

新世纪
高等职业教育规划教材

C语言程序设计

孙永林 主编



 **机械工业出版社**
CHINA MACHINE PRESS

新世纪高等职业教育规划教材

C 语言程序设计

主 编 孙永林

副主编 田银磊

参 编 余明艳

主 审 刘 波

机械工业出版社

北京机械工业出版社

C语言是计算机专业学生必须学习的语言,而且也是一种通用的程序设计语言。现在流行的 VC++和 C++Builder 面向对象的程序设计,都是以C语言为基础的语言。

本书共分十章,介绍了C语言的基本概念、语法以及C语言程序设计方法,并通过实例进行说明。在本书的各章之后,都提供了课后的练习题和上机题。

本书是高职高专系列教材,为了能更好地适应高职高专学生的特点,在编写中着重考虑了把本书编写得深入浅出,易于掌握。由于高职高专计算机专业学生大多数都要参加全国计算机等级考试和计算机专业软件水平考试中的初级程序员考试,为了适应学生的考试要求,本书还介绍了 Turbo C 2.0 的使用方法和编译错误信息表及错误分析,并在综合应用举例中引用了考试题型进行程序设计分析。

本书是针对C语言初学者而编写的教材,内容安排合理、概念清晰、例题丰富、通俗易懂,是一本适应高职高专学生学习C语言的好教材,也可作为自学使用。

图书在版编目(CIP)数据

C语言程序设计/孙永林主编.—北京:机械工业出版社,2003.5
新世纪高等职业教育规划教材
ISBN 7-111-11760-3

I. C… II. 孙… 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(2003)第018389号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
责任编辑:王小东 卢若薇 版式设计:霍永明 责任校对:唐海燕
封面设计:张静 责任印制:闫焱
北京瑞德印刷有限公司印刷·新华书店北京发行所发行
2003年4月第1版·第1次印刷
1000mm×1400mm B5·8.625印张·331千字
定价:22.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
本社购书热线电话(010)68993821、88379646
封面无防伪标均为盗版

新世纪高等职业教育规划教材编审委员会

主任委员	李维东	广东白云职业技术学院	常务副院长
副主任委员	陈周钦	广东交通职业技术学院	院长
	石令明	广西柳州职业技术学院	院长
	蔡昌荣	广州民航职业技术学院	副院长
	覃洪斌	广西职业技术学院	副院长
	姚和芳	湖南铁道职业技术学院	副院长
	韩雪清	机械工业出版社教材编辑室	副主任
委	郑伟光	广东机电职业技术学院	院长
员	张尔利	广西交通职业技术学院	院长
	谈向群	无锡职业技术学院	副院长
	刘国生	番禺职业技术学院	副院长
	陈大路	温州职业技术学院理工学区	主任
	邹 宁	广西机电职业技术学院	副院长
	修德明	济源职业技术学院	副院长
	管 平	浙江机电职业技术学院	副院长
	韦荣敏	广西柳州市交通学校	校长
	田玉柯	遵义航天工业学校	校长
	黄秀猛	厦门市工业学校	校长
	韩书平	新乡市高级技工学校	校长
	张毓琴	广东白云职业技术学院	兼委员会 秘书

编写说明

20世纪90年代以来,我国高职高专教育为社会主义现代化建设事业培养了大批急需的各类专门人才,提高了劳动者的素质,对于建设社会主义的精神文明,促进社会进步和经济发展起到了重要作用。中共中央、国务院《关于深化教育改革,全面推进素质教育的决定》指出:“要大力发展高等职业教育”,教育部在《教育部关于加强高职高专教育人才培养工作的意见》中明确指出:“高职高专教育是我国高等教育的重要组成部分,培养拥护党的基本路线,适应生产、建设、服务第一线需要的,德、智、体、美等方面全面发展的高等技术应用性专门人才;学生应在具有必备的基础理论知识和专门知识的基础上,重点掌握从事本专业领域实际工作的基本能力和基本技能。”加入WTO以后,我国将面临人才资源的全球竞争,其中包括研究开发型人才的竞争,也包括专业技能型优秀人才的竞争。高等职业教育要适应我国现代化建设的需要,适应世界市场和国际竞争的需要,培养大批符合市场需求的、有熟练技能的高等技术应用性人才。

