



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

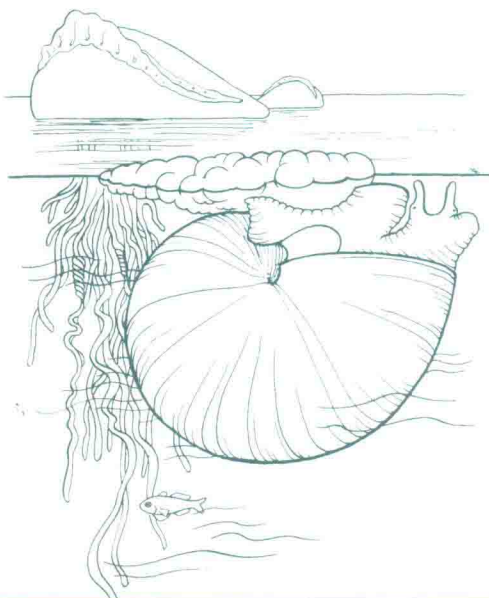
C语言程序设计 实验指导 (第2版)

The Practice of C Programming (2nd Edition)

马晓艳 许婷婷 马华 主编

张兰华 李聪 孙静 副主编

- 实验内容丰富
- 理论与实践结合
- 提升实际应用能力



高校系列



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化普
21 世纪高等学校计
21st Century University Planned Te

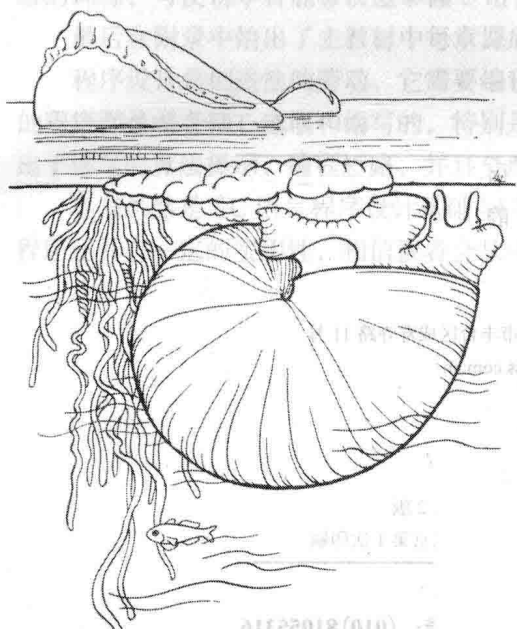
创教材立项项目

C语言程序设计 实验指导 (第2版)

The Practice of C Programming (2nd Edition)

马晓艳 许婷婷 马华 主编

张兰华 李聪 孙静 副主编



高校系列

人民邮电出版社

北京

图书在版编目(CIP)数据

C语言程序设计实验指导 / 马晓艳, 许婷婷, 马华主
编. — 2版. — 北京: 人民邮电出版社, 2015. 9
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-39721-8

I. ①C… II. ①马… ②许… ③马… 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(2015)第165780号

内 容 提 要

本书是《C语言程序设计基础》(第2版)配套使用的实验指导教材。

全书分为4个部分: 第1部分为 Visual C++6.0 上机指导, 详尽介绍了 VC++ 6.0 的编程环境、C 语言程序调试方法和编程风格。第2部分为基本实验, 共安排了15个实验, 在第1版的基础上扩展了实验题目, 以进一步巩固各章节的知识点。第3部分为综合性实验, 共安排了3个实验, 通过综合实验让学生熟练掌握结构化程序设计的思想和方法, 培养学生综合应用知识的能力。第4部分为提高性实验, 共安排了5个实验, 目的是希望读者巩固 C 语言的基本知识, 加强多知识点交叉实验的训练。最后在附录中给出了教材课后习题参考答案及提高性实验的参考程序。

本书实验安排恰当, 例题、习题丰富, 分析透彻, 与主教材配套, 便于自学。本书可作为高等学校“计算机程序设计”课程的实验教材, 也可作为各类考试和社会读者的自学辅助用书。

-
- ◆ 主 编 马晓艳 许婷婷 马 华
副 主 编 张兰华 李 聪 孙 静
责任编辑 许金霞
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 10.75 2015年9月第2版
字数: 281千字 2015年9月北京第1次印刷
-

定价: 28.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一，同时又是一门实践性很强的课程。为了提高本课程的教学质量，改善 C 语言难讲、难学、难以掌握的现状，我们配合教材《C 语言程序设计基础》(第 2 版)，编写了《C 语言程序设计实验指导》(第 2 版)。

我们在第 1 版的基础上增加了 Visual C++6.0 上机指导、C 语言程序的调试方法和 C 语言编程风格等内容，扩充了基本实验内容；还增加了提高性实验并附有参考程序。

