



“十二五”普通高等教育本科国家级规划教材

教育部—英特尔精品课程配套教材

辽宁省精品课程配套教材

高等学校计算机基础教育规划教材

程序设计基础（C语言） 实验指导与测试（第2版）

高克宁 李金双 焦明海 张昱 编著



清华大学出版社

013060403

TP312C
1689-2



“十二五”普通高等教育本科国家级规划教材

教育部—英特尔精品课程配套教材

辽宁省精品课程配套教材

高等学校计算机基础教育规划教材

程序设计基础（C语言） 实验指导与测试（第2版）

高克宁 李金双 焦明海 张昱 编著



清华大学出版社



北航

C1667784

TP312C

1689-2

内 容 简 介

本书作为《程序设计基础(C语言)》的配套教材,旨在帮助学生掌握程序设计的基本技能。

本书介绍了 Visual C++ 6.0 集成开发环境;精心设计了 15 个实验,每个实验由实验目的、实验指导和实验内容三部分构成,其中实验指导部分给出了详细的实验步骤和讲解,对自行开展实验活动具有重要的指导意义;提供了 7 个综合案例,每一个案例都给出了详尽的指导,有利于进一步培养学生的综合能力;另外,针对课程的学习给出了大量的基本概念测试题,包括选择题、阅读程序和程序填空三类题型,这对加深和巩固基本概念的理解有很大的帮助。

本书适合作为高等院校理工科专业教材,也可作为计算机培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础(C语言)实验指导与测试 / 高克宁等编著. --2 版. --北京:清华大学出版社, 2013.8
高等学校计算机基础教育规划教材
ISBN 978-7-302-33068-4

I. ①程… II. ①高… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 150220 号

责任编辑:袁勤勇

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:14 字 数:321 千字

版 次:2009 年 9 月第 1 版 2013 年 8 月第 2 版 印 次:2013 年 8 月第 1 次印刷

印 数:1~2000

定 价:28.00 元

产品编号:046003-01

前言

程序设计课程是一门实践性很强的课程,仅仅通过理论学习不足以完全掌握程序设计的精髓,必须通过大量的程序设计实践来提高对程序设计的认知。本实验教材作为《程序设计基础(C语言)》的配套教材,旨在帮助学生掌握程序设计的基本技能。

本书共5个部分。第一部分即第1章,重点介绍 Visual C++ 6.0 集成开发环境。第二部分即第2章,针对教材的15章精心地设计了15个实验,每个实验包括实验目的、实验指导和实验内容三部分,其中实验指导给出了详细的实验步骤和讲解,对自行开展实验活动具有重要的指导意义。第三部分即第3章,给出了7个综合案例,每一个案例都提供了详尽的指导,以进一步培养学生的综合实践能力。第四部分即第4章,针对课程的学习给出了大量的基本概念测试题,由基本概念、阅读程序和程序填空构成,对于加深和巩固C语言基本概念的理解有很大帮助。第五部分为附录,主要由常用C语言库函数、常见编程错误和实验报告样例构成。

参与本书编写的主要人员包括高克宁、李金双、焦明海、张昱、李封、李凤云、赵长宽、柳秀梅等老师,本书的基本概念测试参考答案见 <http://www.neu.edu.cn/cxsj>。

本书是教育部—英特尔精品课程配套教材、辽宁省精品课程配套教材,本书配套的主教材是“普通高等教育‘十二五’国家级规划教材”。

由于作者水平有限,书中难免会有错误或疏漏之处,真诚地欢迎各位专家和读者批评指正,以帮助我们进一步完善此教材。

作者的联系方式:

