

高职高专计算机系列教材

主编 谭浩强

面向对象语言程序设计

谭浩强 王智广 编著

清华大学出版社

编辑委员会

摇摇摇摇主摇摇任摇摇谭浩强

副 主 任摇摇焦金生 陈 明 丁桂芝

委 员摇摇(按姓氏笔画排序)：

王智广摇摇刘荫铭摇摇朱桂兰摇摇李文英

李 琳摇摇李志兴摇摇孙 慧摇摇武绍利

张 玲摇摇张克善 郝 玲 袁 玫

訾秀玲摇摇薛淑斌摇摇谢 琛

序

《高职高专计算机系列教材》

摇摇摇摇摇

到 21世纪,计算机将成为人类的常用现代工具,每一个有文化的人
摇摇**到** 都应当了解计算机,学会使用计算机,并用它来处理面临的事务。

学习计算机知识有两种不同的方法:一种是侧重知识的学习,从原理入手,注重理论和概念;另一种是侧重应用的学习,从实际入手,注重掌握其应用方法和技能。不同的人应根据其具体情况选择不同的学习方法。对大多数人来说,计算机是作为一种工具来使用的,主要以应用为目的,以应用为出发点。对于高职和高专的学生,显然应当采用后一种学习方法。

传统的理论课程采用以下的三部曲:提出概念——解释概念——举例说明,这适合前面第一种方法。对于侧重应用的学习者,我们在教学实践中摸索出新的三部曲:提出问题——介绍解决问题的方法——最后归纳出一般规律或概念。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。我们认为,后一种方法对高职、高专和成人高教是很合适的。

本系列教材是针对高职和高专的特点组织编写的,包括了高职高专的计算机专业和非计算机专业的教材和参考书。不同专业可以从中选择所需的部分。本系列教材包含的内容比较广,除了可作为正式教材外,还可作为某些专业的选修课或指定自学的教材。

应当指出,检查学习好坏的标准,不是“知道不知道”,而是“会不会用”,学习的目的全在于应用。因此,希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有一些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而应指定学生通过上机掌握。这样做可以培养学生的自学能力,启发学生的求知欲望。

本系列教材是由“浩强创作室”组织北京和天津一些普通高校和高职大学的老师们编写的,他们对高职高专的教学特点有较多的了解,有较多的实践经验。相信本系列教材的出版会有助于高职高专的教材建设和教学改革。

由于我国的高职教育正在蓬勃发展,许多问题有待深入讨论,新的经验将会层出不穷,对如何进行高职教育将会有更新更深入的认识,本系列教材的内容也将会不断丰富和调整。我们只是为了满足许多高职高专学校对教材的急需,才下决心抓紧编写了这套系列教材,以期抛砖引玉。清华大学出版社克服了许多困难,使本系列教材在较短的时间内得以出版。

本系列教材肯定会有不足之处,请专家和读者不吝指正。

《高职高专计算机系列教材》主编
全国高等院校计算机基础教育研究会理事长

谭浩强

1999年 10月 1日

前言

摇摇摇摇摇

现代计算机技术正在以惊人的速度发展着。在计算机的各种应用中,程序设计是至关重要的。程序设计语言的选择随着不同的工作领域、不同基础的人员而有所不同。BASIC语言是一种国内外广泛流行的计算机高级语言,它易于学习,易于掌握,又有实际使用价值。尤其是对于广大的初学者,BASIC是最适合的选择。事实已经证明,很多人以学习BASIC为起点,不断提高,迈入了计算机的应用领域。BASIC语言在计算机普及和计算机应用中起了重要的作用,并将继续发挥其作用。

