



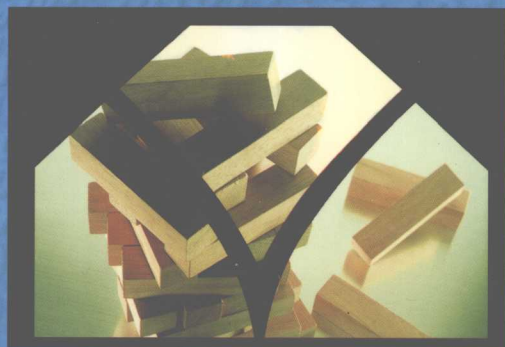
· 全国高职高专电子信息类系列规划教材 ·

C语言程序设计

项目化教程

C Yuyan Chengxu Sheji Xiangmuhua Jiaocheng

● 陈兴无 编著



华中科技大学出版社

<http://www.hustp.com>

C语言程序设计

第 1 版

清华大学出版社



清华大学出版社



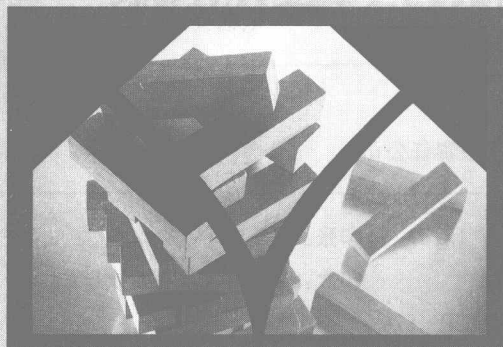
· 全国高职高专电子信息类系列规划教材 ·

C语言程序设计

项目化教程

C Yuyan Chengxu Sheji Xiangmuhua Jiaocheng

陈兴无 编著



华中科技大学出版社

中国·武汉

图书在版编目(CIP)数据

C 语言程序设计项目化教程/陈兴无 编著. —武汉:华中科技大学出版社, 2009 年 3 月

ISBN 978-7-5609-5137-9

I. C… II. 陈… III. C 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 025072 号

C 语言程序设计项目化教程

陈兴无 编著

策划编辑:张 毅

责任编辑:张 毅

责任校对:刘 竣

封面设计:刘 卉

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87542624

录排:武汉正风图文照排中心

印刷:武汉市新华印刷有限责任公司

开本:787mm×1092mm 1/16 印张:18

字数:420 000

版次:2009 年 3 月第 1 版

印次:2009 年 3 月第 1 次印刷

定价:29.00 元

ISBN 978-7-5609-5137-9/TP·675

(本书若有印装质量问题,请向出版社发行科调换)

前言

目前,高职院校的计算机专业及其相关专业大都将 C 语言程序设计作为一门重要的基础课程。在多年的教学实践中,我们体会到要真正掌握 C 语言,学习者的难度较大。一方面课程自身有一定的难度,另一方面是现有教材不能很好地将教学过程中出现的知识、技能与实际软件开发所需要的知识、技能结合起来,学不能用、学不为用,学习者的学习积极性和主动性不能得到充分发挥。我们本着“职业活动导向、任务驱动、项目载体”的教学原则,结合 C 语言程序设计自身的特点,用设计独特、编排新颖、通俗易懂的方法编写了本书。

本书具有以下几个特点。

本书以完成“班级学生成绩管理系统”软件开发为项目,系统地描述了软件开发的全过程。为了更好地完成该项目,我们将其分解成 15 个任务,合理地安排到相关单元中,做到开发项目需要什么知识就讲解什么知识,并做适当地知识扩展。摒弃了以往学习结束后采用集中实训来完成的方式。

之所以选择“班级学生成绩管理系统”作为贯穿全书的项目,其原因如下。

(1) 项目本身不陌生,有一种亲切感

软件开发应当由专业工程师和程序开发工程师共同完成。读者要开发某个软件时,他实际上同时承担了这两种职责。一个熟悉的项目,有利于读者不用过多地纠缠专业细节,有利于读者专注于开发程序相关知识的学习。