教材建设是整个高职高专教育教学工作中的重要环节,在贯彻国家教育教改精神、保证人才质量方面起着重要作用。改革开放以来,各地已出版了一批高职高专教材,但从整体上看,具有高职高专教育特点的教材极其匮乏,教材建设仍滞后于高职高专教育的发展需要。为此,根据目前高等职业教育发展的要求,机械工业出版社组织全国多所在高等职业教育办学有特色、在社会上影响较大的高职院校成立了“新世纪高等职业教育规划教材编审委员会”,选择教学经验丰富、实践能力强的骨干教师,组织、规划、编写了这套“新世纪高等职业教育规划教材”,教材首批四个系列36本(书目附后)。它凝聚着全体编审人员、编委会委员的大量心血,同时得到了各委员院校的大力支持,在此表示衷心感谢。

本套教材的作者队伍是经编审委员会严格遴选确定的,他们来自高等职业教育的第一线,教学经验丰富、业务上乘、文笔过硬,大多是各校学科和专业的带头人。他们对本专业的课程设置、教学大纲、教学教改都有深刻的认识和独到的见解,对高职教育的特色把握能力强,有较高的编写水平。这些都为编写出具有创新性、适用性强的教材打下了良好基础。

本套教材的编写以保证基础、加强应用、体现先进、突出以能力为本位的职教特色为指导思想,在内容上遵循“宽、新、浅、用”的原则。所谓“宽”,即知识面宽,适用面广;所谓“新”,就是要体现新知识、新技术、新工艺、新方

法；所谓“浅”，是指够用为度、通俗易懂；所谓“用”，就是要注重应用、面向实践。

本套教材的出版，促进了高等职业教育的教材建设，将对我国高等职业教育的发展产生积极的影响。同时，我们也希望在今后的使用中不断改进、完善此套教材，更好地为高等职业教育服务，为经济建设服务。

新世纪高等职业教育规划教材编审委员会

前 言

C语言是计算机专业的基础语言，也是计算机专业学生学习必不可少的一个重要部分。在科学高度发展的今天，虽然现在计算机程序设计已流行面向对象程序设计的系统VC++和C++ Builder，但都离不开C语言作为基础语言。C语言功能齐全、表达能力强、应用面广、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。现在，用汇编语言编写的软件都具有被C语言取代的趋势，流行的UNIX操作系统，就是用C语言编写而成的。

由于C语言的概念比较复杂、规则繁多、语法灵活，因此，初学者在其编程的过程中容易出错。这一点也是自学者感到学习时难度较大的原因之一。本书的编者都是具有丰富高职高专教学经验的教师，在本书的编写过程中，能考虑到初学者学习的需要，更能考虑到适应高职高专教学方法和特点，在教材的编写中力图突出和强调可操作性这一特点。

本教材在内容上作了如下尝试：

- 本书的读者群是广大初学者。读者可能是在从来没有接触过计算机专业的基础上学习C语言，应使他们能够在入学之初的C语言学习中尽快学会C语言这门课程。

- 考虑到本书的主要对象是高职高专学生，本书在编写过程中，力图对概念讲解详细，对难点进行分解，并用实例加以解释和说明，使概念更加清晰、易懂。

- 在编写本书过程中，考虑到高职高专学生加强动手能力和操作实践培养的特点，尽量突出可操作性。举例分析时避免复杂、尖涩的数学问题，使学生在学习中专注于C语言本身的操作和使用技巧。在书中，给出了Turbo c 2.0系统的使用方法，并通过实践性环节的练习，加深对C语言概念的理解。