BASIC是第三代的BASIC语言,同时它也是完全结构化的BASIC语言。它与原有的一些BASIC版本(如QBASIC、GW-BASIC等)相比,在功能上进行了许多扩充,主要扩充内容有:扩充了变量和常量的类型,除了保留原有BASIC的数据类型外,还增加了长整型、定长字符串型变量;可以定义数值常量和字符串常量,使数据类型更加丰富;还提供了块级结构,实现了多分支选择结构的CASE语句;提供了灵活方便的各种形式的循环语句,为程序设计带来了极大的方便;提供了将子程序和函数作为单独的程序模块,不需要行号,采用交互式的编辑环境,界面友好;提供了“分步执行”、“跟踪”等调试工具;采用较为先进的解释执行方式;提供了全局变量与局部变量;可以实现递归,有较强的作图功能;可以使用键盘和鼠标;并且提供了功能丰富的联机“帮助”。因此,BASIC是QBASIC和GW-BASIC的比较理想的更新版本。BASIC是没有程序设计经验的人员学习程序设计的首选语言。出于与QBASIC和GW-BASIC等版本的BASIC兼容的考虑,本书保留了QBASIC和GW-BASIC中的某些语句,如GOTO语句等。

BASIC既从其他高级语言中吸取了现代语言的特点,又很好地保持了BASIC语言的一贯特点——为用户着想,容易学习和使用。BASIC是学习用BASIC编程和开发应用程序的有力工具。

本书是专门为高职高专学生编写的教材,全书共包含15章、10个综合练习及10个附录。第1章介绍了程序设计的基础知识。第10章介绍了BASIC

语言运行环境。第 猿章介绍了 四粤税源程序的基本结构、基本数据类型等内容。第 源章至第 员章讲述了 四粤税语言的程序设计语句、子程序、数组类型、字符串的有关函数以及屏幕作图和文件等内容。最后给出了涉及全书所学内容的综合练习。

在本书编写的过程中得到了石油大学(北京)陈明教授的关心和帮助。陈明教授审阅了全书。石油大学(北京)计算机系的赵建辉老师和赵鹏翔老师参加了本书的部分工作。

由于作者水平有限,加上时间紧迫,书中肯定存在不少缺点和错误,敬请读者批评指正。

作者

四粤年 源月于北京

目录

第 1 章 程序设计的基础知识	1
1.1 计算机语言	1
1.1.1 低级语言	1
1.1.2 高级语言	2
1.1.3 BASIC 语言的发展和 QBASIC 语言的特点	4
1.2 算法的概念	6
1.3 常用的算法描述工具	7
1.3.1 用自然语言表示算法	7
1.3.2 用传统流程图表示算法	7
1.3.3 用 N-S 流程图表示算法	9
1.3.4 算法举例	12
1.4 结构化程序设计方法	15
1.5 程序设计的任务	18
习题	19
第 2 章 QBASIC 程序开发环境初步	20
2.1 QBASIC 的开发环境	20
2.1.1 如何进入 QBASIC 环境	20
2.1.2 QBASIC 的工作窗口	21
2.1.3 QBASIC 环境的其他成分	22
2.2 QBASIC 程序的编辑和运行	24
2.2.1 从键盘输入 QBASIC 源程序	24
2.2.2 QBASIC 程序的运行	25
2.2.3 修改和编辑源程序	26
2.2.4 退出 QBASIC	30

习题	31
▶ 第3章 QBASIC 语言基础知识	34
3.1 概述	34
3.2 QBASIC 源程序的结构	35
3.3 常量	37
3.3.1 数值常量	37
3.3.2 字符串常量	40
3.3.3 符号常量	40
3.4 变量	41
3.4.1 变量名和变量的值	41
3.4.2 变量的类型	42
3.5 标准函数	45
3.6 算术运算符和算术表达式	46
3.6.1 算术运算符	47
3.6.2 算术表达式	47
3.6.3 不同类型数据的混合运算	48
3.7 编写并运行一个简单的 QBASIC 程序	49
3.7.1 上机前的准备工作	49
3.7.2 编写并向计算机输入一个新程序	50
3.7.3 将程序以文件的形式保存在磁盘上	51
3.7.4 打开已存盘的文件	52
习题	53
▶ 第4章 顺序结构程序设计	55
4.1 数据输出语句	55
4.1.1 PRINT 语句的使用方法	55
4.1.2 PRINT 语句的输出格式	57
4.1.3 LPRINT 语句	62
4.2 LET 语句	63
4.3 INPUT 语句	65
4.4 READ 语句和 DATA 语句	68
4.5 RESTORE 语句	70
4.6 END 语句和 STOP 语句	73
4.6.1 END 语句	73

