

高等学校
计算机教材

面向应用与实践系列

郭有强 等编著

C语言程序设计

清华大学出版社





高等学校
计算机教材

面向应用与实践系列

郭有强 等编著

C语言程序设计

藏书

清华大学出版社
北京

内 容 简 介

本书采用通俗易懂的语言,全面系统地介绍了C语言程序设计的基本概念、面向过程的编程思想和方法以及解决实际问题的技巧。注重C语言在学科中的基础地位,在整体结构上进行了精心的构思,侧重于对专业发展有影响的内容。在章节结构上做了合理的安排,将概念、知识点与案例相结合,应用实例贯穿于本书始终,尽可能地贴切读者的接受能力,力求将复杂的概念用简洁浅显的语言来描述,做到深入浅出。

本书是集众多长期从事C语言教学工作的一线教师的经验和体会,并参考大量的国内外有关资料编写而成。本书配有《C语言程序设计实验指导与课程设计》辅助教材,给出了详细的主教材习题解答。

本书可作为高等院校各专业计算机程序设计教学用书,以及计算机爱好者自学用书和各类工程技术人员的参考书。

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 13701121993

图书在版编目(CIP)数据

C语言程序设计/郭有强等编著. —北京:清华大学出版社,2009.2
(高等学校计算机教材——面向应用与实践系列)

ISBN 978-7-302-19271-8

I. C… II. 郭… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2009)第004588号

责任编辑:袁勤勇 李玮琪

责任校对:焦丽丽

责任印制:孟凡玉

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:北京鑫海金澳胶印有限公司

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260

印 张:20.25

字 数:567千字

版 次:2009年2月第1版

印 次:2009年2月第1次印刷

印 数:1~4000

定 价:28.00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:031242-01

本书特色

本书根据作者长期从事教学和科研的丰富实践经验编写而成。全书由浅入深地介绍了 C 语言的基本理论、基本知识以及编程的基本技能和方法,使读者能全面、系统地理解和掌握用 C 语言进行程序设计的方法。

本书注重 C 语言在学科中的基础地位,在整体结构上进行了精心的构思,侧重于对专业发展有影响的内容;在章节结构上做了合理的安排,内容全面,概念清晰,层次分明,实例丰富,将概念、知识点与案例相结合;在重难点概念的阐述上,语言清晰,解释到位;讲述力求理论联系实际、循序渐进、深入浅出、通俗易懂,尽可能地贴切读者的接受能力,并注重培养良好的程序设计风格和习惯。

本书注重培养读者分析问题和实际编程能力,给出了两个典型的综合应用。

与本书配套的实践教材为《C 语言程序设计实验指导与课程设计》(ISBN: 978-7-302-19272-5),可以从 <http://computer.bboxy.edu.cn/czdkc/zhongdian.htm> 上下载本书的电子课件、所有例程源代码、习题解答及相关实验和课程设计等内容。

序

1. 关于本书

C语言是一种在世界范围内被普遍采用的优秀的程序设计语言,是现代最流行的通用程序设计语言之一。其功能丰富、使用灵活、可移植性好,既有高级语言的优点,又有低级语言的许多特点;既可以用来编写系统软件,又可用于编写应用软件。因此,C语言被迅速地推广和普及。

C语言作为计算机专业的一门专业基础课程,其专业地位是很重要的,对其掌握的程度如何,直接影响到后继相关课程的学习和掌握。在教学过程中,学习者普遍反映C语言难学难懂,其实并不是C语言本身难,而是教和学的方法不是很得体。首先,教师在教学过程中应整体把握C语言的课程体系,让学生理解相关的结构性概念的作用和意义,在此基础上讲透重点和难点。如在编程语言中为什么会有数组的概念,没有数组能不能编程,数组的本质是什么,在学习数组内容时重难点在哪里;为什么要有指针一章,没有指针可不可以,指针的作用是什么,学习指针的方法应怎样;为什么要有结构体,没有这个概念行不行,如果没有应该怎么做,有什么不方便等;在讲解某个具体的典型程序时,要讲清楚为什么要这样设计,为什么要定义这么多相关的变量,所使用的算法思想及其实现等。总之要有针对性地按照提高其编程能力的目标进行教学。另一方面要掌握好学习方法,先理解相关的概念,再注重对典型程序的学习和掌握,每一个典型程序的学习过程应分为4个阶段:听懂到看懂、理解到熟悉、基本掌握到掌握、完全掌握。如果通过学习能够“完全掌握”书中的十几个典型程序,应该说教学是成功的。鉴于以上理解,结合十多年本课程的教学经验和体会,编写了本教材。

