

高等学校计算机基础教育规划教材

程序设计技术 (C语言)

尚展垒 司丽娜 孟牒 郑远攀 等 编著



清华大学出版社

高等学校计算机基础教育规划教材

程序设计技术 (C语言)

尚展垒 司丽娜 孟牒 郑远攀 等 编著

清华大学出版社
北京

内 容 简 介

本书以 Visual C++ 6.0 作为开发平台,利用 C 语言描述程序设计的基本思想和方法,同时借助 EasyX 介绍图形绘制的方法和原理。全书共分 15 章,主要介绍 C 语言基础知识,程序的控制结构,C 语言中的构造数据类型,常用算法,指针型数据类型,位运算,程序中的文件以及图形的绘制等内容,第 15 章讲述了如何利用软件工程的方法指导读者开发大型软件。

本书适合作为大学计算机专业和非计算机专业的程序设计基础课程教材,也可供自学的读者使用。

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

程序设计技术: C 语言/尚展垒等编著. —北京:清华大学出版社,2019

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-51430-5

I. ①程… II. ①尚… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 242399 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:时翠兰

责任印制:丛怀宇

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:三河市龙大印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:28.25

字 数:688 千字

版 次:2019 年 3 月第 1 版

印 次:2019 年 3 月第 1 次印刷

定 价:69.00 元

产品编号:080412-01

前 言

C语言从诞生之日起就一直保持着旺盛的生命力,不断发展壮大、日臻完善,已经成为目前使用最广泛的编程语言之一。与其他高级语言相比,C语言处理功能丰富,表达能力强,使用灵活方便,执行程序效率高,可移植性强;具有丰富的数据类型和运算符,语句非常简单,源程序简洁清晰;可以直接处理硬件系统和对外围设备接口进行控制。C语言是一种结构化的程序设计语言,支持自顶向下、逐步求精的结构化程序设计技术。另外,C语言程序的函数结构也为实现程序的模块化设计提供了强有力的保障。因此纵然有C++、Java这样的后继者,但到目前为止,它们依然没有取代C的迹象。尤其C99标准发布以后,C语言的旺盛生命力再次得到了保持和延续。

本书的编写者全部是一直战斗在高等学校教学一线,承担“C语言程序设计”课程的教学任务的大学教师,有丰富的教学和C语言编程工作经验,有将自己积累的C语言程序设计经验介绍给大家的强烈愿望,因为在教学实践中,作者感受最深的就是,学习者普遍反映C语言难学、难懂,而事实上,学习者感觉难的往往并不是C语言的核心内容,学习的过程就是学习者与教师、与教材交互的过程,只要遵照一定的学习规律,这个交互过程并不难达成。学习者应该明白,选择一本交互性好的教材是至关重要的。不可否认,一些经典的C语言教材在教学中所起的巨大作用。但是,传统教材过细的内容组织也让学习者迷失了方向。本教材以指针为主线,贯穿于始终。另外,本教材也特别强调实践能力的培养,学习者首先应该学会用适当的编程工具编制、调试程序。学习者在编程实践中不断遇到问题,不断解决问题,自然就会明白许多细节。本书在介绍核心语法的基础上,以培养动手编程能力为首要目标,把那些烦琐的内容留待以后慢慢研究。

本书共分15章,将C语言的特色内容“指针”贯穿于始终,将这一难点分散到相关章节,避免了难点集中造成学习者丧失学习的兴趣。同时,借助于第三方工具的支持,讲述了在Visual C++中的绘图方法,由于绘图的引入,不但可以提高学生学习的兴趣,还能增加C语言的实际应用。