电子邮件: gkn@cc.neu.edu.cn

通讯地址:(110819)辽宁 沈阳 东北大学计算中心 高克宁

作 者

2013年4月 于东北大学

目录

第 1 章 Visual C++ 6.0 开发环境	1
1.1 Visual C++ 6.0 的主窗口	1
1.2 工程与项目工作区	2
1.3 集成开发环境的菜单栏	2
1.3.1 File 菜单	3
1.3.2 Edit 菜单	3
1.3.3 View 菜单	4
1.3.4 Insert 菜单	4
1.3.5 Project 菜单	5
1.3.6 Build 菜单	5
1.3.7 Tools 菜单	5
1.3.8 Window 菜单	6
1.3.9 Help 菜单	6
1.4 集成开发环境的工具栏	6
1.4.1 Standard 工具栏	7
1.4.2 Build MiniBar 工具栏	7
1.4.3 Debug 工具栏	8
1.5 程序调试	8
1.5.1 查找源程序中的语法错误	8
1.5.2 跟踪调试程序	9
1.5.3 调试器	10
第 2 章 C 语言实验	12
实验 1 熟悉实验环境	12
实验 2 数据类型、运算符和表达式	18
实验 3 基本输入输出	22
实验 4 顺序和条件控制语句	27
实验 5 循环控制语句	33
实验 6 数组	37

实验 7 函数	42
实验 8 指针	49
实验 9 结构体与共用体	53
实验 10 预编译处理和位处理	57
实验 11 文件	60
实验 12 程序设计思想及范例	62
实验 13 面向对象程序设计	68
实验 14 并行程序设计与 MPI	74
实验 15 个体软件开发过程管理(PSP)	80
第 3 章 综合案例设计解析	84
案例 1 城市人口数据处理系统	84
案例 2 数据滤波并行计算系统	91
案例 3 小型学生通讯录管理系统	94
案例 4 飞机订票信息系统	98
案例 5 个人财务管理系统	102
案例 6 小型超市管理系统	105
案例 7 图书借阅管理系统	108
第 4 章 基本概念测试	114
测试 1 数据类型与表达式	114
测试 2 控制结构	120
测试 3 数组	142
测试 4 函数	159
测试 5 指针	176
测试 6 结构体与预定义	189
测试 7 文件	194
附录 A 常用 C 语言库函数	199
附录 B 常见编程错误	204
附录 C 实验报告样例	212

第 1 章

Visual C++ 6.0 开发环境

Visual C++ 6.0 开发环境是一个基于 Windows 操作系统的可视化、面向对象的集成开发环境(Integrated Development Environment, IDE)。在该环境下用户可以开发有关 C 和 C++ 的各种应用程序。应用程序包括建立、编辑、浏览、保存、编译、链接和调试等操作,这些操作都可以通过单击菜单选项或工具栏按钮来完成,使用方便、快捷。它还提供了项目工作区(Workspace)、应用程序向导(AppWizard)、类操作向导(ClassWizard)和 WizardBar 等实用编程工具。

1.1 Visual C++ 6.0 的主窗口

在已安装 Visual C++ 6.0 的计算机上,可以直接从桌面双击 Microsoft Visual C++ 6.0 图标,进入 Visual C++ 6.0 IDE,或者单击“开始”→“程序”菜单,选择 Microsoft Visual Studio 6.0 中的 Microsoft Visual C++ 6.0 菜单项,进入 Visual C++ 6.0 IDE,如图 1-1 所示。



图 1-1 Visual C++ 6.0 集成开发环境

Visual C++ 6.0 集成开发环境主要由标题栏、菜单栏、工具栏、项目工作区、编辑区和输出区组成。

1.2 工程与项目工作区

在 Visual C++ 集成开发环境中,把实现程序功能的一组相互关联的 C++ 源文件、资源文件以及支撑这些文件的类的集合称为一个工程。工程是 Visual C++ IDE 开发程序的基本单位,一个工程至少包含一个工程文件,工程文件的扩展名为 dsp。工程文件保存了工程中所用到的源代码文件和资源文件的信息,如文件名和路径等。同时,工程文件还保存了工程的编译设置等信息,如调试版(debug)和发布版(release)。另外,根据工程类型的不同,一个工程包含有不同的源文件、资源文件和其他类别的文件。