2. 本书结构

本书全面地、系统地讲述了C语言的基础知识、语法规则以及编程方法。全书共分11章。第1章为C语言概述,主要内容包括程序与程序设计语言、算法概述、C语言概况和C程序的开发过程与开发环境。第2章为C程序设计基础,主要内容包括C语言的语法基础、数据与数据类型、基本数据类型、数据的输入与输出、基本运算符和表达式。第3章为程序流程控制,主要内容包括选择结构程序设计、循环结构程序设计、几种控制语句和程序设计举例。第4章为数组,主要内容包括一维数组、二维数组与多维数组、字符串数组和字符串和程序设计举例。第5章为模块设计,主要内容包括模块的实现——函数、函数调用中数据的传递、变量的存储类别与作用域、函数的嵌套调用与递归调用、程序设计举例和编译预处理。第6章为指针,主要内容包括地址与指针的概念、指针变量的定义与引用、指针的运算、指针变量作为函数参数、指针与数组、指针与字符串、指针数组、指针与函数和程序设计举例。第7章为结构体与共用体,主要内容包括结构体类型的定义、结构体类型变量的定义及其初始化、结构体类型变量的引用、结构体数组、指向结构体的指针、结构体与函数、链表、共用体类型、枚举类型、用typedef重定义类型名和程序设计举例。第8章为位运算,主要内容包括几个基本概念、位运算符和位运算、位域(位段)和程序设计举例。第9章为文件,主要内容包括文件指针、文件的打开与关闭、文件的读写、文件的随机读写、文件检测函数和程序设计举例。第10章为综合实训,目的是为了提高读者的综合编程能力。第11章为C++语言概述,主要内容包括面向过程和面向对象编程概述、C++语言的发展过程、将C源程序简单改写成C++源程序、面向过程和面向对象的程序设计方法比较。附录A为“标准ASCII码表”,附录B为“运算符的优先级和结合性”,附录C为“C语言的库函数”。

3. 本书特点

本书采用通俗易懂的语言,全面系统地介绍了 C 语言程序设计的基本概念、结构化编程思想和方法以及解决实际问题的技巧。注重 C 语言在学科中的基础地位,在整体结构上进行了精心的构思,侧重于对专业发展有影响的内容。在章节结构上做了合理的安排,先讲概念再讲方法,从用法中加深对概念的理解,将概念、知识点与案例相结合,应用实例贯穿于本书始终,力求将复杂的概念用简洁浅显的语言来描述,做到深入浅出。通过综合实训,注重实际编程能力的培养提高。此外,还具备以下特点:

(1) 针对性强。本书定位在初学高级语言的读者。通过本书的学习,使读者能够掌握一种计算机中较复杂的高级语言的用法。掌握了 C 语言再学习其他高级语言就容易多了。

(2) 可读性好。本书内容实际上是一线教师十多年讲述 C 语言课程的教案的整理、经验和体会的提炼。语言通俗易懂、重点突出、难点讲述清楚。本书尽可能地贴切学生的接受能力,从学习者的角度去思考每部分内容,对可能产生的疑问进行解释并结合实例进行详细的分析和说明。

