高等学校试用教材

电子计算机与算法语言

下　册

华南工学院高等数学教研室
电子计算机软件教研组合编

人民教育出版社

高等学校试用教材

电子计算机与算法语言

下　册

（算法语言部分）

华南工学院高等数学教研室
电子计算机软件教研组　合编

人民教育出版社

本书分上下册。上册为电子计算机部分，介绍电子计算机的基本原理。下册为算法语言部分，分别介绍 ALGOL 60、FORTRAN 和 BASIC 三种语言的基本知识。三种语言的前面有一个引言，简略介绍算法语言的产生和概况。

本书的电子计算机部分由钟沃坚执笔；ALGOL 60 和 BASIC 语言部分由郑威义执笔；FORTRAN 语言部分由邓自立、李锡权执笔。

高等学校试用教材

电子计算机与算法语言

下　册

（算法语言部分）

华南工学院高等数学教研室
电子计算机软件教研组合编

*

人民教育出版社出版

新华书店北京发行所发行

朝阳六六七厂印刷

*

开本 787×1092 1/32　印张 13 2/16　字数 310,000

1978 年 9 月第 1 版　1979 年 4 月第 1 次印刷

书号 15012·0100　　定价 1.10 元

目　录

下册　算法语言

第二篇 程序语言 FORTRAN

第三篇 BASIC 语言(130 机)

下册　算法语言

引言

从机器语言到算法语言

使用电子计算机解题，必须先编好程序。所谓程序，就是根据计算问题，由人事先安排好的计算步骤。把程序和计算问题中的原始数据，制作成穿孔纸带或穿孔卡片，然后输入计算机，计算机根据程序和原始数据，才能算出问题的结果。

既然要编程序，这就存在着用什么形式来表达程序的问题，说得更一般些，就是用什么"语言"来表达程序的问题。

我们从本书的上册"电子计算机"中已经看到，计算机进行计算的过程也就是执行一系列指令的过程。每种型号的计算机，都有自己规定的各种指令。指令的全体称为指令系统。每条指令规定一项操作，计算机执行一系列操作，就能完成一定的计算任务。计算机能接受并执行它自己的指令。指令可以看作是计算机的"语言"，这种语言称为机器语言。

在计算机问世的初期，人们在使用它来进行计算时，就是直接根据机器指令，亦即机器语言来编写程序的。用机器语言编写的程序，称为机器指令程序或手编程序。它由一条条指令组成。

我们从本书的上册"电子计算机"中还知道，计算机的每条指令，通常具有如下的基本格式：

操作码 θ	地址码 D

一般来说，这样一条指令的功能是：从存贮器的第D号地址中取出操作数，与运算器中累加寄存器的数进行操作码θ所规定的操作，操作结果保留在累加寄存器中。操作码和地址码都是一些由二进制数字表示的二进制代码。只是为了方便，一般用八进制数或十六进制数来书写。因此，可以想到，机器指令程序，必是由许许多多的二进制代码(用八进制数或十六进制数书写)所组成。

例如，看一个很简单的算式

$$F=a\times b+c$$

其中，已知a,b,c的具体数值(称为原始数据)，试求F的值(F称为结果变量)。

假设在某台计算机的指令系统中，与上述算式有关的几条指令的操作码如下：

操作种类	操作码	
	(二进制代码)	(八进制表示)
加法	0001	01
乘法	0011	03
取数	0101	05
送数	0110	06

现在，我们来编写计算上式的机器指令程序。

首先，应在计算机的内存贮器中，给程序中的常数和变量分配存贮单元，亦即给程序中的每一个常数和变量分别指定一个存贮单元的地址。这里，我们假定上式中的原始数据a,b,c分别分配在地址码为1001，1002，1003的三个存贮单元中；结果变量F分配在地址码为1010的存贮单元中。于是，用指令代码编写的机器指令程序如下：

05	1001
03	1002
01	1003
06	1010

显然，这些代码是很难记忆和掌握的。一个如此简单的算式，其程序就要这样几个代码。可以想象，一个复杂的问题，其程序的代码就更多更繁了。

机器指令程序还有其它缺点。编写这种程序是极其繁琐的工作，需要耗费大量的人力和时间，而其中很大一部分是机械的、重复的工作。这种程序很不直观，容易写错，写错了又不容易查出，查出了也难以修改。程序设计和检查错误所花费的时间往往比机器解题所需的时间多几十倍甚至几百倍。由于指令系统因机器不同而异，同一个题目，改用另一种型号的机器计算时，原有的程序就不适用了，必须重新编写。

