

新世纪

计算机基础教育丛书

丛书主编

谭浩强

QBASIC

程序设计

谭浩强 主编
张基温 编著
薛淑斌

谭浩强
马素霞



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

QBASIC
程序设计

谭浩强

主编

清华大学出版社



新世纪
计算机基础教育丛书

丛书主编

谭浩强

QBASIC 程序设计

主 编

谭浩强

编 著

谭浩强 张基温

马素霞 黄淑斌



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

(京)新登字 158 号

内 容 简 介

QBASIC 是近年来在国内外得到广泛推广和使用的、完全结构化和模块化的计算机高级语言,是没有程序设计经验的人学习程序设计的最佳选择。教育部考试中心举办的“全国计算机等级考试”和“全国计算机应用技术证书考试(NIT)”均已将 QBASIC 列入考试内容。学习 QBASIC 程序设计可为读者将来进行程序开发工作打下初步基础。

本书是在《True BASIC 程序设计》(清华大学出版社出版)一书成功的基础上编写的。除了介绍 QBASIC 语法外,还较详细地介绍了算法设计的知识。本书可作为高等学校非计算机专业(尤其是理工科)学生学习程序设计的正式教材,也可作为准备参加“全国计算机等级考试(二级)”和“全国计算机应用技术证书考试(NIT)”程序设计模块的培训式自学教材,并可供广大初学者自学参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: QBASIC 程序设计

作 者: 谭浩强 主编

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市丰华印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 20 字数: 473 千字

版 次: 2000 年 1 月第 1 版 2001 年 9 月第 8 次印刷

书 号: ISBN 7-302-03695-0/TP·2095

印 数: 61001~76000

定 价: 23.50 元

21世纪终于来临了,在新的世纪,人们自然对未来有许多美好的愿望和设想。现代科学技术的飞速发展,改变了世界,也改变了人类的生活。作为新世纪的大学生,应当站在时代发展的前列,掌握现代科学技术知识,调整自己的知识结构和能力结构,以适应社会发展的要求。新世纪需要具有丰富现代科学知识、能够独立解决面临任务、充满活力、有创新意识的人才。

掌握计算机知识和应用无疑是培养人才的一个重要环节。计算机既是现代科学技术的结晶,又是大众化的工具。学习计算机知识不仅是为了掌握一种技能,更重要的是:它能启发人们对先进科技的向往,激发创新意识,推动对新知识的学习,培养自学能力,锻炼动手实践的本领。因而它是高等学校全面素质教育中极为重要的一部分。

自20世纪80年代初以来,高等学校中计算机教育(尤其是非计算机专业中的计算机教育)发展迅速,计算机教育的内容不断扩展,程度不断提高,它所起的作用也愈来愈显著。

在实践中,大家已认识到,计算机应用人才队伍是由两部分人组成的:一部分是计算机专业出身的计算机专业人才,他们是计算机应用人才队伍中的骨干力量;另一部分是各行各业中应用计算机的人员。这后一部分人一般并非从计算机专业毕业,他们人数众多,既熟悉自己所从事的专业,又掌握计算机的应用知识,善于用计算机作为工具去解决本领域中的任务。他们是计算机应用人才队伍中的基本力量。事实上,大部分应用软件都是由非计算机专业出身的计算机应用人员研制的。他们具有的这个优势是其他人难以代替的。从这个事实可以看到在非计算机专业中深入进行计算机教育的必要性。

非计算机专业中的计算机教育,无论目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同,决不应该照搬计算机专业的模式和做法。全国高等院校计算机基础教育研究会自1984年成立以来,始终不渝地探索高校计算机基础教育的特点和规律,在80年代中期,最早提出了按层次进行教育的方案。计算机应用是分层次的,不同的人在不同的层次上使用着计算机,同样,计算机教育也是分层次的,以适应不同应用层次的要求。全国有一千多所高等学校,好几百个专业,学校的类

型、条件和基础差别很大,不可能按同一模式、同一要求、同一内容进行教学。按层次组织教学,可以使不同专业、不同学校能够根据自己的情况选择教学内容,做到“各取所需”。

经过十多年的实践,几经调整,许多高校形成了按以下三个层次组织教学的方案:第一层次为计算机公共基础,学习计算机基本知识和基本操作;第二层次为计算机技术基础,内容包括程序设计、数据库、网络和多媒体等;第三层次为计算机应用课程,结合专业应用的需要学习有关计算机应用课程。每一层次中设立若干门课程,包括必修课和选修课。

1988年起,我们根据层次教学方案,组织编写了“计算机基础教育丛书”,邀请有丰富教学经验的专家学者先后编写了20多种教材,由清华大学出版社出版。丛书出版后,迅速受到广大高校师生的欢迎,对高等学校的计算机基础教育起了积极的推动作用。广大读者反映这套教材定位准确、内容丰富、通俗易懂,符合广大非计算机专业学生的特点。许多高校都采用了我们编写的教材。丛书总发行量达到700多万册,这在全国是罕见的。

在新世纪来临之际,我们在该丛书成功的基础上组织了这套“新世纪计算机基础教育丛书”,以适应新形势的要求。本丛书有以下特点:

(1) 内容新颖。根据新世纪的需要,重新确定丛书的内容,以符合计算机科学技术的发展和教学改革的要求。本丛书除保留了原丛书中经过实践考验、且深受群众欢迎的优秀教材外,还新编写了许多新的教材,在这些教材中反映了近年来迅速得到推广应用的一些计算机新技术,以后还将根据发展不断补充新的内容。

(2) 适合按层次组织教学的需要。在新世纪大多数学校是采用层次教学模式的,但不同的学校和专业所达到的层次不同,本丛书采用模块形式,提供了各种课程的教材,内容覆盖高校计算机基础教育的三个层次。既有供理工类专业用的,也有供文科和经济类专业用的;既有必修课的教材,也包括一些选修课的教材供选用。各类学校都可以从中选择到合适的教材。

(3) 符合大学非计算机专业学生的特点。本丛书针对非计算机专业学生的特点,以应用为目的,以应用为出发点,强调实用性。本丛书的作者都是长期在第一线从事高校计算机基础教育的教授和副教授,对学生的基础、特点和认识规律有深入的研究,在教学实践中积累了丰富的经验,可以说,每一本教材都是他们长期教学经验的总结。在教材的写法上,既注意概念的严谨和清晰,又特别注意采用读者容易理解的方法阐明看来深奥难懂的问题,做到例题丰富,通俗易懂,便于自学。这一点是本丛书一个十分重要的特点。书是写给读者看的,读者如果看不懂,只能算失败。

(4) 采用多样化的形式。除了文字教材这一基本形式外,有些教材还配有习题解答和上机指导,我们还准备采用现代教学方式,陆续制作电子出版物,以利于学生自学。

总之,本丛书的指导思想是:内容新颖、概念清晰、实用性强、通俗易懂、层次配套。简单概括为:“新颖、清晰、实用、通俗、配套”。我们经过多年实践形成的这一套行之有效的创作风格相信会受到广大读者欢迎。判别一本书的优劣,读者最有发言权。

本丛书多年来得到各方面人士的指导、支持和帮助,尤其是得到全国高等院校计算机基础教育研究会的各位专家和各高校的老师们的支持和帮助,我们在此表示由衷的感谢。

本丛书肯定有不足之处,竭诚希望得到广大读者的批评指正。

丛书主编

全国高等院校计算机基础教育研究会理事长

谭浩强

2000年1月1日

前言

程序设计是计算机应用人员的一项基本功,是高等学校理工科各专业的一门必修课。学习程序设计能使人们了解计算机是如何进行工作的,因为,计算机的一切工作都是由程序驱动的,只有学习了程序设计才会懂得计算机的“思维”方式和工作方式,不会对计算机的“神奇”功能感到神秘莫测。只有具备了程序设计的知识和能力,才能有效地进行软件的应用开发工作。

学习程序设计必须选择一种计算机语言。应当选择什么语言呢?有两种不同的看法。一种意见认为:应当选择当前最先进的,以便学生学了以后能直接用于程序的开发。另一种意见认为:在学校中学习程序设计的目的是进行程序设计的基本训练,高级语言种类繁多,而且在不断发展,不可能学了一种高级语言后就一劳永逸。无论学哪一种语言都应当能够举一反三,在需要时能很快地掌握其他所需的语言,因为许多计算机高级语言的规则是大同小异的。至于具体选择哪一种语言,要综合考虑各方面的因素,包括:该语言的功能和适用领域、学习的难易程度、当前设备条件、学习者的基础和具体使用要求、学时数多少以及师资条件等。其中有些因素是互相矛盾的,例如,有的语言功能很强,但很难掌握,对设备条件要求较高。实际上不可能有十全十美的方案,实用的方案往往是各种因素的折衷。对广大初学者来说,是否容易学习以及语言的功能能否满足学习要求,这两点是最重要的。作者同意上面的第二种看法。

其实,大学非计算机专业的学生在一门课程中学习的只能是程序设计的初步知识,不可能要求学生在学习本课程后就立即能编写出规模较大的、能供实际使用的、完整的程序。要做到这一步,还需要学习其他一些知识(例如,数据结构、软件工程、计算方法、操作系统等),但是学习本课程可为以后的学习打下初步的基础。在语言的选择上,不同类型不同条件的学校、不同的专业、不同的年级、不同基础的学生,可能会选择不同的语言,我们认为这是正常的。学习哪种语言不是什么大的原则问题,只要能学好、能用好就行。

BASIC 语言是广大初学者喜爱并容易学习的计算机高级语言。自从 1964 年 BASIC 语言问世以来,30 多年来长盛不衰,这不是偶然的。每一个事物的发生、发展和衰亡都有自身的规律,不以某些人的意志为转移。不论哪一种语言,如果它不符合人们的需要,停滞不前,落后于形势,就必

然会被淘汰。BASIC 语言产生后经历了四个发展阶段:第一代是早期的 BASIC,开始时只有 14 条语句。后来增加到 17 条语句,功能相对简单,主要供学生学习使用,被认为是基本的 BASIC。但是,BASIC 语言的“为大众着想”的指导思想在广大群众中倍受称赞,大得人心,为计算机的普及提供了正确的思路。第二代是 20 世纪 70 年代中期出现的微机 BASIC,它的功能有了很大的扩展。除了可供学习之外,还可用于解决一些小型的实际问题。如 TRS-80, APPLE BASIC, MS-BASIC (即 BASICA), GW-BASIC 等都属于第二代 BASIC。第三代是 80 年代初期和中期出现的结构化 BASIC。如 True BASIC, Turbo BASIC, Quick BASIC, QBASIC 等。第四代是图形界面的 BASIC,即 Visual BASIC,它是在 Windows 平台上使用的、面向对象的 BASIC 语言。事实证明 BASIC 语言是有广阔发展前景的。它既保持了易学好用的优点,又跟上了计算机技术发展的趋势,功能愈来愈强。

本书介绍的是第三代 BASIC 语言——QBASIC。它是一种完全结构化和模块化的高级语言,是非计算机专业学生学习程序设计比较理想的一种语言。学习 QBASIC 要比学习 C 语言容易得多,学习 QBASIC 也是学习 Visual BASIC 的必要准备。许多专家认为:QBASIC 是没有程序设计经验的人学习程序设计的最佳选择。在我国近年来学习 QBASIC 的人愈来愈多,教育部举办的全国计算机等级考试已将 QBASIC 定为二级的考试内容之一,全国计算机应用技术证书考试(NIT)也已将 QBASIC 作为培训和一个模块。随着计算机应用的普及,相信 QBASIC 将会在我国迅速推广。

1988 年我们曾编写过一本《True BASIC 程序设计》,由清华大学出版社出版,受到了广大读者的欢迎,至今已发行了近 30 万册。考虑到 BASIC 语言发展和应用的情况,我们决定在该书的基础上再编写一本《QBASIC 程序设计》,以满足广大读者的需要。本书除了介绍 QBASIC 的语法以外,还着重介绍算法设计,与作者编写的其他 QBASIC 书籍相比程度稍深一些,在算法设计方面叙述较为详细,可供基础较好的读者学习。原来使用《True BASIC 程序设计》作为教材的学校很容易过渡到使用本书作为教材。

应当强调,学习本课程必须十分重视上机练习,光靠听课和看书是难以掌握 QBASIC 程序设计的。

参加本书编写和改写工作的有:谭浩强教授、张基温教授、马素霞副教授、薛淑斌高级工程师。由于我们水平有限,本书也许有某些错误和不妥之处,敬请批评指正。

谭浩强

1999 年 5 月于北京

目 录

catalog catalog catalog catalog

1

程序设计的初步知识

1.1 用计算机解决实际问题的过程	1
1.2 计算机算法	2
1.2.1 算法的概念	2
1.2.2 算法举例	3
1.2.3 算法的表示	5
1.2.4 算法设计技术	13
1.3 程序设计语言	20
1.3.1 程序设计语言的发展	20
1.3.2 程序设计语言的功能	23
1.3.3 高级程序设计语言的使用	24
习题	25

2

QBASIC 程序设计初步

2.1 概述	27
2.1.1 QBASIC 简介	27
2.1.2 QBASIC 程序的组成	30
2.1.3 QBASIC 的使用环境	32
2.1.4 编辑和运行 QBASIC 程序	36
2.2 数据描述	41
2.2.1 数据类型	41
2.2.2 常量	41
2.2.3 变量	44
2.3 QBASIC 表达式	47
2.3.1 标准函数	47
2.3.2 算术运算符	48
2.3.3 算术表达式	48
2.3.4 不同类型数据的混合运算	49
2.4 数据传送	49

2.4.1	输出语句(PRINT)	50
2.4.2	赋值语句(LET).....	57
2.4.3	键盘输入语句(INPUT)	58
2.4.4	读数语句(READ)和置数语句(DATA)	60
2.4.5	恢复数据指针语句(RESTORE)	62
2.4.6	数据传送语句小结.....	63
2.5	结束语句(END)和暂停语句(STOP)	63
2.5.1	结束语句(END)	63
2.5.2	暂停语句(STOP)	64
2.6	分步执行和设置断点	64
2.6.1	分步执行.....	64
2.6.2	设置断点.....	65
	习题.....	66



选择结构程序设计

3.1	关系表达式和逻辑表达式	69
3.1.1	关系运算符和关系表达式.....	69
3.1.2	逻辑运算符和逻辑表达式.....	70
3.2	二路分支选择结构	71
3.2.1	行 IF 语句	71
3.2.2	块 IF 语句	74
3.2.3	ELSEIF 结构	79
3.3	多路分支选择结构	80
3.3.1	SELECT CASE 语句	81
3.3.2	多分支转移语句(ON GOTO)	87
3.3.3	用不同语句实现多分支选择结构的比较.....	88
	习题.....	89



循环结构程序设计

4.1	WHILE 循环结构	91
4.2	DO 循环结构	93
4.2.1	最简单的 DO 循环格式.....	93
4.2.2	用 EXIT DO 语句终止循环	94
4.2.3	带条件的 DO 循环.....	95

4.2.4 DO 循环的嵌套	100
4.3 FOR 循环结构	101
4.3.1 FOR 循环的一般格式	101
4.3.2 FOR 循环的执行过程	102
4.3.3 FOR 循环的嵌套	106
4.3.4 EXIT FOR 语句	110
4.4 程序设计举例	111
4.4.1 穷举	111
4.4.2 迭代	117
4.4.3 模拟	123
习题	131



数组与记录

5.1 一维数组	135
5.1.1 单下标变量与一维数组的概念	135
5.1.2 一维数组的建立与引用	135
5.1.3 一维数组的应用举例	139
5.2 二维数组	145
5.2.1 二维数组的定义与引用	145
5.2.2 二维数组的应用举例	146
5.3 多维数组	151
5.4 静态数组和动态数组	152
5.5 排序	155
5.5.1 选择排序	156
5.5.2 插入排序	159
5.5.3 交换排序	165
5.6 记录	168
5.6.1 记录的定义与使用	168
5.6.2 记录数组	170
习题	173



函数与子程序

6.1 模块化程序设计概述	178
6.2 函数	179

6.2.1	自定义函数(DEF)	179
6.2.2	外部函数(FUNCTION)	185
6.2.3	函数的嵌套与递归	194
6.3	块内子程序——子例程	197
6.3.1	GOSUB—RETURN 语句	197
6.3.2	ON GOSUB—RETURN 语句	198
6.3.3	ON KEY(n) GOSUB—RETURN 语句	200
6.4	独立模块的子程序	202
6.4.1	子程序的定义和调用	202
6.4.2	在 QBASIC 环境中有关子程序的操作	204
6.4.3	带参子程序——虚实结合	205
6.4.4	递归子程序与递归程序设计	211
6.5	全局变量与局部变量	221
6.5.1	局部变量	221
6.5.2	全局变量	223
6.6	程序的连接与数据传递	226
6.6.1	程序的连接	226
6.6.2	数据传递	226
	习题	228



字符串

7.1	概述	231
7.2	字符串常量	231
7.3	字符串变量	232
7.3.1	变长字符串变量	232
7.3.2	定长字符串变量	233
7.4	字符串传送	234
7.4.1	用 LET 语句传送字符串数据	234
7.4.2	用 READ/DATA 或 INPUT 语句传送字符串 常量	234
7.4.3	用 LINE INPUT 语句传送字符串常量	235
7.5	字符串的比较	236
7.5.1	字符比较的规则	236
7.5.2	字符串比较的规则	236
7.6	字符串数组	238

7.7 字符串的连接·····	239
7.8 子字符串·····	239
7.9 有关字符串的函数·····	243
7.9.1 测字符串长度的函数·····	243
7.9.2 字符串转换函数·····	245
7.9.3 字符串重复函数·····	250
7.9.4 日期和时间函数·····	251
7.9.5 暂停函数·····	252
习题·····	252



8 屏幕控制与图形

8.1 显示模式·····	254
8.1.1 显示模式·····	254
8.1.2 屏幕坐标系统·····	255
8.2 屏幕控制语句·····	255
8.2.1 LOCATE 语句·····	255
8.2.2 WIDTH 语句·····	255
8.2.3 CLS 语句·····	256
8.2.4 CSRLIN 和 POS 函数·····	256
8.2.5 SCREEN 语句·····	257
8.2.6 COLOR 语句·····	257
8.3 绘图语句·····	258
8.3.1 画点语句·····	258
8.3.2 画线语句·····	259
8.3.3 画圆、椭圆和圆弧语句·····	260
8.4 图形着色语句·····	264
8.5 综合练习·····	265
习题·····	267



9 数据文件

9.1 文件的概念·····	269
9.1.1 文件与记录·····	269
9.1.2 文件的分类·····	270
9.1.3 文件标识·····	271

9.1.4	文件缓冲区	273
9.1.5	文件位置指针	273
9.1.6	文件结束标志	274
9.2	顺序文件	274
9.2.1	顺序文件的打开和关闭	274
9.2.2	对顺序文件的输出(写文件)	276
9.2.3	从顺序文件输入(读文件)	280
9.2.4	对顺序文件的修改	282
9.2.5	与顺序文件有关的函数	284
9.3	随机文件	286
9.3.1	随机文件的打开与关闭	286
9.3.2	对随机文件的读写	287
9.3.3	按字节读写二进制文件	291
	习题	293
附录 A	基本 ASCII 码表	295
附录 B	QBASIC 保留字	297
附录 C	QBASIC 语句一览表	298
附录 D	QBASIC 函数一览表	302
附录 E	PRINT USING 语句的格式字符	304
参考文献		305