(2) 项目不宜过大、过繁

过大的项目不适应初学者。一个真正的学生成绩管理系统软件,包含的内容十分丰富,涉及面太广,读者一时难以抓住课程的精髓。我们在项目名前加上“班级”二字,其目的是简化软件功能,有利于基础知识与基本技能的学习。因此,我们开发的是一个“学习型”的软件。

(3) 知识的覆盖面能满足学习的需要

第一,“班级学生成绩管理系统”虽然是学习型的软件,但它还是包含了学习 C 语言程序设计的大部分知识点和技能点,只有少部分内容不会涉及。我们将暂时没有涉及的内容安排在“扩展知识与理论”一节中,可供自由选择。

第二,C 语言教科书中也有把学生成绩管理系统作为实践内容的,但它们大多是安排在最后,作为集中实训项目;有的教科书用某个项目来贯穿整个教学过程,但它们较多地改变了 C 语言前后关联的知识结构,这都不利于初学者学习。合理地安排各单元任务,尽量保证知识本来的结构体系,能极大地提高读者的学习兴趣,提高学习的积极性,达到事半功倍的效果。

第三,本书在每个单元中都安排了与本单元有关的能力训练任务,其目的是帮助读者更好地完成项目。

第四,本书安排了大量的实例。大量的实例有利于读者接触各种程序设计方法,有些实例还用不同的方法进行了多次设计,这样做会使读者更好地了解 and 掌握程序开发的灵活性。全部实例都用 VC++6.0 调试通过。

第五,本书结合编者多年的教学实践,在除第一单元之外的其他各单元中增加了初学者常见错误及处理方法一节,其目的就是想使读者少走弯路,尽快掌握程序开发方法。

第六,本书每单元结束后都安排了一定量的实训内容,便于读者巩固所学的知识。

第七,本书的教学课件、深入训练答案和项目程序,可在 <http://www.eszy.edu.cn> 网站中下载。

段昌盛、文晓华、杨平华、高寿斌四位同志参加了本书编写大纲的讨论,并提出了许多宝贵意见,在此表示感谢。

本书在编写过程中,还得到了恩施职业技术学院领导、教务处和计算机与信息工程系的大力支持,在此一并表示衷心的感谢。

由于水平和能力有限,本书难免有不当之处,恳请读者多加批评指正。

编 者
2009 年元月

目 录

单元 1 系统设计

- 1.1 任务 1:“班级学生成绩管理系统”总体规划设计 (1)
- 1.2 必备知识与理论 (3)
 - 1.2.1 C 语言的特点 (3)
 - 1.2.2 几个简单的 C 语言程序 (4)
 - 1.2.3 VC++6.0 开发工具介绍 (6)
- 1.3 扩展知识与理论 (10)
 - 1.3.1 算法的概念 (11)
 - 1.3.2 算法的描述方法 (11)
- 1.4 深入训练 (12)
- 习题 1 (13)

单元 2 项目数据设计与数据运算

- 2.1 任务 2:“班级学生成绩管理系统”中相关数据设计 (14)
- 2.2 必备知识与理论 (15)
 - 2.2.1 数据类型概述 (15)
 - 2.2.2 常量与变量 (16)
 - 2.2.3 简单数据类型 (17)
 - 2.2.4 数据运算符及其表达式 (23)
 - 2.2.5 不同数值型数据间的混合运算 (31)
- 2.3 扩展知识与理论 (33)
 - 2.3.1 位运算符和位运算 (33)
 - 2.3.2 常见错误及处理方法 (36)
- 2.4 深入训练 (37)
- 习题 2 (37)

