

21世纪高等学校规划教材 | 计算机科学与技术

山东省精品课程配套教材

C语言程序设计

——理论、方法与实践

张 磊 主编
郑喜珍 李竹健 刘海慧 副主编



清华大学出版社

013058179

21世纪高等学校规划教材 | 计

TP312C-43

822



C语言程序设计

——理论、方法与实践

张磊 主编
郑喜珍 李竹健 刘海慧 副主编

TP312C-43
822



北航

C1668811

清华大学出版社
北京

内 容 简 介

本书面向程序设计初学者编写,以“学习基本理论、掌握基本方法、培养基本实践能力”为编写指导思想,突出C语言课程的应用性、实践性特点,综合运用案例教学、任务驱动、启发式教学等多种教学方法,对C语言程序设计的语言知识和程序设计的方法过程进行系统介绍,特别适合将C语言程序设计作为第一门程序设计课程的高校学生学习使用。

全书共有9章,分别为程序设计概述、简单程序设计、分支结构程序设计、循环结构程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计和文件程序设计。每章均设有实验指导,分为基础实验和综合实验两部分,与理论教学内容密切衔接,相辅相成。本书配有教学课件、例题程序源代码和试题库等丰富的教学资源。

本书适合作为高等院校C语言程序设计等相关课程的教材,也可用作程序设计人员及程序设计爱好者的自学参考书。

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 13701121933

图书在版编目(CIP)数据

C语言程序设计:理论、方法与实践/张磊主编.--北京:清华大学出版社,2013.8

21世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-32320-4

I. ①C… II. ①张… 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(2013)第092458号

责任编辑:付弘宇 薛 阳

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:刘海龙

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 刷 者:北京富博印刷有限公司

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:185mm×260mm

印 张:21.75

字 数:530千字

版 次:2013年8月第1版

印 次:2013年8月第1次印刷

印 数:1~2000

定 价:36.00元

产品编号:053661-01

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。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

前言

本书是在《C 语言程序设计(第 3 版)》(ISBN 978-7-302-27323-3)的基础上编写的,主要在以下两方面做了改进。

第一,程序结构标准化。在这之前的版本中,程序主函数使用非标准的“void main()”结构,而本书使用标准的“int main(void)”结构。这其实是在“实用”和“标准”之间做出的选择。众所周知,C 语言标准将无参数的 main()函数定义为如下形式:

```
int main(void)
{
    函数体
    return 0;
}
```

以下是一个公认的经典的简单 C 语言程序:

```
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

一般而言,按照标准 C 教学的课程,会在教学的第一时间展示这个程序。同学们惊叹之际,往往会发问:“return 0”是什么?

在 C 语言教学研讨会中,老师们对“return 0”感慨甚多,甚至认为,这个小辫子简直是 C 语言教学之初的麻烦制造者。事实确实如此!面对程序设计初学者,“return 0 是什么?”并不是一个容易回答的问题。

幸运的是,有不少 C 语言编译平台支持不使用“return 0”的 main()函数结构,例如,应用极其广泛的 Microsoft Visual C++6.0 平台即对如下结构提供完全支持。