本书共分为 4 部分。

第 1 部分为 Visual C++6.0 上机指导，首先介绍了 Visual C++6.0 开发环境，然后介绍了程序调试的方法，最后介绍了 C 语言的编程风格。

第 2 部分为基本实验。共安排了 15 个实验项目，每个实验项目包括：实验项目名称、实验目的和实验内容。实验目的指出对学生的知识要求和能力要求，详细指明了应该达到的具体目标。实验内容则包含 3 部分的内容：验证性实验、改错性实验和设计性实验。验证性实验训练学生阅读程序，熟悉相应章节的知识点，体会良好的程序书写风格。改错性实验主要训练学生根据出错信息运用程序调试方法进行程序调试，提高学生的程序调试能力。设计性实验给出题目和部分题目的解题提示，让学生自己设计算法，并用 N-S 图描述，然后编程并调试，提高学生运用知识解决问题的能力。

第 3 部分为综合性实验，通过综合性实验的设计，让学生熟练使用结构化程序设计思想和方法，培养学生综合应用知识的能力。

第 4 部分为提高性实验。希望读者通过练习，巩固 C 语言的基本知识，加强多知识点交叉实验的训练，可使初学者能够快速掌握 C 语言程序设计。

最后在附录中给出了主教材中每章课后习题的参考答案以及提高性实验的参考答案。

程序设计是创造性的劳动，它需要编程人员的全身心投入，需要充分发挥主观能动性。书中的程序都是编者精心选取和编写的，特别是部分改错题都给出了错误提示行，部分程序设计题给出了详尽的算法提示、编程思路，并且全部在 Visual C++ 6.0 环境下调试通过。