单元 3 项目封面与菜单的初步设计

- 3.1 任务 3:用输入/输出函数初步设计项目封面与菜单····· (41)
- 3.2 必备知识与理论····· (45)
 - 3.2.1 C 语句····· (45)
 - 3.2.2 格式化输入/输出函数····· (46)
- 3.3 扩展知识与理论····· (53)
 - 3.3.1 单个字符输入/输出函数····· (53)
 - 3.3.2 常见错误及处理方法····· (55)
- 3.4 深入训练····· (55)
- 习题 3····· (56)

单元 4 项目封面、菜单的顺序执行设计

- 4.1 任务 4:项目封面、菜单的顺序执行设计····· (59)
- 4.2 必备知识与理论····· (61)
 - 4.2.1 顺序结构程序设计····· (61)
 - 4.2.2 顺序结构特点····· (62)
- 4.3 深入训练····· (62)
- 习题 4····· (63)

单元 5 项目菜单的选择执行设计

- 5.1 任务 5:用 if 语句实现菜单的选择执行设计····· (66)
- 5.2 任务 6:用 switch 语句实现菜单的选择执行设计····· (68)
- 5.3 必备知识与理论····· (69)
 - 5.3.1 if 语句和条件运算····· (69)
 - 5.3.2 switch 语句····· (74)
- 5.4 常见错误及处理方法····· (78)
- 5.5 深入训练····· (78)
- 习题 5····· (79)

单元 6 项目菜单的循环选择执行设计

- 6.1 任务 7:用循环语句实现项目主菜单的选择执行设计··· (82)
- 6.2 任务 8:用循环语句实现项目主、子菜单的选择执行设计····· (85)
- 6.3 必备知识与理论····· (88)
 - 6.3.1 for 语句····· (89)
 - 6.3.2 while 语句····· (91)
 - 6.3.3 do-while 语句····· (92)
 - 6.3.4 循环的嵌套····· (93)
 - 6.3.5 break 语句与 continue 语句····· (95)
- 6.4 扩展知识与理论····· (97)
 - 6.4.1 良好的源程序书写习惯····· (97)

6.4.2 常见错误及处理方法	(99)
6.5 深入训练	(100)
习题6	(100)

单元7 项目的整体框架设计

7.1 任务9:项目的整体框架设计	(103)
7.2 必备知识与理论	(109)
7.2.1 结构化程序设计思想与函数分类	(109)
7.2.2 函数的定义与调用	(110)
7.2.3 函数的嵌套调用和递归调用	(114)
7.2.4 函数调用中的参数传递	(117)
7.3 扩展知识与理论	(119)
7.3.1 变量的作用域	(119)
7.3.2 变量的生存期	(123)
7.3.3 预处理命令	(126)
7.3.4 常见错误及处理方法	(131)
7.4 深入训练	(132)
习题7	(133)

单元8 项目中数组的应用

8.1 任务10:初步完善学生最高、最低等成绩查找	(136)
8.2 任务11:初步完善学生成绩排序	(140)
8.3 必备知识与理论	(142)
8.3.1 数组概述	(142)
8.3.2 一维数组的定义及其应用	(142)
8.3.3 数组作函数参数	(145)
8.3.4 字符数组的定义及其应用	(150)
8.4 扩展知识与理论	(157)
8.4.1 二维数组的定义及其应用	(157)
8.4.2 常见错误及处理方法	(162)
8.5 深入训练	(163)
习题8	(163)

单元9 项目中指针的应用

9.1 任务12:用指针实现学生最高、最低等成绩查找	(166)
9.2 任务13:用指针实现学生成绩排序	(169)
9.3 必备知识与理论	(170)
9.3.1 内存地址与数据指针的概念	(170)
9.3.2 指向变量的指针变量	(172)
9.3.3 数组指针和指向数组的指针变量	(181)
9.3.4 字符串的指针访问法	(186)
9.4 扩展知识与理论	(190)

9.4.1	二维数组元素的指针访问方式	(190)
9.4.2	指针数组与带参数的 main 函数	(193)
9.4.3	常见错误及处理方法	(198)
9.5	深入训练	(200)
	习题 9	(200)