Visual C++ IDE 以项目工作区来组织应用程序的工程,项目工作区文件的扩展名为 dsw,这种类型的文件在 Visual C++ 中级别是最高的。项目工作区含有工作区的定义和工程中所包含文件的所有信息。因此,要打开一个工程,只需打开对应的项目工作区文件(*.dsw)即可。

项目工作区窗格一般位于屏幕左侧,包含 ClassView(类视图)、ResourceView(资源视图)及 FileView(文件视图)三种视图。

ClassView 用于显示工程中定义的类。展开文件夹将显示工程中所有的类,包括系统生成的和用户自定义的。单击类名左边的+符号,就可以看到类的数据成员和成员函数,如图 1-1 所示。

ResourceView 用于显示工程中所包含的资源文件。展开文件夹可显示所有的资源类型,显示的资源类型包括 Accelerator(加速键)、Dialog(对话框)、Icon(图标)、Menu(菜单)、StringTable(串表)、Toolbar(工具条)、Version(版本)等。

FileView 用于显示所创建的工程。展开文件夹后可以看到工程中所包含的文件。除了查看,还可以管理文件,包括增加、删除、移动、重命名和复制文件等。单击文件类型左边的+符号可看到工程中该种类型的所有文件,双击一个文件即可打开该文件。一个应用程序工程主要包含实现源文件(*.cpp)、头文件(*.h)和资源文件(*.rc)等文件类型。

1.3 集成开发环境的菜单栏

Visual C++ IDE 中的菜单栏由 File、Edit、View、Insert、Project、Build、Tools、Window 和 Help 九个主菜单组成,如图 1-2 所示。每个主菜单又由多个菜单项和子菜单组成。菜单项几乎可以完成 Visual Studio 的所有功能。

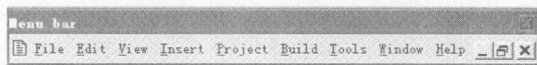


图 1-2 Visual C++ 6.0 IDE 的菜单栏

与 Windows 的操作方法相似,菜单选择可以通过两种方法来进行:一是鼠标单击所选的菜单;另一种是键盘操作,即通过相应的热键来选择。选中某个菜单后,就会出现相应的下拉式子菜单。在下拉式子菜单中,有些菜单项的右边对应着相应的快捷键。

除了主菜单,在 Visual C++ IDE 窗口中的不同地方右击,还可弹出相应的快捷菜单。另外,在 Visual C++ IDE 中,用户可以选择定制的方式开发自己需要的菜单组合,并加上不同的功能。

1.3.1 File 菜单

File 菜单主要包括一些与文件有关的操作命令,其主要的菜单项及快捷键和功能说明如表 1-1 所示。

表 1-1 File 菜单中的主要菜单项和功能

菜 单 项	快捷键	功 能 说 明
New	Ctrl+N	创建一个新的文件或者工程
Open	Ctrl+O	打开一个已存在的文件
Close		关闭当前打开的文件
Open Workspace		打开一个已经存在的工作区
Save Workspace		保存当前打开的工作区
Close Workspace		关闭当前工作区
Save	Ctrl+S	保存当前文件
Save As		以新文件名保存当前文件
Save All		保存所有打开的文件
Page Setup		页面设置
Print	Ctrl+P	打印文件的全部或选定部分
Recent Files		最近的文件列表
Recent Workspace		最近的工作区列表
Exit		退出 IDE

1.3.2 Edit 菜单

Edit 菜单主要包括一些与文件编辑有关的操作命令,它的作用是为用户提供了一种编辑当前打开文件的手段。Edit 菜单中的主要菜单项及快捷键和功能说明见表 1-2。