第1章介绍了程序设计的基本知识,详细阐述了软件的编制方法,使学习者对软件的编制有一个概念上的认识和理解,并能将这些方法应用于后续章节的学习中。

第2章介绍了C语言的基本知识,重点阐述了变量的声明方法,使学习者掌握变量、地址、存储数据之间的关系,同时还讲述了运算符。

第3章介绍了C语言标准库函数和顺序结构程序设计的基本方法,使学习者能够使用库函数编写简单的顺序结构程序。

第4章介绍了选择结构的相关语法,使学习者能够解决选择结构所涉及的问题。

第5章介绍了循环结构的相关语法,使学习者能够解决循环结构所涉及的问题。

第6章介绍了函数的相关知识,重点介绍了基本使用方法、函数参数的传值调用和传址调用,使学习者能够综合使用函数参数的传值调用和传址调用来解决实际问题。

第7章介绍了C语言中数值型一维数组和二维数组的相关知识,使学习者能够熟练使

用函数来解决数组的相关问题。

第8章介绍了排序、查询、方程求解、求积分等内容,使读者能够利用所学的知识解决一些实际问题。

第9章介绍了字符串与字符数组的相关知识,使学习者能够处理与字符串和字符数组相关的问题。

第10章介绍了结构和联合的基本概念,使学习者能够综合使用数组、指针以及结构和联合来解决一些实际问题。

第11章介绍了指针的高级使用,使学习者能够处理涉及指针数组、函数指针和指针函数的相关问题。

第12章介绍了位运算的相关知识,使学习者对位运算有一个较为系统的认识并能够使用所学的位运算知识解决相关问题。

第13章介绍了文件的概念以及处理文件问题所涉及的函数,使学习者在掌握常用文件函数的基础上来解决一些实际问题。

第14章介绍了如何借助于第三方工具 EasyX,很方便地在 Visual C++ 6.0 下绘制图形以及动画,进而编制简单的游戏。

第15章介绍了用软件工程的方法指导大型软件的开发方法,以提高软件的开发效率、降低开发成本、提高软件正确性等。

以上各部分都可以独立教学,自成体系。教师可根据情况适当取舍。

本书由郑州轻工业学院的尚展垒、司丽娜、孟牒、郑远攀等编著,其中,尚展垒任主编,司丽娜、孟牒、郑远攀任副主编,参加本书编写的还有郑州轻工业学院的杨学冬、张江伟和艾怡静。第1章、第10章和附录由尚展垒编写,第2、3章由司丽娜编写,第4、5章由孟牒编写,第6、7章由郑远攀编写,第8~10章由杨学冬编写,第11、12、15章由张江伟编写,第13、14章由艾怡静编写。尚展垒负责本书的组织工作,司丽娜负责本书的统稿工作。

本书的顺利出版得到郑州轻工业学院和清华大学出版社的大力支持。在本教材的编写过程中从许多同行的著作中得到启发,在此表达感谢之情。由于作者学识所限,难免存在疏漏,恳请各位读者批评指正。

编者

2018年8月

目 录

第 1 章 程序设计技术概述	1
1.1 程序设计语言概述	1
1.1.1 程序设计语言	1
1.1.2 计算思维	4
1.2 算法	4
1.2.1 算法的概念	4
1.2.2 算法的特性	6
1.2.3 算法的描述	6
1.3 软件的编制步骤	10
1.4 C 程序设计语言的产生与特点	12
1.5 C 语言程序的运行环境(Visual C++ 6.0 编译环境)	13
1.5.1 C 语言程序上机步骤	13
1.5.2 建立 C 程序的步骤	13
1.5.3 Visual C++ 6.0 集成环境	18
1.5.4 程序的调试和运行	19
1.6 错误解析	21
练习 1	22
第 2 章 程序设计基础	24
2.1 C 程序概述	24
2.1.1 一个简单的 C 程序	24
2.1.2 C 语言的字符集	26
2.1.3 C 语言词汇	26
2.2 基本数据类型	28
2.2.1 常量与变量	29
2.2.2 整型数据	32
2.2.3 实型数据	36
2.2.4 字符型数据	39
2.2.5 变量赋初值	44
2.3 运算符与表达式	45
2.3.1 C 语言运算符简介	45
2.3.2 算术运算符和算术表达式	46

2.3.3	关系运算符与关系表达式	48
2.3.4	逻辑运算符与逻辑表达式	49
2.3.5	赋值运算符和赋值表达式	50
2.3.6	逗号运算符和逗号表达式	55
2.3.7	自增、自减运算符	56
2.3.8	条件运算符和条件表达式	58
2.4	不同类型数据之间的转换	59
2.5	错误解析	62
练习 2	64
第 3 章	标准库函数	66
3.1	C 标准库函数的分类	67
3.2	常用数学库函数	68
3.3	printf() 函数	69
3.4	scanf() 函数	77
3.5	putchar() 函数	82
3.6	getchar() 函数	83
3.7	随机函数	84
3.8	错误解析	86
练习 3	86
第 4 章	选择结构	88
4.1	复合语句	88
4.2	if 语句	90
4.2.1	if 语句中的表达式	90
4.2.2	单分支 if 语句	91
4.2.3	双分支 if 语句	93
4.2.4	多分支 if 语句	95
4.2.5	if 语句的嵌套	98
4.2.6	条件运算符实现选择结构	102
4.3	switch 语句	103
4.4	应用程序举例	106
4.5	错误解析	111
练习 4	113
第 5 章	循环控制结构	117
5.1	while 语句	117
5.2	for 语句	124
5.3	do...while 语句	130
5.4	多重循环结构	135

