位置单击右键,在出现的快捷菜单中单击不想用的某工具栏,则该工具栏名前的符号√消失,且该工具栏将从主菜单中消失。

下面介绍几个与实现 C++ 语言源程序相关的工具栏所含工具项的功能和用法,其余的工具栏不再介绍。

(1) 标准(Standard)工具栏。该工具栏中共有 15 个工具项按钮,如图 1-3 所示。

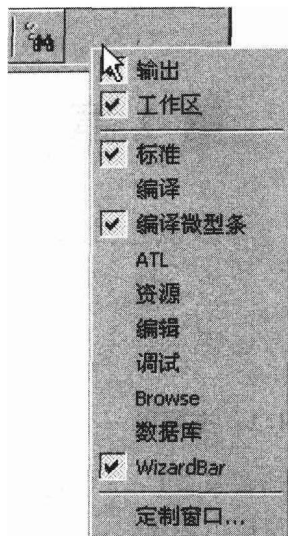


图 1-2 快捷菜单



图 1-3 Standard 工具栏

自左至右各按钮的功能介绍如下:

- New Text File: 创建新的文本文件。
- Open: 打开已有文档。
- Save: 保存当前文档内容。
- Save All: 保存所有打开的文档。
- Cut: 将选定的文档内容从文档中删除,并将其复制到剪贴板中。
- Copy: 将选定的文档内容复制到剪贴板中。
- Paste: 在当前插入点处粘贴剪贴板中的内容。
- Undo: 取消最近一次编辑操作。
- Redo: 恢复前一次取消的编辑操作。
- 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。

(2) Build 工具栏。该工具栏中有 8 个工具项按钮,如图 1-4 所示。

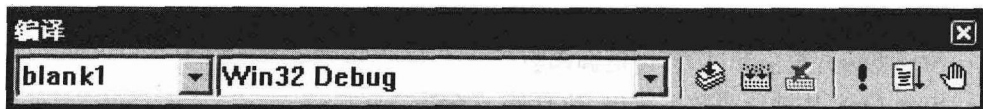


图 1-4 Build 工具栏

这些工具项按钮用来对已建好的应用文件或项目进行编辑、连接和运行。自左至右各工具项按钮的功能介绍如下:

- Select Active Project: 选择当前活动工程。
- Select Active Configuration: 选择活动的配置。VC 6.0 提供了 Win32 Release 和 Win32 Debug 两种活动配置,前者是基于 Win32 平台的发行版,后者是基于 Win32 平台的调试版。

- Compile: 编译文件(此按钮很重要)。
- Build: 构建可执行文件(此按钮很重要)。
- Stop Build: 停止构建可执行文件(此按钮很重要)。
- Execute Program: 执行程序(此按钮很重要)。
- Go: 执行程序到设置的断点处。
- Insert/Remove Breakpoint: 插入或删除断点。

(3) Build MiniBar 工具栏。该工具栏中包含 6 个工具项按钮,如图 1-5 所示。

该工具栏的 6 个工具项按钮依次为:

- Compile。
- Build。
- Stop Build。
- Execute Program。
- Go。
- Insert/Remove Breakpoint。



图 1-5 Build MiniBar 工具栏

它们的功能与工具栏 Build 中的相应按钮功能相同。

(4) Debug 工具栏。该工具栏中含有 16 个工具项按钮,如图 1-6 所示。

这些工具项按钮可用来调试已编译的 C++ 语言源文件及项目,查找所存在的问题。

只有处于调试运行状态时它们才有效。

第一行自左至右各个工具项按钮的功能如下:

- Restart: 重新启动程序,并处于调试状态。

- Stop Debugging: 停止调试运行的程序。

- Break Execution: 中断程序的执行。

- Apply Code Change: 使用改变代码进行调试。

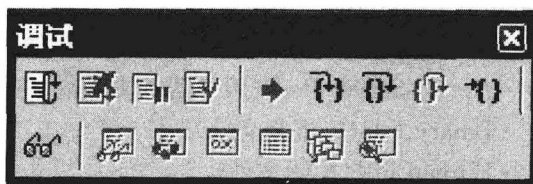


图 1-6 Debug 工具栏

- Show Next Statement: 显示下一条要执行的语句。
- Step Into: 单步调试, 进入被调函数内。
- Step Over: 单步调试, 跳过被调函数。
- Step Out: 从被调函数中跳出, 执行下一条语句。
- Run to Cursor: 运行到当前光标处。