- 本书注意在各章节内容中采用逐步引入新概念的方法，循序渐进，使学生在在学习过程中不会感到困难，可增强学习的信心。

- 考虑到高职高专学生大部分都要参加全国计算机等级考试或计算机专业水平的初级程序员考试，本书列举了一些典型的全国计算机等级考试题和计算机专业水平考试的初级程序员试题。

本书的第1、2、3章由余明艳编写，第4、5、6章由田银磊编写，第7、8、

9、10章由孙永林编写。全书由孙永林任主编，刘波主审。

由于作者水平有限，经验不足，在编写过程中肯定会有不少缺点和错误，诚恳接受专家、老师们以及读者的指正。

编 者

目 录

前言

第 1 章 绪论	1
1.1 C 语言的发展与特点	1
1.2 结构化程序设计	4
1.3 结构化设计所用图表工具简介	6
1.4 简单的 C 程序实例解析	7
练习题	10
第 2 章 数据类型、运算符和表达式	13
2.1 C 的数据类型概述	13
2.2 常量	13
2.3 变量	17
2.4 运算符和表达式	22
2.5 Turbo C 2.0 使用简介	32
2.6 经典实例解析	38
练习题	39
上机题	40
第 3 章 C 程序流程控制的实现	42
3.1 C 语句概述	42
3.2 表达式和表达式语句	42
3.3 输入/输出函数和语句	43
3.4 复合语句	51
3.5 if 语句	53
3.6 switch 语句	56
3.7 循环语句	57
3.8 转移语句	66
3.9 空语句	69
3.10 程序举例	70
练习题	74
上机题	76

第 4 章 函数	77
4.1 函数的定义	79
4.2 函数的返回值	82
4.3 函数的调用及参数传递	84
4.4 变量的作用域	92
4.5 变量的存储类别	95
4.6 编程常见错误解析	99
练习题	102
上机题	106
第 5 章 数组	108
5.1 一维数组	108
5.2 二维数组	113
5.3 字符数组	117
5.4 程序设计举例	123
5.5 编程常见错误解析	128
练习题	130
上机题	131
第 6 章 指针	132
6.1 指针的含义及指针变量	132
6.2 指针的运算	138
6.3 C 语言的内存动态分配函数	143
6.4 指针与数组	145
6.5 指向函数的指针	154
6.6 Turbo C 命令行参数	155
6.7 编程常见错误解析	157
练习题	159
上机题	162
第 7 章 结构体及用户自定义数据类型	163
7.1 结构体	163
7.2 结构体数组	168

7.3 将结构体传递给函数	174	9.5 随机访问文件和 lseek()	222
7.4 结构体指针	176	练习题	223
7.5 结构体内部的数组和结构体	179	上机题	224
7.6 结构体的自我引用	180	第 10 章 综合应用示例	226
7.7 位域	185	10.1 完善程序	226
7.8 共用体	188	10.2 编制程序	234
7.9 枚举	189	附录	245
7.10 类型定义 typedef	191	附录 A ASC II 码表	245
练习题	193	附录 B 运算符及优先级表	246
上机题	194	附录 C ANSI C 标准库函数	247
第 8 章 编译预处理	196	C.1 数学函数	247
8.1 C 的预处理指令	196	C.2 字符函数和字符串函数	248
8.2 预定义的宏替换名	204	C.3 输入输出函数	250
练习题	205	C.4 动态存储分配函数	253
第 9 章 文件	206	附录 D 编译出错信息	254
9.1 文件与流	206	D.1 灾难性错误	254
9.2 文件类型指针	208	D.2 一般性错误	254
9.3 缓冲型文件	208	D.3 警告	261
9.4 UNIX 型文件系统	218	参考文献	263

第 1 章 绪 论

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

早期的操作系统等系统软件，主要是用汇编语言编写的，它依赖于计算机硬件，程序的可读性和可移植性都很差。若改用高级语言来编写，又难以实现汇编语言能直接对硬件进行操作的某些功能。为此，人们开始寻求一种既具有一般高级语言特性，又具有低级语言特性的语言。于是，C 语言就在这种情况下应运而生。