5.5	break 语句和 continue 语句	139
5.5.1	break 语句	139
5.5.2	continue 语句	140
5.6	应用程序举例	143
5.7	错误解析	151
	练习 5	155
第 6 章	函数	157
6.1	C 程序与函数概述	157
6.1.1	模块化程序设计	157
6.1.2	C 程序的一般结构	158
6.2	函数的定义与调用	159
6.2.1	函数的定义	159
6.2.2	函数的调用	162
6.2.3	函数的参数传递	165
6.3	函数的传址引用	167
6.3.1	地址的存储与使用	167
6.3.2	指针说明和指针对象的引用	168
6.4	局部变量与全局变量	172
6.4.1	局部变量	173
6.4.2	全局变量	174
6.5	变量的存储类型	176
6.5.1	存储类型区分符	176
6.5.2	自动变量	177
6.5.3	静态变量	179
6.5.4	外部变量	181
6.5.5	寄存器变量	182
6.5.6	存储类型小结	183
6.6	函数的嵌套与递归调用	185
6.6.1	函数的嵌套调用	185
6.6.2	函数的递归调用	187
6.7	编译预处理	190
6.7.1	宏定义	191
6.7.2	文件包含	194
6.7.3	条件编译	196
6.8	错误解析	197
	练习 6	198
第 7 章	数组	200
7.1	一维数组的定义及使用	200

7.1.1	一维数组的定义	200
7.1.2	一维数组的引用	202
7.1.3	一维数组的初始化	204
7.1.4	程序举例	206
7.2	一维数组与指针运算	207
7.2.1	一维数组的数组名	207
7.2.2	一维数组的下标与指针	208
7.2.3	作为函数参数的一维数组的数组名	212
7.3	二维数组的定义及使用	215
7.3.1	二维数组的定义	215
7.3.2	二维数组元素的引用	216
7.3.3	二维数组的初始化	216
7.3.4	二维数组应用举例	218
7.4	二维数组与指针运算	221
7.4.1	二维数组与元素指针	221
7.4.2	二维数组与行指针	222
7.4.3	作为函数参数的二维数组的数组名	225
7.5	使用内存动态分配实现动态数组	228
7.5.1	动态内存分配的步骤	228
7.5.2	动态内存分配函数	228
7.6	错误解析	231
	练习 7	232
第 8 章 常用算法		235
8.1	算法的概念	235
8.1.1	算法描述	235
8.1.2	算法的特性	236
8.1.3	算法的评估	236
8.2	排序算法	237
8.2.1	冒泡排序算法	237
8.2.2	选择排序算法	239
8.2.3	插入排序算法	242
8.2.4	基于二维数组的排序	243
8.3	查找算法	245
8.3.1	顺序查找	245
8.3.2	二分查找	246
8.3.3	基于二维数组的查找算法	247
8.3.4	其他查找方法	250
8.4	基本数值算法	250
8.4.1	基本数值算法概述	250

8.4.2	求一元非线性方程实根	250
8.4.3	求一元函数定积分的数值	255
练习 8		259
第 9 章	字符数组与字符串	261
9.1	字符数组	261
9.1.1	字符数组的定义与赋值	261
9.1.2	字符数组的初始化	262
9.1.3	字符数组的引用	264
9.2	字符串	266
9.2.1	字符串的定义及其输入与输出	266
9.2.2	字符串的处理与字符串处理函数	269
9.2.3	字符串与指针运算	273
9.3	字符数组与字符串应用举例	276
9.4	错误解析	278
练习 9		280
第 10 章	结构和联合	282
10.1	结构类型的定义与引用	282
10.1.1	结构类型的定义	282
10.1.2	结构变量的引用	285
10.2	结构数组的声明、引用和初始化	287
10.3	联合	290
10.3.1	联合的定义	290
10.3.2	联合变量的说明	291
10.3.3	联合变量的使用	292
10.4	枚举类型	294
10.5	定义类型说明符	296
10.6	应用程序举例	297
10.7	常见错误解析	299
练习 10		300
第 11 章	指针	302
11.1	数组、地址与指针	302
11.1.1	数组、地址与指针的关系	302
11.1.2	一维数组中的地址与指针	303
11.1.3	二维数组中的地址与指针	305
11.2	指针数组与指向指针的指针	306
11.2.1	指针数组	306
11.2.2	指向指针的指针	310

