



高等院校规划教材

王成端 徐翠霞 主 编
魏先民 于春花 崔玲玲 副主编

C语言程序设计

强调程序设计方法和思路，引入典型程序设计案例
注重程序设计实践环节，培养程序设计项目开发技能



含教学光盘一张



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校规划教材

C 语言程序设计

王成端 徐翠霞 主编

魏先民 于春花 崔玲玲 副主编

中国水利水电出版社

内 容 提 要

C 语言是一种结构化的程序设计语言, 它既具有高级语言的特点, 又具有低级语言的特点。

本书共分 11 章, 内容包括: 程序设计的基本概念, 基本数据类型、运算符和表达式, C 语句和结构化程序设计, 数组, 函数, 编译预处理, 指针, 结构体、共用体和枚举, 位运算, 文件, 库函数及其应用。

本书内容安排合理, 讲解通俗易懂, 对重点难点做了十分系统、清楚和详细的阐述, 并辅以大量的典型例题, 使广大读者能够通过学习本书掌握 C 语言的基本知识, 并灵活运用所学知识编写应用程序。

本书每章都配有丰富的习题, 主要包括选择题、填空题和编程题三种类型, 大部分习题都精选于全国计算机等级考试历年的考试真题, 具有典型性和启发性, 使学生能随时自我检查学习效果并获得相应的反馈信息。与本书配套的还有《C 语言程序设计实训——题解、实验、课程设计与样题》, 主教材中所有习题的答案及解析均在实训教材中给出。本书配有教学光盘, 其中包括课堂教学课件、程序源代码、习题解答、实验指导、课程设计、模拟试题等, 形成了主教材、实训教材、配套光盘组成的立体教学体系。

本书可作为高等院校计算机专业或相关专业的教材, 也可作为全国计算机等级考试的培训教程。

图书在版编目 (CIP) 数据

C 语言程序设计/王成端, 徐翠霞主编. —北京: 中国水利水电出版社, 2005
(21 世纪高等院校规划教材)

ISBN 7-5084-2813-7

I. C… II. ①王…②徐… III. C 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 028067 号

书 名	C 语言程序设计
作 者	王成端 徐翠霞 主编 魏先民 于春花 崔玲玲 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 19.75 印张 440 千字
版 次	2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷
印 数	0001—5000 册
定 价	30.00 元 (含教学光盘一张)

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

序

随着计算机科学与技术的飞速发展,计算机的应用已经渗透到国民经济与人们生活的各个角落,正在日益改变着传统的人类工作方式和生活方式。在我国高等教育逐步实现大众化后,越来越多的高等院校会面向国民经济发展的第一线,为行业、企业培养各级各类高级应用型专门人才。为了大力推广计算机应用技术,更好地适应当前我国高等教育的跨越式发展,满足我国高等院校从精英教育向大众化教育的转变,符合社会对高等院校应用型人才要求的各类要求,我们成立了“21 世纪高等院校规划教材编委会”,在明确了高等院校应用型人才模式、培养目标、教学内容和课程体系的框架下,组织编写了本套“21 世纪高等院校规划教材”。

众所周知,教材建设作为保证和提高教学质量的重要支柱及基础,作为体现教学内容和教学方法的知识载体,在当前培养应用型人才中的作用是显而易见的。探索和建设适应新世纪我国高等院校应用型人才体系需要的配套教材已经成为当前我国高等院校教学改革和教材建设工作面临的紧迫任务。因此,编委会经过大量的前期调研和策划,在广泛了解各高等院校的教学现状、市场需求,探讨课程设置、研究课程体系的基础上,组织一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人、科研人员和主要从事该课程教学的骨干教师编写出一批有特色、适用性强的计算机类公共基础课、技术基础课、专业及应用技术课的教材以及相应的教学辅导书,以满足目前高等院校应用型人才培养的需要。本套教材消化和吸收了多年来已有的应用型人才培养的探索与实践成果,紧密结合经济全球化时代高等院校应用型人才培养工作的实际需要,努力实践,大胆创新。教材编写采用整体规划、分步实施、滚动立项的方式,分期分批地启动编写计划,编写大纲的确定以及教材风格的定位均经过编委会多次认真讨论,以确保该套教材的高质量和实用性。