因此，用机器指令代码编写程序的方式，大大阻碍了计算机的广泛应用。随着计算机的蓬勃发展和计算机计算速度的迅速提高，改变编写程序的方式显得尤为迫切。

这是五十年代初期面临的问题。

人们开始考虑，既然计算机能自动地、迅速地完成许许多多的计算工作，那么，编写程序的工作本身，是否也能部分地交给计算机去完成呢？这是程序设计自动化的思想。

人们很快就发现，可以用一些简单而又形象的符号来代表操作码和地址码，并把这些符号称为记忆码或符号语言。不直接用指令代码来编写程序，而改用符号语言来编写程序，这样的程序称为符号程序。

符号程序与机器指令程序不同。计算机并不认识符号程序中

的符号，因此，在符号程序输入计算机之前，必须先将符号程序中的符号(记忆码)，逐个代回计算机所能认识的代码，即把符号程序重新换成机器指令程序。这时计算机才能进行计算。粗略地说，将记忆符号换成指令代码的工作称为"代真"。只要有一张记忆符号与指令代码的对照表，代真完全是机械的工作。因此，只要先用机器语言为此编一个机器指令程序，并先交给计算机，则代真完全可由计算机来完成。指挥计算机完成代真工作的这个用机器指令编写的程序，称为汇编程序。符号程序通过汇编程序而生成机器指令程序，这个机器指令程序称为目标程序(或目的程序)。

符号语言也称为汇编语言。用汇编语言编写的符号程序也称为源程序。机器不能直接执行源程序。源程序通过汇编程序而生成目标程序。机器能直接执行目标程序。

仍以上述算式为例：

$$F=a\times b+c$$

现在我们来编写其符号程序，即源程序。

设在某型号计算机中，操作码与符号码对照如下：

操作码	符号码
01 (加法)	+
03 (乘法)	×
05 (取数)	←
06 (送数)	→

又程序中的变量 a，b，c 和 F 的地址码分别用它们本身的符号 a，b，c 和 F 来表示。于是计算上述算式的符号程序可编写如下：

←	a
×	b
+	c
→	F

显然，符号程序比机器指令程序直观一些了，也比较容易记忆了。此外，由于符号程序中的变量不是以其地址码而是以其记忆码(上例中是以其本身的符号)出现在程序中，因此，给这些变量分配存贮单元的工作实际上是通过汇编程序由计算机自动完成的。

然而，符号程序仍与数学公式(如例中的 $F=a\times b+c$)差别很大。符号程序仍依赖于特定的计算机。符号语言只不过是使程序设计人员摆脱计算机特有的一些细节（如代真和分配存贮单元)。它的自动化程度还很低。它与机器语言一样，都是为特定的机器服务的，所以称为面向机器的语言。

程序设计自动化必须继续发展。实践要求人们创造出自动化程度更高的语言。

五十年代中期，这样的语言终于相继提出来了。这就是算法语言以及其它各式各样的程序自动化语言。如 FORTRAN, ALGOL60, COBOL, BASIC, PL/1, ALGOL68, LISP, APL 等等。

算法语言·编译程序与解释程序

算法语言已经是较高级的程序自动化语言了。它既与数学公式非常接近，又与计算机的内部逻辑结构无关。算法语言使得程序设计人员可以不用考虑机器的内部逻辑结构，因而能集中精力去推敲解题方法的逻辑和计算过程的描述。

还是以上述算式为例:

$$F=a\times b+c$$

用算法语言来写，在 ALGOL60 中是

$$F:=a\times b+c$$

在 FORTRAN 中是

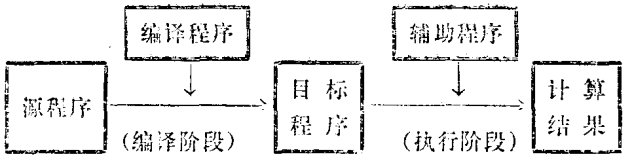
$$F=A*B+C$$

很明显，它与数学公式的形式几乎一致。

与汇编语言一样，用算法语言编写的程序也称为源程序。这种源程序也与用汇编语言写的源程序一样，机器是不能直接执行的，必须先将它换成用机器指令表示的程序即目标程序，计算机才能直接执行。

这里通常采用两种略为不同的做法。

一种做法是，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。当源程序输入计算机后，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。机器在其它一些辅助性程序的协助下执行目标程序，最后得出计算结果。编译程序和有关的辅助性程序合称为编译系统。编译系统如下图所示。



在编译程序对源程序进行编译的过程中，除了包括给源程序中的变量分配存贮单元和代真，从而最终形成目标程序以外；编译程序还对源程序进行语法检查，并提供出错的信息和修改错误的手段。

另一种做法是，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。当源程序输入计算机后，解释程序不是象编译程序那样，把源程序整个地翻译成目标程序以后再执行目标程序，而是对源程序进行边解释边执行。在这个过程中，同样包括语法检查、分配存贮单元和代真等工作。解释程序连同有关的一些辅助性程序合称为解释系统。

一般来说，解释程序的处理方法比编译程序的处理方法要多花费机器时间，但却能少用存贮空间。

使用算法语言实际上机算题的步骤大致如下：对写好了的源程序及其需要的原始数据，先用穿孔机把它们的每一个符号穿孔在纸带或卡片上(按照具体机器所规定的编码系统，每一个符号将对应一组不同的孔)。然后，用输入设备把穿孔纸带或卡片上的编码信息输进计算机。这时，对于编译系统来说，源程序经过编译生成目标程序，机器执行目标程序得出计算结果，最后把结果通过输出设备输出机外。对于解释系统来说，源程序经过解释和执行，得出计算结果，最后同样把结果通过输出设备输出机外。

但是，这是最顺利的情况。对源程序进行编译或解释以及在目标程序的执行过程中，常常会发现源程序中的错误。这时，机器通常会打印出错的信息。程序设计者根据这些出错信息，检查、修改源程序和纸带。然后重新上机计算，直至把题目算出为止。

对于不同型号的计算机以及不同的编译系统或解释系统来说，纸带的穿孔方法、上机操作方法、计算结果的输出格式以及出错信息的规定等等，通常是各不相同的。用户可从具体机器的使用手册中找到这些资料。

计算机程序语言概况

算法语言及其它各式各样的程序自动化语言，通常被称为计算机程序语言，简称计算机语言或程序语言。

目前，世界上的计算机语言已有几百种之多，而且还在继续发展。

这些语言，可以从不同角度加以分类。

从应用范围来看，有专门为进行各种数值计算而设计的语言，如 ALGOL60、FORTRAN 和我国自己设计的 BCY 语言等；有专门为进行数据处理和商业、企业管理而设计的语言，如 COBOL 语言等。这些语言适用于一类问题，可称为通用语言。

也有专门为处理某一特定问题或适应某一特定部门的需要而设计的语言，如数控语言APT、系统模拟语言MIMIC和SIMULA67以及诸如机翼设计语言、船体放样语言、建筑设计语言、地震勘探语言、银行报表语言、数理统计语言和力学结构语言，等等。这些语言可称为专用语言。

从使用方式来看，计算机语言又可分为交互式与非交互式语言。为了提高计算机的使用效率，克服计算机速度高与人在控制台上操作慢的矛盾，发展了分时终端设备。一台计算机可以有几个、几十个甚至几百个分时终端。许多人可以在各个分时终端上同时使用一台计算机。计算机把时间分成一小段一小段，轮流地用来执行各个人的程序。由于机器速度快，因此使用者一点也感觉不出有其他人在使用同一台计算机，就好象自己独占整台机器一样。这样，使用者就可以在终端设备上不慌不忙地演算他的题目，调试他的程序，甚至可以随时向机器提出问题，随时试一试某些计算方案，而让机器马上作出答复。反之，机器也可随时向使用者提出要求，让使用者现场回答。这样，就好象人和机器在"互相交谈"一样。为这种特殊应用方式设计的语言，称为交互式语言或会话式语言。如BASIC、APL和LISP等。

从语言功能不同的角度看，计算机语言也可分为面向过程的语言和面向问题的语言。可用于各种机器计算各种题目的语言(如ALGOL60、FORTRAN和BASIC等)，称为面向过程的语言。使用这种语言时，不仅要把计算的题目告诉计算机，而且要把题目的解法和计算过程描述出来。对于面向问题的语言，则只需把题目告诉计算机，就可得到所需的计算结果。这种语言如报表语言和判定语言等。

随着计算机应用领域的日益扩大，对计算机语言提出越来越多的要求，于是，在继续研制各种特殊用途的专用语言的同时，人

们又致力于建立"无所不包"的大型通用语言。这种语言有人把它叫做"公共汽车"语言。它企图兼收并蓄各种语言的功能和特点，达到"凡是你想要的，我这里都有"的效果。这种语言的代表作可推 PL/1 和 ALGOL68. 前者的构思原则是，吸收各种语言的特点，使之汇集在同一个基础语言中。因此，称为"汇集型语言"。后者的构思原则是，建立一个基础语言(核心语言)，并提供一种可扩充手段。使用者以基础语言为核心，利用提供的扩充手段，根据自己的需要，可拼凑起各种功能的语言。因此，称为"可扩充语言"。

计算机程序语言还在继续发展。可以预计，八十年代必将出现更加完善、更加适用的语言。

软　件

如果说，组成计算机的物质设备（如本书上册所介绍的运算器、存贮器、控制器和输入输出设备等），我们称它为计算机的硬件，那么，使用计算机和发挥计算机效率功能的各种程序，我们便称它为计算机的软件。计算机软件也称计算机程序系统。

大致来说，软件包括如下三大类：

一、面向用户的，包括语言加工系统(即语言及其编译程序、解释程序或汇编程序)、辅助系统(如调整程序、装配编辑程序等)、应用程序库和资料库。

二、面向管理人员的，包括诊断修复系统(如调机程序、诊断程序等)和日常事务管理系统(如运行记录、用户会计记录等)。

三、面向计算机本身的，包括故障处理系统、输入输出控制系统、管理程序和操作系统。

值得注意的是管理程序和操作系统。

管理程序是为提高计算机使用效率以及方便计算机用户而设计的一套程序。管理程序的主要功能是中断的处理、外部设备的

管理、多道程序的调度、编辑程序的调用、提供操作命令以及实现不停机的人机联系，等等。管理程序的进一步发展和扩充就是操作系统。

操作系统是为全面管理计算机资源、自动调度用户的作业程序从而使多个用户能有效地共用一套计算机装置的一套程序。这里，计算机资源是指中央处理机的时间、存贮空间、外部设备以及各种信息。操作系统具有组织整个计算机的工作流程、检查程序和机器的故障以及实现计算机网络的通讯等功能。一台在操作系统管理之下的计算机，可以同时或先后地处理用不同语言写成的成批的程序。这时，不仅一个题目的计算过程不需要人的干预而自动地进行，而且题目之间的调度和衔接也是自动进行的。在现代大型高速的计算机系统中，人的直接干预只能降低整个计算机系统的效率。算题人员将越来越少与计算机直接见面。对算题人员来说，计算机只不过是一个"窗口"，只须把穿好的源程序和原始数据纸带或卡片以及一份作业说明书（它本身往往也要穿成纸带或卡片）送进去，隔一段时间去取回结果就成了。操作系统为用户提供了方便的使用方式和令人满意的服务质量。

随着计算机硬件和软件的发展，六十年代初期终于形成了一门新的科学分支——计算机科学。

使用电子计算机解题的全过程

使用电子计算机解决实际问题，事实上包括提出和分析问题、建立数学模型、选择计算方法、编写和调试程序、准备原始数据、实际上机计算以及评价计算结果的全过程。

任何实际计算问题都是由一定的自然规律支配的运动过程，反映在运动中变动着的量与量之间有一定的数值关系。用一组数学公式来描述这个运动过程，并尽可能真实地反映出量与量之间

的数值关系，这一工作称为建立数学模型。例如，求半径为 r_1，r_2，…，r_{100} 的 100 个圆铁板的面积之和，可归结为下列数学公式的计算：

$$S=\pi\sum_{i=1}^{100}r_i^2$$

写出这个数学公式就是建立这个问题的数学模型。

数学模型建立了，但并不一定适于用计算机直接求解。因此，需要选择适于用计算机直接求解的计算方法。人们经过长期的实践，已经积累了大量计算各类问题的行之有效的方法。这些方法就是一般计算方法书籍中讨论的内容。

在动手编写程序之前，下功夫思考和分析自己的问题，整理计算公式，选择恰当的计算方法，是整个解题过程中最体现人的主观能动性的阶段。

用程序语言把计算过程描述出来的工作就是程序设计。程序设计只是上述全过程中的一个环节。程序编好之后应该细心审查，直到再也查不出错误以后才进行穿孔。通过穿孔纸带或卡片，把组成程序的一串文字符号变成纸带或卡片上的一串编码。只有编码形式的信息，计算机才可接受。程序执行时还可能要有原始数据的输入，这些原始数据也要按规定穿孔。

上机算题一般是先调试程序，调试通过后才正式计算。计算得出的结果是否正确，要细心分析。能否得出正确的结果，可能与下列因素有关，即数学模型是否正确？计算方法是否正确？程序是否正确？机器运行是否正确？

使用电子计算机解题的全过程通常是一个循环反复、渐趋正确的过程。