表 1-2 Edit 菜单中的主要菜单项和功能

菜 单 项	快捷键	功 能 说 明
Undo	Ctrl+Z	撤销上一次编辑操作
Redo	Ctrl+Y	恢复被撤销的编辑操作
Cut	Ctrl+X	剪切选定的内容,并移到剪贴板中
Copy	Ctrl+C	将被选定的内容复制到剪贴板中
Paste	Ctrl+V	将剪贴板中的内容粘贴到当前位置
Delete	Del	删除选定的文本或者光标后面的字符

菜单项	快捷键	功能说明
Select All	Ctrl+A	一次性选定窗口中的全部内容
Find	Ctrl+F	在当前文件中查找指定的字符串
Find in Files		在多个文件中查找指定的字符串
Replace	Ctrl+H	替换指定的字符串
Go To	Ctrl+G	将光标转移到指定的位置
Bookmarks	Alt+F2	给文本文件加标签
Breakpoints	Alt+F9	在指定位置设定断点

1.3.3 View 菜单

View 菜单中的菜单命令项主要用于改变窗口的显示方式和激活指定的窗口。它的作用是让用户设置窗口的显示方式,并提供观察调试的窗口。View 菜单中的主要菜单项及快捷键和功能说明见表 1-3。

表 1-3 View 菜单中的主要菜单项和功能

菜单项	快捷键	功能说明
Class Wizard	Ctrl+W	对工程中的类进行维护浏览
Resource Symbols		编辑资源文件中的符号常量
Resource Includes		编辑、修改资源符号文件名及预处理命令
Full Screen		在窗口的全屏方式和正常方式之间切换
Workspace	Alt+O	显示或隐藏工作区窗口
Output	Alt+2	显示或隐藏输出窗口
Debug Windows		调试程序所需的一些辅助查看命令
Refresh		更新选定的内容
Properties	Alt+Enter	编辑当前被选定对象的属性

1.3.4 Insert 菜单

Insert 菜单主要包括一些与添加有关的操作命令,用于向工程中插入类、资源、对象等。Insert 菜单中的主要菜单项及快捷键和功能说明见表 1-4。

表 1-4 Insert 菜单中的主要菜单项和功能

菜单项	快捷键	功能说明
New Class		创建新类并加入到工程中
New Form		创建新窗体并加入到工程中
Resource	Ctrl+R	创建各种资源并加入到工程中
Resource Copy		将所选资源的副本加入到工程中
File As Text		将所选文件以文本形式插入到文件中

1.3.5 Project 菜单

Project 菜单主要包括一些与工程管理有关的操作命令。Project 菜单中的主要菜单项及快捷键和功能说明见表 1-5。

表 1-5 Project 菜单中的主要菜单项和功能

菜 单 项	快捷键	功 能 说 明
Set Active Project	Alt+F7	从多个工程中设置当前活动工程
Add To Project		在当前工程中增加新文件、文件夹等
Source Control		源代码控制管理选项
Settings		设置当前工程中的各种相关配置
Insert Project into Workspace		将已存在工程插入到当前工作区窗口中

1.3.6 Build 菜单

Build 菜单主要包括一些与建立可执行程序有关的操作命令，能够提供编译、链接、调试等功能。Build 菜单中的主要菜单项及快捷键和功能说明见表 1-6。

表 1-6 Build 菜单中的主要菜单项和功能

菜 单 项	快捷键	功 能 说 明
Compile	Ctrl+F7	编译源代码文件
Build	F7	编译、链接工程，生成可执行文件
Rebuild All		编译和链接所有文件及资源，生成可执行文件
Batch Build		一次编译和链接多个工程
Clean		清除工程的中间文件和输出文件
Start Debug	Ctrl+F5	开始或继续调试程序
Execute		运行程序，如未生成可执行文件则先编译链接
Configurations		编辑工程的配置

1.3.7 Tools 菜单

Tools 菜单中的菜单命令主要用于启动 Visual Studio 6.0 的实用工具和定制 Visual C++ 6.0 IDE 界面。Tools 菜单中的主要菜单项及快捷键和功能说明见表 1-7。

