



普通高等教育“十二五”应用型人才培养规划教材

C语言程序设计

C Language Programming

◎ 主编 胡成松 黄玉兰 李文红



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”应用型人才培

C 语言程序设计

主 编 胡成松 黄玉兰 李文红

副主编 卢云霞 葛 蓁 王 微

参 编 崔欢欢 顿煜卿 肖丹丹 邹 静

主 审 孙宝林



机械工业出版社

本书包含 C 语言概述,数据类型、运算符和表达式,简单程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,结构体、共用体与枚举,文件等共 10 章内容。全书内容以案例“学生信息管理系统”贯穿,旨在帮助学生掌握完整的课程脉络,并将该案例涉及的知识点分解到各个章节。每章通过案例和问题引入知识点,重点讲解程序设计的思想和方法,并介绍相关的语法知识,注重培养学生分析问题和解决问题的能力,每章末尾给出了该案例的具体实现参考程序代码。

本书可供普通高等院校学生学习“C 语言程序设计”课程时作为教材使用。

本书配有电子课件,欢迎选用本书作教材的老师登录 www.cmpedu.com 注册下载,或发 jinaomp@163.com 索取。

图书在版编目 (CIP) 数据

C 语言程序设计/胡成松,黄玉兰,李文红主编. —北京:机械工业出版社,2015.8

普通高等教育“十二五”应用型人才培养规划教材

ISBN 978-7-111-50465-8

I. ①C… II. ①胡…②黄…③李… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 138583 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:吉玲 责任编辑:吉玲

责任校对:刘怡丹 封面设计:张静

责任印制:乔宇

唐山丰电印务有限公司印刷

2015 年 8 月第 1 版第 1 次印刷

184mm×260mm·15.75 印张·390 千字

标准书号:ISBN 978-7-111-50465-8

定价:34.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:010-88379833

机工官网:www.cmpbook.com

读者购书热线:010-88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面防伪标均为盗版

金书网:www.golden-book.com

前言

随着社会快速进入信息化时代,软件毫无疑问地变得越来越重要,而程序设计的能力极大地支配着软件开发的效果和效率,因此具备一定的程序设计能力已经是对当代相关专业大学生最基本的要求之一。

C语言是世界上最流行、使用范围最广的程序设计语言之一,它具有其他语言所没有的特点和优势,它经常被学生选择作为第一门编程语言来学习。掌握该语言的好坏程度决定了学生日后在程序开发领域所能达到的高度。

本书根据作者多年的教学经验和教学改革成果编写而成,以案例“学生信息管理系统”进行贯穿,旨在帮助学生掌握完整的课程脉络,并将该案例涉及的知识点分解到各个章节,每章通过案例和问题引入知识点,重点讲解程序设计的思想和方法,并介绍相关的语法知识。本书注重培养学生分析和解决程序设计问题的能力,每章末尾给出了该案例的具体实现参考程序代码。本书讲述力求理论联系实际、循序渐进,通过大量例题来验证语法和说明程序设计方法,学生既能迅速掌握C语言的基础知识,又能很快地学会C语言的编程技术,以培养解决实际问题的能力。

本书中的程序都是在 Visual C++6.0 编译环境下调试运行的,读者可以选用该编译环境作为学习本教材的开发工具。

本书由武汉工商学院胡成松、葛蓁、肖丹丹,武昌工学院黄玉兰、李文红、卢云霞、邹静,华中农业大学楚天学院王微、崔欢欢、顿煜卿编写,由胡成松、黄玉兰、李文红对稿件进行审查整理,由胡成松统稿,最后由孙宝林教授审稿并提出了宝贵的修改意见。

本书在编写过程中,得到了武汉工商学院、武昌工学院、华中农业大学楚天学院等学校领导的大力支持,在此一并表示感谢。另外,在本书编写过程中,编者参考了大量有关C语言程序设计的书籍和资料,在此对这些参考文献的作者表示感谢。