单元 10 项目中结构体的应用

10.1	任务 14:用结构体实现数据的增加、删除、修改和显示	(203)
10.2	必备知识与理论	(210)
10.2.1	结构体概述	(210)
10.2.2	结构体类型的应用	(211)
10.2.3	结构体数组的应用	(223)
10.3	扩展知识与理论	(229)
10.3.1	结构体变量作函数类型	(229)
10.3.2	共用体类型的应用	(231)
10.3.3	枚举类型的应用	(238)
10.3.4	常见错误及处理方法	(240)
10.4	深入训练	(241)
	习题 10	(242)

单元 11 项目中学生数据的存储与重用

11.1	任务 15:项目中学生数据的存储和重复使用	(245)
11.2	必备知识与理论	(247)
11.2.1	文件的概念	(247)
11.2.2	文件的基本操作	(248)
11.2.3	文件的定位	(257)
11.3	扩展知识与理论	(261)
11.3.1	读/写字符串和格式化读/写数据函数	(261)
11.3.2	文件状态检测	(267)
11.3.3	常见错误及处理方法	(268)
11.4	深入训练	(269)
	习题 11	(269)
	附录	(274)
	参考文献	(279)

单元1 系统设计

能力目标

- 学会与人打交道,完成调查任务。
- 能够启动 Visual C++ 6.0,并能正确进入编程窗口。
- 能根据实例完成三种类型的简单 C 语言程序编写。
- 能初步掌握“班级学生成绩管理系统”工作模块构成。

知识目标

- 了解 C 语言的特点和 C 语言程序的开发步骤。
- 初步了解 C 程序的组成结构和主函数的作用。
- 初步掌握 C 语言流程图和 N-S 图的图例特点与属性。

学习提示

本单元主要介绍“班级学生成绩管理系统”的设计要点和主要模块,介绍 C 语言程序设计中的一些基本概念以及开发应用程序的主要步骤。通过本单元的学习,使读者对用 C 语言开发程序有一个概括性的了解,并能够模仿例题编写一些简单程序。

1.1 任务 1:“班级学生成绩管理系统”总体规划设计

为了使学习者对开发应用软件有较为系统的了解,我们设计了一个贯穿整个教学过程的程序设计任务——班级学生成绩管理系统。之所以在“学生成绩管理系统”前加上“班级”二字,就是为了简化学习的难度,将学习的注意力放在主要功能的实现上。

学生成绩的统计与管理,是学校教学管理中的重要管理内容,它关系到学生是否能正常毕业。随着学校规模的扩大和管理要求的提高,传统的手工管理方法就不适应当前学校教学管理的需要。另外,学生学习成绩的计算机自动化管理也是衡量一个学校管理水平高低的标志,它能高效、方便地管理学生成绩。

要开发一个班级学生成绩管理系统软件,可以先走访本校的教务管理部门,了解学生

成绩管理方法,经过分析得出管理流程,按管理流程设计出管理模块。如果是正式开发管理软件,这个工作要经过与用户单位充分的讨论、论证,最后得出一致的意见。

下面就学习型的“班级学生成绩管理系统”的主要功能进行说明。

“班级学生成绩管理系统”共设计了六大功能模块。

- (1) 打开文件模块,能够打开保存在磁盘上的学生成绩文件。
 - (2) 保存文件模块,能将一个班 40 个学生的学号、姓名、三门课程的学习成绩和总成绩以及平均成绩全部保存在磁盘文件中。
 - (3) 编辑成绩模块,能编辑学生信息和学生成绩,并能进行相应的增加、删除、修改等操作。
 - (4) 显示成绩模块,能显示全部学生信息、指定学生信息、不及格学生信息和按总成绩排序后的学生信息。
 - (5) 计算模块,能对学生成绩进行总成绩与平均成绩计算,能找出全班学习成绩最好的和最差的学生。
 - (6) 程序说明模块,能对软件的版本、功能、使用方法、开发者信息等进行相应说明。
- 另外,本系统只有一个出口,程序只能通过该出口正常结束,以保证安全退出系统。六大功能模块可以用图 1.1 表示。