1960 年出现的 ALGO60 是一种面向问题的高级语言，它离硬件较远，不适宜用来编写系统程序。1963 年，英国的伦敦大学和剑桥大学在 ALGOL60 的基础上推出了 CPL (Combined Programming Language) 语言，它接近硬件一些，但规模大，难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 作了简化，推出了 BCPL(Basic Combined Programing Language)语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又作了进一步的简化，设计出了简单且很接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言编写了第一个 UNIX 操作系统，在 PDP-7 上实现。1971 年又在 PDP-11/20 上实现了用 B 语言编写的第一个 UNIX 操作系统。但 B 语言过于简单，功能有限。1972 年美国贝尔实验室的 D.M.Ritchie 在 PDP-11 机上实现了 C 语言（取自 BCPL 的第二个字母）。他以 B 语言为基础，既保持了 BCPL 的 B 语言精练、接近硬件的优点，又克服了它们过于简单、数据无类型等缺点。1973 年，K.Thompson 和 D.M.Ritchie 合作用 C 语言重写了 UNIX 操作系统，实现了 UNIX 第五版。它比原先的版本更易于理解、修改和扩充，并增加了多道程序设计功能，开创了 UNIX 系统发展的新局面。图 1-1 反映了 C 语言的发展历史。

从 C 语言发展历史来看，C 属于 ALGOL 语言族系。C 语言在发展过程中与 UNIX 相辅相成，开创了 UNIX 的新局面，使其获得巨大的成功；UNIX 促成了 C 语言的大发展，使其得到了迅速推广，从而形成一对繁荣与共的孪生兄弟。1978 年以后，C 语言逐渐独立于 UNIX 系统，独立于 PDP-11 机而蓬勃发展。以 1978 年发表的 UNIX 第七版中的 C 编译程序为基础，Brian W.Kernighan 和

Dennis M.Ritchie(合称 K&R)合作出版了名著《The C Programming Language》,由它产生了 C 语言版本的基础,称为标准 C。1983 年,美国国家标准化协会(ANSI)将标准 C 作了扩充和发展,制定了新的标准,称为 ANSI C。1988 年 K&R 按照 ANSI C 标准重写了他们的名著。目前人们常将 1978 的标准 C 称为旧标准,将 ANSI C 称为新标准。随着 C 的发展,现在 C 语言已风靡全世界,成为世界上应用非常广泛的新型的现代主程序设计语言,成为微、小、超小、大、超大和巨型等各类计算机共同使用的语言,广泛应用于系统软件(如操作系统、编译系统等)、应用软件(如图形处理)、数据处理(如企业管理)以及数值计算等各个领域。

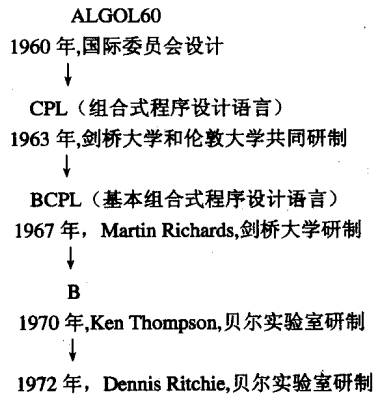


图 1-1 C 语言发展历史

1.1.2 C 语言的特点

C 语言之所以具有强大的生命力,成为国际上公认的最重要的少数几种通用程序设计语言之一,关键在于它自身的特点。概括起来,C 语言的特点是:简洁、灵活,表达能力强,代码质量高,结构化程序,可移植性好。具体可分为下列几点:

1. 语言简洁

C 语言是被公认的比较强有力的语言,经精心选用,它总共只有 32 个关键字,9 种控制语句,压缩了一切不必要的成分。C 的表示方法简洁,基本组成部分紧凑,使用一些简单、规整的方法就可以构造出相当复杂、功能很强的数据类型、语句和程序结构。如:用{、}代替 begin、end 作复合语句或函数体的“括号”;用++表示加 1;--表示减 1;运算符省写等。