由于编者水平有限,不足之处在所难免,恳请广大读者批评指正。

编 者

目录 Contents

前言

第1章 C语言概述 1

学习要点 1

1.1 C语言概况 1

1.1.1 C语言的发展 1

1.1.2 C语言的特点 1

1.2 C语言程序的开发与运行 2

1.2.1 C语言程序的开发过程 2

1.2.2 VC开发环境介绍 3

1.2.3 常见错误提示及解决方法 7

1.3 简单的C语言程序 8

1.4 算法 10

1.4.1 算法概述 10

1.4.2 算法的表示 10

1.5 小结 11

习题 12

第2章 数据类型、运算符和表达式 14

学习要点 14

导入案例 14

2.1 数据类型 14

2.1.1 数据类型概述 14

2.1.2 整型 15

2.1.3 实型 16

2.1.4 字符型 16

2.2 标识符、常量与变量 16

2.2.1 标识符 16

2.2.2 常量 17

2.2.3 变量 20

2.3 运算符与表达式 23

2.3.1 算术运算符与算术表达式 24

2.3.2 赋值运算符与赋值表达式 25

2.3.3 自增/自减运算符 27

2.3.4 逗号运算符与逗号表达式 28

2.3.5 位运算符 28

2.3.6 数据类型转换 31

2.4 知识点强化与应用 32

2.5 小结 33

【案例分析与实现】 34

习题 35

第3章 简单程序设计 37

学习要点 37

导入案例 37

3.1 C语言语句 38

3.2 C语言程序结构简介 39

3.3 数据输入与输出 43

3.3.1 字符数据的输入与输出 43

3.3.2 格式化输出函数 printf 44

3.3.3 格式化输入函数 scanf 46

3.4 知识点强化与应用 47

3.5 小结 50

【案例分析与实现】 50

习题 52

第4章 选择结构程序设计 53

学习要点 53

导入案例 53

4.1 关系运算符与关系表达式 54

4.1.1 关系运算符 54

4.1.2 关系表达式 54

4.2 逻辑运算符与逻辑表达式 55

4.2.1 逻辑运算符 55

4.2.2 逻辑表达式 56

4.3 if语句 57

4.3.1 单分支if语句 58

4.3.2 双分支if语句 60

4.3.3 多分支选择结构 61

4.3.4 if语句的嵌套 62

4.3.5 条件运算符和条件表达式 65

4.4 switch语句 67

4.5 知识点强化与应用 72

4.6 小结 75

【案例分析与实现】 76

习题 79

第5章 循环结构程序设计	82	导入案例	129
学习要点	82	7.1 函数概述	129
导入案例	82	7.2 函数的定义	130
5.1 while 循环	82	7.2.1 函数定义的一般形式	130
5.1.1 while 语句的基本格式	82	7.2.2 函数的参数	131
5.1.2 while 语句的应用	83	7.2.3 函数的返回值	133
5.2 do-while 循环	85	7.3 函数的调用	134
5.2.1 do-while 语句的基本格式	85	7.3.1 函数调用的一般形式	134
5.2.2 do-while 语句的应用	86	7.3.2 函数的嵌套调用	138
5.3 for 循环	87	7.3.3 函数的递归	139
5.3.1 for 语句的基本格式	87	7.4 函数的声明	141
5.3.2 for 语句的应用	89	7.5 数组作为函数参数	142
5.4 循环嵌套	90	7.5.1 数组元素作为函数的实参	142
5.5 break 语句、continue 语句	92	7.5.2 数组名作为函数的实参	143
5.5.1 break 语句	92	7.6 变量的作用域	144
5.5.2 continue 语句	93	7.7 变量的生命周期	146
5.6 知识点强化与应用	95	7.8 内部函数和外部函数	147
5.7 小结	99	7.9 知识点强化与应用	148
【案例分析与实现】	100	7.10 小结	155
习题	101	【案例分析与实现】	156
第6章 数组	103	习题	161
学习要点	103	第8章 指针	163
导入案例	103	学习要点	163
6.1 一维数组	104	导入案例	163
6.1.1 一维数组的定义	104	8.1 指针的基本概念	163
6.1.2 一维数组元素的引用	104	8.1.1 指针的概念	163
6.1.3 一维数组元素的初始化	105	8.1.2 指针变量的定义与初始化	164
6.2 二维数组	107	8.1.3 指针的引用及运算	166
6.2.1 二维数组的定义	107	8.1.4 指针变量作为函数参数	166
6.2.2 二维数组元素的引用	108	8.2 指针与数组	168
6.2.3 二维数组元素的初始化	109	8.2.1 指针与一维数组	168
6.3 字符数组和字符串	110	8.2.2 指针与二维数组	171
6.3.1 字符数组的定义	110	8.2.3 指针与字符串	174
6.3.2 字符数组的初始化	110	8.2.4 指针数组	176
6.3.3 字符数组元素的引用	111	8.2.5 指向指针的指针	178
6.3.4 字符串的存储	111	8.2.6 指针数组作为 main() 函数的 参数	179
6.3.5 字符数组的输入/输出	112	8.3 指针与函数	181
6.3.6 常用字符串处理函数	114	8.3.1 指向函数的指针	181
6.4 知识点强化与应用	116	8.3.2 返回指针的函数	184
6.5 小结	125	8.4 知识点强化与应用	186
【案例分析与实现】	126	8.5 小结	192
习题	128	【案例分析与实现】	193
第7章 函数	129	习题	196
学习要点	129		

第 9 章 结构体、共用体与枚举	197	10.1.1 文件的概念	217
学习要点	197	10.1.2 文件的分类	217
导入案例	197	10.2 文件操作	218
9.1 结构体	197	10.2.1 文件类型指针	218
9.1.1 结构体类型的定义	198	10.2.2 文件的打开操作	219
9.1.2 结构体变量的定义	199	10.2.3 文件的关闭操作	220
9.1.3 结构体变量的初始化	200	10.2.4 文件的读/写操作	221
9.1.4 结构体变量的引用	201	10.3 文件的定位	223
9.1.5 结构体数组	202	10.3.1 rewind 函数	223
9.1.6 结构体指针	203	10.3.2 fseek 函数	223
9.1.7 动态内存分配	205	10.3.3 ftell 函数	224
9.2 共用体	206	10.3.4 文件的错误检测	224
9.2.1 共用体类型的定义	206	10.4 编译预处理	224
9.2.2 共用体变量的说明	206	10.4.1 宏定义	225
9.2.3 共用体变量的引用	207	10.4.2 文件包含	229
9.3 枚举类型	208	10.4.3 条件编译	229
9.3.1 枚举类型的定义	208	10.5 知识点强化与应用	231
9.3.2 枚举变量的说明及引用	208	10.6 小结	232
9.3.3 枚举类型的应用	209	【案例分析与实现】	233
9.4 用户定义类型	209	习题	234
9.5 知识点强化与应用	210	附录	235
9.6 小结	213	附录 A C 语言中的关键字 (32 个)	235
【案例分析与实现】	214	附录 B C 语言常用数据类型	235
习题	215	附录 C 常用字符与 ASCII 码对照表	236
第 10 章 文件	217	附录 D 运算符和结合性	237
学习要点	217	附录 E C 常用库函数	239
导入案例	217	参考文献	246
10.1 文件概述	217		