4.6.2	STOP 语句	73
4.7	单步执行和设置断点	74
4.7.1	功能键提示行	74
4.7.2	单步执行	74
4.7.3	设置断点	75
4.8	REM 语句	76
4.9	SWAP 语句	77
	习题	78

▶ 第5章 选择结构程序设计

5.1	概述	81
5.2	选择条件的描述方法	81
5.2.1	关系运算符和关系表达式	81
5.2.2	逻辑运算符和逻辑表达式	82
5.3	单行 IF 语句和块 IF 语句	84
5.3.1	单行 IF 语句	84
5.3.2	块 IF 语句	86
5.3.3	块 IF 语句的应用举例	87
5.3.4	块 IF 语句的嵌套	90
5.3.5	使用包含 ELSEIF 子句的块 IF	92
5.4	多分支选择结构的实现——SELECT CASE 语句	94
5.4.1	最基本的 SELECT CASE 结构	94
5.4.2	在 CASE 子句中使用“TO”指定一个连续值的范围	97
5.4.3	在 CASE 子句中使用“IS”指定关系条件	98
5.4.4	在 CASE 子句中使用多个条件	99
5.4.5	SELECT CASE 结构的一般格式及其说明	100
5.4.6	SELECT CASE 和 IF - THEN - ELSE 语句的比较	103
	习题	104

▶ 第6章 循环结构程序设计

6.1	概述	106
6.2	WHILE - WEND 循环结构	106
6.2.1	WHILE - WEND 循环结构及其执行过程	106
6.2.2	WHILE - WEND 循环程序举例	107
6.3	FOR - NEXT 循环结构	111

6.3.1	FOR – NEXT 循环结构	111
6.3.2	FOR – NEXT 循环的执行过程	112
6.3.3	EXIT FOR 语句	114
6.3.4	FOR – NEXT 循环程序举例	115
6.4	DO – LOOP 循环结构	118
6.4.1	DO – LOOP 循环结构格式及其执行过程	118
6.4.2	最简格式的 DO 循环(与 EXIT DO 配合)	118
6.4.3	带 WHILE 子句的 DO 循环	120
6.4.4	带 UNTIL 子句的 DO 循环	122
6.5	循环的嵌套	125
6.6	程序举例	128
	习题	136

▶ 第 7 章 函数与子程序

7.1	概述	139
7.2	自定义函数	140
7.2.1	单行自定义函数	140
7.2.2	多行自定义函数	143
7.3	块内子程序	144
7.3.1	GOSUB – RETURN 语句	144
7.3.2	ON GOSUB – RETURN 语句	146
7.3.3	ON KEY(n) GOSUB – RETURN 语句	147
7.4	模块子程序	148
7.4.1	建立一个模块子程序	148
7.4.2	QBASIC 环境下的子程序操作	151
7.4.3	带参数的子程序	152
7.4.4	调用子程序时的参数传递	154
7.4.5	程序举例	157
7.5	模块化的函数	159
7.5.1	模块化函数的建立及在 QBASIC 环境下的操作方法	159
7.5.2	函数的调用及参数传递	160
7.5.3	STATIC 选项	162
7.5.4	程序举例	163
7.6	局部变量与全局变量	164
7.6.1	局部变量	164