11.3	main()函数的参数	312
11.4	函数指针	315
11.5	指针函数	318
11.6	链表	321
11.6.1	链表的概念	321
11.6.2	链表的实现	321
11.6.3	单向链表的操作	322
11.6.4	链表的建立	322
11.6.5	链表的输出	325
11.6.6	链表结点的插入与删除	325
11.7	应用程序举例	326
11.8	错误解析	328
	练习 11	330
第 12 章	位运算	334
12.1	位运算的概念	334
12.1.1	字节与位	334
12.1.2	补码	334
12.2	二进制位运算	336
12.2.1	二进制位运算	336
12.2.2	位复合赋值运算符	344
12.3	应用程序举例	344
12.4	错误解析	345
	练习 12	345
第 13 章	文件操作	347
13.1	文件概述	347
13.2	文件的使用	348
13.2.1	文件的声明	348
13.2.2	文件的打开与关闭	349
13.2.3	文件的读写	351
13.3	随机文件的读写	361
13.4	应用程序举例	362
13.5	错误解析	368
	练习 13	368
第 14 章	绘制图形	370
14.1	绘图简介	370
14.2	EasyX 的下载与安装	373
14.2.1	EasyX 的下载	373

14.2.2	安装 EasyX	373
14.3	绘图前的准备	374
14.3.1	颜色	375
14.3.2	坐标	375
14.3.3	设备	376
14.4	绘图函数	376
14.4.1	绘图环境相关函数	376
14.4.2	颜色模型相关宏及函数	377
14.4.3	图形颜色及样式设置相关函数	378
14.4.4	图形绘制相关函数	382
14.4.5	文字输出相关函数	385
14.5	绘图举例	388
14.6	错误解析	393
练习 14	394
第 15 章	项目开发	395
15.1	软件工程概述	395
15.1.1	软件工程的基本概念	395
15.1.2	分析阶段	397
15.1.3	设计阶段	397
15.1.4	实现阶段	398
15.1.5	测试阶段	399
15.1.6	软件维护	400
15.1.7	文档	400
15.2	客户信息管理系统	401
15.2.1	用软件工程方法指导软件开发	401
15.2.2	客户信息管理系统的实现	403
15.3	俄罗斯方块	414
15.3.1	俄罗斯方块简介	414
15.3.2	俄罗斯方块的实现代码	415
练习 15	425
参考文献	426
附录 A	ASCII 编码	427
附录 B	C 语言的运算符	428
附录 C	C 语言的库函数	429
附录 D	EasyX 的库函数	436

第 1 章 程序设计技术概述

随着科学技术的飞速发展,计算机程序设计语言层出不穷,作为程序设计初学者,首先应该了解什么是程序,什么是程序语言,以及如何进行程序设计。作为程序设计的入门教材,本书将以 C 语言程序设计为主线,详细讲述基本概念、语法规则和基本设计方法。本章首先对程序设计技术的基本知识进行概述,然后对程序设计语言、程序设计技术等概念进行介绍。本章的重点是算法的概念、特征、算法描述方式、软件开发的过程、C 语言的特点及其运行环境等内容。

本章知识点:

- (1) 程序设计语言和程序设计的概念。
- (2) 算法的概念、特征以及描述方式。
- (3) 软件编制的基本步骤。
- (4) C 语言的特点及其运行环境。

1.1 程序设计语言概述

为了使编程者更方便地开发程序,已出现了上千种编程语言,不同的语言具有不同的优点。选择一种适合的语言是非常重要的。下面介绍程序及程序设计语言的基本概念。

1.1.1 程序设计语言

1. 程序

像飞机、汽车或割草机等机器一样,计算机也是一台机器,它由多个电子元器件组成。一台计算机只有运行相应的控制程序,才能完成要做的任务。区分计算机和其他类型机器的关键就是看它们是如何执行任务的。例如,一辆汽车是由司机控制和驾驶的,而一台计算机的“驾驶员”就是程序。所谓程序就是按某种顺序排列的,能使计算机执行某种任务(例如解题、检索数据或对一个系统进行控制等)的连续执行的指令集合,也就是说,程序是计算机指令的序列,编制程序的工作就是为计算机安排指令序列。