第 1 章

C 语言概述

学习要点

1. 了解 C 语言的发展特点
2. 掌握 C 程序的编辑、编译、连接和执行过程
3. 熟悉 Visual C++ 6.0 开发环境
4. 编译和调试 C 语言源程序时，能辨别常见的错误
5. 了解算法的概念及流程图概念

1.1 C 语言概况

众所周知，所有的软件都是用计算机语言编写的，而 C 语言也是众多计算机语言中的一种。本章主要介绍 C 语言的发展历程、C 语言的特点，以及相对于其他高级语言的优势。

1.1.1 C 语言的发展

C 语言是国际上流行的计算机高级程序设计语言之一。与其他高级语言相比，C 语言的硬件控制能力和运算表达能力强，可移植性好，效率高（目标程序简洁，运行速度快），因此应用面非常广，许多大型软件都是用 C 语言编写的。

1960 年，艾伦·佩利（Alan J. Perlis）在巴黎举行的一次全世界一流软件专家参加的讨论会上，发表了《算法语言 Algol 60》报告，确定了程序设计语言 Algol 60；1962 年，艾伦·佩利又对 Algol 60 进行了修正；1963 年，英国剑桥大学在 Algol 语言的基础上进行了改进，推出 CPL 语言；1967 年，剑桥大学的马丁·理查德对 CPL 简化，产生了 BCPL 语言；1970 年，美国贝尔实验室的肯·汤姆逊对 BCPL 语言进行了修改，取其名称的第一个字母提出了 B 语言；1972 年，美国贝尔实验室的布朗·W·卡尼汉和丹尼斯·M·利奇在 B 语言基础上发展成 C 语言（取 BCPL 的第二个字母）；为了避免因各种 C 语言版本的差异导致的不一致情况，1988 年美国国家标准研究所制定了 ANSI 标准，即形成了现在流行的 C 语言。自 1972 年投入使用之后，C 语言成为 UNIX 和 Linux 操作系统的主要编程语言，是当今最为广泛使用的程序设计语言之一。

1.1.2 C 语言的特点

C 语言之所以迄今为止仍被人们普遍使用，不仅与它产生的环境和历史背景相关，关键在于它本身的优点，C 语言具有以下优点。

1. 简单易学

C 语言是现有程序设计语言中规模最小的语言之一。C 语言一共有 32 个关键字，有 9 种控制语句，程序员书写自由，表达方式简洁，主要用大小写字母表示，使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

2. 程序设计结构化，符合现代编程风格

C 语言是以函数形式提供给用户的，这些函数可方便地被调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

3. 表达方式灵活实用

利用 C 语言提供的多种运算符，可以组成各种表达式，还可采用多种方法来获得表达式的值，从而使用户在程序设计中具有更大的灵活性。C 语言的语法规则不太严格，程序设计的自由度比较大，程序的书写格式自由灵活。程序主要用小写字母来编写，而小写字母是更容易阅读的，这些充分体现了 C 语言灵活、方便和实用的特点。

4. 表达能力强

C 语言具有丰富的数据类型，可以根据需要采用整型、实型、字符型、数组类型、指针类型、结构类型、联合类型、枚举类型等多种数据类型来实现各种复杂数据结构的运算。C 语言还具有 34 种运算符，灵活使用各种运算符可以实现其他高级语言难以实现的运算。

5. 语言生成目标代码质量高

众所周知，相对汇编语言而言，很多高级语言生成代码的质量比较低。但是，针对同一问题用 C 语言编写的程序生成代码的效率仅比用汇编语言编写的程序低 10% ~ 20%。

6. 可移植性强

大家知道汇编语言因为依赖机器硬件，所以不可移植。一些高级语言，如 FORTRAN 等的编译程序也不可移植，它们从一种机器搬到另外一种机器，大多数都要根据国际标准进行重新编写，而 C 语言基本上不做修改就可以运行于各种型号的计算机和各种操作系统。

尽管 C 语言具有很多的优点，但和其他任何一种程序设计语言一样也有其自身的缺点，如不能自动检查数组的边界、各种运算符的优先级别太多、某些运算符具有多种用途。但总的来说，C 语言的优点远远超过了它的缺点。经验表明，程序设计人员一旦学会使用 C 语言之后，就会对它爱不释手，使用起来得心应手是 C 语言迅速成为人们进行程序设计和软件开发语言的原因。