表 1-7 Tools 菜单中的主要菜单项和功能

菜 单 项	快捷键	功 能 说 明
Source Browser	Alt+F12	查询当前选定对象和上下文的信息
Customize		定制集成开发环境的用户界面
Options		改变集成开发环境的各项设置
Macro		运行、编辑和录制宏

1.3.8 Window 菜单

Window 菜单主要包括一些与窗口显示有关的操作命令, Window 菜单中的主要菜单项及快捷键和功能说明见表 1-8。

表 1-8 Window 菜单中的主要菜单项和功能

菜单项	快捷键	功能说明
New Window	Alt+F12	为当前文档打开一个新的窗口
Split		分隔窗口
Docking View	Alt+F6	切换窗口组合显示的开关
Close		关闭当前打开的窗口
Close All		关闭所有打开的窗口
Cascade		以层叠的方式显示所有窗口
Tile Horizontally		水平平铺所有窗口
Tile Vertically		垂直平铺所有窗口
Windows		管理当前打开的窗口

1.3.9 Help 菜单

Help 菜单包括了有关 Visual C++ 6.0 帮助的操作命令, Help 菜单中的主要菜单项及快捷键和功能说明见表 1-9。

表 1-9 Help 菜单中的主要菜单项和功能

菜单项	快捷键	功能说明
Contents		启动 MSDN, 显示所有帮助信息的内容列表
Search		显示搜索联机文档的界面
Index		显示索引联机文档的界面
Keyboard Map		显示所有键盘命令
About Visual C++		显示应用程序信息和版本号

1.4 集成开发环境的工具栏

虽然通过菜单可以调用各种命令, 但是菜单的操作相对比较繁琐, 而快捷键又需要用户记忆, 所以有时候还显得不够方便。Visual C++ 6.0 提供了一批预先定义好的工具栏, 只要单击其中相应的按钮就能访问那些经常使用的命令。

Visual C++ 6.0 IDE 的工具栏以停靠窗口的形式出现, 工具栏的位置可以通过鼠标拖曳的方法来改变, 并可以根据需要在显示与隐藏之间进行切换。一般的方法是执行

Tools 菜单中的 Customize 菜单命令,打开 Customize 对话框,然后单击 Toolbars 标签,并根据需要在 Toolbars 列表中选择要放到工具栏上的按钮图标,如图 1-3 所示。也可以在菜单栏或工具栏的空白处右击,然后在一个弹出式菜单中选择要显示或隐藏的工具,如图 1-4 所示。

