

21世纪高职高专规划教材

计算机应用系列

C语言程序设计教程

卢宇清 主编



清华大学出版社

21 世纪高职高专规划教材

计算机应用系列

C 语言程序设计教程

卢宇清 主编

清华大学出版社

北 京

内 容 简 介

本书采用“任务驱动、突出实践应用能力培养”的模式,注重理论联系实际的教学原则,由浅入深地对C语言程序设计内容进行了详细的讲述。全书共分13章,包括C语言程序设计概述;基本数据类型、运算符及表达式;基本输入输出语句及顺序结构设计;选择结构程序设计;循环结构程序设计;数组;函数及编译预处理;指针;结构体及共用体;位运算;文件;面向对象程序设计基础;学生管理系统综合实训。每章均以课前导读、课堂讲解、上机实战、课后练习的结构讲述。

本教材不仅可以作为高职高专院校程序设计课程的教材,也可以作为成人高校、广播电视大学等各类高等院校的教材,同时还可作为相关等级考试的教材,以及程序设计爱好者的自学用书。

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

C语言程序设计教程/卢宇清主编. —北京:清华大学出版社,2009.5

21世纪高职高专规划教材. 计算机应用系列

ISBN 978-7-302-19606-8

I. C… II. 卢… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第022196号

责任编辑:张龙卿(sdzlq123@163.com)

责任校对:李梅

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

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

印 刷 者:北京市世界知识印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260

印 张:21

字 数:507千字

版 次:2009年5月第1版

印 次:2009年5月第1次印刷

印 数:1~4000

定 价:34.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。
联系电话:010-62770177 转 3103 产品编号:029317-01

前 言

为适应我国高等职业教育对教学改革和教材建设的需要,根据教育部《关于加强高职高专教育人才培养工作的意见》、《关于加强高职高专教育教材建设的若干意见》和《关于全面提高高等职业教育教学质量的若干意见》的精神,由清华大学出版社组织全国有关高职高专院校编写了此套全国高职高专计算机类规划教材。《C 语言程序设计教程》是系列教材之一。

本教材按照高等职业教育“基础理论以应用为目的,以必须、够用为度”的原则,力求从实际应用的需要出发,尽量减少枯燥死板的理论概念,加强应用性和可操作性的内容,改革教学方法和手段,融“教、学、做”为一体,强化学生能力的培养,坚持理论、操作、实训并重,基础、技巧、经验并举,让学生学以致用,学有所成,为社会主义现代化建设培养千百万高素质技能型专门人才。

本教材的特点是:

1. 有针对性的讲解。本教材充分考虑了高职高专类学生的特点,对于基本知识的讲解遵循深入浅出、循序渐进的原则,让学生能够轻松地掌握 C 语言的语法和程序构成。对于重点和难点采用分析问题、展开问题、归纳总结的方法进行说明,让学生能够对知识灵活运用,为将来自学其他计算机课程打下良好的基础。

2. 采用任务驱动教学。通过“任务”来诱发、加强和维持学习者的成就动机。对于重点章节,采用了问题引出法,让学生带着疑问去学习新知识,从而增加了知识的趣味性,学生的学习目标也会更加明确。

3. 注重实际应用。注重培养学生的学习能力、实践能力,着力提升创新能力和管理能力。通过大量典型例题、每章节后配备的实验环节以及第 13 章的学生管理系统综合实训,让学生更快地将所学理论知识和实践应用相结合。

4. 结构布局新颖。每章均以课前导读、课堂讲解、上机实战、课后练习的结构讲述。课前导读指出每课内容的基础、重点、难点与学习方法,便于指导读者自学,方便教师讲授;课堂上详细讲解每课的知识;上机实战紧密结合课堂讲解的内容给出实例,指导读者边学边用;课后练习结合每课内容给出填空题、选择题、问答题、判断题等练习题,通过课后练习,读者可达到巩固每课知识的目的。另外,前 11 章全面讲解 C 语言程序设计内容;第 12 章讲解面向对象程序设计基础,为后续课程做好铺垫,第 13 章以学生管理系统作为综合实训项目,增强学生的实践能力。