1.2 C 语言程序的开发与运行

本节主要介绍 C 语言的开发过程，包含编辑、编译、连接、运行 4 个步骤，并对 Visual C++ 6.0（简称 VC）开发环境进行介绍。VC 是微软公司出品的高级可视化计算机程序开发工具，界面友好，使用方便，可以识别 C/C++ 程序，使用此环境进行程序设计十分方便。

1.2.1 C 语言程序的开发过程

C 语言采用编译方式将源程序转换为二进制的目标代码。一个完整的 C 语言程序的开发主要包括编辑、编译、连接和运行 4 个步骤。

1. 编辑

所谓编辑,主要包含以下几点:

1) 利用 C 语言编译系统自带的编辑器将源程序逐个字符输入计算机内存。

2) 编辑、修改源程序代码。

3) 编辑好的代码保存在磁盘文件中。

编辑的对象是源程序,它是以 ASCII 码的形式输入和存储的,不能被计算机所识别。

2. 编译

编译是将已经编辑好的源程序翻译成计算机能够识别的二进制目标代码,即将 .c 或者 .cpp 文件编译成 .obj 文件。在编译的过程中,要及时对源程序进行语法检查,如有报错,将会显示在屏幕上,此时需要重新进入编辑状态对源程序进行编辑、修改,直至准确无误为止。

3. 连接

连接是指将各个模块的二进制代码与系统标准模块进行连接、测试处理后,生成一个可供运行的可执行程序,即由 .obj 文件生成 .exe 文件。

4. 运行

运行是将编译好的可执行文件运行。通过运行程序可以查看程序执行输出结果。图 1-1 所示为 C 语言编辑、编译、连接和运行的全过程。

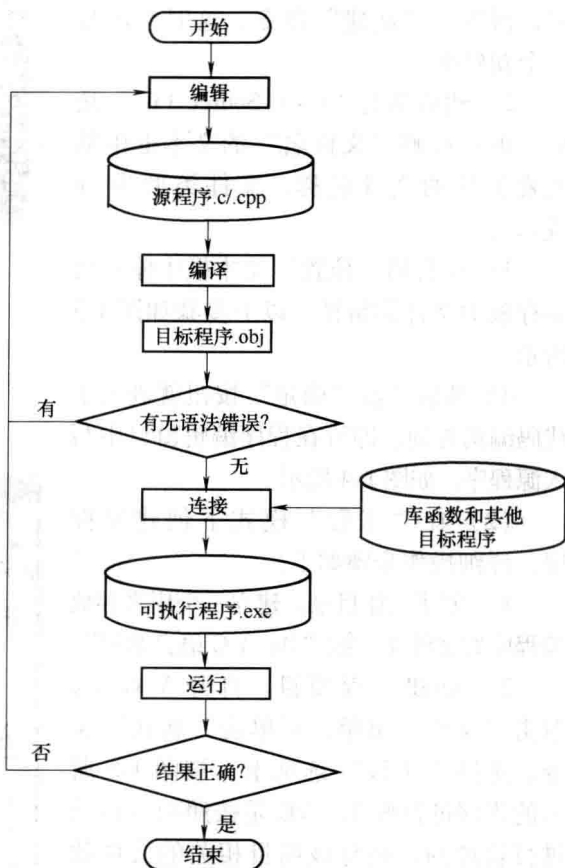


图 1-1 C 语言开发过程

1.2.2 VC 开发环境介绍

VC 可以在“独立文件模式”和“项目管理模式”下使用。一般只有一个文件的时候,可以使用独立文件模式;如果是大型的程序,程序又由多个文件组成时,就用项目管理模式,在这种情况下,所有的源程序合在一起共同构成一个程序,在 C++ 中称作一个“项目”。