本书虽然是《C 语言程序设计基础》(第 2 版)教材的配套参考书，但也可以单独使用，有些程序还具有一定的实用性，相信读者会从中得到启发。

编 者

2015 年 4 月

目 录

第 1 部分 Visual C++6.0 上机指导.....1

1.1 Visual C++6.0 开发环境.....1

1.1.1 Visual C++6.0 集成开发环境简介.....1

1.1.2 启动 Visual C++6.0 集成开发环境.....1

1.1.3 开始一个新程序.....1

1.2 C 语言程序调试方法.....6

1.2.1 错误的类型.....6

1.2.2 查错的方法.....7

1.2.3 Visual C++ 6.0 中常用的 程序调试工具.....7

1.3 C 语言的编程风格.....8

1.3.1 编程风格.....8

1.3.2 变量名、函数名的命名规则.....9

1.3.3 注释.....10

第 2 部分 基本实验.....11

一、实验要求.....11

二、实验报告的内容.....11

实验 1 初识 C 程序.....11

一、实验目的.....11

二、实验内容.....12

三、实验注意事项.....14

实验 2 C 语言基础.....14

一、实验目的.....14

二、实验内容.....14

三、实验注意事项.....17

实验 3 顺序结构程序设计.....17

一、实验目的.....17

二、实验内容.....18

三、实验注意事项.....22

实验 4 选择结构程序设计.....22

一、实验目的.....22

二、实验内容.....22

三、实验注意事项.....30

实验 5 循环结构程序设计(一).....31

一、实验目的.....31

二、实验内容.....31

三、实验注意事项.....37

实验 6 循环结构程序设计(二).....37

一、实验目的.....37

二、实验内容.....37

三、实验注意事项.....45

实验 7 函数(一).....45

一、实验目的.....45

二、实验内容.....45

三、实验注意事项.....58

实验 8 函数(二).....58

一、实验目的.....58

二、实验内容.....58

三、实验注意事项.....63

实验 9 数组(一).....64

一、实验目的.....64

二、实验内容.....64

三、实验注意事项.....76

实验 10 数组(二).....76

一、实验目的.....76

二、实验内容.....77

三、实验注意事项.....85

实验 11 指针(一).....85

一、实验目的.....85

二、实验内容.....85

三、实验注意事项.....90

实验 12 指针(二).....90

一、实验目的.....90

二、实验内容.....91

三、实验注意事项.....97

实验 13 结构体.....98

一、实验目的.....98

二、实验内容.....98

三、实验注意事项.....102

实验 14 文件.....102

一、实验目的	102	一、填空题实验	131
二、实验内容	102	二、改错题实验	132
三、实验注意事项	106	三、设计型实验	134
实验 15 链表	107	提高性实验 4	135
一、实验目的	107	一、填空题实验	135
二、实验内容	107	二、改错题实验	136
三、实验注意事项	107	三、设计型实验	138
第 3 部分 综合性实验	108	提高性实验 5	139
实验 1 学生成绩管理	108	一、填空题实验	139
一、实验要求	108	二、改错题实验	140
二、实验提示	108	三、设计型实验	141
三、参考程序	109	附录 参考答案	143
实验 2 约瑟夫环问题	118	配套教材课后习题参考答案	143
一、实验要求	118	第 1 章 C 语言概述	143
二、实验提示	119	第 2 章 基本 C 语言程序设计	143
三、参考程序	119	第 3 章 选择结构程序设计	144
实验 3 双向链表的综合运算	120	第 4 章 循环结构程序设计	148
一、实验要求	120	第 5 章 函数	149
二、实验提示	120	第 6 章 数组	150
三、参考程序	120	第 7 章 指针	157
第 4 部分 提高性实验	124	第 8 章 结构体	159
提高性实验 1	124	第 9 章 文件	162
一、填空题实验	124	提高性实验参考答案	162
二、改错题实验	125	提高性实验 1 参考答案	162
三、设计型实验	126	提高性实验 2 参考答案	163
提高性实验 2	127	提高性实验 3 参考答案	164
一、填空题实验	127	提高性实验 4 参考答案	164
二、改错题实验	128	提高性实验 5 参考答案	165
三、设计型实验	130	参考文献	166
提高性实验 3	131		

第 1 部分

Visual C++6.0 上机指导

1.1 Visual C++6.0 开发环境

1.1.1 Visual C++6.0 集成开发环境简介

Visual C++ 6.0, 简称 VC 或者 VC6.0, 是微软推出的一款 C++ 编译器, 将“高级语言”翻译为“机器语言 (低级语言)”的程序。Visual C++ 是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++1.0 后, 随着其新版本的不断问世, Visual C++ 已成为专业程序员进行软件开发的首选工具。虽然微软公司推出了 Visual C++.NET (Visual C++7.0), 但它的应用有很大的局限性, 只适用于 Windows 2000、Windows XP 和 Windows NT4.0, 所以实际中, 更多的是以 Visual C++6.0 为平台。Visual C++6.0 不仅是一个 C++ 编译器, 而且是一个基于 Windows 操作系统的可视化集成开发环境 (integrated development environment, IDE)。Visual C++6.0 由许多组件组成, 包括编辑器、调试器以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。

1.1.2 启动 Visual C++6.0 集成开发环境

安装好微软的 Visual Studio 6.0 后, 在系统的“开始”菜单的“程序”中可以启动 Visual C++6.0, 见图 1-1, 也可以在桌面上为菜单项 Visual C++6.0 建立一个快捷图标, 如图 1-2 所示。可选择这两个方法之一启动 VC。

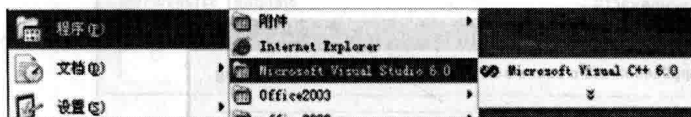


图 1-1 VC++启动



图 1-2 VC++图标

1.1.3 开始一个新程序

一、建立工程

启动 VC 环境后, 选择“File”菜单中的“New”项, 弹出图 1-3 所示的对话框。其标签项自动选择为“Projects” (工程)。作为初学者, 在左侧的列表中选择倒数第 3 项“Win32 Console

Application”(win32 控制台应用), 在右侧“Project Name”文本框中输入欲建立的工程名称, 如: Example01。在“Location”(位置)中选择工程的存放位置(最好为课程建一个总文件夹), 单击“OK”按钮, 系统出现建立工程的导航对话框, 如图 1-4 所示。