人们需要计算机完成什么工作,就要将每个步骤用指令的形式描述出来,并把指令存放于计算机内部存储器中,在需要结果时就向计算机发出一个简单的命令,计算机就会按照指定的顺序自动逐条执行命令,直到全部指令执行完毕并得到预期的结果。这些编写程序的指令是只有计算机才能理解的二进制编码 0 和 1。这种编码方式编写的程序让人不好掌握且难以记忆,不利于程序编写和软件发展,因此计算机科学家研制了各种计算机能够识别且接近于人类自然语言的计算机语言,这就是编写软件的程序设计语言。

2. 程序设计语言

程序设计语言又称为编程语言,是一组用来定义计算机程序的语法规则。一种程序设计语言能够准确地定义计算机需要使用的数据,精确定义在不同情况下所应当采取的操作。

程序设计语言按照使用的方式和功能的不同可分为低级语言和高级语言。低级语言包括机器语言、汇编语言等,高级语言包括面向过程、面向对象的语言。

(1) 机器语言。机器语言(Machine Language)是直接由二进制编码指令表示的计算机语言,即机器指令的集合,它与计算机同时诞生,属于第一代计算机语言,其指令是由 0 和 1 组成的一串代码,有一定的位数并被分成若干段,各段的编码表示不同的含义。机器语言也称为面向机器的语言,用机器语言编写的程序称为机器语言程序或指令程序(机器码程序),其机器本身能直接识别和执行这种目标程序机器码。不同型号的计算机,机器语言一般是不同的。

例如,16 位的计算机指令 1011011000000000,它表示让计算机进行一次加法操作;而指令 1011010100000000 则表示进行一次减法操作。早期的程序设计均使用机器语言。程序员们先把用数字 0、1 编成的程序代码打在纸带或卡片上,1 表示打孔,0 表示不打孔,再将程序通过纸带机或卡片机输入计算机进行运算。例如,应用 8086 处理器完成运算 $s=768+12288-1280$,机器码如下:

```
1011000000000000000000000011
0000010100000000000110000
0010110100000000000000101
```

假如将程序错写成以下面这样,就很难被发现。

```
1011000000000000000000000011
0000010100000000000110000
0001011010000000000000101
```

机器语言的二进制编码指令不易记忆、不易查错、不易修改,可移植性和重用性差。仅上面这个简单的代码,就暴露了机器码晦涩难懂、不易查错的缺点。为了克服上述缺点,便出现了一种采用一定含义的符号英文单词缩写指令助记符来表示指令的程序语言——汇编语言。

(2) 汇编语言。汇编语言(Assembly Language)是面向机器的程序设计语言。在汇编语言中,用助记符(Mnemonic)代替操作码,用地址符号(Symbol)或标号(Label)代替地址码。这种用符号代替机器语言二进制码的计算机语言被称为符号语言。

汇编语言编写的程序,计算机不能直接识别,必须用一种软件将汇编语言翻译成机器语言,这种具有翻译功能的软件就是汇编软件。汇编软件把汇编语言翻译成机器语言的过程称为汇编,如图 1-1 所示。

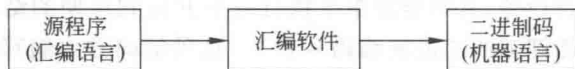


图 1-1 汇编语言源程序汇编的过程

汇编语言比机器语言易读、易写、易记,它与机器语言指令是一一对应的,所以汇编语言不具有 Pascal、C 语言等高级语言通用性强的特点,它与计算机内部硬件结构密切相关,为某种计算机独有。

尽管汇编语言具有执行速度快、易于实现对硬件的控制等优点,但它仍存在着机器语言的缺点:汇编语言依赖于硬件体系,与 CPU 的硬件结构紧密相关,不同 CPU 的汇编语言不尽相同,因此用汇编语言编写的程序可移植性差;其次,进行汇编语言程序设计时,必须了解所使用硬件的结构与性能,对程序设计人员的要求较高,因此这种语言难以普及应用,为了克服以上缺点,便产生了接近人类自然语言的高级语言。

(3) 高级语言。高级语言的语法结构更类似普通英文,更关键的是它不再依赖特定计算机的结构与指令系统,用同一种高级语言编写的源程序,在不同的计算机上运行后基本能获得一样的结果。