本教材由河南农业职业学院的卢宇清老师任主编,河南农业职业学院的郭小粉老师和新乡学院的张同光老师任副主编,河南经贸职业学院的孙琪老师、河南农业职业学院的左艳丽、赵小丽、周志强、方自远老师,以及秦皇岛职业技术学院的廉志凯老师、河南农业大学的



周小刚老师也参加了编写。具体编写分工为:卢宇清编写第6章,郭小粉编写第2章,张同光编写第12和13章,孙琪编写第3和4章,左艳丽编写第1、5章,赵小丽编写第7章,周志强编写第8章,方自远编写第10和11章,廉志凯编写第9章,周小刚编写附录,最后由卢宇清总纂成书。

在本教材的编写过程中,得到了各参编老师所在院校领导的大力支持,在此一并表示感谢。同时,在编写过程中,我们参考了有关教材、著作、某些网站的资料,在此也一并表示感谢。

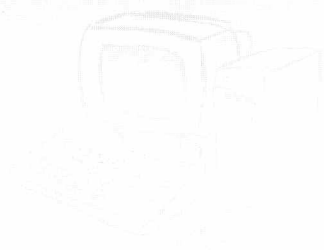
本教材的编写成员都是从事本课程教学工作的一线教师,由于教学、教研、科研任务繁重,时间仓促,书中难免存在不足和疏漏之处,敬请广大读者和同行不吝指正。编者电子邮箱为 hnaclyq@126.com。

编 者

2009年1月



目 录



第 1 章 C 语言程序设计概述	1
1.1 C 语言概述	1
1.1.1 程序设计概述	1
1.1.2 C 语言的发展	2
1.1.3 C 语言的特点	3
1.2 简单的 C 程序构成及格式	4
1.2.1 C 程序实例	4
1.2.2 C 程序的基本组成	5
1.3 C 语言的执行过程及上机步骤	6
1.3.1 C 语言的执行过程	6
1.3.2 C 语言开发环境介绍	7
1.3.3 C 语言的上机步骤	11
本章小结	11
实验	11
思考与练习	13
第 2 章 基本数据类型、运算符及表达式	15
2.1 C 语言的数据类型	15
2.2 标识符、常量和变量	16
2.2.1 标识符	16
2.2.2 常量	18
2.2.3 变量	19
2.3 整型数据	20
2.3.1 整型常量	20
2.3.2 整型变量	21
2.4 实型数据	23
2.4.1 实型常量	23
2.4.2 实型变量	24
2.5 字符型数据	25



2.5.1	字符型常量	25
2.5.2	字符型变量	27
2.5.3	字符串常量	27
2.6	C 语言的运算符和表达式	28
2.6.1	C 语言运算符简介	28
2.6.2	算术运算符及算术表达式	29
2.6.3	关系运算符及关系表达式	32
2.6.4	逻辑运算符及逻辑表达式	33
2.6.5	条件运算符及条件表达式	35
2.6.6	赋值运算符及赋值表达式	36
2.6.7	逗号运算符及逗号表达式	37
2.7	数据类型转换	38
2.7.1	自动类型转换	38
2.7.2	强制类型转换	39
	本章小结	40
	实验	41
	思考与练习	43
第 3 章	基本输入输出语句及顺序结构程序设计	48
3.1	C 语言的语句	48
3.2	数据输入输出的基本概念	50
3.3	字符数据的输入输出	50
3.3.1	字符输入函数 getchar	50
3.3.2	字符输出函数 putchar	50
3.4	格式输入输出	51
3.4.1	格式输入函数 scanf	51
3.4.2	格式输出函数 printf	56
3.5	C 程序的三种基本结构	64
3.6	综合应用举例	65
	本章小结	68
	实验	69
	思考与练习	71
第 4 章	选择结构程序设计	72
4.1	选择结构的引出	72
4.2	if 语句	73
4.2.1	if 语句的三种形式	73
4.2.2	if 语句的嵌套	77
4.3	switch 语句	79