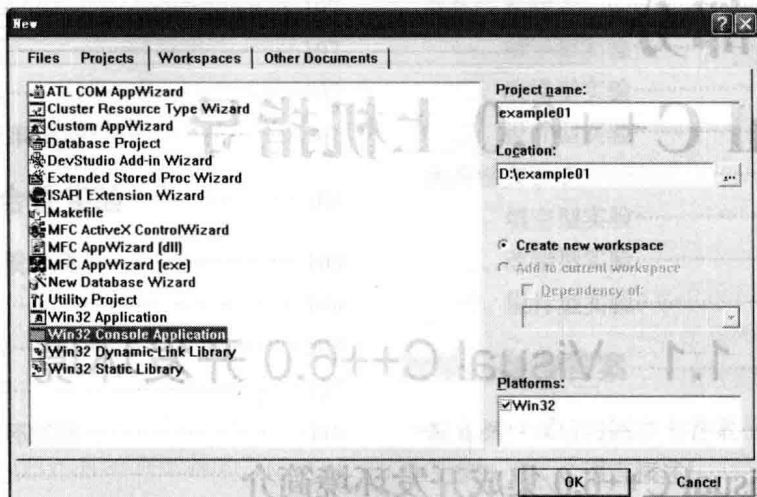


图 1-3 创建控制台应用

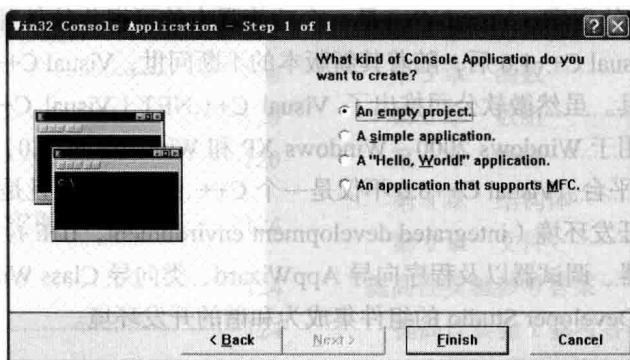


图 1-4 创建工程的导航对话框

在图 1-4 的导航对话框中选择默认的“An empty project”(空的工程), 然后单击“Finish”(完成)。系统弹出相关的创建工程的信息, 如图 1-5 所示。单击“OK”按钮后, 系统创建工程, 建立相关的文件夹, 不过这些文件夹都是空的, 系统界面类似于图 1-6。

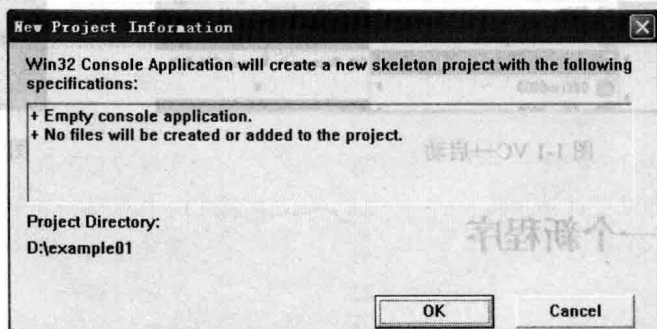


图 1-5 创建工程的信息

在图 1-6 中,窗口左侧为“Workspace”(工作空间)窗口,下部窗口为“Output”(输出)窗口。在 workspace(工作空间)窗口中有两个视图标签:ClassView(类视图)和 FileView(文件视图)。前者按照 C++“类”的管理方式展现 C/C++ 的源代码,后者按照文件的组织方式展现 C/C++ 的源代码。单击标签的名称,可以在二者之间切换显示。

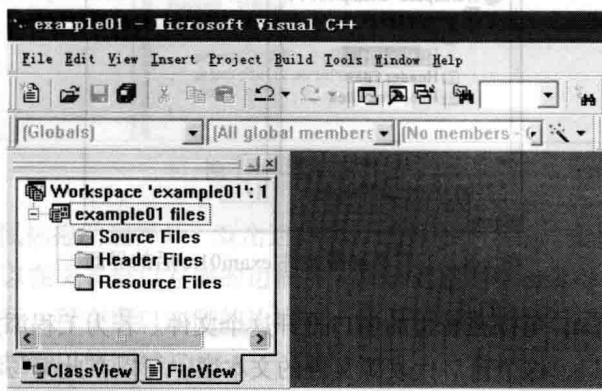


图 1-6 创建工程后的界面

接下来的操作是为工程增加一个源文件,具体步骤见添加源程序部分内容。

二、添加源程序

1. 源文件的添加

按照方法一建立工程后,整个工程是一个完全空的架子,没有任何源文件。此时,单击菜单“File”,再次选择“New”,系统弹出与图 1-3 一样的对话框,不过,此时的对话框默认的标签是“Files”,如图 1-7 所示。

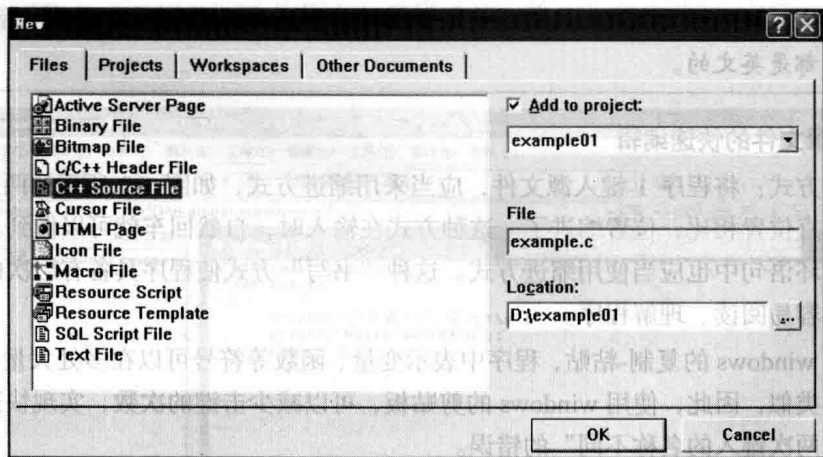


图 1-7 为工程建立源文件的对话框

在图 1-7 中,选择“C++Source Files”(C++源文件),在右边的 File 文本框中填写文件名(一定要带有扩展名 c 或 cpp,它们的编译器不同),然后单击“OK”,系统为使用者建立源文件并打开,右边的文本编辑区,就是展示源文件内容的窗口。图 1-8 是为工程 example01 添加了源文件 exam01.c 后,并展开了 Source Files 文件夹(单击其前面的“+”号)后的界面。