(3) 体系结构合理。本书从提高读者的实际编程能力出发,合理安排各章节的内容,突出重点和难点。注重 C 语言对后继课程的基础作用,书中加强了结构体部分的内容,是出于为数据结构和面向对象语言课程提供基础的考虑。书中精选了大量典型例题和习题,突出实践环节,以期读者深刻理解编程思想和增强实际动手编程能力。通过本书的学习,使读者不仅掌握 C 语言的基本语法和规则,还能够体会结构化程序设计的思想和方法。

本书所有的例程都在 Turbo C 2.0 和 Visual C++ 6.0 环境下调试通过。本书配有《C 语言程序设计实验指导与课程设计》教材,其中附有主教材中的习题参考答案,给出了实验内容及课程设计样例等,供学习时参考。本书配有电子教案,并提供例题程序源代码,以方便自学。

4. 适用对象

本书是集众多长期从事 C 语言教学工作的一线教师的经验和体会,并参考大量的国内外有关资料编写而成。其内容丰富、结构紧凑、概念阐述清楚、注重能力培养,是一本内容全面的教材。可作为高等院校各专业计算机程序设计教学用书,以及计算机爱好者自学用书和各类工程技术人员参考书。

本书由郭有强担任主编(编写第 9~11 章),负责总体设计并统稿;姚保峰(编写第 6 章)和戚晓明(编写第 1、8 章)担任副主编,负责本书全部例题源代码的测试和制作电子讲稿。参加编写工作的还有周会萍(编写第 2 章)、刘娟(编写第 3 章)、谢娜(编写第 4 章)、何爱华(编写第 5 章)和王磊(编写第 7 章)。以上人员不仅在本书内容编写、程序测试、文字校对等工作中付出了辛勤劳动,还参与完成了与本书配套的《C 语言程序设计实验指导与课程设计》的内容编写和文字工作。电子讲稿和本书全部例题的源代码一起放在课程网站 <http://computer.bbxy.edu.cn/czdkc/zhongdian.htm> 上,供下载。

本书的编写大纲及内容均经袁兆山教授审阅,在此谨致谢忱。

感谢胡学钢教授和王浩教授,他们对该书给予了极大的关注和支持,提出了宝贵的建设性意见。感谢本书所列参考文献的作者。感谢清华大学出版社各位编辑,他们为本教材的出版倾注了大量的心血和热情,也正是由于他们前瞻性的眼光和忙碌,才让读者有机会看到本教材。

由于作者水平有限,加之时间仓促,错误与疏漏之处在所难免,敬请读者不吝赐正。在使用该书时如遇到问题需要与作者商榷,或需要索取其他相关资料,请与作者联系。联系方式:bbxyguo@163.com。

郭有强

2008 年 12 月

目 录

第 1 章 C 语言概述	1
1.1 程序与程序设计语言	1
1.1.1 程序	1
1.1.2 程序设计语言	2
1.1.3 程序设计	3
1.1.4 结构化程序及设计步骤	4
1.2 算法概述	4
1.2.1 算法的含义	5
1.2.2 算法的特性	6
1.2.3 算法的表示	7
1.3 C 语言概况	8
1.4 C 程序的开发过程与开发环境	11
1.4.1 C 程序的开发过程	11
1.4.2 C 程序的调试	11
1.4.3 Turbo C 开发程序方法	12
1.4.4 Visual C++ 6.0 开发程序方法	13
本章小结	16
习题 1	17
第 2 章 C 程序设计基础	19
2.1 C 语言语法基础	19
2.1.1 基本字符集	19
2.1.2 C 语言词汇	19
2.1.3 语句	21
2.1.4 标准库函数	22
2.2 数据与数据类型	23
2.3 常量	24
2.3.1 数值常量	24
2.3.2 字符常量	26
2.3.3 字符串常量	27
2.3.4 符号常量	27
2.4 变量	28
2.4.1 变量的定义	28
2.4.2 整型变量	29
2.4.3 实型变量	30
2.4.4 字符变量	31