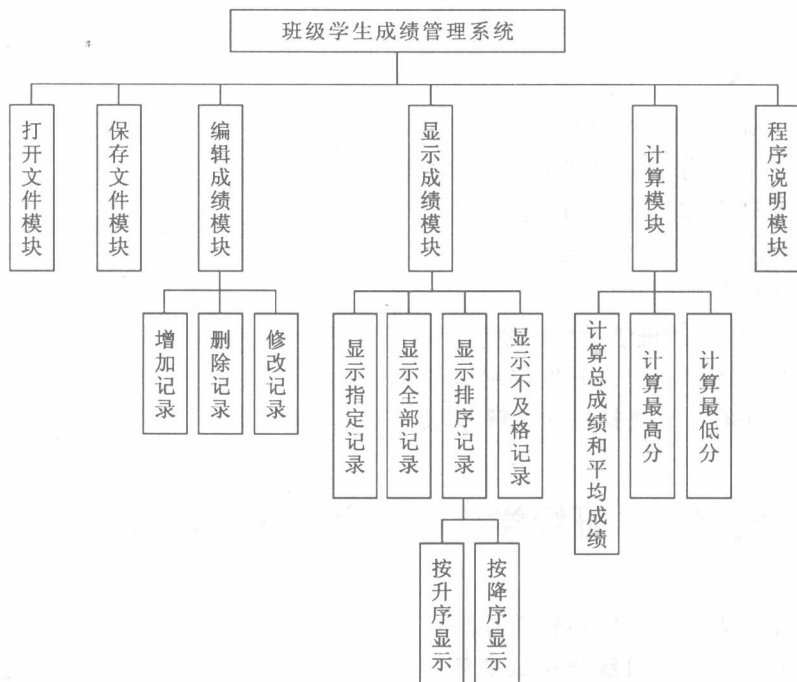


图 1.1 “班级学生成绩管理系统”功能模块示意图

开发“班级学生成绩管理系统”项目遵循“软件开发工作流程”和“循序渐进”原则,分任务实施。本书各单元首先以一至两个任务为驱动,围绕完成任务设计必备知识与理论,力争程序开发步骤与知识水平、能力紧密结合,使学习与应用融为一体,学用结合,学用相长。通过学习的深入逐步完善程序功能,最后形成一个完整的程序。随着知识的逐渐增

多,学习者也可以自己增加新模块,使程序更趋完善,更加实用。

项目开发实施方案分别如下。

任务 1:“班级学生成绩管理系统”总体规划设计。

任务 2:“班级学生成绩管理系统”中相关数据设计。

任务 3:用输入/输出函数初步设计项目封面与菜单。

任务 4:项目封面、菜单的顺序执行设计。

任务 5:用 if 语句实现菜单的选择执行设计。

任务 6:用 switch 语句实现菜单的选择执行设计。

任务 7:用循环语句实现项目主菜单的选择执行设计。

任务 8:用循环语句实现项目主、子菜单的选择执行设计。

任务 9:项目整体的框架设计。

任务 10:初步完善学生最高、最低等成绩查找。

任务 11:初步完善学生成绩排序。

任务 12:用指针实现学生最高、最低等成绩查找。

任务 13:用指针实现学生成绩排序。

任务 14:用结构体实现数据的增加、删除、修改和显示。

任务 15:项目中学生数据的存储和重复使用。

单元能力训练任务分别如下。

任务 A:调查了解本校学生成绩管理的工作流程。

任务 B:画出本校学生成绩管理工作流程模块图。

任务 C:正确安装 Visual C++ 6.0 软件。

任务 D:模仿任务 A 设计“学生通讯录”工作流程模块图。

1.2 必备知识与理论

C 语言是一种中级语言,它既具有高级语言便于学习与开发,接近人类自然语言的特点,又具有低级语言能直接操作计算机硬件的特点。这两大特点使它成为开发系统软件和应用软件强有力的工具。C 语言是目前国际上广泛流行的一种结构化的程序设计语言。

