

高等学校电子信息学科“十三五”规划教材

C语言程序设计

—— 实验与案例

主编 周信东

高等学校电子信息学科“十三五”规划教材

C 语言程序设计

——实验与案例

主 编 周信东

副主编 帅剑平 侯杏娜 陈辉金

西安电子科技大学出版社

内 容 简 介

本书紧密配合教材内容,兼容了大部分C语言程序设计教材的章节编排顺序,可供实践教学和上机选用。书中设计的实验内容涵盖了C程序的开发环境、数据类型与表达式、控制结构、数组与字符串、函数、指针、结构体和共用体、位运算、文件操作等。全书包含14个C语言基础实验和5个典型综合案例,提供了近100个实验题目,旨在提高读者分析问题、解决问题、程序实现的能力。

本书从初学者的角度出发,循序渐进地组织和安排实验内容,具有突出重点、化解难点、注重编程能力的培养等特点。书中主要介绍了Visual C++环境下的实验和调试方法,附录中还介绍了跨平台软件Code::Block的使用,方便读者在其它平台上进行实验。此外,还给出了C语言常用库函数、常见错误信息等内容,方便读者在实验过程中查阅要应用的函数或解决调试错误中遇到的问题。

本书具有基础性、实用性、系统性,适合作为高等院校“C语言程序设计”课程的实验教材,也可供报考计算机等级考试者和其他自学者参考。

图书在版编目(CIP)数据

C语言程序设计:实验与案例 / 周信东主编. — 西安:西安电子科技大学出版社,2018.8

ISBN 978-7-5606-5048-7

I. ① C… II. ① 周… III. ① C语言—程序设计 IV. ① TP312.8

中国版本图书馆CIP数据核字(2018)第174776号

策划编辑 陈 婷

责任编辑 胡 婷 陈 婷

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

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2018年8月第1版 2018年8月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 11.5

字 数 271千字

印 数 1~4000册

定 价 26.00元

ISBN 978-7-5606-5048-7 / TP

XDUP 5350001-1

***** 如有印装问题可调换 *****

前 言

C语言是一种通用的、模块化的编程语言，由于它具有高效、灵活、功能丰富、表达力强和可移植性好等特点，因此被广泛应用于系统软件和应用软件的开发之中，深受程序员的青睐。“C语言程序设计”始终是高等学校的一门基本的计算机课程，也是众多青年学生跨入“程序设计”殿堂的首选语言。然而，由于C语言涉及面广、内容丰富、使用灵活，要学好它并不容易。C语言程序设计是一门实践性很强的课程，不仅要学习基本概念、方法、语法规则，更重要的是要进行大量的上机实践，只有通过大量实践体验，积累编程经验，才能真正提高程序设计的能力。作者结合多年来的教学实践经验和体会，编写了本书作为课程配套的实验教程，以指导学生上机实践。

本书分为三大部分：C语言上机指导、C语言实验安排、典型程序设计案例分析。本书强调实用性、系统性，因此对三大部分内容的编排有如下考虑。

(1) 内容全面、系统，适用于高校C语言程序课程实验教学。本书在实验内容选取上包含了C语言的数据描述、程序控制结构、数组、字符处理、函数、指针、结构体和联合体等基本内容，还包含了位运算、文件等提高内容，这些内容分布在14个实验中，教师可根据教学需要每次选用1~2个实验内容安排。本书针对“C语言程序设计”这门课程的主要内容安排了比较全面、系统的实验项目，有助于读者全面认识和掌握C语言；在内容安排上由易到难、由浅入深，有利于实验的渐进实施和重要内容的消化掌握；在编排顺序上尽量与大部分高校C语言程序设计教材的章节编排顺序保持一致，适合作为“C语言程序设计”的配套实训教程。

(2) 从初学者的角度出发，循序渐进组织、安排内容，突出重点，化解难点，注重编程思维的培养。本书通过上机指导、实验安排、典型案例分折三大部分，使读者先了解和熟悉实验上机环境，然后经过实验项目训练，再发展到分析应用复杂程序设计。所有基础实验，均包含了实验目的和要求、知识要点、实验案例、实验内容和课外练习五部分内容。“实验目的和要求”明确规定了实验的目标和要求；“知识要点”对实验涉及的基本知识和常用算法等进行阐述；“实验案例”给出知识点的应用例程，并对例程进行问题分析，提出编程思路并给出源程序代码、运行结果，供教师上课时讲解分析，也帮助读者能更好地理解所学的知识；“实验内容”部分为安排的课内实验项目，这些题目又分为基础题、增强题和提高题三类，供不同学习层次的读者选作，也为教师布置实验内容增加了选择的余地；“课外练习”的题为难度较大的题目，供编程能力较强的读者课外选作。