```
void main()
{
    printf("Hello world!\n");
}
```

显然,这种结构更适合程序设计初学者,它割掉了“return 0”这个小辫子,不需要回答“return 0 是什么?”。

尽管“void main()”结构与标准 C 的“int main(void)”结构有所不同,但它易于初学者学习,而且不会影响 C 语言基本知识和技能的教學。因此,不少 C 语言教材中选用“void main()”结构。

本书程序结构的转变源于作者与一位小有成就的学生的谈话,他谈到技术标准对他事业的影响时感慨万千,这促使本书程序结构实现了由“实用”到“标准”的转变。

第二,教材内容调整优化。一是对设计性较强的重点实例加强了算法设计、关键语句、关键程序段以及程序执行过程的分析讲解,以便更具学习参考性;二是对算法不够典型、梯度过大、前后衔接不强的例题进行了剔除更新,例题设置更加科学合理;三是对实用性不强的理论知识进行了删减,适当充实了实用性、启发性教学内容,突出了课程的应用性特点。

全书共有9章,分别为程序设计概述、简单程序设计、分支结构程序设计、循环结构程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计和文件程序设计。每章之后均设有实验指导内容。

本书主要有以下特点。

(1) 面向程序设计初学者编写,以程序设计为主线,突出主干知识教学,突出C语言课程的应用性、实践性特点,注重基本应用能力的培养,特别适合将C语言作为第一门程序设计课程的高校学生。

(2) 以实例引领为教学内容,符合认识规律。凡是适合以程序实例开始的新知识,均通过程序实例和程序说明予以引导,首先建立感性认识,然后展开相关内容,进行知识的系统介绍。

(3) 实例丰富,讲解重点突出。本书有大量设计性实例,以培养学生程序设计能力。每个实例均按照问题分析与算法设计、程序实现、程序说明及进一步讨论等内容进行系统讲解,重点加强算法设计、关键语句、关键程序段以及程序讨论的分析说明,力求重点难点讲解透彻,启发性内容留有余地。

(4) 部分典型实例在相关章节保持了连续性,前后衔接,逐步扩展,既便于教师讲解,又便于学生学习理解。

(5) 数据类型、运算符与表达式等语言知识穿插在相关章节介绍,随用随讲,既突出了程序设计主线,又有利于提高教学效率和学生的学习兴趣。

(6) 每章之后均设有实验指导,分为基础实验和综合实验两部分,实验内容与理论教学内容密切衔接,相辅相成。基础实验内容紧密结合课程中的例题程序设置,以“理解→验证→完善→扩充→提高”为实验教学线索,重在巩固课堂教学知识,提高调试程序和完善程序的能力;综合实验以算法设计和程序实现为重点,训练运用所学知识解决实际问题的能力。

本书作者秉承“建设精品教材,培养优秀人才”的教育理念,广泛吸收借鉴其他优秀教材的长处,在前期教材建设的基础上,融入多年的教学实践经验和教学研究成果,编写完成了该教材,本教材力求深入浅出,循序渐进,语言流畅,通俗易懂,便于讲解,便于学习。但由于作者水平所限,书中难免存在不足之处,敬请读者批评指正。

本书由张磊任主编,郑喜珍、李竹健、刘海慧任副主编,参加本书编写的还有冯伟昌、黄忠义、魏建国、张元国、王桂东、王金才、张文、高永存、王涛、薛莹、徐英娟、马明祥、滕秀荣、彭玉忠、潘振昌、徐思杰等。全书由张磊统稿并定稿。

本书有配套的教学课件、程序源代码以及试题库等丰富的教学资源,读者可以从清华大学出版社网站 www.tup.com.cn 下载。

作者联系邮箱: Mail16300@163.com。

编者

2013年5月

目 录

第 1 章 程序设计概述	1
1.1 程序设计语言	1
1.2 算法	2
1.2.1 算法的概念与算法描述	3
1.2.2 算法的逻辑结构	4
1.2.3 算法的特性	7
1.2.4 算法评价	8
1.3 程序设计与实现	8
1.3.1 程序设计的基本过程	8
1.3.2 使用 Visual C++ 6.0 实现 C 语言程序	9
1.3.3 程序设计示例	11
1.4 C 语言程序的基本结构	14
1.4.1 程序的函数化结构	14
1.4.2 程序中的常量和变量	17
1.4.3 程序中的基本语句	17
1.4.4 标识符与保留字	17
1.4.5 程序风格	18
小结	19
习题 1	20
实验 1 编辑运行 C 语言程序	22
第 2 章 简单程序设计	24
2.1 用 printf() 函数输出数据	24
2.1.1 固定数值的加法程序	24
2.1.2 printf() 函数	26
2.2 用 scanf() 函数输入数据	30
2.2.1 任意数值的加法程序	30
2.2.2 scanf() 函数	31
2.3 输入输出字符数据	34
2.4 语言知识补遗	35
2.4.1 数据类型	35
2.4.2 常量详解	36

2.4.3	简单变量详解	38
2.4.4	算术运算	40
2.4.5	赋值运算	41
2.4.6	宏命令	43
2.5	程序设计举例	48
2.5.1	计算三角形面积	48
2.5.2	字母转换	49
2.5.3	鸡兔同笼问题	50
*2.6	表达式中数据类型的自动转换	52
*2.7	用 typedef 命名数据类型	53
*2.8	使用 C++ 命令输入输出数据	53
2.8.1	简单的 C++ 程序	54
2.8.2	C++ 的输入和输出	55
小结		57
习题 2		58
实验 2	简单程序设计	62
第 3 章	分支结构程序设计	64