教材编委会分析研究了应用型人才与研究型人才在培养目标、课程体系和内容编排上的区别,分别提出了 3 个层面上的要求:在专业基础类课程层面上,既要保持学科体系的完整性,使学生打下较为扎实的专业基础,为后续课程的学习做好铺垫,更要突出应用特色,理论联系实际,并与工程实践相结合,适当压缩过多过深的公式推导与原理性分析,兼顾考研学生的需要,以原理和公式结论的应用为突破口,注重它们的应用环境和方法;在程序设计类课程层面上,把握程序设计方法和思路,注重程序设计实践训练,引入典型的程序设计案例,将程序设计类课程的学习融入案例的研究和解决过程中,以学生实际编程解决问题的能力为突破口,注重程序设算法的实现;在专业技术应用层面上,积极引入工程案例,以培养学生解决工程实际问题的能力为突破口,加大实践教学内容的比重,增加新技术、新知识、新工艺的内容。

本套规划教材的编写原则是:

在编写中重视基础,循序渐进,内容精炼,重点突出,融入学科方法论内容和科学理念,反映计算机技术发展要求,倡导理论联系实际和科学的思想方法,体现一级学科知识组织的层次结构。主要表现在:背靠计算机学科的科学体系,明确目标定位,分类组织实施,兼容互补;理论与实践并重,强调理论与实践相结合,突出学科发展特点,体现学科

发展的内在规律；教材内容循序渐进，保证学术深度，减少知识重复，前后相互呼应，内容编排合理，整体结构完整；采取自顶向下设计方法，内涵发展优先，突出学科方法论，强调知识体系可扩展的原则。

本套规划教材的主要特点是：

(1) 面向应用型高等院校，在保证学科体系完整的基础上不过度强调理论的深度和难度，注重应用型人才的专业技能和工程实用技术的培养。在课程体系方面打破传统的研究型人才培养体系，根据社会经济发展对行业、企业的工程技术需要，建立新的课程体系，并在教材中反映出来。

(2) 教材的理论知识包括了高等院校学生必须具备的科学、工程、技术等方面的要求，知识点不要求大而全，但一定要讲透，使学生真正掌握。同时注重理论知识与实践相结合，使学生通过实践深化对理论的理解，学会并掌握理论方法的实际运用。

(3) 在教材中加大能力训练部分的比重，使学生比较熟练地应用计算机知识和技术解决实际问题，既注重培养学生分析问题的能力，也注重培养学生思考问题、解决问题的能力。

(4) 教材采用“任务驱动”的编写方式，以实际问题引出相关原理和概念，在讲述实例的过程中将本章的知识点融入，通过分析归纳，介绍解决工程实际问题的思想和方法，然后进行概括总结，使教材内容层次清晰，脉络分明，可读性、可操作性强。同时，引入案例教学和启发式教学方法，便于激发学习兴趣。

(5) 教材在内容编排上，力求由浅入深，循序渐进，举一反三，突出重点，通俗易懂。采用模块化结构，兼顾不同层次的需求，在具体授课时可根据各校的教学计划在内容上适当加以取舍。此外还注重了配套教材的编写，如课程学习辅导、实验指导、综合实训、课程设计指导等，注重多媒体的教学方式以及配套课件的制作。

(6) 大部分教材配有电子教案，以使教材向多元化、多媒体化发展，满足广大教师进行多媒体教学的需要。电子教案用 PowerPoint 制作，教师可根据授课情况任意修改。相关教案的具体情况请到中国水利水电出版社网站 www.waterpub.com.cn 下载。此外还提供相关教材中所有程序的源代码，方便教师直接切换到系统环境中教学，提高教学效果。

总之，本套规划教材凝聚了众多长期在教学、科研一线工作的教师及科研人员的教学科研经验和智慧，内容新颖，结构完整，概念清晰，深入浅出，通俗易懂，可读性、可操作性和实用性强。本套规划教材适用于应用型高等院校各专业，也可作为本科院校举办的应用技术专业的课程教材，此外还可作为职业技术学院和民办高校、成人教育的教材以及从事工程应用的技术人员的自学参考资料。

我们感谢该套规划教材的各位作者为教材的出版所做出的贡献，也感谢中国水利水电出版社为选题、立项、编审所做出的努力。我们相信，随着我国高等教育的不断发展和高校教学改革的不深入，具有示范性并适应应用型人才培养的精品课程教材必将进一步促进我国高等院校教学质量的提高。