2.5	各类数值型数据间的混合运算及数据类型转换	32
2.6	数据的输入与输出	34
2.6.1	格式输出函数 printf	34
2.6.2	格式输入函数 scanf	38
2.6.3	单字符输入输出函数	41
2.7	基本运算符和表达式	43
2.7.1	运算符、表达式、优先级和结合性	43
2.7.2	算术运算符与算术表达式	45
2.7.3	赋值运算符与赋值表达式	46
2.7.4	自增、自减运算符与表达式	49
2.7.5	关系运算符与关系表达式	50
2.7.6	逻辑运算符与逻辑表达式	51
2.7.7	逗号运算符及逗号表达式	53
2.7.8	条件运算符与条件表达式	54
2.7.9	sizeof 运算符	54
	本章小结	55
	习题 2	56
第 3 章	程序流程控制	59
3.1	顺序结构程序设计	59
3.2	选择结构程序设计	59
3.2.1	if 语句	59
3.2.2	switch 语句	64
3.3	循环结构程序设计	66
3.3.1	while 语句	66
3.3.2	do-while 语句	69
3.3.3	for 循环	69
3.3.4	循环的嵌套	71
3.3.5	几种循环的比较	73
3.4	几种控制语句	73
3.4.1	break 语句	73
3.4.2	continue 语句	74
3.4.3	return 语句	75
3.4.4	goto 语句	75
3.5	程序设计举例	76
	本章小结	81
	习题 3	82
第 4 章	数组	86
4.1	一维数组	86
4.1.1	一维数组的定义、初始化	86

4.1.2 一维数组元素的引用	87
4.2 二维数组与多维数组	90
4.2.1 二维数组定义、初始化	90
4.2.2 二维数组元素的引用	92
4.2.3 多维数组	94
4.3 字符串数组和字符串	95
4.3.1 字符数组的定义与初始化	95
4.3.2 字符串与字符数组	96
4.3.3 字符数组的输入和输出	97
4.3.4 字符串处理函数	99
4.4 程序设计举例	103
本章小结	105
习题4	106
 第5章 模块设计	112
5.1 函数的概念	112
5.2 模块的实现——函数	114
5.2.1 函数的定义	114
5.2.2 函数的调用	118
5.2.3 函数的声明	118
5.3 函数调用中参数的传递	119
5.3.1 值传递方式	120
5.3.2 地址传递方式	121
5.3.3 数组作为函数参数	121
5.4 变量的存储类别与作用域	125
5.4.1 变量的作用域：局部变量和全局变量	125
5.4.2 变量的存储类别	129
5.5 函数的嵌套调用与递归调用	134
5.5.1 函数的嵌套调用	134
5.5.2 函数的递归调用	136
5.6 程序设计举例	137
5.7 编译预处理	139
5.7.1 文件包含	139
5.7.2 条件编译	140
5.7.3 宏定义与宏替换	141
本章小结	145
习题5	145
 第6章 指针	149
6.1 地址与指针的概念	149
6.2 指针变量的定义与引用	151

6.2.1 指针变量的定义	151
6.2.2 指针变量的引用	151
6.3 指针的运算	153
6.4 指针变量作为函数参数	155
6.5 指针与数组	158
6.5.1 指向一维数组的指针变量	158
6.5.2 数组指针作函数参数	161
6.5.3 指向二维数组的指针变量	164
6.5.4 内存的动态分配	167
6.6 指针与字符串	169
6.6.1 字符串的指针表示	169
6.6.2 字符串指针作函数参数	171
6.6.3 字符串指针变量与字符数组的区别	172
6.7 指针数组	173
6.7.1 指针数组的概念	173
6.7.2 带参数的 main 函数	175
6.8 指针与函数	177
6.8.1 指针型函数	177
6.8.2 指向函数的指针变量	178
6.8.3 指向函数的指针变量作函数参数	179
6.9 程序设计举例	181
本章小结	187
习题 6	190
第 7 章 结构体与共用体	196
7.1 结构体类型的定义	196
7.2 结构体类型变量的定义及其初始化	197
7.2.1 结构体变量的定义	197
7.2.2 结构体变量的初始化	199
7.3 结构体类型变量的引用	200
7.4 结构体数组	202
7.4.1 结构体数组的定义与初始化	202
7.4.2 对结构体数组元素的操作	203
7.4.3 结构体数组作为函数参数	204
7.5 指向结构体的指针	206
7.5.1 指向结构体变量的指针	206
7.5.2 指向结构体数组的指针	207
7.6 结构体与函数	209
7.7 链表	212
7.7.1 链表的概念	212
7.7.2 单链表的建立	212

7.7.3 单链表的基本操作	216
7.8 共用体类型	221
7.8.1 共用体的定义与引用	221
7.8.2 共用体与结构体的不同	223
7.9 枚举类型	224
7.10 用 typedef 重定义类型名	226
7.11 程序设计举例	227
本章小结	234
习题 7	234
第 8 章 位运算	241
8.1 几个基本概念	241
8.2 位运算符和位运算	242
8.3 位域(位段)	245
8.4 程序设计举例	246
本章小结	248
习题 8	249
第 9 章 文件	250
9.1 文件及文件指针	250
9.1.1 文件的概念	250
9.1.2 文件指针	251
9.2 文件的打开与关闭	252
9.3 文件的读写	253
9.3.1 字符读写函数 fgetc 和 fputc	253
9.3.2 字符串读写函数 fgets 和 fputs	256
9.3.3 数据块读写函数 fread 和 fwrite	257
9.3.4 格式化读写函数 fscanf 和 fprintf	258
9.4 文件的随机读写	260
9.4.1 文件定位	260
9.4.2 文件的随机读写	261
9.5 文件检测函数	262
9.6 程序设计举例	263
本章小结	265
习题 9	265
第 10 章 综合实训	269
10.1 通讯录管理程序	269
10.1.1 项目要求及分析	269
10.1.2 总体设计	269
10.1.3 代码实现	270

10.1.4	测试结果	277
10.2	学生成绩管理系统	278
10.2.1	项目要求及分析	278
10.2.2	总体设计	278
10.2.3	代码实现	279
10.2.4	测试结果	291
	本章小结	292
第 11 章 C++ 语言概述		293
11.1	面向过程和面向对象编程概述	293
11.2	C++ 语言的发展过程	295
11.3	将 C 源程序简单改写成 C++ 源程序	295
11.4	面向过程和面向对象的程序设计方法比较	296
	本章小结	298
	习题 11	298
附录 A 标准 ASCII 码表		300
附录 B 运算符的优先级和结合性		301
附录 C C 语言的库函数		303
参考文献		308

第 1 章 C 语言概述

学习目标

- (1) 了解程序、程序设计的概念。
- (2) 理解算法的含义,掌握算法设计的基本思想及其表示方法。
- (3) 了解 C 语言的发展历史,掌握 C 语言的特点和结构。

C 语言是广泛流行的高级程序设计语言,它适宜作为系统描述语言。本章主要介绍程序和程序设计、算法、C 语言的发展历史、特点和程序结构。通过本章的学习将对 C 程序开发设计有较深入的感性认识,进而强化对计算机语言和程序的理解,为进一步学习 C 语言程序设计打下基础。

1.1 程序与程序设计语言

1.1.1 程序

尽管当今的计算机系统已具有相当高的水准,但仍采用冯·诺依曼(Von Neumann,1903—1957)的体系结构,即存储程序结构,这说明计算机的执行必须有程序的控制,因此利用计算机解决问题,首先要编写计算机程序。程序是由数据和处理数据的操作组成的。数据是操作的对象,操作的目的是对数据进行加工处理,以得到期望的结果。计算机程序是许多指令的集合,每一条指令让计算机执行完成一个具体的操作,一个程序所规定的操作全部执行完后,就能产生计算结果。

以多媒体软件播放电影为例,程序运行的基本过程如图 1.1 所示。

(1) CPU 发出指令读硬盘或外存中数据(可以在多媒体软件的打开文件对话框中输入电影文件名)。

(2) 从硬盘读数据到内存中,为程序的下一步运行做数据准备(打开文件后,读电影数据的一部分)。

(3) CPU 发出指令处理内存中的数据(多媒体播放软件处理内存中电影数据)。

(4) CPU 计算后的结果放入内存中(处理后的电影画面和音频数据暂存于内存)。

(5) 显示内存中的结果到显示器(即将内存中的电影画面在显示器中显示,音频数据送到音箱输出声音)。

(6) 继续处理内存中的数据(电影数据)。

这一系列步骤由多媒体播放软件控制,多媒体播放软件由一系列程序组成。

计算机系统由软件系统和硬件系统两大部分组成。完成各种不同的任务,需要不同的软件。程序当然属于软件,但程序与软件不是同一个概念,利用计算机解决问题,需要从算法到程序,再到软件。

一个程序应包括如下信息。

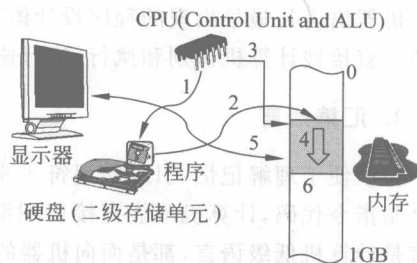


图 1.1 程序的运行过程

(1) 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(Data Structure)。

(2) 对操作的描述,即操作的方法和步骤,也就是算法(Algorithm)。

数据是操作的对象,操作的目的是对数据进行加工处理,以便得到结果。例如,程序设计就像厨师做菜肴时需要的菜谱,通常菜谱包括两个方面。

① 配料:制作菜肴所需的原料(类似于程序设计中的数据结构)。

② 操作的方法和步骤:制作菜肴时将原料按规定的步骤加工成所需的菜肴(类似于程序设计中的算法)。

因此,著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式:

$$\text{数据结构} + \text{算法} = \text{程序}$$

通常认为:

$$\text{程序} = \text{算法} + \text{数据结构} + \text{程序设计方法} + \text{语言工具和环境}$$

由此可见,编写程序是让计算机解决实际问题的关键。一般编制计算机程序必须具备两个基本条件:一是掌握一门计算机语言;二是要掌握解题的方法和步骤。

1.1.2 程序设计语言

编写程序所使用的语言称为程序设计语言,它是人与计算机之间进行信息交流的工具。从1946年第一台计算机ENIAC诞生至今,计算机技术突飞猛进地发展,程序设计语言的发展从低级到高级经历了机器语言、汇编语言、面向过程和面向对象语言4个阶段。

1. 机器语言

机器语言是最早出现的程序设计语言,是用0和1组成的二进制指令(机器指令)集合。程序指令直接被计算机识别和执行,运行速度快,但指令难于理解和记忆。

2. 汇编语言

为了便于理解记忆,引入助记符号来表示机器指令(如ADD来表示加法)。助记符替代了二进制指令代码,计算机不能直接识别汇编语言,需要编译后才能识别运行。机器语言和汇编语言是计算机低级语言,都是面向机器的语言。

3. 高级语言

高级语言采用接近自然语言的命令或语句进行编程。高级语言独立于机器,编程不需要了解计算机指令系统和硬件结构,只需要掌握解决问题的方法和高级语言的语法规则。20世纪50年代末和60年代初研制出的高级程序设计语言BASIC、FORTRAN、AL-GOL60和COBOL号称计算机语言发展史上的4个里程碑。

高级语言的运行方式有3种:解释执行、编译执行、先编译后解释。第一种是通过语言解释程序将源程序解释翻译一句,执行一句,如BASIC、JavaScript、VBScript、PHP、ASP、Perl和Python等语言,这些语言的共同特点是运行速度慢,但简单。第二种是通过编译程序将源程序编译成二进制目标代码程序,最后执行的是目标代码程序,如C语言。显然,后者程序执行效率高、速度快,并且能够脱离语言环境独立运行。第三种是一种新型的语言,是先编译后解释执行的,如Java和C#语言。