当用户安装好 VC 后,通过“开始”→“所有程序”菜单,找到“Microsoft Visual C++ 6.0”单击进入即可,VC 界面如图 1-2 所示。

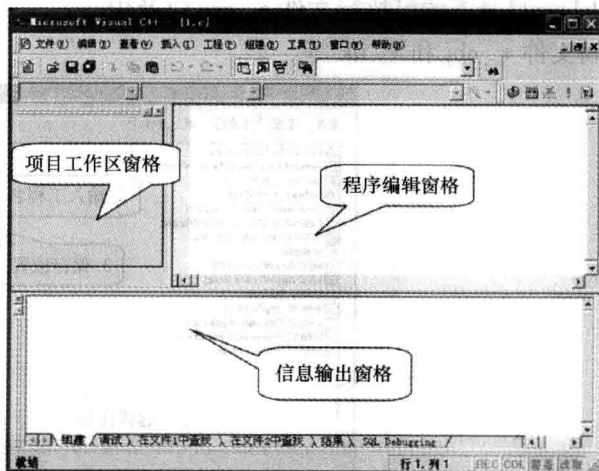


图 1-2 Microsoft Visual C++ 6.0 程序设计窗口

1. 创建源程序

(1) 在“文件”模式下创建源程序，详细操作步骤如下。

1) 打开 V C++，单击“文件”菜单，再单击“新建”命令，这时会出现一个新建框。

2) 然后单击“C++ Source File”选项，再在右侧“文件名”的文本框中输入要创建的文件名称，文件类型为 .c 或 .cpp。

3) 在右侧“位置”文本框中输入所要存放的文件的路径，以上步骤如图 1-3 所示。

4) 然后单击“确定”按钮就进入了代码编辑界面，即可在程序编辑窗口中写入源程序，如图 1-4 所示。

(2) 在“工程”模式下创建源程序，详细操作步骤如下。

1) 创建工作目录。建立一个用来存放源程序的文件夹，如“D:\C 语言案例”。

2) 创建工程项目。打开 V C++，单击“文件”菜单，再单击“新建”命令，选择“工程”选项卡，按图 1-5 所示的步骤进行操作。VC 是按照项目目录进行管理的，把与该项目相关的文件放在一个目录下，包括工程文件 *.dsp 和 *.dsw、程序源文件 *.c 和 *.cpp、在 debug 目录下的可执行文件 *.exe 以及中间文件 *.obj 和 *.ilk 等。