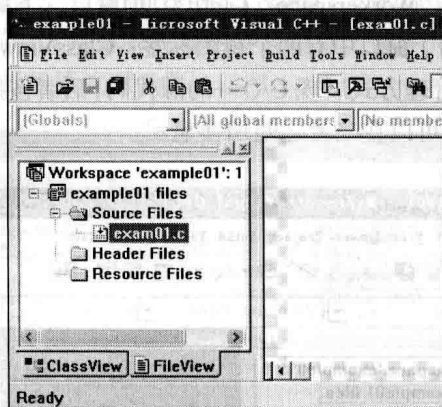


图 1-8 工程添加源文件 exam01.c 后的界面

双击文件名 exam01.c, 可以在右边的窗口打开这个文件。若为工程添加了多个文件, 则每个打开的文件对应一个窗口, 这个窗口代表源文件的文本区, 可以在此编写程序。例如, 编辑如下代码:

```
/* 程序 1 */
#include "stdio.h"
int main()
{
    printf("这是第一个 c 程序!\n");
    printf("hello world!\n");
    return 0;
}
```



输入汉字后要及时切换回到英文输入方式, 因为 C 语言使用的 `\n`、引号、分号、括号等都是英文的。

2. 程序源文件的快速编辑

采用缩进方式: 将程序 1 键入源文件, 应当采用缩进方式, 如图 1-9 所示的两个 `printf` 行, 与大括号的垂直位置相比, 位置缩进了。这种方式在输入时, 自然回车就可以做到。以后的编程中, 分支、循环语句中也应当使用缩进方式。这种“书写”方式使程序具备有层次的美感, 增强逻辑感, 让人容易阅读、理解程序。

积极使用 windows 的复制-粘贴, 程序中表示变量、函数等符号可以在多处大量出现, 甚至一些语句也极其类似, 因此, 使用 windows 的剪贴板, 可以减少击键的次数, 实现快速输入, 还可以减少出现“两次键入的名称不同”的错误。

使用剪贴板, 需要事先选定文本, 其设计的方法如下。

光标定位: 除了使用鼠标定位外, 可以使用键盘上的光标移动键 (箭头键)、home 键、end 键、快捷键等进行快速光标定位。

键盘右侧区域的箭头键 `←`、`→` 是水平 (光标所在位置左右) 移动一个字符。

键盘右侧区域的箭头键 `↑`、`↓` 是光标在垂直位置移动一行 (上下行移动)。

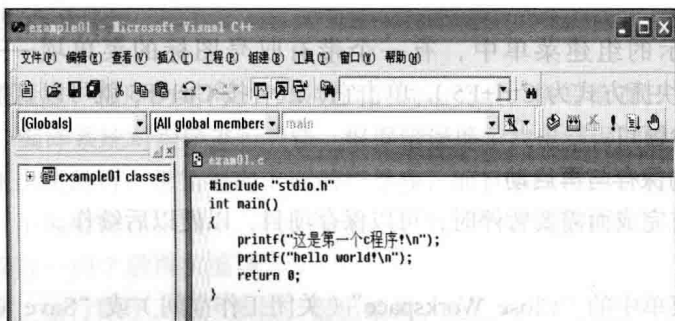


图 1-9 编写程序 1

文本选择：除了用鼠标拖动选择外，双击选择可以选择一个词汇（变量名、函数名等）；如果打算选择单行代码，可以在文本窗口左侧的边框上、该行的前面单击鼠标左键，即可完成单行的选择。如果打算选择多行，可在窗口左侧的边框上，按下鼠标左键进行拖动。

全文操作：全文选择可以使用键 Ctrl+A。

全文缩进：整个源程序按照格式进行缩进可以使用快捷键 Alt+F8，当然，必须事先选择全文。

取消与恢复：前次的操作可以用快捷键 Ctrl+Y 取消；而刚刚被取消的操作可以用快捷键 Ctrl+Z 恢复。

三、编译与运行

1. 编译

当将程序 1 键入之后，可单击菜单“组建”中的“编译”，系统会对源文件及整个工程进行编译，编译完成后再进行连接，最终生成可执行程序（exe）；也可以单击菜单第 2 项或按 F7 生成可执行程序，如图 1-10 所示。