4. 面向对象语言

面向对象语言比面向过程语言更接近于自然语言。程序员设计程序时,基本避开“怎么做”的解题过程,而是更多地考虑计算机“做什么”,计算机就会完成具体的解题过程。面向对象语言一般是在面向过程的计算机语言上发展起来的,如C++是由C语言发展而来的。

程序设计技术是随着编程语言的发展而发展的,编程语言的发展趋于相互同化,往往彼此相似,所以熟练掌握一种语言后,再学其他语言就不太难了。

1.1.3 程序设计

计算机毕竟不是人,要计算机干什么事情,人们都要事先替它想好解决问题的方法(叫做算法),再用计算机能理解的语言(计算机语言或程序设计语言)详细描述出来(这个描述过程即程序设计)。在计算机科学中,把解决问题的方法、步骤描述出来,这个描述的过程就是程序设计。

程序设计与软件设计不是一个概念,它们的侧重点不同。如果说软件设计是工程项目设计,则程序设计是工程中核心部分的实施过程。从另一个角度看,程序设计与软件设计之间又难以找出明显的分界线,交叉地带有很多共同的问题。整个程序设计过程为软件开发的一部分。程序设计方法是程序设计中要关心的,当今典型的结构化程序设计方法有:自顶向下、逐步细化、模块化设计、结构化编码。

下面介绍设计简单程序时的步骤,如果进行系统工程开发,则操作步骤比较复杂。对于程序设计人员来说,必须掌握更多的知识,例如软件工程方面的知识等。程序设计的一般步骤如下。

1. 确定数据结构

依据所需要处理的任务要求,规划输入的数据和输出的结果,确定存放数据的数据结构。由于在C语言中数据结构集中体现在数据类型上,因此,在进行C语言程序设计时,应统筹规划程序中所使用的变量、数组、指针以及它们的类型等。这是很重要的,如果在此期间选择了不合适的变量或者数组,将来修改起来就会比较困难。

2. 确定算法

算法是指为解决某一特定问题而采取的确定的有限的步骤。对同一个问题,每一个人确定的算法都不应该完全相同。算法有优有劣,其评判的标准本书不做介绍。对于程序设计人员来说,应该学习比较优秀和比较经典的算法,本书实例中介绍的均为一些常用的算法。

3. 编写程序

在充分论证数据结构和算法以后才能考虑编写程序,编写程序需要结合程序设计方法(面向过程的或是面向对象的)和程序设计语言(C语言、C++语言和Delphi等),当然即使是C语言,还有许多集成开发环境,例如Turbo C或是Microsoft C等,不同的集成开发环境写出的程序代码是有区别的。

4. 程序调试

程序开发人员编写的程序称为源程序或源代码,源代码不能直接被计算机执行。源代码要经过编译程序编译,生成目标程序,然后链接其他相应的代码,最后生成可被计算机执行的可执

行文件(EXE 或 COM 文件)。一个源代码有时要经过多次的修改才能编译通过,因此这一步有时是很困难的。程序在编译时,如果不能通过,则会有错误提示信息,程序员要根据错误提示信息调试程序。

5. 整理源程序并总结资料

有人以为程序调试成功就万事大吉了,其实这种思想是错误的。对于程序设计人员来说,平时的归纳和总结是很重要的。程序员应将平时的源程序进行归类保存,以方便今后查找,同时一定要注意保留文字资料。

1.1.4 结构化程序及设计步骤

结构化程序的概念首先是从以往编程过程中无限制地使用转移语句而提出的。转移语句可以使程序的控制流程强制性地转向程序的任一处,导致程序流程无序可寻,程序结构杂乱无章,这样的程序是令人难以理解和接受的,并且容易出错。为此提出了结构化程序设计的 3 种基本结构。