(3) 突出实用性, 提高程序分析和调试能力。学好 C 语言的前提是加强上机实践, 解决实际问题, 掌握一些实用技术。本书在实验中穿插安排了 VC++ 环境下的 C 语言程序设计排错和调试技术, 在前几个实验中安排了大量的程序改错题, 目的是通过改错题的练习, 让读者掌握排查程序错误的方法和程序的调试技巧, 提高读者分析问题、解决问题的能力, 同时通过调试程序更深层次地了解程序的运行过程, 进一步理解 C 语言各种语句的用法及其语法规则和常用算法。

(4) 注重编程练习, 提高综合编程能力。本书从实验一开始到实验十四的“综合程序设计”, 由浅入深逐步安排了编程题练习, 始终将编程作为 C 语言实验的主要学习目标, 同时拓展了 C 语言程序设计的一些常用算法。在数组、函数、指针、结构体和联合体等实验中引入了常用排序算法, 在位运算和文件实验中引入了简单加/解密算法, 实验十四安排了难度较大的综合练习题目, 并在本书第三部分安排了 5 个典型的应用案例及其分析, 相当数量的编程训练和典型应用程序分析有利于拓展读者思路, 提高读者综合编程的能力, 最终学会用 C 语言编制程序解决实际问题。

此外, 在附录中安排了跨平台软件 Code::Block 上机指南, 让读者可以在不兼容 VC++ 的系统下安装使用; 安排了常见错误处理及常见函数库, 方便读者在做实验过程中遇到问题时查阅。

本书凝集着桂林电子科技大学多位从事 C 语言程序设计实验教学老师的辛勤汗水, 书中介绍的实验方法都是由几位老师从多年的实践教学总结提炼得来的, 绝大多数实验题目都经过了实践验证。本书第 2.1、第 2.7~2.10 节及 3.3 节由帅剑平编写; 第 2.2~2.6 节及 3.2 节由侯杏娜编写; 第 2.11~2.14 节及第 3.1、3.4 节由周信东编写; 第一部分及附录资料的收集和整理由陈辉金完成; 全书由周信东统稿。

本书具有基础性、实用性、系统性, 同时充分考虑了和其它教材内容的兼容性, 适合作为高等院校“C 语言程序设计”课程的实验指导书, 也可以作为读者学习 C 语言的参考资料。

由于作者水平有限, 加上写作时间仓促, 不当之处在所难免, 恳请读者批评、指正。

编者

2018 年 6 月于桂林电子科技大学

目 录

第一部分 C 语言上机指导	1
1.1 在 Visual C++ 环境下运行 C 语言程序	1
1.2 Visual C++ 环境下的程序调试	6
1.3 C 语言程序设计的基本步骤	12
1.4 程序设计风格	13
第二部分 C 语言实验安排	16
2.1 实验一 C 语言的数据描述和上机环境	16
2.2 实验二 顺序结构基本操作	22
2.3 实验三 分支结构程序设计	28
2.4 实验四 循环结构程序设计	36
2.5 实验五 数组的使用	42
2.6 实验六 字符处理	47
2.7 实验七 函数的使用	51
2.8 实验八 指针(一)——指针的定义和引用	57
2.9 实验九 指针(二)——指针数组和函数指针	61
2.10 实验十 结构体和联合体	64
2.11 实验十一 编译预处理	74
2.12 实验十二 位运算	80
2.13 实验十三 文件	85
2.14 实验十四 综合程序设计	91
第三部分 典型程序设计案例分析	96
3.1 图书管理系统	96
3.2 电子通讯录管理系统	120
3.3 独轮车问题	133
3.4 输出万年历	139

3.5 贪吃蛇游戏.....	144
附录.....	153
附录 1 在 Code::Blocks 下运行 C 语言程序.....	153
附录 2 C 语言程序常见错误类型及解决方法.....	160
附录 3 Visual C++ 常见错误信息对照表.....	164
附录四 C 语言常用标准库函数.....	172
参考文献.....	178

第一部分 C 语言上机指导

1.1 在 Visual C++ 环境下运行 C 语言程序

1.1.1 Visual C++ 6.0 概述

Visual C++ 是 Microsoft 公司推出的基于 Windows 平台的可视化编程环境，它不仅提供控制台应用程序供用户学习和运行标准 C 程序，而且提供“可视”的资源编辑器与 MFC 类以及应用程序向导，为快速、高效开发 Windows 应用程序提供了极大的方便。此外，利用 Visual C++ 还可进行 Internet、数据库及多媒体等多方面的程序开发。由于其功能强大、具有良好的界面和可操作性，被用户广泛应用。2000 年以后，微软全面转向 .NET 平台，Visual C++ 6.0 成为支持标准 C/C++ 规范的最后版本。

Visual C++ 6.0 分为标准版、专业版和企业版 3 种，其基本功能是相同的。下面以企业版为编程环境，对如何在 Visual C++ 环境下调试和运行标准 C 程序进行简单的介绍。