目前,常用的高级语言有 BASIC、FORTRAN、COBOL、Pascal、PL/M、C 等。一般来说,高级语言在编程时不需要对机器结构及其指令系统有深入的了解,除此之外,用高级语言写的程序通用性好、便于移植。相对面向机器的低级语言,高级语言具有以下优点。

① 高级语言更接近人类自然语言,易学、易掌握,一般工程技术人员只要几周时间的培训就可以胜任程序员的工作。

② 高级语言为程序员提供了结构化程序设计的环境和工具,使设计出来的程序可读性好、可维护性强、可靠性高。

③ 高级语言与具体的计算机硬件关系不大,因而所写出来的程序可移植性好,重用率高。

④ 可将烦琐的事务交给编译程序做,自动化程度高,开发周期短,使程序员可以集中时间和精力去提高程序的质量。

高级语言完全采用了符号化的描述形式,用类似自然语言的形式描述对问题的处理过程,可使程序员能更加专注于分析问题的求解过程,而不用了解和关心计算机的内部结构和硬件细节,更易于被人们理解和接受。20 世纪 80 年代以来,众多的第四代非过程化语言、第五代智能化语言相继推出,第四代语言将原来程序员告诉计算机怎么做,变成了程序员告诉计算机做什么,这是一种全新的开发方式,第四代语言就是被人们称为的面向对象语言。

(4) 面向对象语言。面向对象语言(Object-Oriented Language, OOL)是以对象作为程序基本结构单位的程序设计语言,程序设计的核心是对象,对象是程序运行的基本成分。面向对象语言的发展有两个方向:一种是纯面向对象语言,例如 Smalltalk、Eiffel 等;另一种是混合型面向对象语言,即在过程式语言及其他语言中加入类、继承等成分,例如 C++、Objective-C 等。面向对象语言刻画客观系统较为自然,便于软件扩充与复用。

综上所述,以上 4 种计算机语言各有优缺点。程序员在使用时,需根据应用场合选用。在实时控制系统中,特别是在对程序的空间和时间要求很高,需要直接控制设备的场合,通常采用汇编语言;在应用系统程序设计、多媒体应用、数据库等诸多领域,采用面向对象语言比较合适。面向过程是程序设计的基础,所以程序设计的初学者只有学习好面向过程的程序设计,才能为程序设计的学习打下坚实的基础,所以本书采用 C 程序设计语言为背景,介绍程序设计的基本概念和方法。

1.1.2 计算思维

计算思维(Computational Thinking)是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

2006年3月,美国卡耐基梅隆大学计算机科学系主任周以真(Jeannette M. Wing)教授在美国计算机权威期刊《Communications of the ACM》上提出并定义了计算思维的概念。周教授为了让人们更易于理解,又将它更进一步地定义为,通过化简、嵌入、转化和仿真等方法,把一个看来困难的问题重新阐释成一个人人们知道问题怎样解决的方法;是一种递归思维;是一种并行处理;是一种把代码译成数据又能把数据译成代码的方法;是一种多维分析推广的类型检查方法;是一种采用抽象和分解来控制庞杂的任务或进行巨大复杂系统设计的方法;是基于关注分离的方法(SoC方法);是一种选择合适的方式去陈述一个问题或对一个问题的相关方面建模,使其易于处理的思维方法;是按照预防、保护及通过冗余、容错、纠错的方式,并从最坏情况进行系统恢复的一种思维方法;是利用启发式推理寻求解答,即在不确定情况下的规划、学习和调度的思维方法;是利用海量数据来加快计算,在时间和空间之间,在处理能力和存储容量之间进行折中的思维方法。

计算思维吸取了解决问题所采用的一般数学思维方法,是现实世界中巨大、复杂系统的设计与评估的一般工程思维方法,以及复杂性、智能、心理、人类行为的理解等的一般科学思维方法。计算思维建立在计算过程的能力和限制之上,计算方法和模型使人们敢于去处理那些原本无法由个人独立完成的问题求解和系统设计。计算思维最根本的内容是抽象(Abstraction)和自动化(Automation)。计算思维中的抽象完全超越物理的时空观,可以完全用符号来表示,其中的数字抽象只是一类特例。与数学和物理科学相比,计算思维中的抽象显得更为丰富,也更为复杂。