4.4 综合应用举例	82
本章小结	86
实验	87
思考与练习	89
第5章 循环结构程序设计	91
5.1 循环结构的引出	91
5.2 goto 语句以及 goto 语句构成的循环	92
5.3 while 语句	93
5.4 do-while 语句	94
5.5 for 语句	96
5.6 break 语句和 continue 语句	98
5.6.1 break 语句	98
5.6.2 continue 语句	99
5.7 几种循环语句的比较	100
5.8 循环的嵌套	102
5.9 综合应用举例	103
本章小结	106
实验	106
思考与练习	109
第6章 数组	114
6.1 数组的引出	114
6.2 一维数组	115
6.2.1 一维数组的定义	115
6.2.2 一维数组的引用	116
6.2.3 一维数组的初始化	118
6.2.4 一维数组的应用举例	119
6.3 二维数组及多维数组	121
6.3.1 二维数组的定义	121
6.3.2 二维数组的引用	123
6.3.3 二维数组的初始化	124
6.3.4 二维数组的应用举例	126
6.4 字符数组及字符串	128
6.4.1 字符数组	128
6.4.2 字符串	131
6.4.3 字符串处理函数	132
6.4.4 字符数组应用举例	136
6.5 综合应用举例	138



本章小结·····	140
实验·····	140
思考与练习·····	144
第7章 函数及编译预处理 ·····	152
7.1 函数的引出·····	152
7.1.1 模块化设计思想·····	152
7.1.2 函数的分类·····	153
7.2 函数的定义·····	154
7.2.1 无参函数·····	155
7.2.2 有参函数·····	156
7.2.3 空函数·····	157
7.3 函数的参数及其返回值·····	158
7.3.1 形式参数和实际参数·····	158
7.3.2 函数间参数传递·····	160
7.3.3 函数的返回值·····	161
7.4 函数的调用·····	165
7.4.1 函数调用·····	166
7.4.2 函数调用的方式·····	166
7.4.3 被调用函数的声明·····	167
7.4.4 函数的嵌套调用·····	168
7.4.5 函数的递归调用·····	168
7.5 变量的作用域及存储类别·····	171
7.6 编译预处理·····	176
7.6.1 宏定义·····	176
7.6.2 文件包含·····	180
7.7 综合应用举例·····	181
本章小结·····	183
实验·····	183
思考与练习·····	185
第8章 指针 ·····	190
8.1 指针的引出·····	190
8.2 地址和指针的概念·····	190
8.3 指针和指针变量·····	191
8.3.1 指针和指针变量·····	191
8.3.2 指针变量的定义及初始化·····	192
8.3.3 指针变量的基本运算·····	196
8.4 指针和数组·····	198



8.4.1 指向数组的指针	198
8.4.2 字符指针	202
8.4.3 指针数组	204
8.5 指针与函数	206
8.5.1 指针作函数的参数	206
8.5.2 指针型函数	210
8.5.3 指向函数的指针	211
8.6 综合应用举例	212
本章小结	214
实验	214
思考与练习	215
第9章 结构体及共用体	221
9.1 问题的引出	221
9.2 结构体	223
9.2.1 结构体类型的定义	223
9.2.2 结构体变量的初始化	225
9.2.3 结构体变量的引用	226
9.3 结构体数组	228
9.3.1 结构体数组的定义	228
9.3.2 结构体数组的初始化	229
9.3.3 结构体数组的应用	229
9.4 共用体	232
9.4.1 共用体的概念及特点	232
9.4.2 共用体类型的定义	233
9.4.3 共用体变量的引用	233
9.5 枚举	234
9.6 用 typedef 定义类型	236
9.7 综合应用举例	237
本章小结	238
实验	239
思考与练习	239
第10章 位运算	242
10.1 位运算符和位运算	242
10.1.1 按位与运算符(&)	243
10.1.2 按位或运算符()	243
10.1.3 按位异或运算符(^)	244
10.1.4 按位取反运算符(~)	245