1.1.2 进入 Visual C++ 工作环境

1. 建立个人文件夹

第一次上机时先在本地盘(如 E: 盘)上建立一个以自己学号命名的文件夹，如：1800340136，然后按以下步骤进行操作。

2. 启动 Visual C++ 6.0

双击 Windows 桌面上的 Visual C++ 6.0 图标或单击 Windows 桌面上的“开始”按钮，在“程序”中选择“Microsoft Visual C++ 6.0”运行即可。

1.1.3 编辑、编译、运行标准 C 程序

上机操作的整个过程如图 1-1 所示。

1. 创建一个新的工程文件(Project file)

(1) 在 Visual C++ 的主菜单中单击“文件”菜单，在下拉菜单中选择“新建”命令，屏幕上出现一个“新建”对话框，在该对话框中选择“工程”标签，出现“工程”对话框。创建新的工程文件如图 1-2 所示。

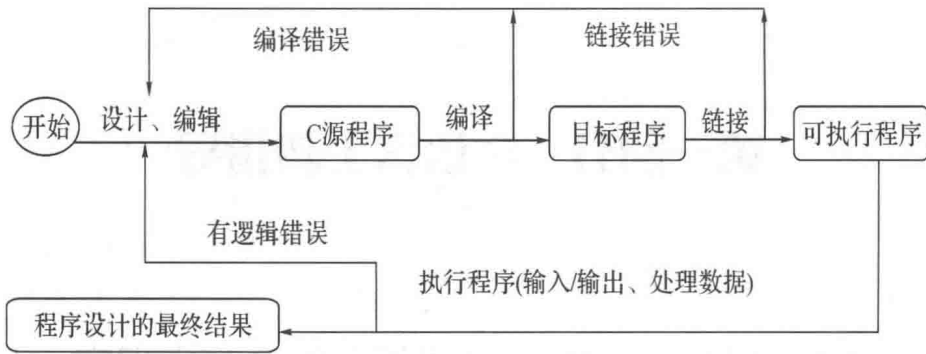


图 1-1 一个 C 程序的上机过程

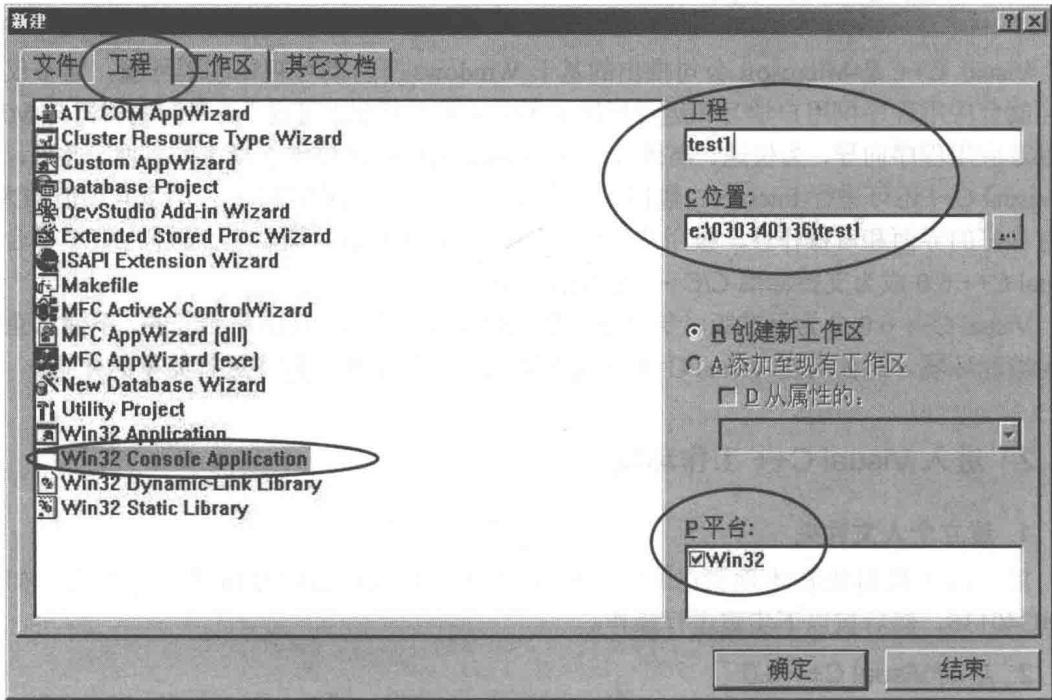


图 1-2 创建新的工程文件

(2) 选择工程类型为“Win32 Console Application”，这时，在右边的“平台”选框中就会出现 Win 32。

(3) 输入工程名字。在“工程”选框中输入所指定的工程文件名字，例如：test1。

(4) 输入路径名。在“位置”选框中，输入你将要把所建立的工程文件放入何处的路径名。例如，要将工程文件放在 E：盘下已建立好的子目录 e:\1800340136 中，则该选取的路径为：e:\1800340136\test1。选择“确定”按钮，该工程文件已建立。