7.6.2 全局变量	165
7.7 函数的嵌套调用及递归调用	166
7.7.1 函数的嵌套调用	166
7.7.2 函数的递归调用	167
7.8 在 QBASIC 环境中同时观察两个模块	171
习题	172

▶ 第 8 章 数组

8.1 数组和数组元素的概念	175
8.2 一维数组	176
8.2.1 一维数组的建立和数组元素的引用	176
8.2.2 一维数组应用举例	180
8.3 二维数组	189
8.3.1 二维数组的建立和数组元素的引用	189
8.3.2 二维数组应用举例	191
8.4 静态数组和动态数组	197
8.5 数组名作为函数或子程序的参数	200
习题	204

▶ 第 9 章 字符串处理

9.1 概述	208
9.2 字符串常量和字符串变量	209
9.2.1 字符串常量	209
9.2.2 字符串变量	210
9.3 字符串变量的赋值	212
9.3.1 使用 LET 语句赋值	212
9.3.2 使用 INPUT 语句赋值	212
9.3.3 使用 LINE INPUT 语句赋值	213
9.3.4 用 READ/DATA 语句给字符串变量赋值	214
9.4 字符串的连接	215
9.5 字符串的比较	215
9.5.1 字符串比较的依据	215
9.5.2 字符串比较的规则	216
9.5.3 字符关系表达式	216
9.6 字符串数组	218

9.7	有关字符串处理的函数	221
9.7.1	子字符串处理函数	221
9.7.2	求字符串长度的函数	226
9.7.3	字符串与数值间的转换函数	226
9.7.4	字符与 ASCII 码间的转换函数	228
9.7.5	大小写字母之间的转换	228
9.7.6	建立由相同字符组成的字符串函数	229
9.7.7	日期和时间函数	230
9.7.8	INKEY \$ 函数	231
9.8	字符串处理应用举例	231
	习题	237

▶ 第 10 章 屏幕控制和简单作图

10.1	显示模式	240
10.1.1	文本模式	240
10.1.2	图形模式	240
10.2	屏幕控制语句及函数	241
10.2.1	CLS 语句	241
10.2.2	WIDTH 语句	241
10.2.3	LOCATE 语句	241
10.2.4	CSRLIN 和 POS 函数	242
10.2.5	SCREEN 语句	243
10.2.6	色彩设置语句 COLOR	244
10.3	简单作图语句	246
10.3.1	画点语句	246
10.3.2	画线	248
10.3.3	画圆、椭圆和圆弧语句	249
10.4	图形着色	252
10.5	图形窗口	253
10.5.1	VIEW 语句	253
10.5.2	WINDOW 语句	254
10.6	程序举例	255
	习题	259

▶ 第 11 章 文件

11.1	文件的概念	261
11.1.1	文件的分类.....	261
11.1.2	数据文件的组成	262
11.1.3	文件名	262
11.1.4	文件的读写和文件缓冲区	263
11.1.5	文件指针	264
11.2	顺序文件.....	264
11.2.1	顺序文件的特点	264
11.2.2	顺序文件的打开与关闭	264
11.2.3	顺序文件的写操作	266
11.2.4	顺序文件的读操作	270
11.2.5	顺序文件的修改	272
11.3	随机文件.....	275
11.3.1	记录型变量	275
11.3.2	随机文件的打开与关闭	277
11.3.3	用于记录变量的读写语句	277
11.3.4	记录变量应用举例	278
11.4	QBASIC 处理文件与目录的语句	281
	习题	283
	综合练习	286
	附录 A 字符与 ASCII 代码对照表	288
	附录 B QBASIC 保留字	289
	附录 C QBASIC 语句一览表	291
	附录 D QBASIC 函数一览表	296
	参考文献	299

第1章

程序设计的基础知识

1.1 计算机语言

我们知道,语言分为自然语言和人工语言两大类。自然语言是人类在自身发展过程中形成的语言(如中文、英文等),是人与人之间传递信息的媒介;人工语言则是为了某种目的而自行设计的语言。计算机语言是用于人与计算机之间通信的语言,是人工语言的一种形式。