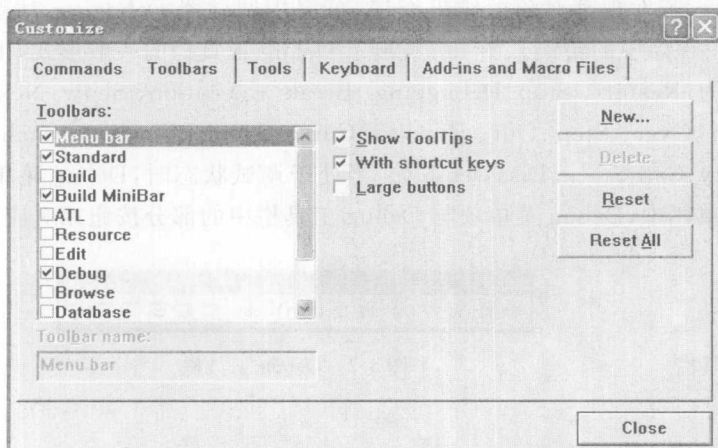


图 1-3 在 Customize 对话框中设置要显示的工具栏

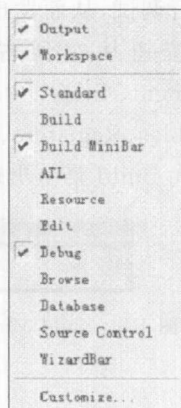


图 1-4 在弹出菜单中设置要显示的工具栏

如果要恢复工具栏或菜单栏到系统原来的默认状态,则在 Customize 对话框中单击 Reset 或 Reset All 按钮即可。

1.4.1 Standard 工具栏

Standard 工具栏中包括一些与文件和编辑有关的常用操作命令,每个按钮的功能与文件(File)、编辑(Edit)和查看(View)等主菜单中的某个菜单项对应。图 1-5 给出了 Standard 工具栏的一般形式,Standard 工具栏按钮从左到右依次对应的菜单项为 New Text File、Open、Save、Save All、Cut、Copy、Paste、Undo、Redo、Workspace、Output、Window List、Find in Files、Find、Search。

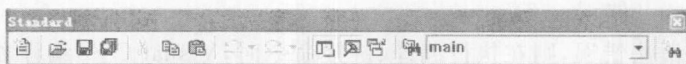


图 1-5 Standard 工具栏

1.4.2 Build MiniBar 工具栏

Build MiniBar 工具栏中主要包括工程的选择、编译、链接和调试等操作命令,每个按钮的功能与编译和调试菜单中某个菜单项对应。图 1-6 给出了编译微型条工具栏(Build MiniBar)的一般形式,Build MiniBar 工具栏按钮从左到右依次对应的菜单项为 Compile、

Build、Stop Build、Execute Program、Go、Insert/Remove Breakpoint。

1.4.3 Debug 工具栏

当程序编译、链接完成后,若发现某对象的结果有误,可启用调试器 Debugger 对程序进行调试,从而发现并改正程序中的错误。图 1-7 给出了 Debug 工具栏的一般形式,工具栏按钮从左到右依次为 Restart、Stop Debugging、Break Execution、Show Next Statement、Step Into、Step Over、Step Out、Run to Cursor、QuickWatch、Watch、Variables、Registers、Memory、Call Stack、Disassembly。当处于调试状态时,Debug 菜单会取代 Build 菜单出现在菜单栏中,Debug 菜单项与 Debug 工具栏中的部分按钮相对应。

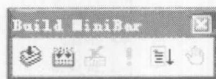


图 1-6 Build MiniBar 工具栏

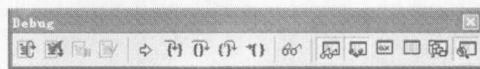


图 1-7 Debug 工具栏

1.5 程序调试

程序调试是程序设计中一个很重要的环节,一个程序一般要经过很多次调试才能保证基本正确。程序调试分为源程序语法错误的修改和程序逻辑设计错误的修改两个阶段,编译器只能找出源程序的语法错误,程序的逻辑设计错误只能靠程序员利用调试工具手工检查和修改。

1.5.1 查找源程序中的语法错误

语法错误分为一般错误(error)和警告错误(warning)。当出现 error 错误时将不会生成可执行程序,而出现 warning 错误时能够生成可执行程序,但程序运行时可能发生错误,严重的 warning 错误还会引起死机现象。warning 错误比 error 错误更难于修改,应该尽量消除 warning 错误。

如果程序有语法错误,则在执行编译、链接命令时,Visual C++ 编译器将在输出窗口中给出语法错误提示信息,但链接错误提示信息不能给出错误发生的具体位置。

在输出窗口中双击错误提示信息或按 F4 键可以返回到源程序编辑窗口,并通过一个箭头符号定位到产生错误的语句。

在运行阶段,当运行环境检测到一个不可能执行的操作时,也会发生错误。例如,除以零的错误。在不可预料的事情发生时或者在程序中一条语句不能正常执行时,发生运行错误。这种错误在应用程序开始执行之后发生。