第二行自左至右各个工具项按钮的功能如下:

- Quick Watch: 快速查看当前的调试状态。
- Watch: 打开一个独立窗口, 用来显示用户要查看的变量值和类型, 当用户输入变量名时, 调试程序自动显示变量的值和类型。
- Variables: 打开一个独立窗口, 该窗口内有 3 个标签, 分别用来显示当前语句和上一条语句所用的变量、正在执行函数的局部变量以及 this 指针所指的对象的信息。
- Registers: 打开一个独立窗口, 显示 CPU 各个寄存器的状态。
- Memory: 打开一个独立窗口, 显示内存的当前状态。
- Call Stack: 打开一个独立窗口, 显示当前语句调用的所有函数, 当前函数在窗口顶部。
- Disassembly: 打开一个独立窗口显示反汇编代码。

1.1.3 菜单栏

VC 6.0 主窗口的菜单栏中包含 9 个主菜单项, 从左到右依次为文件、编辑、查看、插入、工程、编译、工具、窗口和帮助。

下面简单介绍这些菜单项的功能。

1. 文件菜单

打开文件菜单, 将出现如图 1-7 所示的下拉菜单项, 其中共有 14 个选项。

该菜单的各种命令选项的主要功能是用来对文件进行创建、打开、关闭、保存和打印等操作。

(1) 新建(New)命令。选择该命令后, 出现如图 1-8 所示的对话框。

该对话框是用来创建文件、工程、工作区以及其它文档的。它有 4 个标签: 文件(Files)、工程(Projects)、工作区(Workspaces)和其它文档(Other Documents)。

① 文件标签显示出可创建的文件类型, 包含的文件类型有:

- 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++源程序文件)。
- Cursor File(光标文件)。
- HTML Page(HTML 页文件)。
- Icon File(图标文件)。

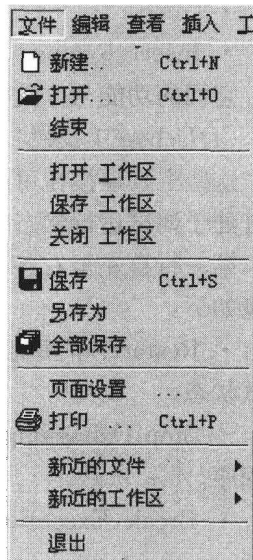


图 1-7 文件菜单

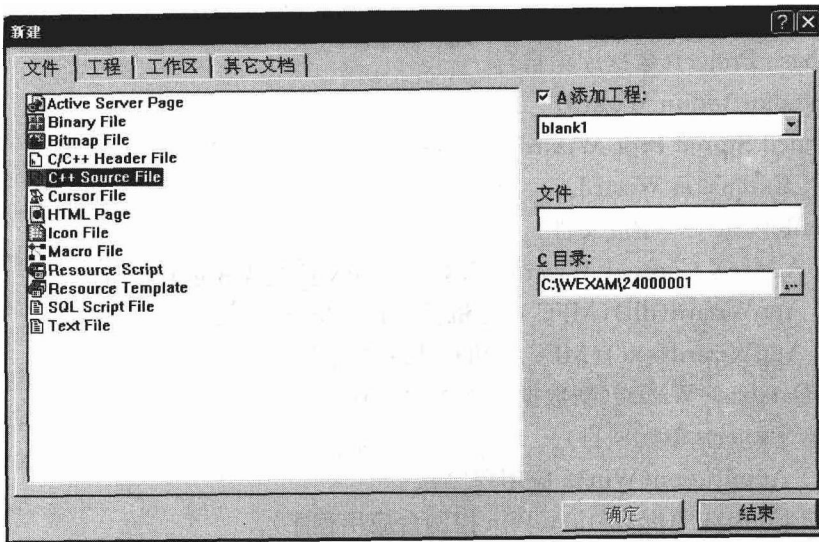


图 1-8 “新建”对话框

- Macro File(宏文件)。
- Resource Script(资源脚本文件)。
- Resource Template(资源模板文件)。
- SQL Script File(SQL 脚本文件)。
- Text File(文本文件)。