系统进入编译时，在下面将出现 output 窗口，其中给出编译和链接过程中的语法检查信息。如果有错，给出错误信息。这些信息包括错误行、类别、错误代号、错误对象名、错误原因。



图 1-10 编译与连接

2. 运行程序

在图 1-10 所示的组建菜单中, 有一个带有叹号图标的项目——Execute (执行) example01.exe, (其快捷方式为 Ctrl+F5), 单击它, 或者按 Ctrl+F5 就可以直接运行程序。图 1-11 就是运行时的控制台窗口。

四、工程项目的保存与再启动

当工程项目没有完成而需要暂停时, 可以保存项目, 以便以后续作。

1. 保存工程

单击“文件”菜单中的“Close Workspace”(关闭工作空间)或“Save Workspace”(保存工作空间)都可以保存工程, 前者在没有保存时, 给出提示保存。

注意: 只有关闭工作空间以后, 才能编写下一个程序, 如图 1-12 所示。

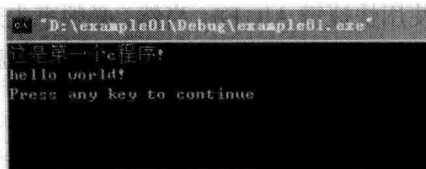


图 1-11 运行结果

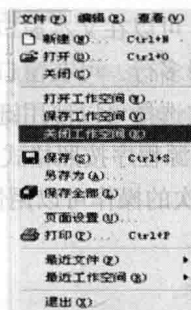


图 1-12 关闭工作空间

2. 打开工程

单击“文件”菜单中的“Open Workspace”(打开工作空间), 在出现的对话框中, 选择正确的工程文件夹, 后打开后缀为.dsw 的文件。

另一个打开已有工程的方法是, 单击“文件”菜单中的“Recent Workspaces”(最近的工作空间), 从列表中选择。

第 3 个方法是在 windows 资源管理器中, 找到并打开相应的工程文件夹后, 双击后缀为.dsw 的文件。

用打开工程的方法打开工程, 工程的状态为上次操作的最后状态, 打开后就可以接着操作。

1.2 C 语言程序调试方法

1.2.1 错误的类型

语法错误: 拼写错, 括号不匹配, 漏写分号, 等等。对于查出的错误 (Error) 必须排除, 否则程序无法运行; 而警告 (Warning) 则应根据情况处理, 否则可能产生运算误差等。

逻辑错误: 编译无误, 有时也有执行结果, 但结果不符合题意。例如 scanf 的参数中漏写地址符, if 语句、for 语句花括号位置错误等, 都会导致此类错误。

运行错误: 运行结果错误也可能是由于输入数据错误、类型不匹配等造成的。例如用户没有按照 scanf 规定的格式输入数据就会造成此类错误。

1.2.2 查错的方法

静态检查：人工检查，程序的结构、各函数间的调用关系，拼写检查。

编译程序：由 C 编译系统对程序进行查错，根据错误提示找出错误的位置并改正。注意提示的出错行未必是真正出错的行，常需要向上面的行寻找；而且系统指出的错误类型也未必是真正的错误，需要分析，不能停留在字面上。代码中有一个错误时，可能产生一大批编译错误，应从下到上逐一改正，修改一两个后再次编译。

排除语法错误后，运行程序，输入数据，得出结果，还应对结果进行分析，看是否符合要求。要准备一些测试数据，有意识地检查结果的正误。

若运行结果错误，通常由于程序中存在逻辑错误，应对照流程图检查算法逻辑。

对于怀疑出错的地方，添加一些 `printf` 函数输出某些变量的值，以找到出错的程序段，缩小查错范围。

1.2.3 Visual C++ 6.0 中常用的程序调试工具

Visual C++ 6.0 提供单步运行、断点跟踪等工具，帮助程序员查错。

一、设置固定断点或临时断点

所谓断点，是指定程序中的某一行，让程序运行至该行后暂停运行，使得程序员可以观察分析程序的运行过程中的情况。这些情况一般包括以下几个方面。

(1) 在变量窗口(Variables)中观察程序中变量的当前值。程序员观察这些值的目的是与预期值对比，若与预期值不一致，则此断点前运行的程序肯定在某个地方有问题，以此可缩小故障范围。

(2) 在监控窗口(Watch)中观察指定变量或表达式的值。当变量较多时，使用 Variables 窗口可能不太方便，使用 Watch 窗口则可以有的放矢、有计划地观察关键变量的变化。