10.1.5 左移运算符(<<)	246
10.2 位段	246
10.3 综合应用举例	249
本章小结	250
实验	250
思考与练习	251
第 11 章 文件	252
11.1 文件概述	252
11.1.1 文件的概念	252
11.1.2 文件的分类	253
11.1.3 文件类型指针	253
11.1.4 缓冲文件系统和非缓冲文件系统	253
11.2 文件的操作	254
11.2.1 文件的打开	254
11.2.2 文件的关闭	256
11.2.3 文件的读写	256
11.2.4 文件的定位	262
11.3 综合应用举例	263
本章小结	265
实验	265
思考与练习	266
第 12 章 面向对象程序设计基础	267
12.1 面向对象程序设计基本概念	267
12.2 类和对象	270
12.3 数据的抽象和封装	273
12.4 继承性	275
12.5 多态性	277
本章小结	279
实验	280
思考与练习	280
第 13 章 学生管理系统综合实训	281
13.1 系统设计要求	281
13.2 系统设计及函数实现	281
13.2.1 系统设计	281
13.2.2 函数设计	282
13.3 参考程序	285
13.4 运行结果	302



本章小结.....	303
实验.....	303
思考与练习.....	304
附录 1 Turbo C 编译错误信息	305
附录 2 C 语言常用关键字及说明	311
附录 3 ASCII 代码表.....	312
附录 4 C 语言运算符及优先级	313
附录 5 Turbo C 常用库函数.....	315
参考文献	323

第1章 C语言程序设计概述

课前导读

C语言是国际上广泛流行的高级程序设计语言,它集高级语言和低级语言于一体,既可以用来编写系统程序,也可以用来编写应用程序。本章主要介绍了C语言的特点和发展,C语言程序的构成,C语言程序的执行过程和C语言开发工具的使用方法。

知识目标

- ☐ 了解C语言的发展和特点。
- ☐ 掌握C语言程序的构成和执行过程。
- ☐ 掌握Turbo C 2.0的使用方法。

能力目标

- ☐ 熟悉C语言程序的执行过程。
- ☐ 能正确使用Turbo C 2.0对C语言程序进行编辑、编译、连接和运行。

1.1 C语言概述

C语言编写的程序既有操作系统、编译程序、汇编程序、数据库管理程序等系统软件,也有数值计算、文字处理、控制系统、游戏等应用软件。

1.1.1 程序设计概述

1. 程序设计

在日常生活中我们可以看到,计算机已经广泛应用在各个领域,用来解决各种类型的问题。计算机之所以能够产生如此大的影响,其原因不仅在于人们发明了机器本身,更重要的是人们为计算机开发出了不计其数的能够指挥计算机完成各种各样工作的程序。正是这些功能丰富的程序给了计算机无穷的生命力。而程序设计就是用某种程序语言编写这些程序的过程。

那么,如何进行程序设计呢?一个简单的程序设计一般包含以下四个步骤:

- (1) 分析问题,建立数学模型。使用计算机解决具体问题时,首先要对问题进行充分的



分析,确定问题是什么,采用什么方法去解决。针对所要解决的问题,找出已知的数据和条件,确定所需的输入、处理及输出对象,建立起解决问题的数学模型。

(2) 确定数据结构和算法。根据建立的数学模型,来确定存放数据的数据结构,并选择合适的算法加以实现。

(3) 编写程序。根据确定的数据结构和算法,使用选定的计算机语言编写程序代码。

(4) 调试程序。在计算机上用各种可能的输入数据对程序进行调试,分析所得到的运行结果,进行程序的测试和调整,直至获得预期的结果。

由此可见,一个完整的程序要涉及四个方面的问题:数据结构、算法、编程语言和程序设计方法。这四个方面的知识都是程序设计人员所必须具备的,其中算法是至关重要的一个方面。

2. 算法

什么是算法? 一个算法,就是一个有穷规则的集合,这些规则规定了一个解决某一特定类型的问题的运算序列。简单地说,任何解决问题的过程都是由一定的步骤组成的,把解决问题确定的方法和有限的步骤称作为算法。一个算法应当具有以下基本特征:

(1) 有穷性。一个算法必须在执行有限个操作步骤后终止。

(2) 确定性。算法中每一步的含义必须是确切的,不可出现任何二义性。相同的输入应输出相同的结果。

(3) 有效性。算法中的每一步操作都应该能有效执行,一个不可执行的操作是无效的。例如,一个数被 0 除的操作就是无效的,应当避免这种操作。

(4) 有零个或多个输入。在算法中用到的数据在大多数情况下需要通过输入来得到。

(5) 有一个或多个输出。在一个完整的算法中至少应有一个输出。

1.1.2 C 语言的发展

对 C 语言的研究起源于系统程序设计的深入研究和发 展。C 语言的前身是 ALGOL60,1963 年,英国的剑桥大学和伦敦大学将 ALGOL60 发展成 CPL(Combined Programming Language,混合编程语言)。1967 年,英国剑桥大学的 M. Richards 在 CPL 语言的基础上,实现并推出了 BCPL(Basic Combined Programming Language,基础混合编程语言)语言。1970 年,美国贝尔实验室的 K. Thompson 以 BCPL 语言为基础,设计了一种类似于 BCPL 的语言,称为 B 语言。他用 B 语言在 PDP-7 机上实现了第一个实验性的 UNIX 操作系统。1972 年,贝尔实验室的 Dennis M. Ritchie 为克服 B 语言的诸多不足,在 B 语言的基础上重新设计了一种语言,由于是 B 语言的后继,故称为 C 语言。到了 1973 年,K. Thompson 和 Dennis M. Ritchie 两个人合作把 UNIX 的 90% 以上内容用 C 语言进行了改写,即大家熟知的 UNIX 第五版。1977 年出现了独立于机器的 C 语言编译文本。1978 年贝尔实验室正式发表了 C 语言。1983 年,ANSI(American National Standards Institute)为 C 语言制定了新的标准,称为 ANSI C。ANSI C 标准于 1989 年被采用,该标准一般称为 ANSI/ISO Standard C。于是,1989 年定义的 C 标准定义为 C89,到了 1995 年,出现了 C 的修订版,其中增加了一些库函数,出现了初步的 C++,在此基础上,C89 成为 C++



的子集。此后,C 语言不断发展,在 1999 年又推出了 C99,C99 在基本保留了 C 的特性的基础上增加了一系列新的特性,随后又几经修改和完善,它也从面向过程的编程语言发展到面向对象的程序设计语言,目前可在微机上运行的 C 语言版本主要有 Microsoft C/C++、Turbo C、Quick C、Visual C/C++ 等版本。C 语言的发展历程如图 1-1 所示。

1.1.3 C 语言的特点

C 语言之所以如此广泛使用,与它具有的其他语言所不可比拟的特点有密切的关系。C 语言的特点如下。

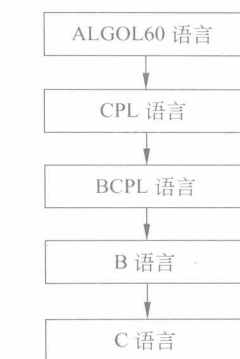


图 1-1 C 语言的发展历程

1. C 语言是结构化程序设计语言

C 语言提供了结构化程序所需的基本控制语句:如用于选择结构的 if 语句和 switch 语句;用于循环结构的 while 语句和 for 语句。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言的源程序由函数组成,每个函数都是独立的模块,可单独编译,生成目标代码,也可以与其他语言连接生成可执行文件,而且调试、维护起来比较方便。

2. C 语言具有丰富的数据类型

C 的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据类型的运算,并引入了指针概念,使程序效率更高。

3. C 语言简洁易读,运算符和库函数丰富

C 语言一共有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示。C 语言还提供了 30 多种运算符,运算能力十分丰富,同时,C 系统提供了大量的标准库函数可直接调用,大大提高了程序设计的效率和质量。

4. C 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而 C 语言允许程序设计人员有较大的自由度。

5. C 程序移植性好

C 语言具有效率高,程序可移植等特点。C 语言程序本身独立于机器硬件,可从一种环境到另一种环境中运行。被广泛地移植到各类计算机上,从而形成了多种版本的 C 语言。

综上所述,C 语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来,不失为一个出色而有效的现代通用程序设计语言。它一方面在计算机程序语言研究方面具有一定价值,由它引出了许多后继语言。另一方面,C 语言对整个计算机工业和应用的发展都起了很重要的推动作用。