1.2.1 C 语言的特点

C 语言与其他许多语言相比,其特点如下。

(1) C 语言总共只有 32 个关键字(见附录 II),9 种控制语句(见附录 III),语言简洁、紧凑,使用方便、灵活。

(2) C 语言共有 34 种运算符(见附录 IV)。运算符丰富,由运算符组成的表达式类型多样化。(一般而言,学习 C 语言的过程就是学习掌握关键字、控制语句和运算符的过程。)

(3) C 语言的类型丰富,既有系统定义的简单类型,如整型、实型、字符型等,又有用户自定义的构造类型,如数组类型、结构体类型、共用体类型等。

(4) C 语言利用三种简单的控制结构(顺序结构、分支结构、循环结构)就能实现任何复杂结构。采用函数作为程序的模块单位,便于实现程序的模块化。

(5) C 语言语法限制不太严格,程序设计自由度大。

(6) C 语言生成目标代码质量高,一般只比汇编语言生成的目标代码效率低 10%~20%。

(7) 用 C 语言编写的程序可移植性好。一般不用修改就能用于各种型号的计算机和各种操作系统。

(8) C 语言学习难度较大,特别是函数、指针、地址等内容难度大,需要认真学习才能掌握。

尽管 C 语言没有其他语言好掌握,但对编写系统软件和应用软件,用 C 语言明显优于其他语言。它还是后续课程(如数据结构、软件工程)的先修课,所以 C 语言是一种十分优秀而又十分重要的语言,如果想成为一名优秀的软件工程师,就必须认真地学好 C 语言。

1.2.2 几个简单的 C 语言程序

为了使学习者对 C 语言有一个初步认识,下面先通过三个简单实例来分析 C 语言程序的结构。

【例 1.1】 一个最简单的 C 语言程序,在屏幕上显示“Hello, world”。

```
1 #include <stdio.h>
2 main( )                               /* 主函数 */
3 {
4     printf("Hello,world\n"); /* 输出函数 */
5 }
```

程序运行结果:

```
Hello,world
```

这是世界上第一个 C 语言程序,整个程序只有一个主函数(main),主函数中只有一条输出语句,这条语句放在一对大括号中,它的功能是在系统默认的输出设备(显示器)上输出“Hello,world”,实现这一功能是使用系统提供的标准库函数 printf()。

注意:为了方便理解和教学,给程序加了行号,但在上机调试时不能带行号。

【例 1.2】 求两数之和。

```
1 #include <stdio.h>
2 main ( )
3 {
4     int a,b,sum;                       /* 定义了三个整型变量 */
5     a=1234;
6     b=5678;
7     sum=a+b;
8     printf("sum=%d\n",sum); //打印 a 和 b 之和
9 }
```


程序运行结果：

```
sum=6912
```

本程序的作用是求两个整数 a 和 b 之和。第 1 行是文件包含命令行；第 2 行 main 表示主函数，函数语句包含在一对大括号中；第 4 行定义三个变量；第 5 行和第 6 行分别给变量 a 和 b 赋值；第 7 行将 a、b 之和赋给 sum；第 8 行使用输出函数输出两数之和。另外，/*、*/ 之间和//之后的文字分别表示注释，为便于理解，用汉字进行注释。

【例 1.3】 从键盘上输入两个整数，比较两数，将大的数输出。

```
1 int max (int x,int y)//定义 max 函数,形式参数 x,y 为整型
2 {
3     int z; /*max 函数中的声明部分,定义变量 z 为整型*/
4     if(x>y)z=x;
5     else z=y;
6     return(z); /*将 z 的值返回,通过函数调用将值带回到调用处*/
7 }
8 void main ( )
9 {
10    int max(int x,int y);      //函数声明
11    int a,b,c;
12    printf("输入两个整型数据:");/*输入提示*/
13    scanf ("%d,%d",&a,&b);    /*输入变量 a 和 b 的值*/
14    c=max(a,b);               /*调用 max 函数,将返回的值赋给 c*/
15    printf("max=%d",c);      /*输出 c 的值*/
16 }
```