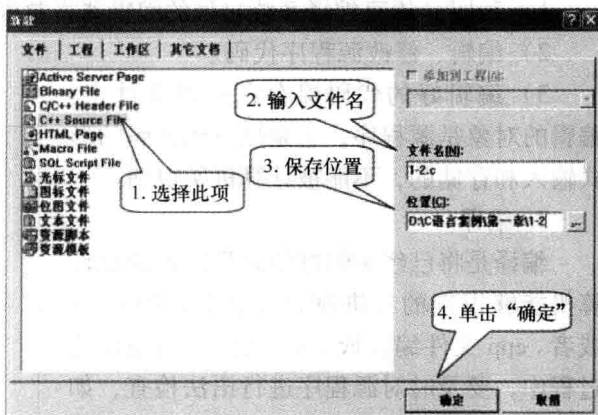


图 1-3 “新建”对话框的“文件”选项卡

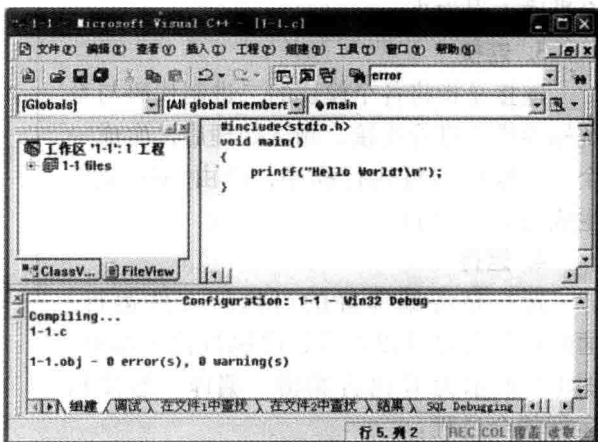


图 1-4 输入源程序代码

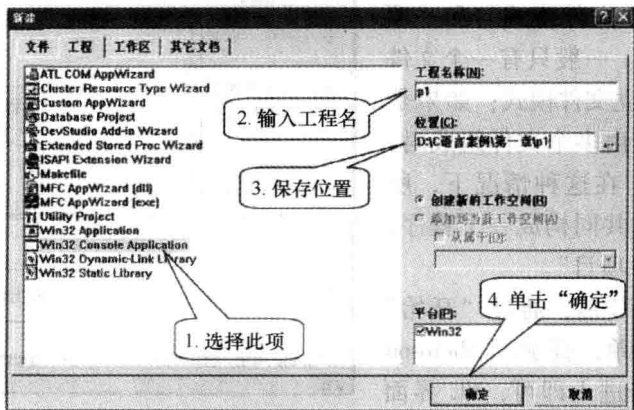


图 1-5 “新建”对话框的“工程”选项卡

3) 选择要创建的程序类型, 如“一个空工程”或是“一个简单的程序”, 单击“完成”按钮, 如图 1-6、图 1-7 所示。

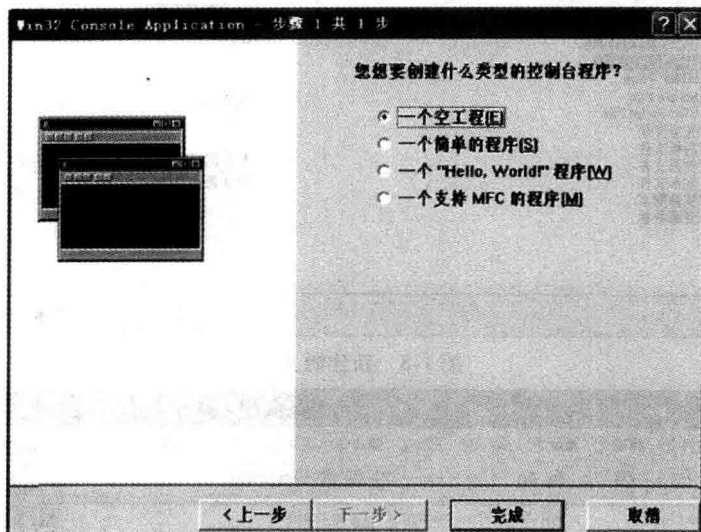


图 1-6 控制台类型对话框

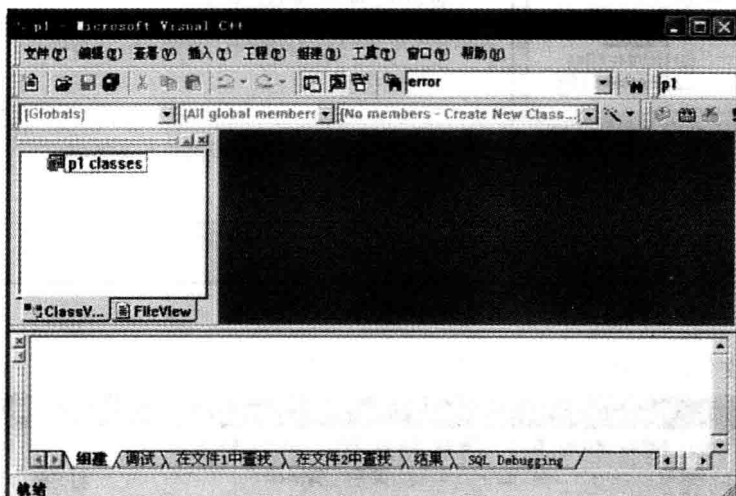


图 1-7 新建工程

4) 单击“文件”菜单, 单击“新建”命令, 选择“文件”选项卡下面的“C++ Source File”选项, 在“文件名”文本框中, 输入后缀为 .c 的文件名, 在“位置”文本框中确认源文件的路径, 并且务必让“添加到工程 (A):”前面的选项框被选中, 如图 1-8 所示。

5) 单击“确定”按钮以后, 在窗口左边的工作区会看到两个选项卡: ClassView 和 FileView。一般默认为 ClassView, 单击 FileView, 才可以从 ClassView 切换到 FileView 选项卡。逐步单击工作区 p1 工程下面的“+”号, 可以在 Source Files 目录下看到刚才新建的 1-1.c 源文件, 这个时候可以在右边的程序编辑窗口中输入源代码, 如图 1-9 所示。

注意: 工作空间可以包含多个工程, 工程又可以包含多个程序文件, 工程与工程之间相互不影响, 工作空间类似于文件夹, 工程类似于文件夹中的文件。

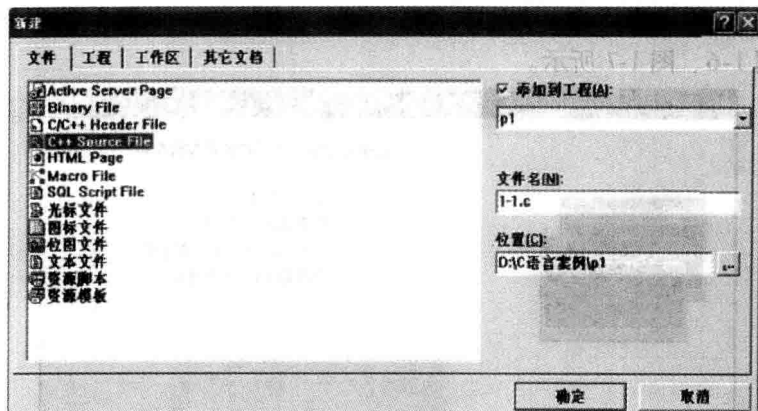


图 1-8 新建源文件

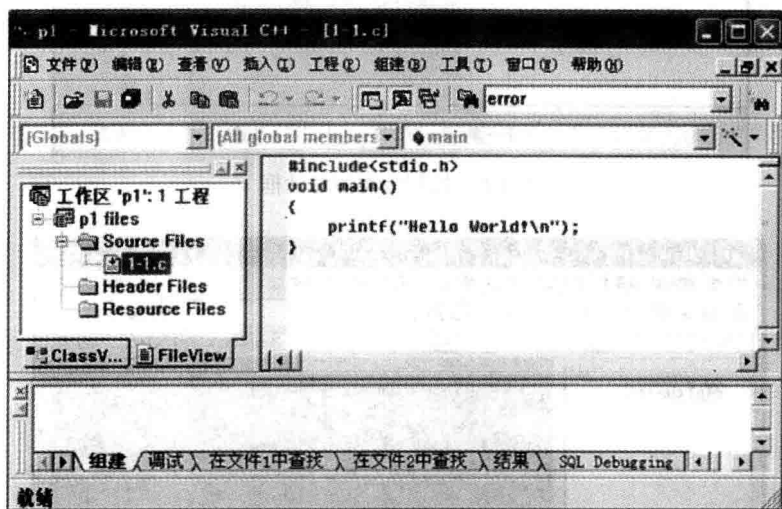


图 1-9 新建源文件代码

连续创建源文件时，要先退出 VC 或选择“文件”→“关闭工作空间”命令，如图 1-10



图 1-10 关闭工作空间

所示,然后再选择“文件”→“新建”命令创建第二个源文件。这是因为如果工作空间里已有打开的工程,再连续创建新的文件,运行时会有多个 main 函数,而 VC 在执行源程序的时候总是从 main 函数处开始,这将会导致程序无法运行。不关闭工作空间连续创建源文件时会产生多个 main 函数导致系统提示错误,如图 1-11 所示。