我们期待广大读者对本套规划教材提出宝贵意见，以便进一步修订，使该套规划教材不断完善。

21 世纪高等院校规划教材编委会
2004 年 8 月

前 言

C 语言是近年来国内外广泛使用的计算机程序设计语言,也是软件开发人员必须掌握的一种语言。各类高等学校中普遍开设 C 语言课程,全国计算机等级考试二级和三级中也包括 C 语言的考试。

本书内容共分 11 章。第 1 章为程序设计和 C 语言概述,主要介绍程序设计的基本概念、C 语言的特点、C 语言程序的基本结构。第 2 章为数据描述与基本操作,主要介绍 C 语言的基本数据类型、运算符和表达式。第 3 章为 C 语言程序的流程控制,主要介绍基本语句、复合语句、控制语句的格式及应用。第 4 章为数组,主要介绍一维数组、二维数组和字符数组及其应用。第 5 章为函数,主要介绍函数定义及调用、变量和函数的作用域。第 6 章为编译预处理,主要介绍宏定义、文件包含和条件编译三种命令的格式、功能和用法。第 7 章为指针,主要介绍指针的概念、指针的定义、指针的基本运算以及指针与数组和指针与函数的关系。第 8 章为结构体、共用体和枚举,主要介绍 C 语言构造类型的定义及应用。第 9 章为位运算,主要介绍各种位运算符及其应用。第 10 章为文件,主要介绍文件的基本操作和使用规则。第 11 章为库函数及其应用,主要介绍常用库函数的应用。

本书的编写力求在体系结构上安排合理、重点突出、难点分散、便于掌握;在语言描述上注重概念清晰、逻辑性强、通俗易懂、便于自学。每章均由学习目标、正文、小结和习题四部分组成。在学习本章之前,通过阅读学习目标能够了解本章的主要内容和学习后应达到的目标,使学习目标明确,增强学习的积极性和主动性;每章最后的小结是对本章内容的总结和归纳,使所学的内容条理化和系统化,进一步加深对内容的理解和认识。

本书的讲解以 87 ANSI C 为基础,采用现代风格定义和声明函数,使程序具有更好的易读性,并使系统对函数和参数的类型进行检查。本书用流程图描述算法,所有程序都按结构化程序设计方法编写,这有助于读者养成良好的程序设计习惯,编写出高质量的程序。本书中例题和习题的源代码程序均在 Turbo C 2.0 环境下通过上机调试。本书在附录中给出 Turbo C 2.0 常用库函数,供读者查阅。

为加强 C 语言程序设计的学习,还编写了配套的《C 语言程序设计实训——题解、实验、课程设计与样题》,教材中所有习题的答案及解析均在实训教材的第一部分中给出,实训的第二部分是 Turbo C 环境介绍,第三部分是实验,第四部分是课程设计,第五部分是模拟试题及参考答案。

本书由王成端、徐翠霞担任主编,魏先民、于春花、崔玲玲担任副主编。其中第 1、6、9 章由王成端编写,第 5、7 章和附录 I、附录 III 由徐翠霞编写,第 2、3 章和附录 II 由崔玲玲编写,第 4、8 章由于春花编写,第 10、11 章由魏先民编写。全书由王成端负责统稿。

由于作者的水平有限、经验不足,编写时间仓促,书中难免存在许多缺点及不足之处,恳请广大读者和同行批评指正。

编者

2005 年 1 月

目 录

序

前言

第 1 章 程序设计及 C 语言概述.....	1
本章学习目标	1
1.1 程序设计的基本概念	1
1.1.1 程序和程序文件	1
1.1.2 程序设计和程序设计语言	1
1.2 算法	3
1.2.1 算法及其特征	3
1.2.2 算法设计的要求	3
1.2.3 算法的组成要素	4
1.2.4 算法的描述	4
1.3 C 语言的特点	6
1.4 简单的 C 语言程序	7
1.5 C 语言程序的开发	8
本章小结	10
习题一	10
第 2 章 数据描述与基本操作	11
本章学习目标	11
2.1 基本标识符	11
2.1.1 关键字	11
2.1.2 预定义标识符	12
2.1.3 用户定义标识符	12
2.2 基本数据类型	13
2.3 常量和变量	14
2.3.1 变量	14
2.3.2 常量	16
2.4 运算符与表达式	20
2.4.1 运算符与表达式简介	20
2.4.2 算术运算符与算术表达式	20
2.4.3 赋值运算符与赋值表达式	23
2.4.4 关系运算符与关系表达式	24
2.4.5 逻辑运算符与逻辑表达式	25