(3) 在输出窗口中观察程序当前的输出与预期是否一致。同样地，若不一致，则此断点前运行的程序肯定在某个地方有问题。

(4) 在内存窗口(Memory)中观察内存中数据的变化。在该窗口中能直接查询和修改任意地址的数据。对初学者来说，通过它能更深刻地理解各种变量、数组和结构等是如何占用内存的，以及数组越界的过程。

(5) 在调用堆栈窗口(Call Stack)中观察函数调用的嵌套情况。此窗口在函数调用关系比较复杂或递归调用的情况下，对分析故障很有帮助。

二、使用功能键或相应的菜单项进行操作。

F9：在当前光标所在行设置断点(Breakpoint，再次使用则取消当前行已有断点)，见图 1-13。

F5 (Go)：调试状态运行程序，程序执行到有断点的地方停下，见图 1-14。

此时左下角的变量窗口(Variables)显示当前的变量值。

如果在右下角的察看窗口(Watch)输入变量名，则可监控该变量值的变化。

F10 (Step Over) 或 F11 (Step Into)：单步执行程序。

F11 和 F10 的区别：如果当前执行语句是函数调用，则 F11 会进入被调用函数里面。

Ctrl+F10：运行到光标所在行。

Shift+F11：跳出当前所在函数。

Shift + F5 (Stop Debugging)：停止调止状态。

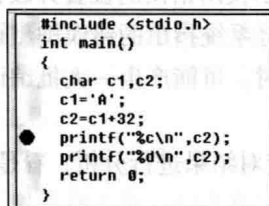


图 1-13 断点

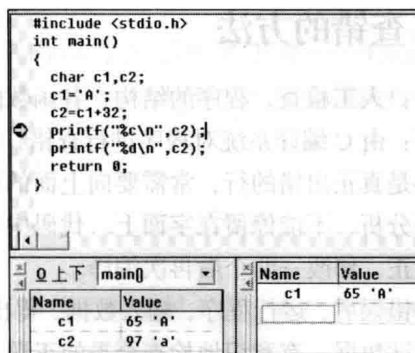


图 1-14 运行到断点处，变量窗口，察看窗口

在调试状态，系统会出现 Debug（调试）工具箱，如图 1-15 所示。其中包括了上面提到的功能的工具按钮。

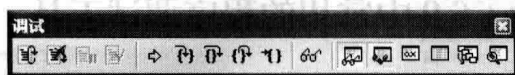


图 1-15 调试工具箱

三、单步运行调试的基本步骤

- (1) 保存 C 或 C++ 文件。
- (2) 根据断点调试找到错误处。
- (3) 采用 F10 或 F11 单步调试找到精确的错误处。可先用 F10，确定函数输入、输出是否与预想的一致；如不相符，则用 F11 进入函数体一步一步调试。
- (4) 调试过程中，需要监视程序中变量值的变化。VC++ 6.0 的 Variables 和 Watch 窗口，就用来设置监视变量。在调试过程中，鼠标轻轻放在变量上（不用单击）也会显示该变量的值。

1.3 C 语言的编程风格

遵循一门语言的编程风格是非常重要的，否则编写的代码将难以阅读，给后期的维护带来诸多不便，例如，一个程序员将许多代码都写在同一行，尽管程序可以正确地编译和运行，但是这样的代码几乎无法阅读，其他程序员无法容忍这样的代码，造成沟通困难。本节将介绍一些最基本的编程风格，希望初学者在关注编程的正确性同时注重自己编写程序的可读性。

在编写程序时，许多地方如函数体、循环语句的循环体及分支语句的分支体都涉及使用一对大括号括起来的多行语句，俗称“代码块”。“代码块”有两种流行（行业内遵循的习惯）的写法——Allmans 风格和 Kernighan 风格。下面分别介绍这两种风格。

1.3.1 编程风格

Allmans 风格也称“独行”风格，即左右大括号各自独占一行，如下列代码所示。

```
#include<stdio.h>
int main()
{
```

```
printf("Programing");
printf("is fun.\n");
return 0;
}
```

当代码量较小时适合使用 Allmans 风格，代码布局清晰，可读性强，本书的大部分例题都采用了这个风格。

Kernighan 风格也称“行尾”风格，既左大括号在上一行的行尾，而右大括号独占一行，如下列代码所示。

```
#include<stdio.h>
/*getMax 函数用于求两个整数中的最大值，
其中参数 a, b 的类型和返回值类型都是整型的。*/
int getMax(int a,int b){
    if(a>b)
        return a;
    return b;
}

int main(){
    int x,y,max;
    printf("Please input two integers: ");
    scanf("%d%d",&x,&y);    //通过键盘输入两个整数
    max=getMax(x,y);        //调用函数实现求两个整型数据的最大值
    printf("The max number is:%d.\n",max);
    return 0;
}
```