(5) 在出现的“Win32 Console Application”对话框中选择“An empty project”，建立一个空的工程文件，并点击“完成”按钮。界面如图 1-3 所示。

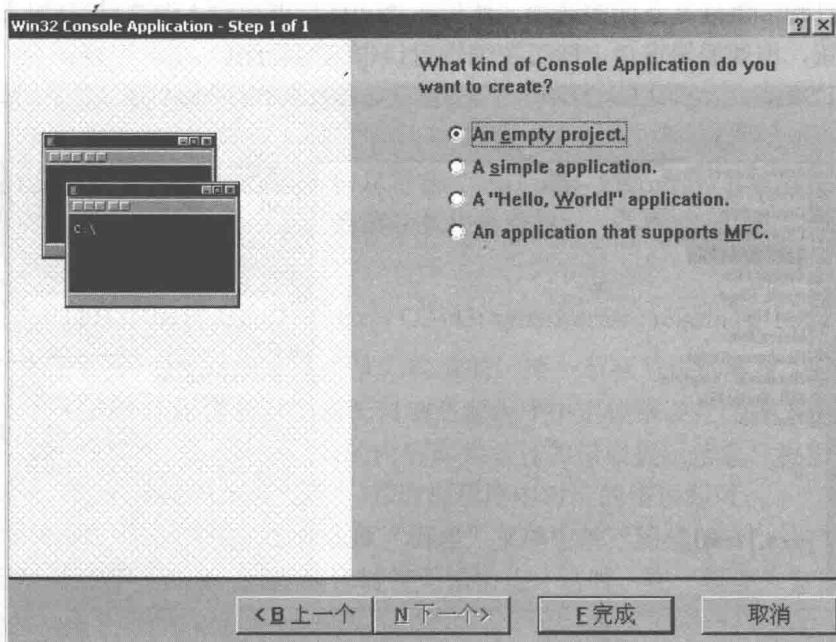


图 1-3 建立空工程

(6) 系统弹出“新建工程信息”对话框，如图 1-4 所示。点击“确定”按钮，即可完成工程的创建。

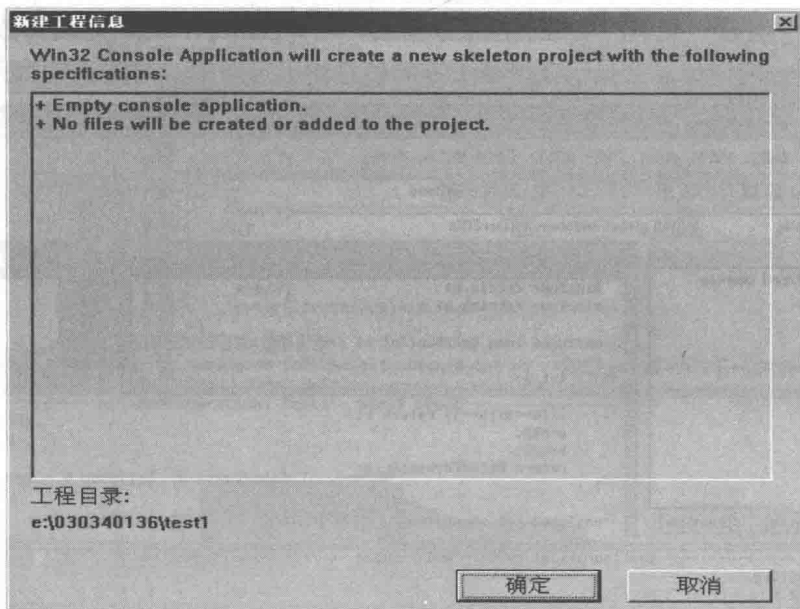


图 1-4 确认工程信息

2. 建立源文件(Source file)

选择“文件”菜单中的“新建”选项，出现如图 1-5 所示的对话框，在 4 个标签中选择“文件”标签，在其中的对话框选项中，选择“C++ Source File”并在右边“添加工程”