2.4.6	条件运算符与条件表达式	27
2.4.7	逗号运算符与逗号表达式	28
2.4.8	其他运算符	28
2.5	数据的类型转换	29
2.5.1	自动类型转换	29
2.5.2	强制类型转换	30
2.6	数据的输入和输出	30
2.6.1	字符输入输出函数	31
2.6.2	格式输入输出函数	32
2.7	应用举例	35
	本章小结	38
	习题二	38
第 3 章	C 语言程序的流程控制	42
	本章学习目标	42
3.1	C 语言程序的三种基本结构	42
3.1.1	顺序结构	42
3.1.2	选择结构	42
3.1.3	循环结构	43
3.2	C 语句概述	44
3.2.1	基本语句	44
3.2.2	复合语句	45
3.2.3	控制语句	45
3.3	选择结构程序设计	45
3.3.1	if 语句（条件语句）	45
3.3.2	switch 语句	49
3.4	循环结构的程序设计	52
3.4.1	for 语句	52
3.4.2	while 语句	53
3.4.3	do-while 语句	54
3.4.4	循环语句的嵌套	55
3.5	流程转向语句	58
3.5.1	break 语句和 continue 语句	58
3.5.2	return 语句	60
3.6	典型应用算法	61
3.6.1	穷举算法	61
3.6.2	迭代算法	62
	本章小结	65

习题三	66
第 4 章 数组	73
本章学习目标	73
4.1 一维数组	73
4.1.1 一维数组的定义	73
4.1.2 一维数组元素的引用	75
4.1.3 一维数组的初始化	76
4.1.4 一维数组应用举例	77
4.2 二维数组	84
4.2.1 二维数组的定义	84
4.2.2 二维数组元素的引用	85
4.2.3 二维数组的初始化	85
4.2.4 二维数组应用举例	87
4.3 字符数组与字符串	88
4.3.1 字符数组的定义和初始化	88
4.3.2 字符串的存储及字符串结束标志	90
4.3.3 字符串的输入和输出	91
4.3.4 字符串处理函数	92
4.3.5 字符数组应用举例	96
本章小结	98
习题四	99
第 5 章 函数	103
本章学习目标	103
5.1 模块化程序设计及 C 程序结构	103
5.1.1 模块化程序设计	103
5.1.2 C 程序结构	104
5.2 函数定义	105
5.3 函数调用	106
5.3.1 函数调用的格式	106
5.3.2 函数调用的方式	106
5.3.3 函数的嵌套调用	108
5.3.4 递归函数及递归调用	111
5.4 函数原型	113
5.5 调用函数和被调用函数之间的数据传递	114
5.5.1 实参—形参之间的数据传递	115
5.5.2 函数调用结果的返回	118
5.6 变量的作用域及其存储类型	121

5.6.1 作用域和生存期	121
5.6.2 局部变量及其作用域和存储类型	122
5.6.3 全局变量及其作用域和存储类型	124
5.7 函数的作用域	128
5.7.1 内部函数	129
5.7.2 外部函数	129
5.8 函数应用举例	129
本章小结	135
习题五	136
第 6 章 编译预处理	142
本章学习目标	142
6.1 概述	142
6.2 宏定义	143
6.2.1 不带参数的宏定义	143
6.2.2 带参数的宏定义	145
6.3 文件包含	147
6.4 条件编译	149
6.4.1 #ifdef 命令	149
6.4.2 #ifndef 命令	150
6.4.3 #if 命令	151
本章小结	152
习题六	153
第 7 章 指针	157
本章学习目标	157
7.1 指针概述	157
7.1.1 指针的基本概念	157
7.1.2 指针变量的定义及初始化	158
7.1.3 指向指针的指针	159
7.2 指针的赋值	160
7.3 指针的运算	162
7.3.1 指针运算符 “*”	162
7.3.2 指针的算术运算	162
7.3.3 指针的关系运算	164
7.3.4 指针的强制类型转换	164
7.4 指针与数组	165
7.4.1 一维数组元素的指针访问方式	165
7.4.2 二维数组元素的指针访问方式	167