人们要使用计算机,使计算机按人们的意志进行工作,就必须使计算机能理解和执行人们赋予它的指令。这就需要找到一种人和计算机都能识别的语言。

1.2 低级语言

1.2.1 机器语言

计算机是不能识别与执行人类的自然语言的。在计算机内部,任何信息和指令都是采用二进制(“0”和“1”)方式编码表示的。计算机只能接受和识别“0”和“1”这样的二进制信息。每一类型的计算机都分别规定了由若干个二进位的信息(即若干个“0”或“1”组成的信息)组成一条指令。例如,某种计算机规定以 00010001 这样的编码作为“加法”指令,让计算机执行一次加法。以 00010000 作为“减法”指令,让计算机执行一次减法操作。以上两条指令只是第 7 位和第 8 位不同(带阴影的两位,设最右边的一位为第 0 位)。可以看出,4 个 0 和 1 最多可以组成 16 个不同的指令或信息。

这种计算机能直接识别和执行的二进制形式的指令称为“机器指令”。例如,前面介绍的 00010001 和 00010000 就是两条机器指令。一条机器指令产生一个相应的机器操作。每一种计算机都确定有若干种指令(如加法指令、减法指令、传送指令、取数指令、存数指令、输入及输出指令等)以实现不同的操作。一种计算机的指令集合称为该计算机的机器语言,或者说该计算机的指令系统。这就如同用算盘算题一样,每一条珠算口诀就是一条“指令”,算盘全部口诀之和就是“珠算语言”。也就是说,“语言”是全部指令的总和。人们为了解决某一问题,可以从该“语言”中选择所需的指令,组成一个指令序列。这个指令序列称为“机器语言程序”。

不同的计算机系统的电路逻辑是不同的,因此,对不同的计算机,即使是执行同一种操作(例如在两类不同的计算机上都执行一次加法操作),它们的指令是不同的。或者说,不同的计算机有不同的指令系统。譬如有的计算机指令的长度为 16 位,有的计算机则为 8 位,也就是说,机器语言是依赖于具体计算机的(而不是各类计算机都通用的),它是“面向机器”的语言。

用机器语言编写出的程序,计算机能直接识别和执行,执行效率比较高。但人们要直接用“0”和“1”这样的二进制代码编程序,实在是难学、难记、难写、难检查、难调试、自然就难以推广。只是在计算机产生初期,计算机专门人员才用机器语言编写程序。

汇编语言

由于用机器语言编写的程序存在上述缺点,我们可以用一些符号(一般是英文单词或英文单词的缩写)作为助记符代替机器语言中的各种二进制的机器指令,形成一种新的程序设计语言,称为汇编语言。汇编语言在一定程度上克服了机器语言存在的缺点。用汇编语言编写的程序称为汇编语言源程序。计算机是不能直接识别和执行汇编语言源程序的,在执行前必须将它翻译成计算机能够直接识别和运行的二进制指令形式的目标程序,如图 1-1 所示。完成这种翻译工作的程序叫做汇编程序,这个翻译过程叫做汇编。

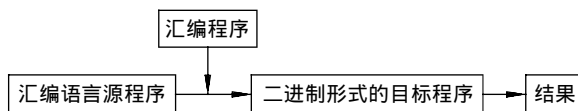


图 1-1 汇编

高级语言

用汇编语言编写的程序虽然比用机器语言编写的程序方便了很多,但它仍然存在难以编写、出错之后难以检查和修改的问题。同时,汇编语言仍然要依赖具体的计算机。为了解决汇编语言的这些缺陷,人们创造了一种各类计算机都通用的、接近于人类“自然语言”和“数学语言”的程序设计语言。譬如写出如下一条指令:

PRINT SIN(30) + COS(30)