的选择框内打钩，激活其下面的选项，然后在“文件”框内输入源文件名(如 1st.c)，单击“确定”按钮，出现编辑窗口，即可在编辑窗口中编写程序。

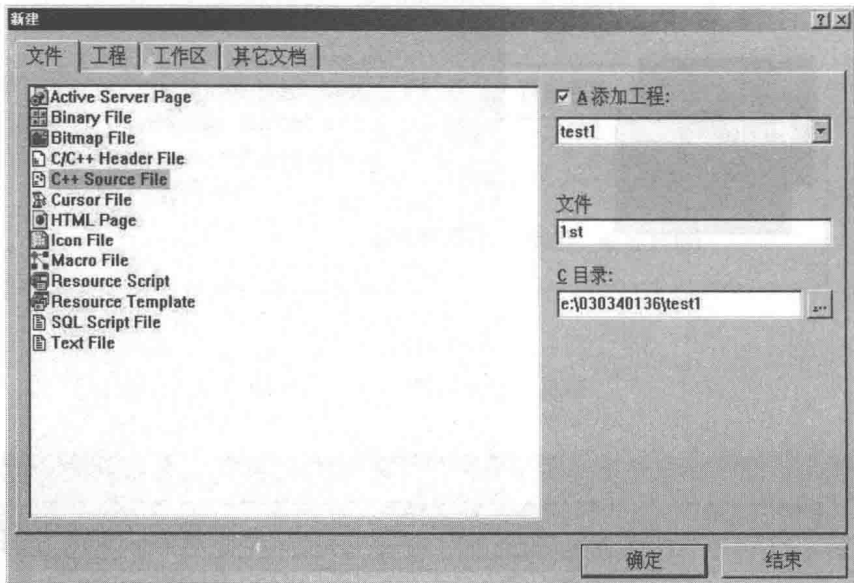


图 1-5 新建源文件

3. 编译链接和运行源程序

程序编好后要进行编译、链接和运行，具体步骤如下：

(1) 选择“组建”菜单，单击下拉菜单中的“编译[1st.c]”，这时系统开始对当前的源程序进行编译，编译信息会显示在屏幕下方的信息输出窗口中，如图 1-6 所示。

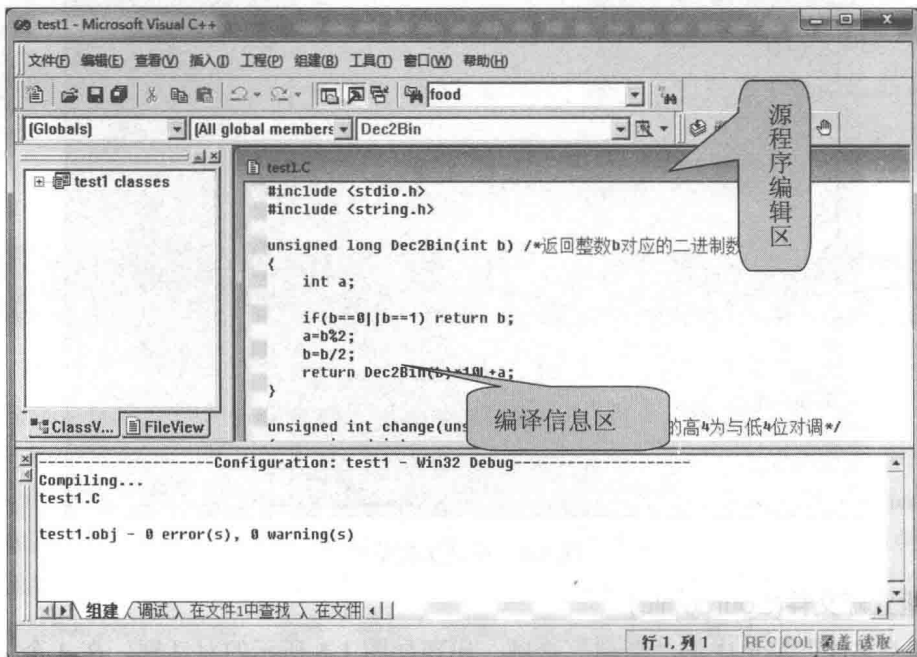



图 1-6 编辑、编译源文件

编译信息的最后一行显示类似“test1.obj - 0 error(s), 0 warning(s)”的信息。若“error(s)”前的数字为 0，表示程序编译没有错误，否则表示存在错误，须根据错误提示修改程序中的错误后再重新编译。如重新编译后还有错误，再继续修改、编译，直到没有错误为止。

信息输出窗口中的编译错误提示信息行在“test1.obj - 1 error(s), 0 warning(s)”语句之前显示，可用鼠标点击信息输出窗口的滚动条往前查找。

错误信息提示行显示如：

```
e:\030340136\test1\lst.c(37) : error C2143: syntax error : missing ';' before 'if'
```

信息行是由冒号(:)分隔的多个“节”构成的，第一节括号中的数字表示错误所在程序中的行号，可双击本错误信息行由系统自动在源程序中定位错误行(在代码编辑窗口错误行前会出现  图形指示)；最后一节的内容即为该行的错误提示信息，根据提示信息(可参照本书附录“常见错误信息对照表”)修改源程序中对应的错误即可。

(2) 编译无误后进行链接，这时选择“组建”菜单中的“组建[test1.exe]”选项。同样要对出现的错误要进行更改，直到编译链接无误为止。这时，在“组建”窗口中会显示如下信息：test1.obj- 0 error(s), 0 warning(s)(如图 1-7 所示)，说明编译链接成功，并生成和工程文件名同名的可执行文件 test1.exe。



图 1-7 编译链接

(3) 运行程序，选择“组建”菜单中的“! 执行 [test1.exe]”选项。这时会出现一个“MS-DOS”窗口，输出结果显示在该窗口中，如图 1-8 所示。



图 1-8 程序运行的输出结果窗口中的结果

(4) 运行结束后，可以回到“文件”菜单，点击“关闭工作空间”选项，关闭当前文件窗口。若要编辑新的源程序，可以再次打开“文件”菜单，重新建立工程文件，重复前述步骤；也可以点击“文件”菜单中的“打开工作空间”选项，打开一个已经存在的源文件。

1.2 Visual C++ 环境下的程序调试

在编写较长的程序时，一次性成功而不含有任何错误绝非易事，这需要进行长期、大量的练习。对于程序中的错误，VC 提供了易用且有效的调试手段。

1. 设置调试器

VC 程序在编译时可生成为 Debug 和 Release 两种版本。Debug 通常称为调试版本，它包含调试信息，并且不作任何优化，体积比较大，便于程序员调试程序。Release 称为发布版本，它往往进行了各种优化，使得程序在代码大小和运行速度上都是最优的，以用户使用。

要调试一个程序，首先必须使程序包含调试信息。若是第一次启用调试器，为了增加调试信息，可以按照下述步骤进行调试器的设置。

(1) 通过 IDE 菜单“Project(工程)|Settings(设置)”或直接按 Alt+F7 快捷键打开 Project Settings 对话框。

(2) 选择 C/C++ 页，在“Category(分类)”中选择“General(常规)”，则会出现一个“Debug Info(工程选项)”列表框，可供选择的调试信息方式如表 1-1 所示。

表 1-1 VC 调试器选项设置

命令行	Project Settings	说 明
无	None	没有调试信息
/Zd	Line Numbers Only	目标文件或者可执行文件中只包含全局和导出符号以及代码行信息, 不包含符号调试信息
/Z7	C 7.0- Compatible	目标文件或者可执行文件中包含行号和所有符号调试信息, 包括变量名及类型、函数及原型等
/Zi	Program Database	创建一个程序库(PDB), 包括类型信息和符号调试信息
/ZI	Program Database for Edit and Continue	除了前面/Zi 的功能外, 这个选项允许对代码进行调试过程中的修改和继续执行。这个选项同时使 #pragma 设置的优化功能无效

(3) 选择“Link(链接)”页, 勾选复选框“Generate Debug Info(产生调试信息)”, 这个选项将使链接器把调试信息写进可执行文件和 DLL。

如果 C/C++ 页中设置了 Program Database 以上的选项, 则“Link incrementally(增量编译)”可以选择。若选中这个选项, 程序则可以在上一次编译的基础上被编译, 而不必每次都从头开始编译。


2. 修正语法错误

调试最初的任务主要是修正一些语法错误, 这些错误包括以下 3 点。

- (1) 未定义或不合法的标识符, 如函数名、变量名和类名等。
- (2) 编写的程序语句不合法, 如使用的前后括号或引号不匹配、语句末尾缺省分号等。
- (3) 数据类型或参数类型及个数不匹配。

对于这些语法错误, 可对程序进行编译。若程序中存在上述错误, 编译后会在信息输出窗口中列出所有错误项, 根据系统提供的错误信息进行修改。

为了能快速定位到错误产生的源代码位置, VC 提供下列一些方法。

(1) 在信息输出窗口中双击某个错误, 或将光标移到该错误处按 Enter 键, 则该错误被高亮显示, 状态栏上显示出错误内容并定位到相应的代码行中, 且该代码行最前面有个蓝色箭头标志 。

(2) 在信息输出窗口中的某个错误项上右击鼠标, 在弹出的快捷菜单中选择“Go To Error/Tag(转到错误/标记)”命令。

(3) 按 F4 键可显示下一错误, 并定位到相应的源代码行。

(4) 若将光标移到信息输出窗口中的错误编号上, 按 F1 键可启动 MSDN 并显示出错误的内容, 从而帮助用户理解错误产生的原因。

语法错误被修正并对程序重新编译和链接后, 信息输出窗口中会出现类似“XXXX.exe - 0 error(s), 0 warning(s)”的提示信息, 说明程序编译和链接已经通过。