1.2 简单的 C 程序构成及格式

1.2.1 C 程序实例

用 C 语言语句编写的程序称为 C 程序或 C 源程序。本节通过两个简单的 C 程序实例,介绍 C 程序的基本组成和结构,使读者对 C 语言和 C 程序的特性有初步的了解。

【例 1-1】 输出三行字符串。

```
/* This is a simple program! */
main ()
{
    printf("*****\n");
    printf(" Happy birthday to you !\n ");
    printf("*****\n");
}
```

【程序结果】

```
*****
Happy birthday to you !
*****
```

【结果分析】

(1) 程序第一行 `/* ..... */` 表示注释部分,注释仅仅是为程序设计人员及程序使用者方便理解程序而附加在程序中的说明信息,对程序的运行功能是不起作用的。注释内容可以用汉字也可以用英语,注释可以放在程序中的任何位置。

(2) 程序第二行是 C 程序的主函数, `main` 为主函数名。 `main` 后的 `()` 是函数的参数部分,括号内可为空,但括号不能省略。每个 C 语言源程序都必须有且只有一个主函数 `main`。

(3) 程序第三行和第七行对应一对花括号 `{ }`,花括号内语句的集合构成函数体,是 C 程序必不可少的重要组成部分。本例中的函数体由三个语句组成,每个语句都以分号结尾。其中 `printf` 是 C 语言提供的标准输出库函数,其作用是将双引号内的字符串原样输出, `"\n"` 是换行控制符。

【例 1-2】 求正方形的面积。

```
#include "stdio.h"           /* 文件包含命令 */
main()                       /* 主函数定义 */
{
    float a, area;           /* 定义局部变量 */
    printf("please input a:");
    scanf("%f", &a);         /* 输入边长 */
    area=a * a;
    printf("area=%f\n", area); /* 输出正方形的面积 */
}
```

**【程序结果】**

```
please input a:3.5 ✓  
area=12.250000
```

【结果分析】

(1) 主函数 main() 前的一行语句称为预处理命令, 这里的 include 称为文件包含命令, 其意义是把 "" 或 <> 内指定的文件包含进来, 成为本程序的一部分。被包含的文件通常是由系统提供, 其扩展名为 ".h", 因此也成为头文件。预处理命令后不能加 ";"。

(2) 主函数体中用到了输入函数 scanf() 和输出函数 printf()。scanf() 和 printf() 是标准输入输出函数, 其头文件为 stdio.h。C 语言规定, 对 scanf() 和 printf() 这两个函数可以省去对其头文件的包含命令, 所以本例中也可省略第一行的包含命令: #include "stdio.h"。

(3) 主函数体由两部分构成: 说明部分和执行部分。

① 说明部分完成变量的类型说明。C 语言规定, 程序中用到的变量都必须先说明, 后使用。本例中函数体的第一行为说明部分。

② 从函数体的第二行到第五行都为执行部分, 执行部分用来完成程序的功能。执行部分的第一行调用 printf() 函数在显示器上输出提示字符串, 提示用户输入变量 a 的值。第二行调用 scanf() 函数, 接受从键盘上输入的数据。第三行用来计算正方形的面积。第四行调用 printf() 函数, 将正方形的面积输出。

1.2.2 C 程序的基本组成

由以上两个例子可以看到, C 程序的一般组成形式如下:

```
包含的头文件  
子函数类型说明  
main()                /* 主函数说明 */  
{ 变量定义            /* 主函数体 */  
  执行语句组  
}  
子函数名 1(参数)      /* 子函数说明 */  
{ 变量定义            /* 子函数体 */  
  执行语句组  
}  
子函数名 2(参数)      /* 子函数说明 */  
{ 变量定义            /* 子函数体 */  
  执行语句组  
}  
:  
子函数名 N(参数)      /* 子函数说明 */  
{ 变量定义            /* 子函数体 */  
  执行语句组  
}
```

其中, “子函数名 1”至“子函数名 N”是用户自定义的函数。

由此可见, 一个完整的 C 程序应符合以下几点: