

结构化
程序设计语言

Turbo BASIC

曹培梁 董小国 编著

科学普及出版社

前 言

1964年,美国达特莫斯学院公布了 BASIC 语言的第一个版本。它是为初学计算机的人设计的会话式语言。由于它简单易学,受到了广大用户的欢迎。因此,在多数计算机系统中,尤其是在微型计算机上都配有 BASIC 语言,使它拥有最众多的用户,成为世界上最为流行、应用最为广泛的计算机语言。

但是,由于 BASIC 语言自身的一些弱点,以及许多 BASIC 版本往往互不兼容,却影响着 BASIC 语言的进一步应用和发展。特别是,程序设计越来越走向工程化,结构化程序设计方法的重要性已逐渐为广大程序设计人员所认识和接受,致使研制新的结构化的 BASIC 版本成为刻不容缓的任务。于是, True BASIC 和 Turbo BASIC 等结构化语言就相继应运而生。从此开创了 BASIC 语言的新纪元。

Turbo BASIC 是美国波朗德 (Borland) 公司于 1987 年推出的一种 BASIC 语言的编译系统。可以在 IBM-PC、长城 0520 及其兼容机上运行。因为它是向上发展,向下兼容,所以原有的 PC BASIC (GW BASIC) 程序一般情况可直接在 Turbo BASIC 中运行 (只要不涉及调用汇编语言子程序的工作)。它既保持了原有 BASIC 易学易用、适用面广的优点,又在功能上作了一些重大的扩充,使之成为一种结构化的程序设计语言。它支持 8087 协处理器;它可使用全内存;可处理二进制格式文件;它将编辑、编译、运行、追踪等工作融为一个整体;它的操作命令和环境参数均以菜单方式提供,它的屏幕可有四个窗口;它能生成 EXE 文件,可直接在 DOS 下运行。因此,它不仅使用方便,而且大大增强了处理复杂问题的能力,缩短了调试程序的时间,可以说它是 PC BASIC 的更新换代的产品,很值得向广大的 BASIC 用户推荐。

但是,目前市场上缺乏 Turbo BASIC 的实用教材,这给它的推广使用带来一定的困难。为此,我们编写了这本《结构化程序设计语言——Turbo BASIC》的教材,试图使初学计算机的人一开始就接受这个结构化的 BASIC 最新版本;同时,又为在广大的旧 BASIC 用户中推广使用 Turbo BASIC 提供了一本有实用价值的参考书。

该书本着由浅入深、循序渐进的原则,全面系统地介绍了 Turbo BASIC 语言。同时,为了帮助读者掌握编写程序的技巧和方法,书中配有大量的各种不同类型的例题和习题,以供读者参考使用,并在书后附有习题参考答案。(应当说明,例题与习题中的程序并不是唯一的,也并不是最佳最完善的程序,只是作为读者学习各章时的程序举例和练习时使用的。这些程序均已上机运行通过。)

本书可作为大专院校及计算机训练班学习 BASIC 语言的教材,也可作为其它部门的工程技术人员、管理干部以及中学教师学习 Turbo BASIC 语言的参考书。

全书共分 12 章,第一章简单概括地介绍了计算机的一般知识和 Turbo BASIC 的特点。第二至第十二章全面系统地介绍了 Turbo BASIC 语言及其上机操作方法,并于第五章第一节专门介绍了结构化程序设计的方法和结构化流程图的画法。书后配有附录,分别介绍了有关的技术资料,供读者参考使用。

本书第一至第六章及第十二章由曹培梁编写,第七至第十一章由董小国编写。由于时间仓促,水平有限,缺点错误在所难免,请专家和读者批评指正。

编 者

1989年5月

目 录

第一章 计算机的一般知识

第一节 计算机的硬件	(1)
第二节 计算机的软件	(2)
第三节 计算机的语言	(3)
第四节 Turbo BASIC 语言的概述	(4)
习题一	(6)

第二章 Turbo BASIC 语言基础

第一节 Turbo BASIC 程序的基本结构	(7)
第二节 Turbo BASIC 的字符集和保留字	(10)
第三节 常量、变量、数组	(11)
第四节 预定义函数	(17)
第五节 表达式和运算符	(18)
习题二	(22)

第三章 Turbo BASIC 的简单程序

第一节 赋值语句 (LET 语句)	(23)
第二节 输出语句 (PRINT 语句和 WRITE 语句)	(25)
第三节 键盘输入语句 (INPUT 语句)	(31)
第四节 读数语句 (READ 语句) 和置数语句 (DATA 语句)	(33)
第五节 恢复数据区语句 (RESTORE 语句)	(36)
第六节 三种提供数据语句的比较	(37)
第七节 其它几个简单语句	(38)
习题三	(39)