当代码量较大时不适合使用 Allmans 风格，因为该风格将导致代码的左半部分出现大量的左右大括号，导致代码清晰度下降，这时应采用 Kernighan 风格。

1.3.2 变量名、函数名的命名规则

1. 变量名的命名规则

变量的命名规则要求用“匈牙利法则”。即开头字母用变量的类型，其余部分用变量的英文意思或其英文意思的缩写，尽量避免用中文的拼音，要求单词的第一个字母大写。

即变量名=变量类型+变量的英文意思（或缩写）对非通用的变量，在定义时加入注释说明，变量定义尽量可能放在函数的开始处。如：

short(int) 用 n 开头 nStepCount

long(LONG) 用 l 开头 lSum

char(CHAR) 用 c 开头 cCount

float(FLOAT) 用 f 开头 fAvg

double(DOUBLE) 用 d 开头 dDeta

void(VOID) 用 v 开头 vVariant

全局变量用 g_ 开头，如一个全局的长型变量定义为 g_lFailCount，即变量名=g_+变量类型+变量的英文意思（或缩写）。静态变量用 s_ 开头，如一个静态的指针变量定义为 s_plPerv_Inst，即变量名=s_+变量类型+变量的英文意思（或缩写）。对枚举类型（enum）中的变量，要求用枚举变量或其缩写做前缀，并且要求用大写。对结构体类型变量的命名要求定义的类型用大写，并要加上前缀，其内部变量的命名规则与变量命名规则一致。

2. 函数名的命名规则

函数的命名应该尽量用英文表达出函数完成的功能。遵循动宾结构的命名法则, 函数名中动词在前, 并在命名前加入函数的前缀, 函数名的长度不得少于 8 个字母。一般第一个字母为小写英文, 以后每个单词的第一个字母大写, 例如:

```
long cmGetDeviceCount(.....);
```

1.3.3 注释

注释是程序员对自己所编写程序的注释性说明, 给代码添加合适的注释是一个良好的编程习惯。注释的目的是有利于代码的维护, 提高代码可读性。C 语言支持两个格式的注释: 单行注释和多行注释。注释的内容不限定语言, 可以是中文也可以是英文。

单行注释使用//表示单行注释的开始, 即该行中从//开始的后续内容都为注释。如

```
scanf("%d%d",&x,&y);    //通过键盘输入两个整数
max=getMax(x,y);        //调用函数实现求两个整型数据的最大值
```

多行注释以/*表示注释开始, 以*/表示注释结束, 其中/*之间不能有空格。例如上例中:

```
#include<stdio.h>
/*getMax 函数用于求两个整数中的最大值,
其中参数 a,b 的类型和返回值类型都是整型的, */
int getMax(int a,int b){
    if(a>b)
        return a;
    return b;
}
```

本书中, 根据需要两种注释都会使用。

编译器在将程序翻译为机器语言代码时, 会忽略所有的注释内容, 也就是说, 注释对程序功能没有任何影响。但注释是编写程序不可缺少的部分, 清晰的注释记录了程序员编写程序时的思路和设计思想, 也便于自己或其他程序员日后查阅、引用。

在上例中, 要求用户输入两个整数前, 会通过 printf("Please input two integers:"); 语句向屏幕上打印一行提示, 非常直观。大家在编写程序时, 也要注意编写程序的界面友好, 输入输出要有提示说明, 对用户错误的输入要有检测和提示, 养成良好的编程习惯。

第2部分

基本实验

本部分为读者设置了15个实验项目，读者可根据需要选择。一些实验项目给出了参考程序，希望读者首先独立思考、独立编程，然后和书中给出的答案进行比较，考察两者的优劣。

一、实验要求

1. 实验前应明确实验目的，阅读实验内容，对于较复杂的题目给出算法的N-S图描述，写好实验题目程序清单。
2. 能够理解调试过程中出现的提示信息，尽量独立改正程序中的各类错误，提高自己调试程序的能力。
3. 随时记录调试程序时遇到的问题，养成良好的交流学风，在交流中找到解决问题的途径和方法。
4. 实验结束时要填写实验报告。
5. 充分利用上机时间，提高上机效率。上机前把需要运行的程序准备好，不要现场编写，因为上机的时间是用来调试程序的。

二、实验报告的内容

本部分设置的实验项目，主要包括以下几个部分。

1. 实验目的
2. 实验内容
3. 实验步骤

对于设计性实验，给出算法说明（复杂的可用N-S图表示）和程序清单。

4. 实验结果
5. 实验总结

包括程序运行时数据的输入、运行结果的输出。

对运行情况以及本次实验所取得的经验进行分析。如调试时出现错误信息，对错误原因进行分析。

实验1 初识C程序

一、实验目的

1. 编辑简单的C程序，以此熟悉Visual C++ 6.0的集成开发环境，并初步认识C程序的组成。