选择某种文件类型后,在对话框右边的 File name 文本框内键入要创建的文件名。

② 工程标签显示出各种可供选择的项目类型,如图 1-9 所示。

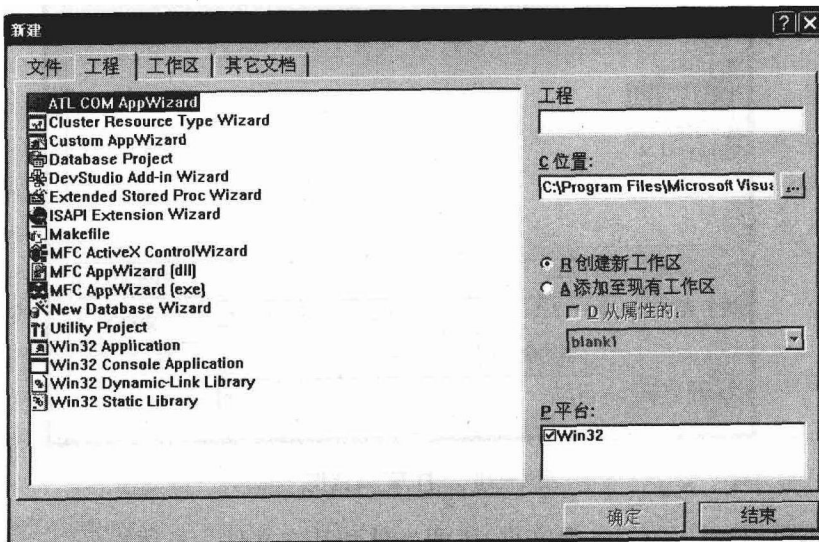


图 1-9 工程标签窗口

可供选择的项目类型有:

- ATL COM AppWizard(ATL 应用程序创建向导)。
- Cluster Resource Type Wizard(簇资源类型创建向导)。

- Custom AppWizard(自定义的应用程序创建向导)。
- Database Project(数据库项目)。
- DevStudio Add-in Wizard。
- Extended Stored Proc Wizard。
- ISAPI Extension Wizard。
- Makefile(C/C++生成文件)。
- MFC ActiveX Control Wizard(MFC ActiveX 控制程序创建向导)。
- MFC AppWizard(dll)(MFC 动态链接库创建向导)。
- MFC AppWizard(exe)(MFC 可执行程序创建向导)。
- New Database Wizard(新数据库创建向导)。
- Utility Project(单元项目)。
- Win32 Application(Win32 应用程序)。
- Win32 Console Application(Win 控制台应用程序)。
- Win32 Dynamic-Link Library(Win32 动态链接库)。
- Win32 Static Library(Win32 静态库)。