2. 表达方式灵活实用

C 语言提供了多种运算符和获得表达式值的方法,对问题的表达也可通过多种途径得到,使用户在程序设计中有更大的主动性;语法限制不太严格,程序设

计自由度大，如对数组下标越界不作检查，对变量类型使用灵活（整型量与字符型数据及逻辑型数据可以通用）等。另外，语言格式自由、限制少，编写程序较自由；程序以小写字母为基础，小写字母易读易写等，充分体现出 C 语言灵活、实用、方便的特点。

3. 表达能力强

C 语言具有丰富的数据结构和运算符。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，包含了现代化语言所要求的各种数据结构。能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。C 的运算符包含的范围很广，共有 34 种运算符（见附录 B），可以实现其他高级语言难以实现的运算。C 语言可直接访问物理地址，能进行位(bit)操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 语言兼有高级语言和低级语言的许多优点，故有人称 C 为“高级语言中的低级语言”或“中间语言”，它既可以用来编写系统软件，又可以用来开发应用软件，是一门通用程序设计语言。

4. 语言生成的目标代码质量高

相对于汇编语言而言，许多高级语言生成的代码质量很低，所以，迄今汇编语言仍是编写系统软件的主要工具。但是，许多实验表明，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写的代码低 10%~20%。由于用 C 语言描述问题比用汇编语言描述编程迅速，工作量小，可读性好，而在代码质量上可与汇编语言相媲美，因此，C 语言迅速成为人们进行程序设计和软件开发的得心应手的工具。

5. 结构化程序设计

C 语言是一种结构化语言，提供了编写结构化程序所需的基本控制流结构语句，如 if-else、switch、while、do-while、for 等；用函数作为程序设计的基本单位，实现程序的模块化，是结构化的理想语言，符合现代编程风格要求。

6. 可移植性好

用 C 语言编写的程序比用汇编语言编写的程序可移植性好，基本上不作修改就可用于各种型号的计算机和各种操作系统。

虽然 C 语言的优点很多，但它也有其弱点，如运算符的优先级较多，有些还与常规约定不同，不便记忆；某些语法部分不易用形式化方法进行描述；各种 C 语言版本之间略有差别，缺乏统一的标准；C 语言语法是强类型的语言，它在强调灵活、高效的同时，在一定程度上牺牲了某些安全性，如类型检验太弱，转换比较随便等。因此，C 语言对程序设计员提出了较高的要求，尤其在使用 C 语言的某些高级手段时更是如此。但是，C 语言的优点远远超过了它的弱点，这些优点使 C 语言具有强大的吸引力。实际经验表明，程序设计人员一旦接触了

这种语言，并且有一定程序设计经验后，就会对它爱不释手。

1.2 结构化程序设计

结构化程序设计的概念最早是由 E.W.Dijkstra 提出来的，它是详细设计的逻辑基础。E.W.Dijkstra 在 1956 年一次学术会议上指出“可以从高级语言中取消 GOTO 语句”、“程序的质量与程序中所包含的 GOTO 语句的数量成反比”。1966 年，Bohm 和 Jacopini 首先证明了只用三种基本的控制结构就能够实现任何单入口单出口的程序。这三种基本的控制结构是“顺序”、“选择”、“循环”。Bohm 和 Jacopini 的这一证明给结构程序设计奠定了基础。IBM 的 Mills 在 1971 年进一步提出“程序应该只有一个入口和一个出口”，从而补充了结构程序的规则。

那么，什么是结构化程序设计呢？目前还没有一个精确的并为所有人普遍接受的定义，一个比较流行的定义是：结构程序设计是一种程序设计技术，它采用自顶向下逐步求精的设计方法和单入口单出口的控制结构。