3.1	简单条件的分支程序	64
3.1.1	判断优等生程序	64
3.1.2	关系表达式	65
3.1.3	if 语句	66
3.2	复合条件的分支程序	73
3.2.1	应用新标准的优等生程序	74
3.2.2	逻辑表达式	75
3.3	用 switch 语句实现分支控制	77
3.4	goto 语句	79
3.5	条件运算	80
3.6	分支结构应用举例	82
3.6.1	闰年问题	82
3.6.2	判断等边三角形	84
3.6.3	求解一元二次方程	85
3.6.4	学生成绩分等显示	86
小结		88
习题 3		88
实验 3	分支结构程序设计	91
第 4 章	循环结构程序设计	96
4.1	while 循环结构程序	96

4.1.1	while 循环程序示例	96
4.1.2	while 语句	97
4.1.3	自增、自减运算	99
4.2	do-while 循环结构程序	101
4.2.1	do-while 循环程序示例	101
4.2.2	do-while 语句	102
4.3	for 循环结构程序	102
4.3.1	for 循环程序示例	102
4.3.2	for 语句	103
4.3.3	逗号表达式	105
4.4	循环体中的控制命令	105
4.4.1	break 命令	106
4.4.2	continue 命令	107
4.5	多重循环程序	108
4.5.1	多重循环程序示例	108
4.5.2	多重循环的一般结构	109
4.6	循环结构程序设计举例	110
4.6.1	字符统计	110
4.6.2	比赛评分	111
4.6.3	学生成绩分等统计	113
4.6.4	最大公约数	116
4.6.5	Fibonacci 数列	116
4.6.6	乘法表	117
4.6.7	搬砖问题	118
4.6.8	找素数	120
4.6.9	哥德巴赫猜想	121
	小结	122
	习题 4	123
	实验 4 循环结构程序设计	128
第 5 章 数组程序设计		131
5.1	一维数组程序设计	131
5.1.1	逆序输出数据程序	131
5.1.2	一维数组的定义	132
5.1.3	数值型一维数组的输入和输出	133
5.1.4	数值型一维数组的初始化	135
5.1.5	字符型一维数组的初始化	137
5.1.6	一维数组的存储	137
5.2	字符串操作	138

5.2.1	字符串的输入输出	138
5.2.2	多字符串操作函数	142
5.3	二维数组程序设计	145
5.3.1	矩阵求和程序	145
5.3.2	二维数组的定义	146
5.3.3	二维数组的输入和输出	147
5.3.4	二维数组的初始化	148
5.3.5	二维数组的存储	150
5.4	数组应用	150
5.4.1	排序	150
5.4.2	查找	153
5.4.3	单词统计	155
5.4.4	矩阵运算	156
5.4.5	成绩处理	160
5.4.6	杨辉三角形	163
	小结	164
	习题 5	165
	实验 5 数组程序设计	169
第 6 章	函数程序设计	172
6.1	函数概述	172
6.2	自定义函数示例	173
6.3	函数定义及调用	175
6.3.1	函数定义	175
6.3.2	函数值和 return 命令	176
6.3.3	函数调用	177
6.4	函数嵌套和递归函数	182
6.4.1	函数嵌套	182
6.4.2	递归函数	184
6.5	关于数组的函数设计	188
6.5.1	数组元素作函数参数	189
6.5.2	一维数组名作函数参数	190
6.5.3	关于二维数组的函数设计	193
6.6	函数应用举例	195
6.6.1	计算长方体的面积	195
6.6.2	求解 Fibonacci 数列	196
6.6.3	排序函数的设计与应用	198
6.7	变量的作用域和存储类型	200
6.7.1	变量的作用域	200

6.7.2 变量的存储类型	202
* 6.8 编译连接多个源文件的 C 程序	204
小结	207
习题 6	207
实验 6 使用自定义函数的程序设计	212
第 7 章 指针程序设计	215
7.1 指针概述	215
7.2 指针变量的定义和使用	216
7.2.1 指针变量程序示例	216
7.2.2 定义指针变量	217
7.2.3 使用指针变量	217
7.3 指针与数组	219
7.3.1 用指针访问一维数组	220
7.3.2 用指针访问二维数组	222
7.3.3 用指针处理字符串	225
7.3.4 指针数组	227
7.4 指针作函数参数	228
7.4.1 简单变量指针作函数参数	228
7.4.2 指向数组的指针作函数参数	230
7.4.3 字符串指针作函数参数	232
7.4.4 指针数组作函数参数	233
* 7.4.5 使用带参数的 main() 函数	235
7.5 指针函数和指向函数的指针变量	236
7.5.1 指针函数	236
* 7.5.2 指向函数的指针变量	238
7.6 指针应用举例	239
小结	244
习题 7	245
实验 7 指针程序设计	251
第 8 章 结构体程序设计	255
8.1 结构体数据概述	255
8.2 结构体类型和结构体变量	256
8.2.1 使用结构体变量存储学生信息	256
8.2.2 定义结构体数据类型	257
8.2.3 结构体变量的定义及使用	258
8.3 结构体数组	262
8.3.1 结构体数组的定义及元素引用	262

8.3.2	结构体数组的初始化	262
