

21世纪普通高校计算机公共课程规划教材

C语言程序设计

韦娜 王俊 袁玲 吴文红 等 编著



清华大学出版社

21 世纪普通高校计算机公共课程规划教材

C语言程序设计

韦娜 王俊 袁玲 吴文红 等 编著

清华大学出版社
北京

内 容 简 介

本书以培养学生程序设计基本能力为目标,较系统地介绍了C语言程序设计的基本知识和程序设计的基本方法。全书共12章,主要内容包括程序设计概述、数据类型、运算符与表达式、顺序结构、选择结构及循环结构程序设计、数组、函数与编译预处理、指针、构造数据类型、文件、综合实例。全书结构清晰,实例丰富,语言流畅,通俗易懂。

本书可作为高校本科、高职高专理工类专业“C语言程序设计”课程的教材,也可作为C语言程序设计的自学用书。

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/韦娜等编著.--北京:清华大学出版社,2016

21世纪普通高校计算机公共课程规划教材

ISBN 978-7-302-42751-3

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

中国版本图书馆CIP数据核字(2016)第020165号

责任编辑:郑寅堃 王冰飞

封面设计:何凤霞

责任校对:时翠兰

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 彩 插:1 字 数:478千字

版 次:2016年3月第1版 印 次:2016年3月第1次印刷

印 数:1~3500

定 价:39.50元

产品编号:067759-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

C 语言以其独特的魅力吸引了众多的软件开发者,是程序设计的一种基本语言,是掌握各种现代编程技术的重要基础。它兼有高级语言和低级语言二者之长,代码简洁高效,功能强大。“C 语言程序设计”课程是大学计算机基础教育的核心课程,是学生计算思维能力培养的重要载体。

为了提高教学质量,贯彻和实施以计算思维为切入点的教学改革,近年来,我们首先开展了计算机基础课程考试改革工作,实现了“C 语言程序设计”课程上机考试。借助考试系统,把期末考核与阶段性测试相结合,以考带练,以练促考;使“教与学”、“学与练”更好地结合起来。实践证明,考试改革既增强了学生应用能力的发展,又培养了学生自主学习能力,使学生在编程中学习知识,在学习过程中拓展思维。随着考试改革的深入,为适应新的教学需求,我们编写了这本新的教材,教材内容更强调“面向问题求解”的思维方法训练。本书的主要特点如下:

(1) 例题经典,分析透彻。选择典型例题,注重算法设计,强化程序设计能力的培养。帮助学生在应用中加深对 C 语言基本语法和程序设计方法的理解。对例题的重点、难点在例题“分析”、“说明”、“注意”中列出。

(2) 设置引例,承前启后。每章设置引例,学习新内容时,引例起承接过渡的作用。通过简单实用的引例使学生对新知识有感性的认识,引例引出语法,语法引入应用。把面向语法为中心的程序设计教学转变为面向问题求解的程序设计教学,突出程序设计思想与方法。

(3) 综合实例,贯穿全篇。以菜单驱动的“学生成绩管理系统”为主线,主要章节的实例部分设置例题围绕着第 12 章综合实例逐步展开。引导学生逐步编写菜单驱动的学生成绩管理系统,提高系统思维能力。该内容可以作为课程设计的参考内容。

(4) 注重实用性与趣味性。特别选择了实用有趣的题目,如鸡兔同笼、猜数游戏、小学生四则运算练习、建立计时器等,引导学生体会利用计算机解决问题的思路和方法,注重培养学生的计算思维和编程兴趣。

为了更好地满足学习者需要,本书编有配套教材《C 语言程序设计学习指导》,辅导学生巩固教材所学,加强上机实践能力。作者为选用本书的老师提供了配套的课件和例题源程序等教学资源,可从清华大学出版社网站(<http://www.tup.com.cn>)下载使用。

本书是在教材《C 语言程序设计》(武雅丽等编著)多年教学实践的基础上,对原书内容进行梳理、精简和充实而编写的。作者都是多年从事 C 语言程序设计教学的高校教师。本书各章节编写分工如下:第 1 章、第 2 章由吕进编写;第 3 章、第 10 章由袁玲编写;第 4

章、第 5 章、第 12 章由韦娜编写；第 6 章、7.1~7.7 节由吴文红编写；第 8 章、第 9 章由王俊编写；第 7.8 节及第 11 章由卢江编写。全书由韦娜、王俊负责统稿。

清华大学出版社对本书的策划、出版做了大量工作，在此表示衷心感谢！

由于编者水平所限，书中难免有错误及疏漏之处，恳请广大读者批评指正。

作 者

2015 年 10 月于西安

目 录

第 1 章 程序设计概述	1
1.1 计算机语言与程序设计	2
1.1.1 计算机语言	2
1.1.2 程序与计算机程序设计	3
1.2 算法	4
1.2.1 算法的定义与特性	4
1.2.2 算法的表示	5
1.3 结构化程序设计方法	9
1.4 C 语言的历史与特点	10
1.4.1 C 语言的产生和发展	10
1.4.2 C 语言的特点	10
1.5 C 语言程序结构	11
1.6 C 语言程序的执行步骤	12
习题	13
第 2 章 数据类型与表达式	15
2.1 字符集、关键字和标识符	16
2.1.1 字符集	16
2.1.2 关键字	16
2.1.3 标识符	16
2.2 C 语言的数据类型及其存储形式	17
2.2.1 C 语言的数据类型分类	17
2.2.2 整型数据	18
2.2.3 实型数据	19
2.2.4 字符型数据	21
2.3 常量与变量	21
2.3.1 常量	22
2.3.2 变量	24
2.4 运算符与表达式	25
2.4.1 C 语言运算符与表达式概述	26

2.4.2	算术运算	26
2.4.3	赋值运算	28
2.4.4	关系运算	29
2.4.5	逻辑运算	30
2.4.6	逗号表达式	31
2.4.7	求字节数运算符	32
2.4.8	圆括号运算符	32
2.4.9	位运算	32
2.5	类型转换	34
2.5.1	自动类型转换	34
2.5.2	强制类型转换	36
习题		36
第 3 章	顺序结构程序设计	38
3.1	C 语言基本语句	39
3.2	数据的输入/输出	40
3.2.1	数据输入/输出的概念	40
3.2.2	格式化的输入/输出	41
3.2.3	字符的输入/输出	51
3.3	顺序结构程序设计	52
3.3.1	顺序结构程序	52
3.3.2	良好的源程序书写风格	53
3.4	顺序结构程序设计实例	55
习题		57
第 4 章	选择结构程序设计	58
4.1	if 语句及嵌套	59
4.1.1	if 语句	59
4.1.2	if 语句的嵌套	61
4.2	条件运算符和条件表达式	66
4.3	switch 语句	68
4.4	选择结构程序设计实例	71
习题		75
第 5 章	循环结构程序设计	77
5.1	while 语句	78
5.2	do...while 语句	81
5.3	for 语句	83
5.4	循环的嵌套	87

5.5	循环语句的特点	89
5.6	与循环有关的控制语句	89
5.6.1	break 语句	89
5.6.2	continue 语句	91
5.7	循环结构程序设计实例	92
	习题	98
第 6 章	数组	100
6.1	数组的基本概念	101
6.2	一维数组	102
6.2.1	一维数组的定义	102
6.2.2	一维数组元素的引用	103
6.2.3	一维数组的初始化	104
6.2.4	一维数组程序举例	105
6.3	二维数组	111
6.3.1	二维数组的定义	111
6.3.2	二维数组元素的引用	112
6.3.3	二维数组的初始化	112
6.3.4	二维数组程序举例	113
6.4	字符数组	115
6.4.1	一维字符数组的定义与初始化	115
6.4.2	一维字符数组的输入/输出	116
6.4.3	二维字符数组	119
6.4.4	字符串处理函数	120
6.4.5	字符数组程序举例	123
6.5	数组程序设计实例	124
	习题	128
第 7 章	函数与编译预处理	129
7.1	程序与函数	130
7.2	函数的定义及调用	131
7.2.1	定义无参函数	131
7.2.2	定义有参函数	131
7.2.3	函数的调用	132
7.3	函数参数及其传递方式	137
7.3.1	变量做函数参数	137
7.3.2	数组做函数参数	139
7.4	函数的嵌套调用	142
7.5	函数的递归调用	143

7.6 变量的作用域及存储类型	146
7.6.1 局部变量	146
7.6.2 全局变量	147
7.6.3 变量的存储方式和生存期	149
7.6.4 小结	152
7.7 函数程序设计实例	153
7.8 编译预处理	158
7.8.1 宏定义	158
7.8.2 文件包含	162
7.8.3 条件编译	163
习题	165

第8章 指针

166

8.1 地址与指针	167
8.1.1 地址的概念	167
8.1.2 指针的概念	168
8.2 指针变量	168
8.2.1 指针变量的定义	168
8.2.2 指针的基本运算	169
8.2.3 指针的初始化及赋值	170
8.2.4 零指针与 void * 类型指针	173
8.3 指针与数组	173
8.3.1 指针与一维数组	174
8.3.2 指针的运算	176
8.3.3 指针与二维数组	179
8.4 指针与字符串	182
8.5 指针数组与指向指针的指针	186
8.5.1 指针数组	186
8.5.2 指向指针的指针	188
8.6 指针与动态内存管理	190
8.6.1 动态内存管理函数	190
8.6.2 动态内存管理的应用	191
8.7 指针程序设计实例	192
习题	195

第9章 指针与函数

196

9.1 指针作为函数的参数	197
9.1.1 指针变量作为函数的参数	197
9.1.2 数组作为函数的参数	199

9.1.3	字符指针变量作为函数的参数	201
9.1.4	指向数组的指针变量作为函数的参数	202
9.1.5	指针数组或二级指针变量作为函数的参数	204
9.2	函数返回指针值	205
9.3	指向函数的指针	206
9.4	带参 main() 函数	209
9.5	指针小结	211
9.5.1	关于指针的数据类型小结	211
9.5.2	指针运算小结	212
9.6	指针与函数程序设计实例	213
	习题	218
第 10 章	构造数据类型	219
10.1	结构体类型	220
10.2	结构体变量	221
10.2.1	结构体变量的定义	221
10.2.2	结构体变量的初始化	223
10.2.3	访问结构体变量成员	224
10.2.4	结构体的嵌套	225
10.2.5	结构体变量与函数	226
10.3	结构体数组	227
10.3.1	结构体数组的定义	227
10.3.2	结构体数组的初始化与引用	227
10.3.3	结构体数组作为函数参数	229
10.4	结构体类型的指针	231
10.4.1	指向结构体变量的指针	231
10.4.2	指向结构体数组的指针	233
10.4.3	结构体类型的指针作为函数参数	234
10.5	链表	236
10.5.1	单链表概述	236
10.5.2	链表的基本操作	237
10.6	共用体	244
10.6.1	共用体类型	244
10.6.2	共用体变量	245
10.6.3	访问共用体变量成员	246
10.7	枚举	247
10.7.1	枚举类型	247
10.7.2	枚举变量	248
10.8	用 typedef 定义类型名	249

10.9 程序设计实例	250
习题	254
第 11 章 文件	256
11.1 文件概述	257
11.1.1 文件的概念	257
11.1.2 缓冲文件系统	258
11.1.3 文件类型指针	258
11.2 文件的打开与关闭	259
11.2.1 文件的打开	259
11.2.2 文件的关闭	260
11.3 文件的顺序读/写操作	260
11.3.1 读/写字符函数	261
11.3.2 读/写字符串函数	263
11.3.3 格式化读/写数据函数	264
11.3.4 读/写数据块函数	266
11.4 文件的随机读/写操作	267
11.4.1 文件指针复位函数	268
11.4.2 文件指针随机移动函数	268
11.4.3 取文件指针当前位置函数	270
11.5 文件出错检测函数	271
11.5.1 ferror()函数	271
11.5.2 clearerr()函数	271
11.6 文件程序设计实例	271
习题	273
第 12 章 综合实例	274
12.1 应用程序开发流程	274
12.2 学生成绩管理系统开发实例	275
12.2.1 功能描述	275
12.2.2 系统设计	275
附录 A 常用字符及其 ASCII 代码	289
附录 B C 语言的运算符及其结合性	292
附录 C C 语言库函数	294
参考文献	300

第 1 章

程序设计概述

计算机技术的发展日新月异,计算机程序设计语言也层出不穷,那么什么是计算机语言?什么是程序设计?如何进行程序设计?本书将以 C 语言为实现工具介绍程序设计的基本思想和方法。作为全书的开篇,本章介绍程序设计的基本知识、C 语言的发展、C 程序的基本结构和执行步骤等。通过本章的学习,使读者对程序设计和 C 语言有一个概要的认识,为以后各章的学习打下基础。

【引例 裁纸奔月】

地球离月球的距离是 386 000km,如果将一张纸裁成两等份,把裁好的两张纸摞起来,再裁成两等份,如此重复下去,多少次后纸的高度就是地球离月球的距离?一张纸的厚度是 0.006cm。

分析:把厚度为 0.006cm 的纸不断地裁剪、层叠,经过若干次操作后,把摞好的纸当作梯子,你就可以沿着纸梯子直奔月球。

利用 C 语言提供的循环语句编程,可以非常轻松地求出裁纸的次数和最终纸的厚度。

程序:

```
#include<stdio.h>
int main(void)
{
    double paper = 0.006;
    int n = 0;
    /* 纸厚小于地球与月球的距离时重复执行 */
    while(paper < 386000000000.0)
    {
        paper = paper * 2;           //摞起来,即纸的厚度 * 2
        n = n + 1;                   //记录摞纸的次数
    }
    printf("the height of paper:%.2f\n", paper);
    printf("times of cutting:%d\n", n);
    return 0;
}
```

程序运行结果:

```
the height of paper: 52776558133.25
times of cutting: 43
```

说明:这就是说,只要裁上 43 次,纸的高度足以让我们沿着这个纸梯子直奔月球。这个例子展示的就是用计算机解决问题的编程方法。

1.1 计算机语言与程序设计

在学习 C 语言程序设计之前需要了解一些程序设计的基本概念,包括计算机语言、程序、程序设计的过程等。

1.1.1 计算机语言

什么是计算机语言?为什么要使用计算机语言?人和人交流用的是双方都能听懂和读懂的自然语言,同样,人和计算机交流也要用人和计算机都容易接受和理解的语言,这就是计算机语言。人用自然语言讲述和书写,目的是给其他人传播信息。同样,人使用计算机语言把人的意图表达给计算机,目的是使用计算机。

计算机语言的发展经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言与汇编语言

机器语言是用二进制代码表示的能被计算机识别和执行的指令集合。机器语言的每一条语句实际上是一条指令。例如计算 $15+4$ 的机器语言程序如下:

10110000 00001111	把 15 送到寄存器 AL 中
00000100 00000100	4 与寄存器 AL 的内容相加,结果仍放入 AL 中
11110100	结束

显然,用机器语言编写的程序难写、难记忆、难阅读。由于不同的计算机指令系统不同,因此机器语言的通用性差。机器语言的优点是编写的代码不用翻译,直接执行,因此占用的存储空间小,执行速度快。

为了克服机器语言的缺点,人们用助记符代替机器指令,于是产生了汇编语言。 $15+4$ 的汇编语言程序如下:

MOV AL, 15	往寄存器 AL 送 15
ADD AL, 4	寄存器 AL 加 15,且送回到 AL 中
HLT	结束

由此可见,汇编语言在一定程度上克服了机器语言难读、难写的缺点,同时保持了占用存储空间少、执行效率高的优点。

机器语言和汇编语言都是面向机器的语言,称为低级语言。

2. 高级语言

因为低级语言是难以普及应用的,因此产生了高级语言,它是用更接近人的自然语言和数学表达式的一种语言,由表达不同意义的“关键字”和“表达式”按照一定的语法、语义规则组成,完全不依赖机器的指令系统。这样的高级语言为人们提供了很大的方便,编制出来的程序易读、易记,也便于修改、调试,极大地提高了编制程序的效率,也大大提高了程序的通用性,便于推广、交流,从而极大地推动了计算机的普及应用。

高级语言离人们的理解愈加接近了,但离计算机的理解愈远了。计算机是不能直接理解那些英语单词、数学表达式的,因此需要求助于翻译程序。翻译程序通常有两种工作方式,即编译方式和解释方式。

编译方式：事先编好一个称为编译程序的机器指令程序并放在计算机中，把用高级语言编写的源程序输入计算机，编译程序便把源程序整个地翻译成用机器指令表示的目标程序。然后执行该目标程序，得到计算结果。

解释方式：事先编好一个称为解释程序的机器指令程序并放在计算机中，把用高级语言编写的源程序输入计算机，它并不是像编译方式那样把源程序整个地翻译成用机器指令表示的目标程序，而是逐句翻译，译出一句立即执行一句，即边解释边执行。这种方式比编译方式多费机器时间，但可少占计算机的内存。

高级语言的种类繁多，目前应用广泛的主要有 FORTRAN、C/C++、Visual Basic、Java 语言和 C# 语言等。

(1) FORTRAN 语言：1954 年推出，它是世界上最早的高级语言，主要用于科学计算。

(2) C/C++ 语言：1972 年推出的 C 语言功能丰富、使用灵活、可移植性强。1983 年，贝尔实验室的 Bjarne Stroustrup 在 C 语言的基础上推出了 C++。C++ 语言进一步扩充和完善了 C 语言，是一种面向对象的程序设计语言，为程序员提供了一种新的编程方式。C/C++ 语言适合各类应用程序的开发，是系统软件的主流开发语言。

(3) Visual Basic 语言：1991 年 Microsoft 公司推出了可视化的、基于对象的 Visual Basic。VB 功能强大、学习简单，给非计算机专业的广大用户开发应用程序带来了便利。

(4) Java 语言：1995 年推出，它是一种跨平台的面向对象程序设计语言，主要用于 Internet 应用开发。Java 语言现已成为移动计算和云计算环境下的主流开发语言。

(5) C# 语言：2000 年推出，它是一种易用的、安全的面向对象程序设计语言，专门为 .NET 应用而设计。它吸收了 C++、Visual Basic、Java 等语言的优点，体现了最新程序设计技术的精华。

1.1.2 程序与计算机程序设计

什么是程序设计？大家在日常生活中可以看到，同一台计算机有时可以画图，有时可以制表，有时可以玩游戏……使用不同的程序，计算机就可以处理不同的问题。在现代社会，计算机已经进入社会生活的各个领域，这正是因为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，也正是这些功能丰富的程序给了计算机无尽的生命力。所谓程序，就是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述，这个描述的过程就称为程序设计。对数据的描述就是指明数据结构形式；对处理方法和步骤的描述也就是 1.2 节要讨论的算法问题，因而数据结构与算法是程序设计过程中密切相关的两个方面。图灵奖的获得者瑞士著名计算机科学家沃思 (Niklaus Wirth) 教授关于程序提出了著名公式：

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

实际上，一个程序除了以上两个要素外，还应当采用结构化程序设计方法进行程序设计，并且用某一种计算机语言表示，因此可以表示为：

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

由此可见，一个完整的程序要涉及 4 方面的问题，即数据结构、算法、编程语言和程序设计方法。这 4 个方面的知识都是程序设计人员所必须具备的。

一个简单的程序设计一般包含以下 4 个步骤。

(1) 分析问题,建立数学模型:在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,解决问题的步骤又是什么。

(2) 确定数据结构和算法:根据建立的数学模型对指定的输入数据和预期的输出结果确定存放数据的数据结构,针对所建立的数学模型和确定的数据结构选择合适的算法加以实现。

(3) 编制程序:根据确定的数据结构和算法用自己所使用的程序语言把这个解决方案严格地描述出来,也就是编写出程序代码。

(4) 调试程序:在计算机上用实际的输入数据对编好的程序进行调试,分析所得到的运行结果,进行程序的测试和调整,直到获得预期的结果。

1.2 算 法

算法是问题求解的方法、思路 and 过程,每一个具体问题都有多种求解方法。尽管人们的思维模式不同,分析和解决问题的方法不同,但算法设计仍然有一些基本规律可循。

1.2.1 算法的定义与特性

广义地说,算法是为解决问题而采取的确定的方法和步骤。在程序设计中,算法是一系列解决问题的清晰指令。

数据结构是指相互间存在一种或多种特定关系的数据元素集合,数据结构应该包括数据元素集合及元素间关系的集合,即数据的组织形式。数据结构与算法有着密切的关系,只有明确了问题的算法才能更好地构造数据结构,但选择好的算法常常依赖于好的数据结构,程序是算法的实现形式,因此编写一个程序的关键就是合理地组织数据和设计好的算法。

先看一个简单的实例。

【例 1.1】 输入 3 个数,求其最大数。

分析: 用自然语言描述的算法如下。

设 num1、num2、num3 存放 3 个数,max 存放其最大值。为求其最大数,只要对 3 个数进行比较,其步骤如下:

(1) 输入 3 个数 num1、num2、num3。

(2) 先把第 1 个数 num1 的值赋给 max。

(3) 将第 2 个数 num2 与 max 比较,如果 $\text{num2} > \text{max}$,则把第 2 个数 num2 的值赋给 max,否则不执行任何操作。

(4) 将第 3 个数 num3 与 max 比较,如果 $\text{num3} > \text{max}$,则把第 3 个数 num3 的值赋给 max,否则不执行任何操作。

(5) 输出 max 的值,即最大值。

一个算法应具有以下 5 个特性。

(1) 有穷性:一个算法应包含有限个操作步骤,而不能是无限的。也就是说,对于一个算法,要求其在时间和空间上是有穷的。例如,例 1.1 中求 3 个数中最大数的算法,经过两次比较就能得出结论,所以具有“有穷性”。

(2) 确定性:算法中的每一步都应当是确定的,而不应当是含糊的、模棱两可的,也就是要求必须有明确的含义,不允许存在二义性。例如,“将成绩优秀的同学名单打印输出”,在