选择某种项目类型后,在该对话框右边的 Project name 文本框中输入项目名,在 Location 框内输入或修改项目所在路径。

该对话框的工作区标签和其它文档标签可以创建各种类型的工作区文件和文档,在这里就不作详细介绍了。

(2)打开(Open)命令。选择该命令后,弹出“打开”对话框,如图 1-10 所示。

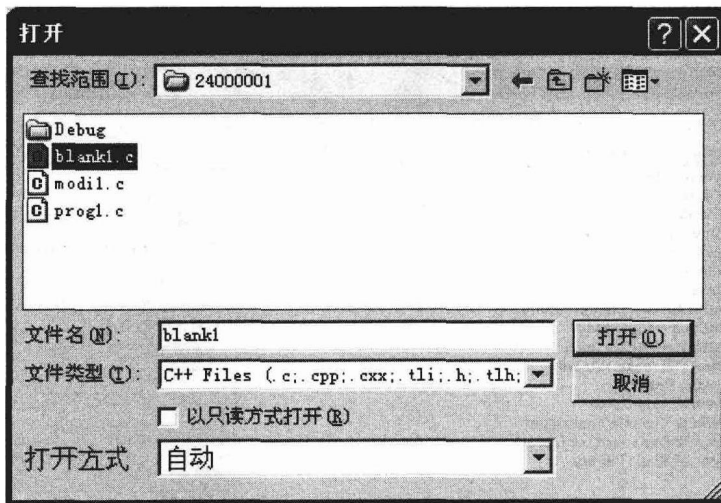


图 1-10 “打开”对话框

该对话框可用来打开 C++ 源文件、工程文件和其它文件。具体操作方法是:先在“搜寻”栏中选定要打开的文件的的路径,再通过“文件类型”栏指定要打开文件的类型。此时,在文件名列表框中会出现所要选的文件名。双击文件名或单击文件名再单击“打开”按钮,都可以打开所选的文件。

(3)关闭(Close)命令。该命令用来关闭在活动窗口中打开的文件。若该文件修改后尚

未保存,系统会提示用户是否保存该文件。

(4)打开工作区(Open Workspace)命令。选择该命令将弹出“Open Workspace”对话框,用来打开工作区的文件,也可打开其它文件。

(5)关闭工作区(Close Workspace)命令。该命令用来关闭当前工作区的文件,选择该命令后,弹出一个对话框,提示用户是关闭所有文件还是保留这些文件,选择“是”关闭所有文件,选择“否”保留这些文件。

(6)保存(Save)命令。该命令用于保留当前窗口中的文件内容,并存放到原文件名中。如果该文件是未命名的新文件,则系统提示“Save As”对话框。

(7)另存为(Save As)命令。该命令用来将已打开的文件保存到一个新的文件名中。选择该命令,出现“Save As”对话框,用户可将新的文件名输入到该对话框中的文件名文本框中。

(8)全部保存(Save All)命令。该命令用来保存当前窗口中所有被打开文件的内容。如果某个文件尚未命名,系统将会提示用户先输入文件名。

(9)页面设置(Page Setup)命令。该命令用来设置和格式化打印结果。选择该命令后,出现“Page Setup”对话框,使用该对话框为打印文档设置标题、脚注和边距等。

(10)新近的文件(Recent Files)命令。该命令用来显示最近打开过的4个文件,单击该文件名,可以将该文件打开。

(11)退出(Exit)命令。该命令用来退出 VC 6.0 编译系统。在退出系统前,应将打开的文件保存。

2. 编辑菜单

该菜单的功能是对文档进行编辑和搜索,如撤销/恢复命令(Redo/Undo)、剪贴命令(Cut/Copy/Paste)等。常用命令如下:

(1)查找(Find)命令。该命令的功能是用来在当前打开的文件中查找指定的字符串。选择该命令后,出现“Find”对话框。在该对话框的 Find What 文本框中输入要查找的字符串,在 Direction 框内选择查找方向为向前(Up)或向后(Down)。在如下4个复选框内根据需要进行选择:

- Match whole word only(仅匹配整个单词)。
- Match case(区分大小写字母)。
- Regular expression(正确表达式)。
- Search all open Documents(搜索所有打开的文档)。

(2)定位(Go To)命令。该命令用来指定如何将光标移到当前活动窗口的指定位置。选择该命令后弹出 Go To 对话框。

(3)书签(Bookmarks)命令。该命令用来设置、命名、删除和读取书签。

(4)断点(Breakpoint)命令。该命令用来设置、删除和查看断点。断点可分为位置断点、数据断点、消息断点和条件断点。断点将告诉调试器何时何处中断程序的执行,以便检查程序代码、变量和寄存器的值。

(5)List Members 命令。该命令用来列出当前光标处对象类属的成员。可以通过双击相应成员名,将该成员添加到光标处。

(6)Parameter Info 命令。该命令用来显示当前光标处函数的参数信息,可在书写函数

调用时提供参考,保证调用函数的实参与形参的一致性,从而减少错误。

3. 查看菜单

该菜单包含调试信息和控制屏幕显示方式的命令项。常用命令如下:

- (1)全屏幕(Full Screen)命令。该命令用来使源代码编辑区扩大到全屏幕。
- (2)工作区(Workspace)命令。该命令用于显示项目工作区窗口。
- (3)输出(Output)命令。该命令用于显示数据输出窗口,在编译时,该窗口将会显示出编译信息,包括出错信息。
- (4)调试窗口(Debug Window)命令。选择该命令出现级联菜单,在级联菜单中列出了调试窗口的若干操作。

4. 插入菜单

该菜单可以创建新类、资源、窗体,并将它们插入到文档中,也可以将文件作为文本插入到文档中,还可以添加新的 ATL 对象到项目中。常用命令如下:

- (1)资源拷贝(Resource Copy)命令。该命令用来复制选定的资源。
- (2)File As Text 命令。该命令用来选择插入到文档中的文件。
- (3)新建 ATL 对象(New ATL Object)命令。该命令用来启动 ATL Object Wizard,将添加新的对象到项目中。

5. 工程菜单

该菜单用于对工程和工作区的管理。可以选择指定工程为工作区中的当前(活动)工程,也可以将文件、文件夹等添加到指定的工程中去,还可以编辑和修改工程间的依赖关系。常用命令如下:

- (1)设置活动工程(Set Active Project)命令。该命令用来选择当前活动工程。
- (2)添加工程(Add To Project)命令。该命令用来将新文件或已有文件或部件及控制添加到指定的工程中去。
- (3)插入工程到工作区(Insert Project into Workspace)命令。该命令用来将工程插入到工作区中。

6. 编译菜单

该菜单包括用于编译、连接和运行应用程序的命令。常用命令如下:

(1)编译(Compile)命令。该命令用于编译显示在源代码编辑窗口中的源文件。在编译过程中,检查源文件中是否有语法错误。如果发现错误(warning 或 error),则将错误信息显示在输出窗口中。使用鼠标双击某行错误信息时,将在源代码编辑窗口中用粗箭头指向出错的代码行,方便用户修改。

(2)构建(Build)命令。该命令用来创建当前文件项目。该命令实际上包含了对源文件或项目的编译和连接,最终生成可执行文件。如果被创建的文件或项目已被编译,则该命令将用来连接,生成可执行文件。在编译或连接中检查出语法错误时,将出错信息显示在输出窗口中,用户修改后,再进行创建,直到生成可执行文件为止。

(3)重建全部(Rebuild All)命令。该命令用来对所有文件进行重新编译、连接,包含已编译过的文件。因此,此项操作耗时稍长。

(4)执行(Execute)命令。该命令用来运行已生成的可执行文件,并将运行结果显示到相应的环境中(例如 MS-DOS, Windows 98 或 Windows NT 等)。

(5)开始调试(Start Debug)命令。选择该命令出现级联菜单,选取该菜单项便可启动调试器。这时,将用 Debug 菜单项代替 Build 菜单项。

下面是 Debug 菜单项:

(1)Go 命令。该命令用在调试过程中,从当前语句启动或者继续运行。

(2)Restart 命令。该命令将重新装载程序到内存中,并且将放弃所有变量的当前值。

(3)Stop Debugging 命令。该命令将中断当前调试过程,并返回到原来的编辑状态。

(4)Step into 命令。该命令用来设置单步执行程序。当程序执行到某一函数调用语句时,进入该函数体,并从第一行语句开始单步执行。

(5)Step over 命令。该命令也是单步操作命令,不同的是当程序执行到某一函数调用语句时,不进入该函数体内,直接执行该调用语句,然后停在该调用语句后面的语句。

(6)Step out 命令。该命令用来在单步执行时从某个函数体内跳出,调试该函数调用语句后面的语句。该命令与 Step into 命令配合使用,先用 Step into 命令将单步执行植入某函数体内,发现不需要对该函数体内进行单步调试时,使用该命令跳出该函数体。

(7)Quick Watch 命令。选择该命令,将弹出 Quick Watch 对话框,通过该对话框可以查看和修改变量和表达式,或将变量和表达式添加到 Watch 窗口中。

7. 工具菜单

该菜单中的命令用来浏览用户程序中定义的符号、定制菜单与工具栏,激活常用的工具或更改选项和变量的设置。

8. 窗口菜单

该菜单的命令用来进行有关窗口的操作,常用命令如下:

(1)新建窗口(New Window)命令。该命令为当前工程文件打开一个新窗口。

(2)拆分(Split)命令。该命令用于将活动窗口进行分区。

(3)关闭(Close)命令。该命令用来关闭当前打开的窗口。

(4)全部关闭(Close All)命令。该命令用来关闭所有的窗口。

(5)后窗(Next)命令。该命令用来显示下一个窗口。

(6)前窗(Previous)命令。该命令用来显示前一个窗口。

9. 帮助菜单

帮助菜单用来获得大量的帮助信息。用户应该养成遇到问题查 Help 的习惯。

1.1.4 工作区

1. 工程(Project)

工程是一些相互关联的源文件的集合,这些源文件组成一个程序,它们被编译、连接后生成一个可执行文件。

创建一个工程后,可以添加任何其他目录的文件到该工程中。添加文件到工程中并不改变文件的位置,而工程只是记录文件的名字和位置。

2. 工程工作区(Project Workspace)

在 VC 语言中,文件、工程和工程配置是由工作区组织起来的。工作区的内容和设置通过工作区文件(.dsw)来描述,在建立一个工作区文件的同时,还生成工程文件(.dsp)和工作区选项文件(.opt),用来保存工作区的设置。

3. 工作区窗口

工作区窗口用来查看和修改工程中的所有元素。该窗口的底部提供了 3 种面板,即类面板(Class View)、资源面板(Resource View)和文件面板(Files View)。它们的功能如下:

(1)类面板。该面板在工作区窗口中显示该工程中所有类及其成员函数。单击“+”号,打开树形结构的每一项,显示出某类的成员函数和数据成员。双击某一项,在右边的源代码编辑窗口中显示该成员的源代码。如果该源代码已被显示,则使用光标进行指示。

(2)文件面板。该面板在工作区窗口中显示工程中的所有文件及其相互关系。单击“+”号,依次打开树形结构的每一页,显示出所有的资源文件、头文件和源代码文件。双击某一项,会在右边的源代码编辑窗口内打开该文件,显示其源代码。

(3)资源面板。该面板在项目工作区窗口中显示项目中的所有资源。单击“+”号,依次打开树形结构的某一项,显示出所有资源,包括字符串表、对话框图符及其版本信息。

1.2 在 Visual C++ 中编辑并运行一个 C 程序

C 语言程序的上机步骤是:输入程序→编辑源程序→对源程序进行编译→连接→运行。在这个过程中,源程序在编译时会检查程序中有没有语法错误,如果没有语法错误,则顺利产生一个扩展名为 .obj 的目标程序;目标程序进行连接,连接时会检查程序有没有连接错误,如果没有连接错误,则产生一个 .exe 的可执行程序;运行可执行程序,如果没有运行错误,则可以看到程序的运行结果。所以,在编译运行程序时,要改正程序的语法错误、连接错误和运行错误,才能顺利看到程序的正确运行结果。

1. 编辑

在 VC 6.0 中编写 C 程序,操作步骤如下:

- (1)单击“文件”菜单的“新建”命令,弹出如图 1-8 所示的“新建”对话框。
- (2)单击对话框中的“文件”标签,选择 C++ Source file 选项。
- (3)在窗口中输入程序。
- (4)单击“文件”菜单的“保存”命令,或单击工具栏中的“保存”按钮,弹出如图 1-11 所示的“保存为”对话框。选择保存路径,输入文件名,注意文件的扩展名为“.c”。

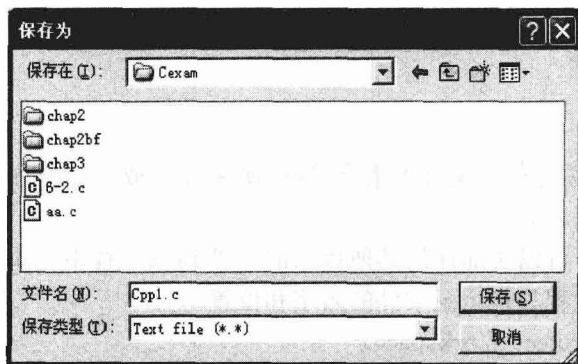


图 1-11 “保存为”对话框