8.3.3	利用结构体数组管理学生信息	263
8.4	结构体指针变量	264
8.4.1	结构体指针变量的定义及使用	264
8.4.2	结构体指针作函数的参数	266
8.5	使用链表存储数据	267
8.5.1	使用链表存储学生信息	267
8.5.2	链表的特点	268
8.5.3	动态内存管理函数	269
8.5.4	定义链表结构	270
8.6	链表的基本操作	271
8.6.1	链表结点的插入	271
8.6.2	链表结点的删除	275
8.6.3	链表结点的查找	276
8.7	结构体应用举例	279
8.7.1	字符串加密	279
8.7.2	学生数据排序	281
8.7.3	Josephus 问题	284
	小结	286
	习题 8	287
	实验 8 结构体程序设计	293
第 9 章	文件程序设计	297
9.1	文件概述	297
9.1.1	文件的概念	297
9.1.2	文件的分类	298
9.1.3	文件的一般操作过程	299
9.1.4	文件的指针	299
9.2	文件的基本操作	300
9.2.1	打开和关闭文件	300
9.2.2	最基本的文件读写函数	302
9.3	文件的数据块读写操作	305
9.4	文件的其他操作	308
9.4.1	文件的格式化读写	308
9.4.2	文件位置指针的定位	310
9.4.3	文件的字符串操作	313
9.5	文件应用举例	314
9.5.1	文件复制	314
9.5.2	存储在文件中的学生数据排序	315

小结	316
习题 9	317
实验 9 文件程序设计	320
附录 A 选择题参考答案	325
附录 B C 语言的运算符	326
附录 C C 语言的经典保留字	328
附录 D 常用的 C 语言标准库函数	329
附录 E 常用 ASCII 码字符对照表	332
参考文献	333

第1章

程序设计概述

• 本章导读

本章是程序设计的概述内容,介绍程序设计语言及程序的概念、算法的概念、算法的设计和描述方法以及算法的逻辑结构知识;介绍 C 语言程序设计的基本过程,包括问题分析、算法设计、编写程序以及使用 VC++ 6.0 编辑运行程序的知识;本章最后,对 C 语言程序的基本结构进行介绍,重点内容是 C 语言程序的函数化结构。

“算法”既是本章的核心概念,也是 C 语言程序设计自始至终的核心内容。程序=数据结构+算法,希望读者充分认识算法在程序设计中的重要性。

• 主要知识点

- (1) 算法的概念、特性,算法设计及描述。
- (2) 程序设计的基本过程。
- (3) 用 VC++ 6.0 编辑运行 C 语言程序。
- (4) C 语言程序的结构特点。

1.1 程序设计语言

程序设计语言是计算机能够理解和识别的一种语言体系,它按照特定的规则组织计算机指令,使计算机能够自动进行各种操作处理。按照程序设计语言的规则组织起来的一组计算机指令,称为计算机程序。不同的程序设计语言具有不同的使用规则,因而编写的计算机程序也不同。

程序设计语言分为三种类型,即机器语言、汇编语言和高级语言。机器语言是一种二进制语言,它直接使用二进制代码描述指令,是唯一能够被计算机硬件直接识别、直接执行的程序设计语言。用这种语言编写的程序很不直观,并且难懂、难写、难记、难以修改和维护。汇编语言用符号代替了机器指令代码,而且助记符与指令代码一一对应。与机器语言相比,汇编语言比较直观、容易记忆,但它的通用性与机器语言一样,都很差。机器语言和汇编语言都属于低级语言。高级语言是接近于自然语言的一种计算机语言,具有很强的描述能力,能够方便地按照处理问题的逻辑思路编写计算机程序。高级语言进一步分为面向过程的设计语言 and 面向对象的程序设计语言。典型的面向过程的程序设计语言有 Pascal、Basic、C 等,典型的面向对象的程序设计语言有 Visual Basic、Visual C++、Java 等。

用计算机高级语言编写的程序要在计算机上运行,必须依靠语言处理程序的支持。计

计算机语言处理程序将用高级语言编写的源程序转换为机器语言代码序列,然后由计算机加以执行。

学习程序设计语言,必须要注意学习它的使用规则,只有正确地使用语言规则,才能编写出正确的计算机程序。

C语言是1972年由美国的Dennis Ritchie设计发明的,并首次在UNIX操作系统的DEC PDP-11计算机上使用。它由早期的编程语言BCPL(Basic Combind Programming Language)发展演变而来。在1970年,AT&T贝尔实验室的Ken Thompson根据BCPL设计出较先进的并取名为B的语言,最后导致了C语言的问世。1983年,美国国家标准化协会(ANSI)根据C语言问世以来各种版本对C的发展和扩充,制定了C的标准,称为ANSI C。1987年,ANSI又公布了新的标准——87 ANSI C。目前流行的C编译系统都是以它为基础的。

在C语言的基础上,1983年,贝尔实验室的Bjarne Stroustrup推出了C++。C++作为C语言的继承和发展,不仅保留了C语言的高度灵活、高效率和易于理解等诸多优点,还包含几乎所有面向对象的特征,成为一种面向对象的程序设计语言。C++所支持的面向对象的概念,容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统的结构化程序设计不同的思维方式和编程方法,因而也增加了整个语言的复杂性。

下面是一个用C语言编写的计算机程序,它通过累加的方法计算1~100的所有自然数的和。