7.4.3	字符指针与字符串	170
7.4.4	指向一维数组的指针	173
7.4.5	指针数组	174
7.5	指针与函数	177
7.5.1	指针作为函数参数	177
7.5.2	指针函数	179
7.5.3	指向函数的指针	180
7.6	带参数的 main 函数及其应用	184
7.6.1	命令行参数	184
7.6.2	带参数的 main 函数	184
7.7	指针应用举例	185
	本章小结	187
	习题七	188
第 8 章	结构体、共用体和枚举	194
	本章学习目标	194
8.1	结构体	194
8.1.1	结构体类型的定义	194
8.1.2	结构体类型变量的定义和初始化	196
8.1.3	结构体成员的引用	198
8.2	结构体和函数	200
8.2.1	结构体类型的变量作为函数参数	200
8.2.2	结构体类型的变量作为函数返回值	202
8.3	链表	202
8.3.1	链表的定义及基本结构	203
8.3.2	动态存储分配函数	204
8.3.3	链表的基本操作	205
8.4	共用体	209
8.4.1	共用体类型的定义	210
8.4.2	共用体变量的定义	210
8.4.3	共用体成员的引用	211
8.5	枚举	213
8.5.1	枚举类型和枚举型变量的定义	213
8.5.2	枚举型变量的使用	215
8.6	typedef 类型定义	216
8.6.1	类型定义的基本格式	216
8.6.2	类型定义的使用说明	217
8.7	综合应用实例	218

本章小结	220
习题八	221
第 9 章 位运算	226
本章学习目标	226
9.1 概述	226
9.2 位运算	227
9.2.1 按位取反运算符	227
9.2.2 按位与运算符	227
9.2.3 按位或运算符	229
9.2.4 按位异或运算符	230
9.2.5 左移运算符	230
9.2.6 右移运算符	231
9.3 位运算应用举例	232
9.4 位段	233
本章小结	235
习题九	236
第 10 章 文件	238
本章学习目标	238
10.1 文件概述	238
10.1.1 文件	238
10.1.2 文件系统	240
10.2 文件指针和文件读写指针	241
10.2.1 文件指针	241
10.2.2 文件读写指针	241
10.3 文件的打开和关闭	242
10.3.1 文件的打开	242
10.3.2 文件的关闭	243
10.4 文件的读写	244
10.4.1 fputc 函数和 fgetc 函数	244
10.4.2 fputs 函数和 fgets 函数	247
10.4.3 fprintf 函数和 fscanf 函数	248
10.4.4 fwrite 函数和 fread 函数	250
10.5 文件的定位	252
10.5.1 fseek 函数	252
10.5.2 rewind 函数	253
10.5.3 ftell 函数	253
10.5.4 文件的随机读写	254

10.6 错误处理	255
10.6.1 ferror 函数	255
10.6.2 clearerr 函数	256
10.7 文件的重定向	256
10.8 文件应用举例	257
本章小结	259
习题十	260
第 11 章 库函数及其应用	264
本章学习目标	264
11.1 Turbo C 标准库	264
11.1.1 库文件和目标文件的比较	264
11.1.2 头文件	264
11.2 数学函数	265
11.2.1 三角函数	266
11.2.2 指数和函数	267
11.2.3 其他数学函数	268
11.3 字符串函数	269
11.3.1 字符串基本处理函数	270
11.3.2 字符串与数值转换函数	271
11.4 字符屏幕处理函数	272
11.4.1 字符窗口的定义	272
11.4.2 字符窗口颜色的设置	273
11.4.3 窗口字符的输入输出函数	275
11.4.4 窗口字符的其他处理函数	275
11.5 图形处理函数	277
11.5.1 图形模式的初始化	277
11.5.2 屏幕颜色的设置和清屏函数	279
11.5.3 基本图形函数	279
11.5.4 封闭图形的填充	281
11.5.5 图形窗口和图形屏幕操作函数	283
11.5.6 图形模式下的字符输出	285
本章小结	287
习题十一	287
附录 I 常用字符与 ASCII 码对照表	288
附录 II 运算符的优先级和结合方向	289
附录 III 库函数	290
参考文献	301