1.2.1 基本的控制结构

1. 顺序结构

如图 1-2 所示，先执行 A 操作，再执行 B 操作，两者是顺序执行的关系。

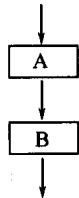


图 1-2 顺序结构

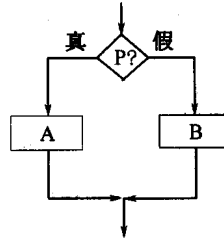


图 1-3 双分支选择结构

2. 选择结构

(1) 双分支选择结构 如图 1-3 所示，其中 P 代表一个条件，当 P 条件成立（或称为“真”）时执行 A，否则执行 B。只能执行 A 或 B 之一，两条路径汇合在一起然后出口。

(2) 多分支选择结构 由选择结构派生出一类多分支选择结构，如图 1-4 所示，根据 K 的值 (K_1, K_2, \dots, K_n) 不同决定执行 A_1, A_2, \dots, A_n 之一。

3. 循环结构

有两种典型的循环结构，即当型循环和直到型循环。

(1) 当型循环结构 见图 1-5, 当 P 条件成立 (“真”) 时, 反复执行 A 操作。直到 P 为 “假” 时才停止循环。

(2) 直到型循环结构 如图 1-6 所示, 先执行 A 操作, 再判断 P 是否为 “假”, 若 P 为假, 再执行 A, 如此反复, 直到 P 为 “真” 为止。

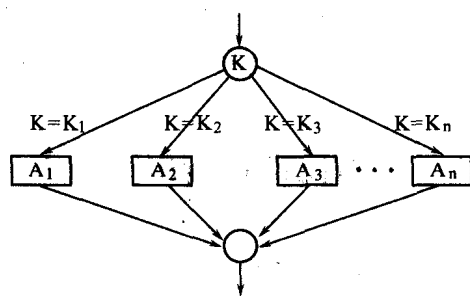


图 1-4 多分支选择结构

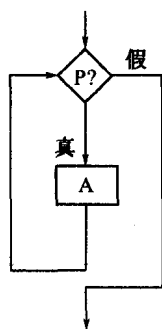


图 1-5 当型循环结构

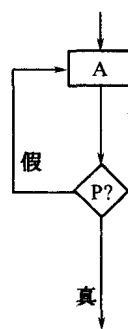


图 1-6 直到型循环结构

1.2.2 逐步求精方法

抽象是处理复杂问题的重要工具, 逐步求精就是一种具体的抽象技术, 是结构化程序设计的一种具体的抽象技术, 是结构化程序设计的一种最基本的方法。

为了解决一个复杂的问题, 人们往往不可能一开始就能了解到问题的全部细节, 而只能对问题的全局作出决策, 设计出问题本身较为自然的、很可能是自然语言表达的抽象算法。这个抽象算法由一些抽象数据及其上的操作组成 (即抽象语句), 仅仅表示解决问题的一般策略和问题解的一般结构。对抽象算法作进一步的求精, 就进入下一层抽象。在每求精一步, 抽象语句和抽象数据都将进一步分解和细化。如此继续, 直到最后的算法能够为计算机所 “理解” 为止。最终得到的算法可能是用某种高级语言或汇编语言表示出来的。