需要说明的是,编译器给出的错误提示信息可能不十分准确,并且一处错误往往会引出若干条错误提示信息。因此,修改一个错误后应该马上进行程序的编译。通过重复的

编译可使程序中的语法错误越来越少,直至所有的语法错误都被修改,从而得到一个可执行程序。

1.5.2 跟踪调试程序

即使源程序没有语法错误,但最后生成的可执行程序也没有像预想的那样运行,这类程序设计上的错误被称为逻辑设计错误或缺陷(bug)。跟踪调试程序是查找此类逻辑设计错误方法中最常采用的动态方法。跟踪调试的基本原理就是在程序运行过程的某一阶段观测程序的状态。在一般情况下程序是连续运行的,所以我们必须使程序在某一点停下来。首先,我们所做的第一项工作就是设立断点;其次,再运行程序,当程序在断点设立处停下来时,再利用各种工具观测程序的状态。程序在断点停下来后,有时需要按用户要求控制程序的运行,以便进一步观测程序的流向。

1. 设置断点

利用 Visual C++ 6.0 提供的集成调试环境,可以设置从简单到复杂的各种断点。选择 Edit 菜单下的 Breakpoints 菜单项后,系统显示 Breakpoints 对话框,如图 1-8 所示。从图中可以看到 Breakpoints 对话框包含三个选项卡,分别为按位置、按表达式的值和按窗口消息三种设置断点的方式。

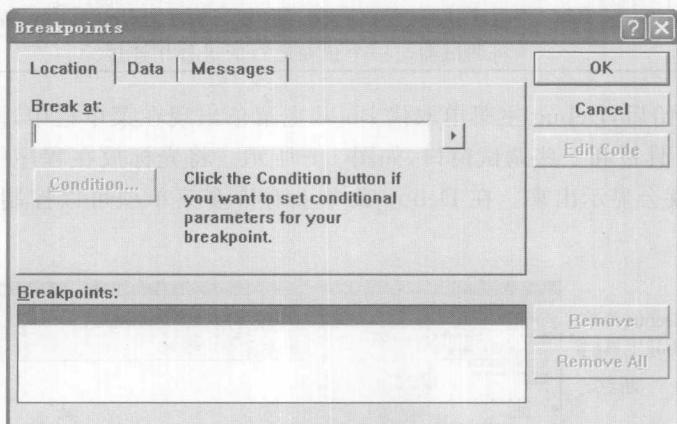


图 1-8 Breakpoints 对话框

Location 选项卡用于设置位置断点。该断点是最常用的一个无条件断点,也是默认的断点类型。程序执行时遇到这种断点,只是简单地停下来。

在 Build MiniBar 工具栏中,单击 Insert/Remove Breakpoint 按钮可以快捷地在某个程序行插入/移除断点。

2. 控制程序运行

当设置完断点后,程序就可以进入 Debug 状态,并按要求来控制程序的运行,其中有

四条命令：Step Over、Step Into、Step Out 和 Run to Cursor。

我们可以通过用鼠标单击工具栏按钮或使用热键来控制程序的运行。

3. 观察数据变化

在调试过程中,用户可以通过 Watch 和 Variable 窗口查看当前变量的值。这些信息可以反映程序运行过程中的状态变化以及变化结果的正确与否,可以反映程序是否有错,再加上人工分析,就可以发现错误所在。

1.5.3 调试器

为了查找和修改程序中的逻辑设计错误,Visual C++ IDE 提供了重要的调试工具——Debug。单击 Build→Start Debug 菜单中的菜单项,可以启动 Debug。Start Debug 子菜单中有 Go、Step Into、Run to Cursor 及 Attach to Process 菜单项,它们的功能见表 1-10。

表 1-10 Start Debug 菜单中的菜单项和功能

菜单项	快捷键	功能说明
Go	F5	开始或继续调试程序,到某断点、程序结尾或需用户输入的地方停止
Step Into	F11	单步执行程序的一个指令,能进入被调用的函数内部
Run to Cursor	Ctrl+F10	程序执行到当前光标处,相当于在光标处临时设置了一个断点
Attach to Process		将调试器与一个正在运行的进程相连接