第 1 章 程序设计及 C 语言概述

本章学习目标

本章主要介绍程序设计的基本概念、C 语言的特点、简单的 C 语言程序及开发 C 语言程序的主要步骤。通过本章的学习，要求做到：

- 了解程序、程序文件、程序设计等基本概念。
- 了解算法的概念及其特征，熟悉常用的描述算法的方法。
- 了解 C 语言对程序设计方法的支持及 C 语言的基本特点。
- 熟练掌握简单 C 语言程序的基本结构，能够编写包含键盘输入、计算和输出等操作的简单 C 语言程序。
- 了解 C 语言程序开发的过程，学会最简单的 C 语言程序开发。

编写软件需要使用程序设计语言，目前可用的程序设计语言很多，各有其特点。有些语言适用于开发数据库应用程序，有些语言适用于开发科学计算程序，有的简便易学，有的功能全面。在本书中，我们将介绍 C 语言。

1.1 程序设计的基本概念

1.1.1 程序和程序文件

程序就是用程序设计语言对要完成的任务进行描述。

一个程序一般应包含以下两方面内容：一是对数据的描述，在程序中要指定数据的类型和数据的组织形式，即数据结构；二是对操作步骤的描述，也就是算法。

程序必须以文件的方式存储在计算机中，存放程序的文件称为程序文件。对于 C 语言，源代码程序文件约定的扩展名为 c。

1.1.2 程序设计和程序设计语言

1. 程序设计

所谓程序设计就是把解题步骤用程序设计语言描述出来的工作过程。简单的程序设计一般包含以下几个部分：

(1) 确定数据结构。

根据任务提出的要求、指定的输入数据和输出结果，确定存放数据的数据结构。

(2) 确定算法。

针对存放数据的数据结构来确定解决问题、完成任务的步骤。

(3) 编写源代码程序。

根据确定的数据结构和算法,使用选定的程序设计语言编写程序代码,简称编程。

(4) 调试和运行程序。

通过对程序的调试消除由于疏忽而引起的语法错误或逻辑错误;用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据能进行适当的处理。

2. 程序设计语言

人们利用计算机解决实际问题,一般要编写程序。程序设计语言就是用户用来编写程序的语言,它是人与计算机之间交换信息的工具。

程序设计语言一般分为机器语言、汇编语言和高级语言三类。

(1) 机器语言。

机器语言是最底层的计算机语言。用机器语言编写的程序,计算机硬件可以直接识别和运行。在用机器语言编写的程序中,每一条机器指令都是二进制形式的指令代码。在指令代码中一般包括操作码和地址码,其中操作码告诉计算机作何种操作,地址码则指出被操作的对象。例如代码 10000000 表示加法操作,而代码 10010000 表示减法操作。

对于不同的计算机硬件(主要是 CPU),其机器语言是不同的。因此,针对一台计算机所编写的机器语言程序不能在另一台计算机上运行。由于机器语言程序是直接针对计算机硬件的,因此它的执行效率比较高,能充分发挥计算机的速度性能。但是,用机器语言编写程序的难度比较大,容易出错,而且程序的直观性比较差,也不容易移植。

(2) 汇编语言。

为便于理解与记忆,人们采用能帮助记忆的英文缩写符号(称为指令助记符)来代替机器语言指令代码中的操作码,用地址符号来代替地址码。例如 ADD 表示加法运算操作码, SUB 表示减法运算操作码。用指令助记符及地址符号书写的指令称为汇编指令(也称符号指令),而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称为符号语言。

汇编语言也是与具体使用的计算机相关的。由于汇编语言采用了助记符,因此,它比机器语言直观,容易理解和记忆。用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。但是,计算机不能直接识别源程序,必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后,计算机才能识别并执行。这种翻译的过程称为“汇编”,负责翻译的程序称为汇编程序。

(3) 高级语言。

机器语言和汇编语言都是面向机器的语言,一般称为低级语言。低级语言对机器的依赖性太大,用它们开发的程序通用性差,普通的计算机用户也很难胜任这一工作。

随着计算机技术的发展以及计算机应用领域的不断扩大,从 20 世纪 50 年代中期开始逐步发展了面向问题的程序设计语言,称为高级语言。高级语言与具体的计算机硬件无关,其表达方式接近于被描述的问题,人们易于接受和掌握。用高级语言编写程序要比低级语言容易得多,并大大简化了程序的编制和调试,使编程效率得到大幅度地提高。高级语言