由此可见, 逐步求精的基本思想可以概括为: 从最能直接反映问题的体系结构的概念出发, 逐步精细化、具体化, 逐步补充细节, 直到设计出可以在机器上

执行的程序。简单的来说，是一种先全局后局部、先整体后细节、先抽象后具体的自顶向下的设计方法。

1.3 结构化设计所用图表工具简介

1.3.1 程序流程

程序流程图又称程序框图，它是历史悠久、使用最广泛的一种图形描述工具，其特点是简单、直观、易学。程序流程图的符号并不统一，图 1-7 给出了几种常用的符号。

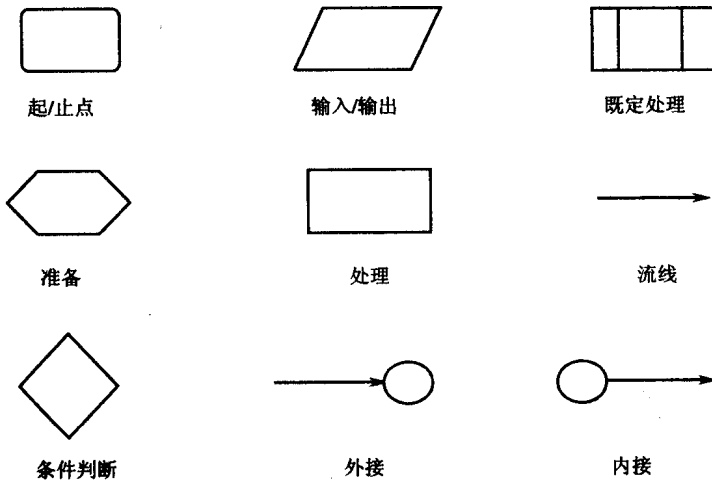


图 1-7 流程图的基本符号

1.3.2 盒图

盒图也称 N-S 图，是由 Nassi 和 Shneiderman 提出的一种符合结构化程序设计原则的图形描述工具。盒图的基本符号如图 1-8 所示。

盒图有以下的特点：①功能域(即一个特定控制结构的作用域)明确。②无法确定全局数据和局部数据的作用域。③容易确定全局数据和局部数据的作用域。④容易表示嵌套关系，也可以表示模块的层次结构。

盒图的缺点是修改比较困难，另外当分支嵌套层次多时往往在一张纸上难以画下。

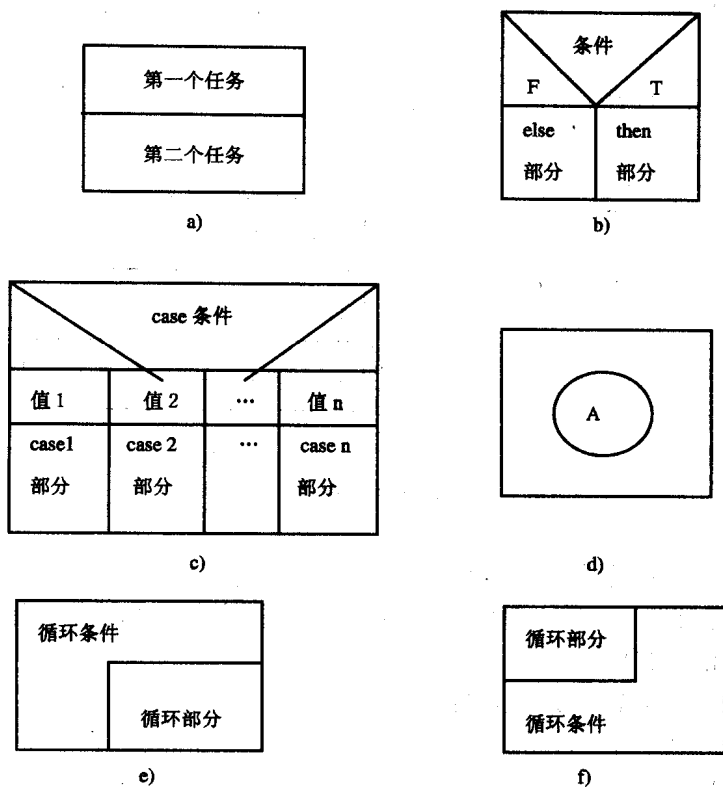


图 1-8 盒图的基本符号

a) 顺序 b) 选择 c) case d) 调用子程序 e) do-while 循环 f) do-until 循环

1.4 简单的 C 程序实例解析

1.4.1 实例解析

```

例 1-1
main()
{ printf("*****\n");
  printf("very good\n");
  printf("*****\n");
}

```

解析 从这个简单的 C 程序的例子可以看出：该源程序是由一个函数（main 函数）构成的；该 main 函数后面紧跟一对圆括号；其函数体部分是由一