计算思维所关注的核心问题是人类思维方式及问题求解能力的培养。在本课程的学习中,一个重要的内容是以系统化、逻辑化的计算思维方式去思考和解决问题,着重培养计算思维能力,强化工程化、系统化程序设计的观念和能力。

1.2 算 法

算法是程序设计的精髓。计算机科学家、Pascal语言的发明者尼克劳斯·沃思(Niklaus Wirth)曾提出一个著名的公式:程序=算法+数据结构。计算机解题的过程中,无论是形成解题思路还是编写程序,都是在实施某种算法。前者是推理实现的算法,后者是操作实现的算法。学习程序设计,还要养成一种严谨的软件开发习惯,熟悉软件工程的基本原则。

1.2.1 算法的概念

什么是算法?当代著名计算机科学家D. E. Knuth在他的一本书中写道:“一个算法,就是一个有穷规则的集合,其中之规则规定了一个解决某一特定类型的问题的运算序列。”简单地讲,算法(Algorithm)就是确定的解决问题方法的有限步骤。

需要明确的是,不是只有科学计算才有算法,在日常生活中做任何一件事情,都是按照

一定规则,一步一步进行的。例如,在工厂中生产一部机器,首先要把零件按工序一道道进行加工,然后,再把各种零件按一定规则组进行组装,这个工艺流程其实就是算法;在农村,种庄稼有耕地、播种、育苗、施肥、中耕、收割等环节,这些栽培技术也是算法。

计算机算法按用途通常可以分为两大类:一类是用于解决数值计算的算法,例如科学计算中用于数值积分、解线性方程的计算方法等;另一类是用于解决非数值计算的算法,例如信息管理、文字处理、图形图像处理、排序、分类、查找等操作的算法。

下面通过3个解决问题的过程来说明一下算法设计的基本思维方法。

例 1-1 求 $1 \times 3 \times 5 \times 7 \times 9$ 。

算法分析:这是一最原始方法。

步骤 1,先求 1×3 ,得到结果 3。

步骤 2,将步骤 1 得到的乘积 3 乘以 5,得到结果 15。

步骤 3,将 15 再乘以 7,得 105。

步骤 4,将 105 再乘以 9,得 945。

这样的算法虽然正确,对于乘数少的还可以,但对于乘数多的计算,步骤太烦琐。

改进的算法如下。

步骤 1,使 $t=1$ 。

步骤 2,使 $i=3$ 。

步骤 3,使 $t \cdot i$,乘积仍然放在变量 t 中,可表示为 $t \cdot i \rightarrow t$ 。

步骤 4,使 i 的值+2,即 $i+2 \rightarrow i$ 。

步骤 5,如果 $i \leq 9$,则重新执行步骤 3 以及其后的步骤 4 和步骤 5;否则,算法结束。

例 1-2 计算函数 $f(x)$ 的值。函数 $f(x)$ 为

$$f(x) = \begin{cases} 3x + a, & x \leq a \\ ax + b + c, & x > a \end{cases}$$

其中, a 、 b 、 c 是常数。

算法分析:本题属于一种函数题目,有两个不同的表达式,根据输入 x 的值决定采用哪个公式计算,使用计算机解题的算法如下。

步骤 1,将 a 、 b 、 c 和 x 的值输入计算机中。

步骤 2,判断 $x \leq a$ 是否成立。如果条件成立,执行步骤 3,否则执行步骤 4。

步骤 3,按照表达式 $3x+a$ 计算出 $f(x)$ 的结果,然后执行步骤 5。

步骤 4,按照表达式 $ax+b+c$ 计算出 $f(x)$ 的结果,然后执行步骤 5。

步骤 5,输出 $f(x)$ 的值。

步骤 6,算法结束。

例 1-3 对给定的两个正整数 m 和 $n(m \geq n)$,求它们的最大公约数。

算法分析:本题是数值运算的算法,利用成熟的算法可以很好地解决此类问题,这个算法就是用辗转相除法求解最大公约数。

例如:假设 $m=35$, $n=12$,余数用 r 表示,它们的最大公约数的求法如下。

步骤 1, $m/n=35/12$ 的余数 r 为 11,将 n 作为新的 m ,以 r 作为新的 n ,继续相除。

步骤 2, $m/n=12/11$ 的余数 r 为 1,将 n 作为新的 m ,以 r 作为新的 n ,继续相除。

步骤 3, $m/n=11/1$ 的余数 r 为 0,当余数 r 为 0 时,此时的 n 就是两数的最大公约数,