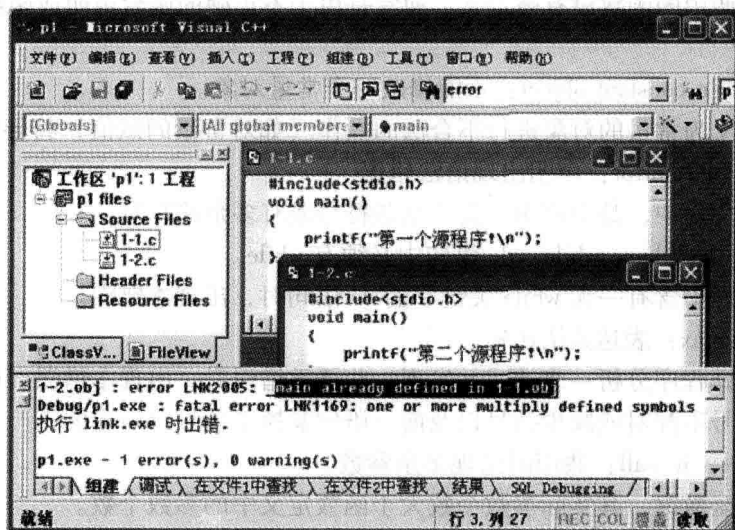


图 1-11 多个源文件连接时的错误界面

2. 编辑

在程序编辑窗口内可以进行类似于 Word 中的一些编辑操作,如复制、粘贴、剪切、撤销与恢复、删除、查找与替换等,这些功能为代码的编辑操作带来了极大的方便,实现途径有 3 种:“编辑”菜单、“标准”工具栏或者相应的快捷键。

VC 还提供了其他编辑功能,例如,通过单击“编辑”菜单→“高级”菜单,可以实现对选择的代码进行大小写转换;通过“工具”菜单→“选项”命令→“制表符”选项卡,可以设置制表符大小,输入时自动实现缩进,可以使代码整齐、规范;同时在程序源代码的编辑过程中可按下 Ctrl + S 组合键进行保存,并可以选定编辑区域中的语句,按下 Alt + F8 组合键可以实现自动排版。

1.2.3 常见错误提示及解决方法

编译和调试源程序时,信息窗口显示诊断信息、警告、出错信息、错误在源程序中的位置。按功能键 F5 可以扩大和恢复信息窗口,按 F6 键或 Alt + E 组合键,光标从信息窗口调到编辑窗口,此时可以根据系统的错误提示对错误进行修改,常见的出错信息及分析如下。

Argument list syntax error: 参数表出现语法错误。

分析: 函数调用的一组参数之间必须以逗号隔开,并以一右括号结束。若源文件中含有一个其后不是逗号也不是右括号的参数,则出现此错。

Array size too large: 数组长度过大。

分析: 定义的数组太长,可用内存不够。

Bad file name format in include directive: 包含命令中文件名格式不正确。

分析: 包含文件名必须用引号“filename.h”或尖括号<filename.h>括起来, 否则将产生此错误。

Call of non-function: 调用未定义函数。

分析: 正被调用的函数没有被定义, 通常是由于不正确的函数声明或函数名拼写错误引起的。

Cannot modify a constant object: 不能修改一个常量对象。

分析: 对定义为常量的对象进行不合法的操作(如对常量的赋值)将导致本错误。

Declaration syntax error: 声明出现语法错误。

分析: 在源文件中, 某个声明丢失了某些符号或有多余的符号。

Do statement must have while: do 语句中必须有 while。

分析: 源文件中含有一无 while 关键字的 do 语句时, 出现本错误。

Expression syntax: 表达式语法错。

分析: 当编译程序分析一表达式并发现一些严重错误时, 出现本错误。通常是由于两个连续操作符、括号不配对或缺少括号以及前一语句漏掉了分号等引起的。

Extra parameter in call: 调用时出现多余参数。

分析: 调用函数时, 其实际参数个数大于函数定义中的参数个数。

Illegal use of point: 指针使用不合法。

分析: 施于指针运算符只能是加、减、赋值、比较。如用其他运算, 则出现本错误。

Non-portable pointer assignment: 不可移植指针赋值。

分析: 源程序中将一个指针赋给一个非指针或相反。但作为特例, 允许把零值赋值给一个指针。如果合适, 强行抑制本错误信息。

Non-portable pointer comparison: 不可移植指针比较。

分析: 源程序中将一个指针和一个非指针(常量零除外)进行比较。如果合适, 强行抑制本错误信息。

Redeclaration of 'xxxxxx': xxxxxx 重定义。

分析: 此标识已经定义过。

Statement missing;: 语句缺少分号。

分析: 编译程序发现一表达式语句后面没分号。

Unable to open include 'xxxxxx.x': 不能打开包含文件“xxxxxx.x”。

分析: 编译程序找不到该包含文件, 可能是由于一个#include 文件包含它本身引起的。

Undefined symbol 'xxxxxx': 符号“xxxxxx”未定义。

分析: 标识符无定义, 可能是由于说明或引用处拼写错误引起的, 也可能是根本就没有定义这个标识符。

1.3 简单的 C 语言程序

【例 1-1】已知长方形的长为 8, 宽为 6, 求该长方形的面积 s。

下面是用 C 语言编写的程序:

```
#include <stdio.h>
void main()
{
    int a,b,s;
    a=8;
    b=6;
    s=a* b;
    printf("s=%d\n",s);
}
```