3. 设置断点

程序编译和链接无误后, 并不是此项目就完全没有错误, 可能还存在“异常(Exception)”、

“断言(Assertion)”等其他错误，而这些错误在编译时是不会显示的，只有当程序运行后才会出现，此类错误需要通过调试程序来进行分析和修正。

设置断点是调试程序的常用手段，如图 1-9 所示。

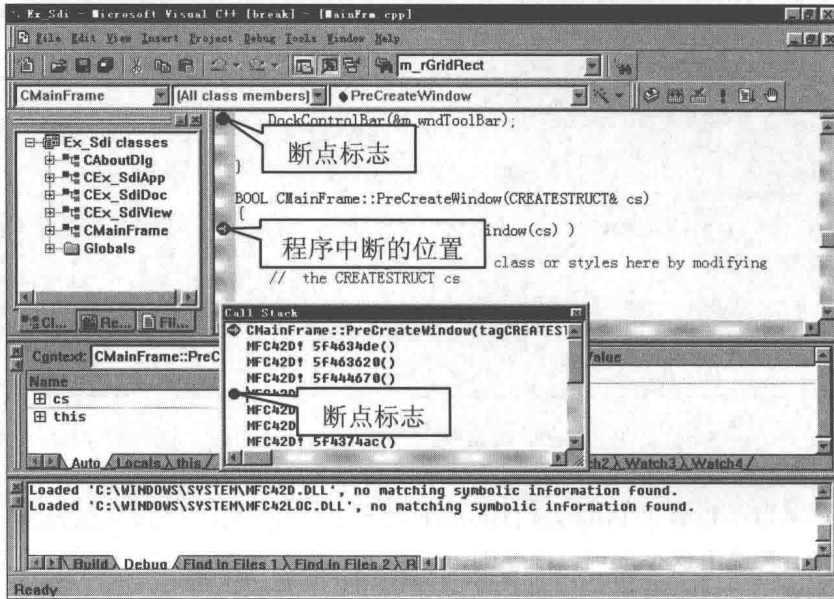



图 1-9 设置的断点

调试的断点可以有如下类型：位置断点、数据断点、条件断点。

(1) 通过以下 3 种快捷方式可以设置位置断点。

- ① 先将光标定位在需要设置断点的程序行位置，再按快捷键 F9。
- ② 先将光标定位在需要设置断点的程序行位置，再在 Build 工具栏上单击  按钮。
- ③ 在需要设置断点的程序行位置右击，选择“Insert | Remove Breakpoint”命令。

(2) 通过 IDE 菜单“编辑|断点...”打开 Breakpoints 对话框，如图 1-10 所示，也可以设置断点。

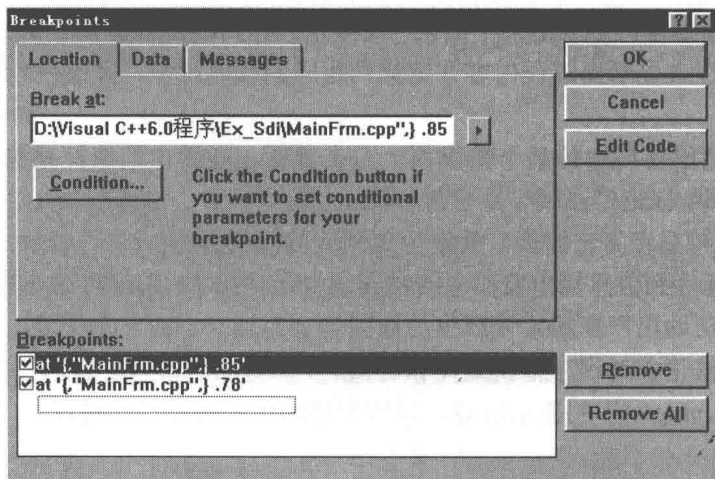


图 1-10 “Breakpoints”对话框

① 凡是可用的断点前均有选中标记(√)。若单击前面的复选框,则该断点被禁止。用按钮[Remove]和[Remove All]分别可清除当前选中的断点或全部断点。

② 在 Location 页面的“Break at”文本框中输入断点的名称,单击[Edit Code]按钮可以查看断点位置处的源代码或目标代码,单击[Condition]按钮,可以输入程序运行中断所需要的表达式条件,从而设置一个条件断点。

③ 在 Data 页面中,VC 提供了一种设置数据断点的方法,如图 1-11 所示。在 Data 页面最上面的编译框中,可以键入任何有效的 C/C++ 表达式,它可以是赋值语句、条件语句或是单独的一个变量名。在程序运行过程中,若变量的值有所改变,或条件表达式变成真时,则程序在该断点处中断。