调试过程开始后,Debug 主菜单取代 Build 主菜单出现在菜单栏中,同时出现一个可停靠的 Debug 工具栏和一些调试窗口,如图 1-9 所示。将光标放在程序中的某个变量名上,它的当前值就会显示出来。在 Debug 菜单上有许多菜单项可以控制程序的执行,见表 1-11。

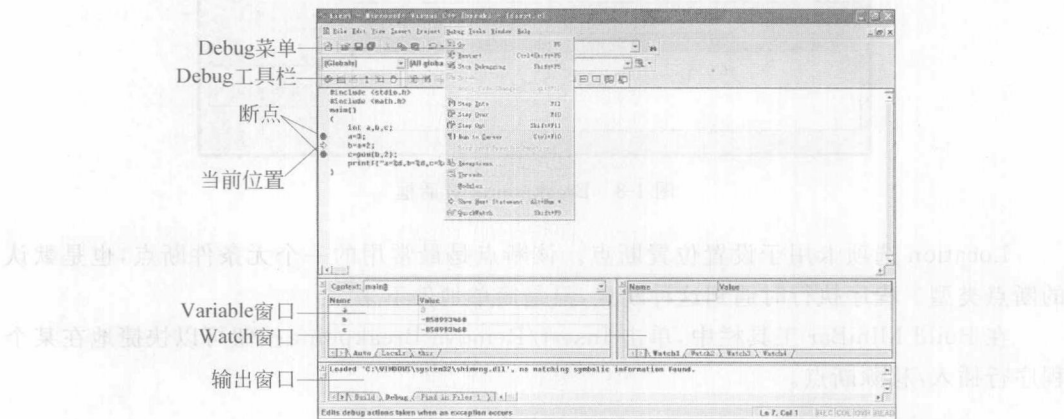


图 1-9 Debug 界面

表 1-11 Debug 菜单中的主要菜单项和功能

菜单项	快捷键	功能说明
Go	F5	与 Start Debug 子菜单中的 Go 命令功能相同
Restart	Ctrl+Shift+F5	重新开始执行程序,并处于调试状态
Stop Debugging	Shift+F5	终止程序的调试,返回到程序编辑的状态
Break		在当前位置暂停程序的执行
Apply Code Changes	Alt+F10	在程序调试时应用代码更改
Step Into	F11	与 Start Debug 子菜单中的 Step Into 命令功能相同
Step Over	F10	单步执行,当遇到一个函数的调用时,该函数被执行,但并不进入该函数内部
Step Out	Shift+F11	运行到当前函数调用返回后的第一条语句;使用这个命令能在已确定错误不在当前函数中时,快速地执行完此函数
Run to Cursor	Ctrl+F10	与 Start Debug 子菜单中的 Run to Cursor 命令功能相同
Step Into Specific Function		执行程序的每一步,并进入指定的函数调用,这个命令可以跟踪任意多的嵌套函数
Show Next Statement	Alt+Num *	显示当前指令所处的代码位置
QuickWatch	Shift+F9	快速查看变量和表达式的值

Variable 窗口用于观察和修改变量的当前值,Debug 可根据当前程序运行过程中变量的变化情况自动选择应显示的变量。用户可以在 Variable 窗口的 Context 下拉框选择要查看的函数,然后 Debug 会在窗口中显示函数局部变量的当前值。该窗口中有三个选项卡,Auto 选项卡中显示当前语句或前一条语句中变量的值和函数的返回值,Locals 选项卡中显示当前函数局部变量的名称、值和类型,this 选项卡以树形方式显示当前类对象的所有数据成员,单击十号可展开 this 指针所指对象。

Watch 窗口用于观察和修改变量或表达式的值。它有 Watch1、Watch2、Watch3 和 Watch4 四个标签,在每个标签中,用户都必须手工设置要观察的变量或表达式。