具体操作步骤如下:

1. 从桌面“开始”菜单打开并进入VC++6.0编程环境及代码编辑窗口。
2. 单击“文件”菜单选择“新建”命令,打开“新建”对话框。
3. 在“文件”选项卡中,选择“C++ Source File”选项。
4. 在右侧“文件名”文本框中输入C程序文件名(扩展名为.c)。
5. 在右侧“位置”文本框中写出所建立的文件的路径。
6. 单击“确定”按钮,即可进入程序编辑窗口,如图1-12所示。

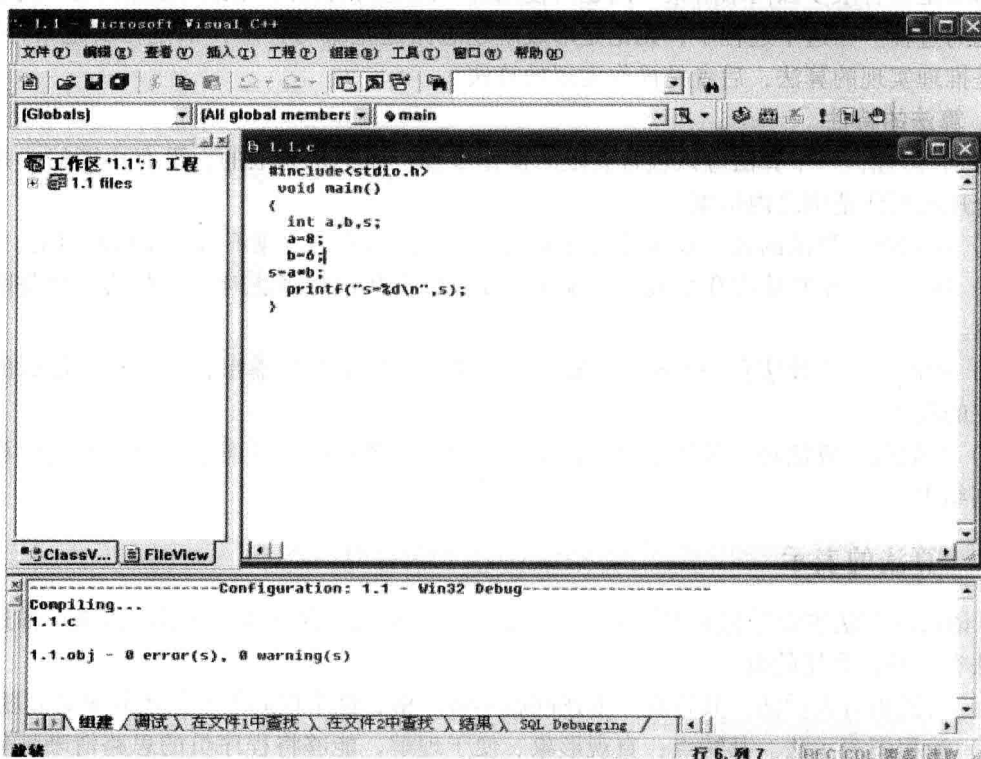


图 1-12 工作界面

这样进入了代码编辑窗口后,输入相应代码,编译、连接无错误后,程序运行结果如图1-13所示。

```
s=48  
Press any key to continue
```

图 1-13 程序运行结果



思考

程序中的 `\n` 起什么作用？去掉后再次运行程序得到什么结果？

1.4 算法

通过对本节的学习，将了解算法的概念、算法的特性，以及算法的几种常用表示方法，并且通过举例说明流程图表示法和 N-S 流程图表示法。

1.4.1 算法概述

1. 算法的概念

算法是在有限步骤内求解某一问题所使用的一组定义明确的规则，通俗点说，就是计算机解题的过程。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是推理实现的算法，后者是操作实现的算法。

2. 算法的特性

- 1) 有穷性：一个算法应该包含有限的操作步骤，而不是无限的、没有止境的，并且应该在合理的操作范围之内结束。
- 2) 确切性：算法的每一步骤必须有准确的定义，而不是含糊不清、模棱两可的。
- 3) 输入：一个算法有 0 个或多个输入，输入是指在执行算法时，需要从外界获取必要的信息。
- 4) 输出：一个算法有一个或多个输出，输出不一定是打印输出，一个算法得到的结果便是它的输出。
- 5) 有效性：算法每一步从原则上讲都应该能够有效的执行，而且通过有效的运算后便可得到结果。

1.4.2 算法的表示

C 语言的算法主要是把人类如何解决问题的方法和思路以某种形式表示出来。描述算法的方法有多种，常用的有：

- 1) 自然语言表示法。其特点：描述问题不够严格，这个仅局限于描述简单的问题。
- 2) 流程图表示法。其特点：直观形象、便于理解，能够将程序员的思路清晰地体现出来，为后期程序的编辑和修改提供了方便，但是同一问题的流程图不唯一。
- 3) N-S 流程图表示法。其特点：结构清晰，但难于修改。
- 4) 伪代码表示法。
特点：结构清晰，代码简单，可读性好，并且类似于自然语言。