程序运行结果：

输入两个整型数据：5,9<回车>（注：本书中用户输入的部分均添加下划线）

```
max=9
```

本程序包括两个函数：主函数和被调用的函数 max。max 函数的作用是比较 x 和 y 的值，并将较大者赋给变量 z，return 语句将 z 的值返回给主调函数 main，main 函数中的 scanf() 是系统提供的标准库函数，其功能是输入 a 和 b 的值，&a 和 &b 中的“&”的含义是“取地址”，即将两个输入的数值分别赋给变量 a 和变量 b 的地址所标识的单元中，也就是将两数值输入给变量 a 和 b，程序第 14 行的功能为调用 max 函数。

综合上述三个例题，可以得到 C 语言程序的基本特点如下。

(1) C 语言程序是由函数构成的。它可以由一个主函数(main)组成，也可以由一个 main 函数和若干个其他函数组成，一个 C 语言源程序有一个，而且只有一个 main 函数。因此，函数是 C 语言程序的基本单位。被调用的函数可以是系统提供的函数(如 printf() 和 scanf())，也可以是用户根据需要自己设计的函数(如 max())。程序的全部工作都是由各个函数来完成的，编写 C 语言程序就是编写一个个函数。

(2) 一个函数由以下两部分组成。

① 函数的首部，即函数的第一行。包括函数名、函数类型、函数属性、函数参数名、参

数类型等。一个函数名后面必须跟一对圆括号,这也是函数的标志,函数参数可以没有,如 `main()`。

② 函数体,即函数首部下面的大括号`{ }`内的部分。如果一个函数内有多个大括号,则最外层的一对`{ }`为函数体的范围。

函数体包括以下两部分。

声明部分:声明所用到的变量或声明所调用的函数。例如,例 1.2 中第 4 行,例 1.3 中第 3、10、11 行。

执行部分:由若干语句组成。例如,例 1.2 中第 5 行至第 8 行。

当然,有时也可以没有声明部分,甚至既没有声明部分,也没有执行部分。例如,

```
dump() { }
```

这是一个空函数,什么都不做,但它是合法的,这样的函数主要起占位作用。

(3) 不论 `main` 函数在程序中的什么位置,C 语言程序总是从 `main` 函数开始执行。一般而言,`main` 函数执行完后程序也就结束了。也就是说,`main` 函数是程序的入口和出口。

(4) C 语言程序书写格式自由,一行内可以有几条语句,一条语句也可以写成多行。C 语言程序没有行号,但每条语句和数据定义的最后必须有一个分号。分号是 C 语言语句的必要组成部分。

(5) C 语言本身没有输入/输出语句。输入/输出是通过库函数 `scanf()` 和 `printf()` 等函数来完成的。

(6) C 语言可以用 `/* */`(块注释)对 C 语言程序中的任何部分作注释,VC++ 中还可以用 `//`(行注释)来给程序加注释,两者的区别在于 `/* */` 可以对多行进行注释,而 `//` 只能对单行进行注释。注释部分是不会参加编译的。

1.2.3 VC++ 6.0 开发工具介绍

1. C 语言程序上机调试步骤

(1) 编辑。编辑源程序,一般用文字处理软件编写,当然也可以用集成化的程序设计软件,其中包括了文字处理部分。C 语言源程序的扩展名为 `.C`。

(2) 编译。源程序编写好之后,可以进行编译。编译是将源程序转换成二进制文件,即目标文件,扩展名为 `.obj`(注意:源程序中的注释是不会被编译的)。在编译过程中,将发现在源程序编写过程中出现的错误,这种错误一般是由书写错误造成的,因此,这种错误我们形象地称为语法错误,这种错误是易于修改的。