的显著特点是独立于具体的计算机硬件，通用性和可移植性好。

必须指出，用任何一种高级语言编写的程序（称为源程序）都要通过编译程序翻译成机器语言程序（称为目标程序）后计算机才能执行，或者通过解释程序边解释边执行。

1.2 算法

学习计算机程序设计语言的目的，是要用语言作为工具，设计出计算机可以运行的程序。对于一个需要求解的问题，怎样才能写出程序呢？除了选定合理的数据结构外，一般来说，十分关键的一步是设计算法，有了一个好的算法，就可以选用一种程序设计语言把算法转换为程序。

1.2.1 算法及其特征

算法是指为解决某个特定问题而采取的确定且有限的步骤。一个算法应当具有以下五个特性。

1. 有穷性

一个算法应包含有限个操作步骤。也就是说，在执行若干个操作步骤之后，算法将结束，并且每一步都在合理的时间内完成。

2. 确定性

算法中每一条指令必须有确切的含义，不能有二义性，并且对于相同的输入必然有相同的执行结果。

3. 可行性

一个算法是可行的，即算法中指定的操作，都可以通过已经实现的基本运算执行有限次来实现。

4. 输入

一个算法一般有零个或多个输入。在计算机上实现的算法，通常是用来处理数据对象的，在大多数情况下这些数据对象是需要通过输入来得到的。

5. 输出

一个算法一般有一个或多个输出。算法的目的是为了求“解”，这些“解”只有通过输出才能得到。

1.2.2 算法设计的要求

通常设计一个好的算法应考虑达到以下目标。

1. 正确性

算法首先是正确的，不含语法错误或逻辑错误。算法对于一切合法的输入数据都能产生满足任务要求的输出结果。

2. 可读性

算法主要是用于人的阅读与交流，可读性好有助于人们对算法的理解与掌握。

3. 健壮性

当输入数据非法时, 算法应适当地作出反应或进行处理, 而不会产生莫名其妙的输出结果。例如, 一个求三角形周长的算法, 当输入的三条边不能构成一个三角形时, 不应继续计算, 而应报告输入出错。

4. 效率与低存储量需求

一般来说, 效率指的是算法执行时间。对于同一个问题如果有多个算法可以解决, 执行时间短的算法效率高。存储量需求指算法执行过程中所需要的最大存储空间。

1.2.3 算法的组成要素

算法是由基本操作和控制结构组成的。

1. 基本操作

每个操作的确定不仅取决于任务的需求, 还取决于所用的程序设计语言。在 C 语言中所描述的操作主要包括: 算术运算、逻辑运算、关系运算、位运算、输入和输出操作等。

2. 控制结构

每个算法都要由一系列的操作组成。同一操作序列, 按不同的顺序执行, 就会得出不同的结果。控制结构即如何控制组成算法的各操作的执行顺序。一个算法只能由三种基本结构组成, 即顺序结构、选择结构、循环结构。

1.2.4 算法的描述

算法可以采用多种描述方法进行描述, 最常用的有自然语言、伪代码、流程图、N-S 图等。

1. 用自然语言描述算法

自然语言是人们在日常生活、工作、学习中通用的语言, 一般不需专门学习和训练就能理解用这种语言所表达的意思。但用自然语言描述程序 (或算法) 的流程时, 一般要求直接而简练, 尽量减少语言上的修饰。

【例 1-1】用自然语言描述 $\text{sum}=1+2+3+\dots+99+100$ 的算法。

- (1) 定义变量 sum , 并置初值为 0。
- (2) 计算 $1+2+3+\dots+99+100$ 的和, 并将和值存于 sum 中。
- (3) 输出变量 sum 的值。

2. 用伪代码描述算法

伪代码是介于自然语言和程序语言之间的一种算法描述语言, 它具有自然语言灵活的特点, 同时又接近于程序设计语言。

算法描述语言与程序设计语言的最大区别在于, 算法描述语言比较自由, 不受语法约束, 而程序设计语言受语法约束。

算法描述语言通常是借助某种高级语言的控制结构, 中间的操作可以用自然语言, 也可以用程序设计语言, 或使用自然语言与程序设计语言的混合体。

【例 1-2】用伪代码表示 $\text{sum}=1+2+3+\dots+99+100$ 的算法。