1. 顺序结构

顺序结构表示程序中的各操作是按照它们出现的先后顺序执行的。这种结构的特点:程序从入口点开始,按顺序执行所有操作,直到出口点处,所以称为顺序结构。顺序结构是一种线性结构,在该结构中,各操作块(简称块,它所对应的计算机语言表述形式是程序段)按照各自出现的先后顺序,依次逐块执行。它是计算机科学用以描述客观世界顺序现象的重要手段,是从简单到复杂的程序的主体基本结构,因为选择结构与循环结构的组成,也离不开顺序结构(即作为它们的子结构)。

2. 选择结构

选择结构表示程序的处理步骤出现了分支,它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择 3 种形式。

3. 循环结构

循环结构表示程序反复执行某个或某些操作,直到某条件为假(或为真)时才可终止循环。在循环结构中最主要的是:什么情况下执行循环,哪些操作需要循环执行?循环结构的基本形式有两种:当型循环和直到型循环。循环结构是一种常见的重要基本结构,是计算机科学中用以描述客观世界广泛存在的循环现象的重要手段,用它可以避免重复不必要的操作,简化程序,节约内存,提高效率。

通过三种基本控制结构可以看到,结构化程序中的任意基本结构都具有唯一入口和唯一出口,并且程序不会出现死循环。在程序的静态形式与动态执行流程之间具有良好的对应关系。理论上已经证明,用三种基本程序结构可以实现任何复杂的算法。

1.2 算法概述

如果说程序设计是把解决问题的方法步骤描述出来的描述过程,那么,程序设计首先要涉及对算法的研究。算法的内容涉及面很广,在这里仅作概念性的介绍。

1.2.1 算法的含义

算法(Algorithm)是一项工作如何被完成的步骤,即算法是指解决问题的方法和步骤。

在生活中,人们到银行取钱的过程就是一个算法,其步骤如下:

- (1) 带上存折去银行。
- (2) 填写取款单并到相应窗口排队。
- (3) 将存折和取款单递给银行职员。
- (4) 银行职员办理取款事宜。
- (5) 拿到钱并离开银行。

在计算机领域,算法就是用计算机解决数值计算或非数值计算问题的方法。著名的图灵理论指出,只要能被分解为有限步骤的问题就可以被计算机执行。这条理论定义了算法一是有有限的步骤,二是能够将这些步骤设计为计算机所执行的程序。

【例 1-1】 设有两个杯子 A 和 B,分别盛放酒和醋,要求将它们互换。

S1: $C \leftarrow A$

S2: $A \leftarrow B$

S3: $B \leftarrow C$

【例 1-2】 从 10 个数中找出最大数,存入 max 中。

S1: $i=1$,令 max 等于第一个数;

S2: $i=i+1$;

S3: 将 max 与第 i 个数进行比较,若前者小于后者,则修正 max 为第 i 个数;否则进行 S4;

S4: 若 $i < 10$,则转到 S2;否则输出 max 并结束。

【例 1-3】 求 1~100 的和。

S1: $\text{sum} \leftarrow 0, t \leftarrow 1$;

S2: $\text{sum} \leftarrow \text{sum} + t$;

S3: $t \leftarrow t + 1$;

S4: 若 $t \leq 100$,则转到 S2;否则转到 S5;

S5: 输出 sum,结束。

【例 1-4】 求 $n!$ 。

S1: $s \leftarrow 1, t \leftarrow 1$;

S2: $s \leftarrow s * t$;

S3: $t \leftarrow t + 1$;

S4: 若 $t \leq n$,则返 S2;否则输出 s 并结束。

1. 衡量算法步骤优劣的标准

一个算法步骤是否优劣的标准主要有以下 3 个方面。

(1) 思路:清晰、正确。

(2) 过程:简单、明了、扼要。

(3) 算法:合适。

【例 1-5】 求 $1+2+3+\cdots+100$ 这个问题可以设计以下两种算法。

算法 1:

$1+2=3, 3+3=6, 6+4=10, \dots, 4851+99=4950, 4950+100=5050$