(3) 链接。编译成功后的文件并不能运行,因为这种程序虽然称为目标文件,但仍是半成品,不能执行。在目标程序中还没有为函数、变量等安排具体的地址,因此也称为浮动程序。所以链接就是将若干目标文件加以归并、整理,为所有的函数、变量分配具体的地址,同时将库函数链接到 `.obj` 文件中,生成可执行的文件,扩展名为 `.exe`。

在链接的过程中也可能发现错误,这种错误是由设计不足或缺陷引起的,一般不易发现,我们称这种错误为逻辑错误。

(4) 运行。根据运行的不同目的,运行可分为应用运行、测试运行和调试运行。

① 应用运行是指程序正式投入使用后的运行,目的是通过程序运行完成预先设定的功能,从而获得相应的效益。

② 测试运行是应用运行前的试运行,是为了验证整个应用系统的正确性,如果发现错误,应进一步判断错误的原因和产生错误的大致位置,以便加以纠正。

③ 调试运行是专门为验证某些函数的正确性而进行的,被运行的主函数通常就是一个调试程序。运行时,通过输入一些特定的数据,观察它是否产生预期的输出结果。如果发现任何不正常的情况,应配合使用程序跟踪等手段,观察程序是否按预期的流程运行,程序中的某些变量的值是否如预期的那样变化,从而判定出错的具体原因和位置,以便加以纠正。

上机调试程序步骤如图 1.2 所示。

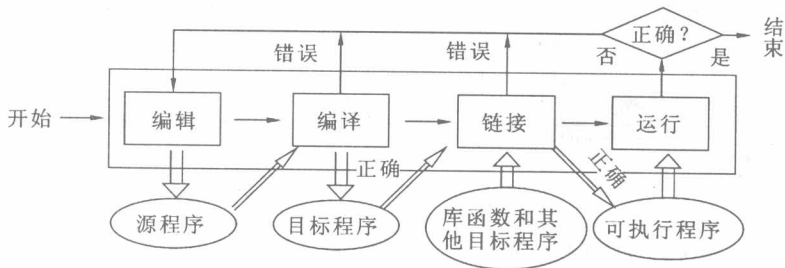


图 1.2 上机调试程序步骤

2. 开发工具介绍

1) 启动 VC++ 程序

在“开始”菜单中选择“程序”菜单,再选择“Microsoft Visual Studio 6.0”菜单,在下拉菜单中点击“Microsoft Visual C++ 6.0”,如图 1.3 所示。

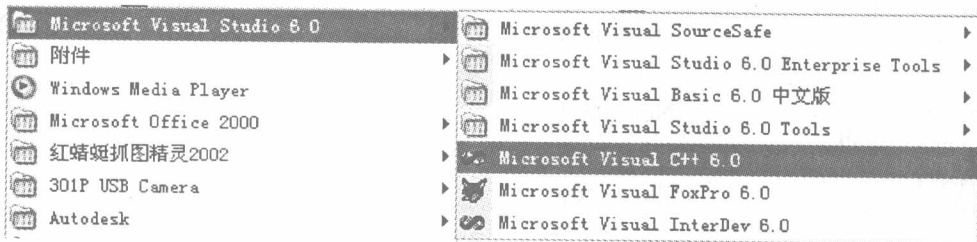


图 1.3 打开 VC++ 6.0

随后进入 VC++ 编辑窗口,如图 1.4 所示。

VC++ 编辑窗口和一般的 Windows 窗口并无太大的区别。它由标题栏、菜单栏、工具栏、工作区、编辑区、调试信息显示区和状态栏组成。在没有编辑文件的情况下,工作区无信息显示,编辑区为深灰色。

2) 菜单栏和工具栏

由于 VC++ 能够编辑 C++ 语言程序,而 C++ 语言程序又是 C 语言的超集,功能