第1章 程序设计的初步知识

1.1 用计算机解决实际问题的过程

目前,电子计算机已广泛地应用于科学计算、企业管理、自动化过程控制等各个方面。无论用计算机来解决哪一方面的问题,我们都必须把对实际问题的解决归结为计算机能够执行的若干个步骤,然后再把这些步骤用一组计算机的指令进行描述,最后交给计算机来执行。这组解决实际问题的指令,就是通常所说的程序。要设计出解决实际问题的程序并不是一个简单的问题,人们要做大量的工作。具体可以归结为以下几个步骤,如图 1.1 所示。

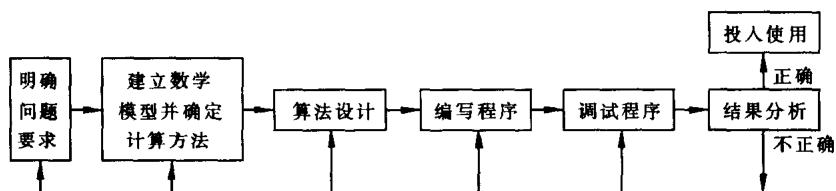


图 1.1 用计算机解决实际问题的过程

1. 明确问题要求

用计算机解决实际问题时,首先要对实际问题进行分析,明确问题的要求是什么,要求计算机做什么,已知一些什么样的数据,需要得到什么样的数据,还有其他哪些方面的要求等等。

分析问题、弄清问题的性质是用计算机解题的出发点。只有弄清问题,才能发现问题的特点和实质,以便采取最有效的方法来解决。

2. 建立数学模型和确定计算方法

为了用计算机来解决实际问题,必须设法用数学方法来描述或模拟实际问题。我们把对错综复杂的实际问题简化抽象,并用合理的数学公式来描述的过程,称为建立数学模型的过程。因此,数学模型是反映问题本质的数学表达式。

在建立数学模型的过程中,我们必须做到比较精确地知道待解问题的目标,明确问题的输入与输出,明确问题的约束限制条件,才能选择、制定好数学模型。建立数学模型是十分关键的一步,同时也是较为困难的一步。

为了利用计算机解决实际问题,除了建立数学模型外,有时还要借助于计算方法。所谓计算方法就是通过数值计算来求解复杂函数的方法。

3. 算法设计

在确定好数学模型及计算方法的基础上,设计出解题步骤(即算法),并用算法描述工

具将算法描述出来。有关算法及算法描述工具的内容见本章 1.2 节。

4. 编写程序

所谓程序即是用计算机语言描述的算法。编写程序的过程即是将用算法描述工具表示的算法转换成用计算机语言来描述。

5. 调试程序及结果分析

经过以上步骤得到的程序并不能保证其正确性,只有通过上机调试,才能比较彻底地发现程序中的语法错误及逻辑错误。即使程序调试通过,得到运行结果,仍不能说明程序是正确的,还要对运行结果进行认真分析,看看输出结果是否满足要求,以及程序所执行的功能是否与要求一致。如果发现有错误或偏差,则找出问题出现在什么地方。在调试程序之前的每一步骤中都有可能出现错误,具体是哪一步的错误,则需要具体分析。相比之下,在编程中出现的错误最易修改,而在程序完成之后,再来修改和纠正前面步骤中的错误,就很困难了,而且错误越是靠前,修改起来就越是困难,消耗的人力及时间也越多。因此,每一步骤都应尽力保证其正确性,错误发现得越早,越易于修改。

1.2 计算机算法

1.2.1 算法的概念

广义地说,做任何事情都需要按一定的过程进行。比如盖房子,需先打地基,后砌墙;看病需先挂号,再看病、开处方、划价、交钱、取药。这些过程都包括一系列基本操作。对于不同的问题及在不同的环境下,对基本操作有不同的定义。无论基本操作如何定义,都将解决问题所要进行的操作序列称为“算法”。如做菜的菜谱、治病的处方、生产流程等。

一般来说,一个算法应该具备以下几个特征:

(1) 有穷性:一个算法应当包括有限个操作步骤,或者说由有穷的操作序列组成。例如,创作一首小提琴曲谱可以称为小提琴算法,如果有人说,他设计了一首永远演不完的小提琴曲谱,则此曲谱不能称为算法。