```
/* program e1-0.c */
#include <stdio.h>
int main(void)
{
    int i=1,s=0;
    while(i<=100)          /* 循环控制 */
    {
        s=s+i;              /* 数据累加 */
        i=i+1;              /* 生成下一个要累加的数 */
    }
    printf("sum = %d\n",s);  /* 输出结果 */
    return 0;
}
```

当然,要计算1~100的所有自然数的和还有多种方法,如等差数列求和法。如何选用和设计有效的方法解决问题,是程序设计的重要内容。

1.2 算法

瑞士科学家、Pascal语言发明者Niklaus Wirth对计算机程序给出了一个著名的定义,即:程序=数据结构+算法。该定义归结了计算机程序的两个核心问题,强调了算法在程序中的重要性。本节对算法的基本知识作简要介绍。

1.2.1 算法的概念与算法描述

1. 算法的概念

算法是逐步求解问题的方法,是在有限步骤内求解某一问题所使用的一组定义明确的规则,是计算机处理问题所需要的具体步骤。算法的最终实现是计算机程序,程序设计人员只有将算法转变为计算机程序,才能利用计算机解决问题。

算法的建立通常会经过由粗略到细化的过程,先找出解决问题的基本思路,把解决问题的基本过程表达出来,确立粗略的算法框架,然后对框架中的内容进行逐步细化,添加必要的细节,形成解决问题的有效算法。

2. 算法的描述方法

算法描述的方法多种多样,可以使用自然语言描述,也可以使用专门的算法表达工具进行描述。为了使算法表达得更清晰,更容易实现程序编写,在进行程序设计时通常使用专门的算法表达工具对算法进行描述,如流程图、N-S图、PAD图、伪代码等。以下是使用自然语言、流程图以及伪代码描述算法的说明。

1) 用自然语言描述算法

问题: 计算 1~100 的所有自然数的和。

最直观的理解,计算 1~100 的所有自然数的和(以下称为“自然数累加”问题),就是求以下代数式的值:

$$1 + 2 + 3 + 4 + \cdots + 99 + 100$$

显然,可以采用逐个自然数累加的方法求和。算法的粗略描述如下: 假若用 i 表示当前要加的数, i 的初值为 1,每加一次, i 的值增加 1; 用 s 表示已经累加取得的结果,初值为 0。那么,问题求解的过程就是不断地将 i 加到 s 中,直到 i 的值超过 100 时,结束累加过程,并将累加的结果显示在计算机屏幕上。

上面一段文字,对问题的求解方法进行了基本描述,但作为算法还不够完整,还需要更明确地表达出求解问题的步骤。下面是包含执行步骤的算法描述,是用自然语言对算法进行描述的常见形式。

步骤① i 和 s 赋初值,使 $i=1, s=0$;

步骤② 判断 i 的值,若 $i \leq 100$ 则执行步骤③; 否则,转步骤⑤;

步骤③ s 加 i ;

步骤④ i 加 1,转步骤②;

步骤⑤ 显示 s 的值,结束。

按照上述算法给定的 5 个步骤,即可求解“自然数累加”问题。若选用一种计算机语言正确描述该算法,就会得到求解“自然数累加”问题的计算机程序,运行程序,将得到“自然数累加”问题的计算结果。

2) 用流程图描述算法

流程图是常使用的一种算法描述工具,其特点是绘制简单,结构清晰,逻辑性强,便于描述,容易理解。表 1-1 列出了常用的流程图符号及其功能。