“PRINT”是一个英文单词,意思是“打印”。“SIN(30)”是一个数学表达式,它的数学含义是分别计算 30 度的正弦值和 30 度的余弦值,把它们相加之后再加 PRINT。以上是一条接近自然语言(英文)和数学语言的指令。如果计算机能执行这样的指令,将为使用者提供极大的方便。

这种人工创造的语言称为“高级语言”,机器语言称为“低级语言”。所谓“低级”,指它贴近机器,而“高级”是指它更贴近人的语言。高级语言是各种计算机都通用的。

世界上最早出现的高级语言是 FORTRAN 语言(1957 年诞生,1958 年正式使用)。它是一种用于数值计算的高级语言。它规定了可以使用的几十个英文单词以及它们对应的操作,并且规定了必须遵循的语法。很显然,计算机不能识别这种人们自己创造的语言,需要有一个“翻译”,把用高级语言编写的程序翻译成用二进制形式表示的机器语言程序

(即由若干条机器指令组成的指令序列),如图 1.1 所示。这个“翻译”工作不是由人来完成的,而是用一个计算机软件来实现的。人们在“创造”一种语言的同时,必须设计出这样一个翻译系统。在运行高级语言程序之前,先将这个翻译系统调入计算机内,然后让它工作,它会将高级语言程序逐条翻译成机器语言程序,然后计算机执行机器语言程序。一般将翻译的方式分为编译方式和解释方式。

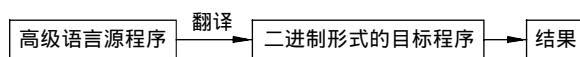


图 1.1 翻译过程

编译方式是把整个高级语言源程序一起翻译成一个目标程序,用一个称为“编译程序”的软件进行编译工作。如果发现语法有错,则不生成目标程序。经用户修改后,再次进行编译,直到不出现语法错误为止,此时生成目标程序。解释方式也是将高级语言源程序翻译为机器指令,但它与编译方式不同,不是把整个高级语言源程序一起翻译成一个目标程序。而是翻译一句,执行一句,不产生整个的目标程序。如果程序没有错误,则一直进行到全部执行完毕,如果在运行过程中发现程序有错,则立即中止“解释”工作,需要修改程序后再重新运行。

解释方式使用灵活方便,占内存较少,但占机器时间多,效率较低。编译方式得到的目标程序经过优化,执行效率高,但占内存多。Pascal 采用的是解释方式。

用高级语言直接编写的程序称为“源程序”,翻译后得到的机器语言程序称为“目标程序”。计算机执行的是目标程序,运行后可得到结果。

目前,国内外使用的高级语言种类很多,应用范围各不相同。比较常用和流行的高级语言主要有以下几种:

- FORTRAN 适用于数值计算
- COBOL 适用于商业和管理领域
- PL/I 最早出现的结构化语言,适于教学使用
- ALGOL 适用于编写系统软件
- BASIC 一种易学易用又有实际使用价值的语言,特别适合于初学者
- LISP 用于人工智能领域

高级语言出现后,人们都能很快地学会使用计算机,而可以完全不顾什么机器指令,也可以不必深入懂得计算机的内部结构和工作原理,就能方便地使用计算机进行各种科学计算或事务管理。可以说,高级语言的出现是计算机发展史中“惊人的成就”。

使用高级语言编写程序还有一个很大的优点,就是它可以适用于不同的计算机,或者说,对不同的计算机具有通用性。用某一种高级语言编写的源程序几乎可以不加修改就能使用在另一类型的计算机上。这就给使用者带来很大的方便。应当指出:即使是同一种高级语言,对不同型号的计算机来说,它所使用的编译系统是不同的。正如把同一篇中文翻译为英文和翻译为法文需要不同的翻译一样。但是这个问题用户不用考虑,在计算机出厂时,已经将该机器所使用的各种语言的编译程序(或解释程序)存储在磁盘上作为计算机系统的软件提供给用户了。