不仅如此,有穷性还常常理解为实际上能够容忍的合理限度。例如,执行一个计算机算法需要让计算机运行 1000 年,便是不能容忍的。

(2) 确定性:算法中的每一步的含义都应该是清楚无误的,不能模棱两可,也就是说不应该存在歧义性(即可做两种以上不同的解释)。例如,“张三对李四说他的孩子考上了大学”,这句话就是“歧义性”的,因为可以理解为张三的孩子考上了大学,也可理解为李四的孩子考上了大学。因此在表示算法时要使用明确的文字或数字语言。

(3) 有效性:组成算法的每一个操作都应该是在特定环境下允许使用的,可以执行的,并且最后能得出确定的结果。只要算法中有一个操作是不可执行的,整个算法就不具有有效性。

(4) 有零个或多个输入:例如,要在一个“人事档案管理系统”中查找某一职工的情况,则需输入待查职工的编号或姓名。有的算法也可以没有输入。例如一盒音乐磁带,只要启动,就会放出音乐,而不需要有输入。

(5) 有一个或多个输出：算法的目的是求解，应该将“解”输出。对于无解的问题也应输出“无解”的信息。没有输出的算法是没有意义的。

基于算法的特征，还可以给出算法的另一种定义：算法是一个过程，这个过程由一套清楚的规则组成，这些规则指定了一个操作序列，以使用有限的步骤提供特定类型问题的解答。这里需要指出的是，并非所有的过程都能称为“算法”。过程可以不受“有穷性”的约束，即它的操作可以是无穷尽的。例如求所有自然数之和，它要执行无穷尽的加法，那么这个工作便是一个“过程”，而不能称为算法。

既然做任何事情都需要按一定的步骤进行，要用计算机解题，也要根据具体问题，制定解题步骤。由于计算机只能接受和理解计算机语言，因此，我们最终必须用计算机语言来描述算法，即解题步骤。用计算机语言描述的算法就称为程序。

一个程序由计算机能够执行的基本操作组成。计算机能够执行以下基本操作：

(1) 数据传送：将数据由输入设备或外存传到内存，将数据由内存送到输出设备或外存，将数据由内存中的一个单元传送到内存的另外一个单元等。

(2) 逻辑运算：与、或、非。

(3) 算术运算：加、减、乘、除、乘方等。

(4) 比较、判断和转移。

无论用计算机处理多么复杂的问题，都需要将解题步骤分解为计算机能够执行的基本操作，即需要用这些简单操作的组合去解决复杂的问题，这就迫使人们不得不花很大的力气去研究计算机算法。

通常把计算机处理的问题分为两大类：一类是“数值运算”，如解一个联立方程，求一个函数的定积分等；另一类是“非数值运算”，如用计算机进行文字处理、档案管理、情报检索、图书管理、辅助设计、辅助制造、过程控制等等。当今，计算机已深入到人类生产和生活的各个领域，在非数值运算领域的应用远远超过了数值运算领域。20世纪50年代初期，计算机在数值运算方面的广泛应用，使得研究计算方法的计算数学迅速发展并成熟。50年代中后期，计算机开始广泛应用于非数值运算领域。非数值运算的特点是处理对象大多数是“离散性”的数据，计算机在非数值运算领域的应用也促进了以研究离散对象为目标的组合算法的研究。

1.2.2 算法举例

下面通过几个实例来说明如何用计算机的基本操作来解决实际问题。

【例 1.1】 有两个变量 a 和 b ，要求将它们的值互换。例如： $a=3, b=4$ ；互换后 $a=4, b=3$ 。

为了进行两个变量的值互换，需引入一个辅助变量 c （正如两个瓶子里的液体互换需要用第三个瓶子作为过渡一样）。其算法可表示如下：

S1: $a \Rightarrow c$ （将变量 a 的值送给变量 c ）。

S2: $b \Rightarrow a$ （将变量 b 的值送给变量 a ）。

S3: $c \Rightarrow b$ （将变量 c 的值送给变量 b ）。

上面用 S1, S2, S3 表示步骤的次序，在写算法时常用这种形式的标记。S 是 step（步）