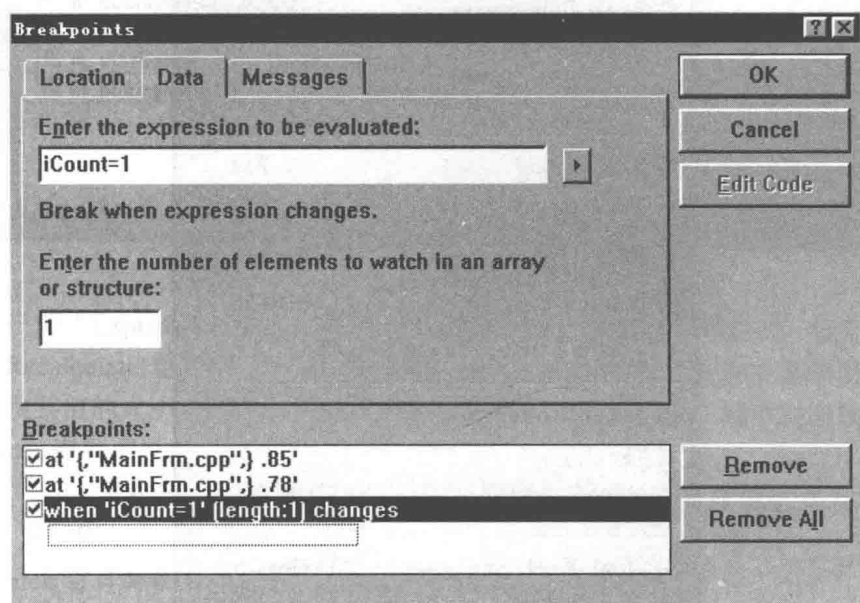


图 1-11 Data 页面

4. 启用调试器

选择“组建 | 开始调试”菜单的“Go”、“Step Into”或“Run to Cursor”命令可以启动调试器,进入调试(Debug)状态。

5. 控制程序运行

当程序开始在调试状态下运行时,程序会由于断点而停顿下来。这时可以看到有一个小箭头指向即将执行的代码。而且,原来的“组建(Build)”菜单就会变成“调试(Debug)”菜单,如图 1-12 所示。

Debug 菜单中有 4 条命令 Step Into、Step Over、Step Out 和 Run to Cursor 是用来控制程序运行的,其含义分别是:

(1) Step Into: 运行当前箭头指向的语句。如果当前箭头所指的语句中包含函数调用,则进入该函数内部进行单步执行函数中的语句。

(2) Step Over: 运行当前箭头指向的语句。此命令只运行当前箭头指向的一条语句,即使语句中包含函数调用,也不进入该函数内部进行单步执行,而是直接返回函数的执行

结果。

(3) Step Out: 如果当前箭头所指向的语句是在某一函数内, 用它使程序运行至函数返回处。

(4) Run to Cursor: 使程序运行至光标所在的代码行。

通过运用上述命令来控制程序的运行, 可了解程序语句执行的顺序, 也可通过观察(Watch)窗口观察程序执行过程中变量值的变化情况, 从而判断程序的执行是否正确。

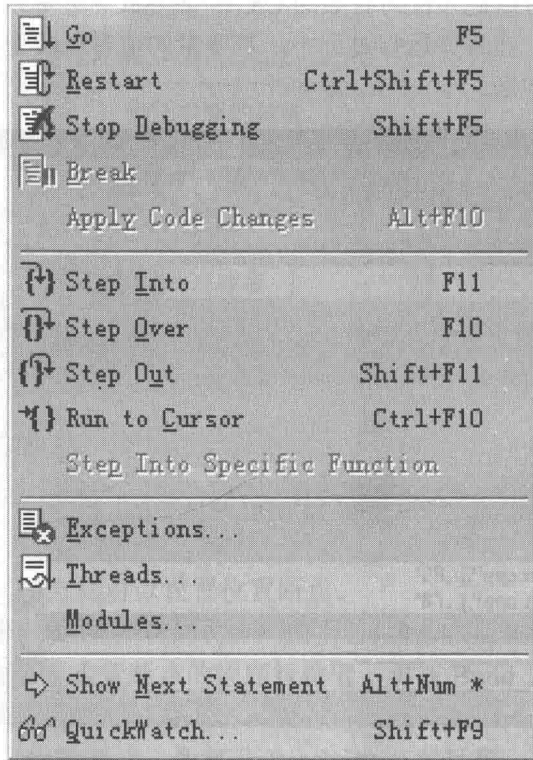


图 1-12 Debug 菜单

6. 查看和修改变量的值

调试器还提供一系列的窗口用来显示各种不同的调试信息。可借助“查看(View)”菜单下的“调试窗口(Debug Windows)”子菜单访问它们。当启动调试器后, VC 的开发环境会自动显示出 Watch 和 Variables 两个调试窗口, 且信息输出窗口自动切换到调试(Debug)页面。

1) QuickWatch 窗口的使用

QuickWatch 窗口可以用来帮助用户快速查看或修改某个变量或表达式的值。若仅需要快速查看变量或表达式的值, 只需要将鼠标指针直接放在该变量或表达式上, 片刻后系统会自动弹出一个窗口显示出该变量或表达式的值。

启动调试器后, 选择“调试(Debug)|QuickWatch”或按快捷键 Shift+F9 即可调出 QuickWatch 窗口, 如图 1-13 所示。