第四章 Turbo BASIC 的上机初步

第一节 Turbo BASIC 系统的安装与启动	(40)
第二节 Turbo BASIC 的工作屏幕	(41)
第三节 Turbo BASIC 程序的建立和运行	(43)
第四节 修改程序	(47)
第五节 使用汉字	(48)

习题四	(48)
-----------	------

第五章 控制结构

第一节 结构化程序设计与结构化流程图	(49)
第二节 选择结构	(52)
第三节 循环结构	(64)
习题五	(89)

第六章 函数与过程

第一节 函数	(92)
第二节 过程	(98)
第三节 关于函数和过程的进一步分析	(105)
习题六	(110)

第七章 字符串处理

第一节 字符串常数与字符串变量	(111)
第二节 字符串变量的赋值和输出	(112)
第三节 字符串表达式	(114)
第四节 字符串函数	(115)
习题七	(125)

第八章 格式输出

第一节 格式函数	(127)
第二节 格式语句	(128)
习题八	(135)

第九章 图 形

第一节 基础知识与 SCREEN 语句	(137)
第二节 文本方式	(139)
第三节 图形方式	(142)
习题九	(158)

第十章 文 件

第一节 文件概念	(159)
第二节 文件管理语句和函数	(160)
第三节 顺序文件	(163)
第四节 随机文件	(168)
第五节 二进制文件	(175)
习题十	(180)

第十一章 声音与音乐

第一节 扬声器发声语句	(181)
第二节 音乐语句	(182)
习题十一	(186)

第十二章 Turbo BASIC 的上机操作

第一节 Turbo BASIC 的菜单系统	(187)
第二节 Turbo BASIC 的编辑器	(197)
习题参考答案	(202)
附录 A 错误信息	(217)
附录 B 与汇编语句的接口	(226)
附录 C 事件捕获	(232)

第一章 关于计算机的一般知识

我们通常所说的计算机，是指通用电子数字计算机，以后简称为计算机。

计算机从 1946 年问世以来，其应用范围一直在迅速地扩大，如今已进入到人类生活的各个领域。历史上从未有过哪一种发明创造，象计算机这样，发展得如此迅速，应用得如此广泛。为了有效地使用计算机，应对它有一个初步的了解。为此，本章将粗浅地介绍一下关于计算机的一般知识。

第一节 计算机的硬件

计算机的硬件由控制器、运算器、存贮器和输入输出设备组成，其相互关系如图 1-1 所示。

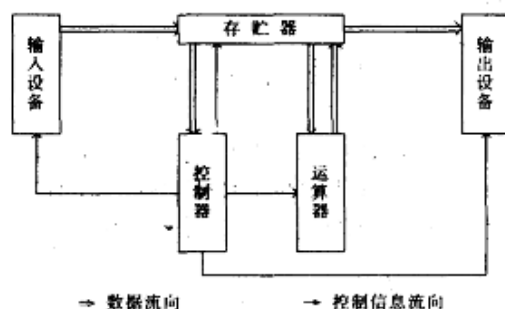


图 1-1 计算机的硬件组成

现对各部分简介如下：

一、控制器

用于控制并协调计算机各部分按人们预先制定的程序自动地进行工作。计算机的每一种基本操作称为一条“指令”，它是人对计算机发出的一道工作命令。控制器的作用是，依次从存贮器中取出指令，分析指令，然后发出适当的电信号，使计算机的有关部分执行相应的动作。

二、运算器

用于进行加、减、乘、除等算术运算及逻辑运算。当需要进行某种运算时，由控制器发出命令，将存放在内存贮器中的数据送到运算器中，在运算器中进行运算，然后将运算结果送回内存贮器中，以备后用或进行输出。

三、存贮器

用于存放程序和数据，是计算机的记忆装置。存贮器分内存贮器和外存贮器两种，前者存取速度快，一般存贮容量小，而后者存取速度慢，通常存贮容量很大。内存贮器是由许多存贮单元组成的，每个存贮单元包括若干个二进制位，每一个二进制位称为一个“比特” (bit)。多数计算机是以 8 个二进制位组成一个字节 (byte)，以一个或几个字节组成一个存贮单元，称为字 (word)，每一个字或字节都有一个编号，称为“地址”。当需要进行某种

运算时，控制器发出相应命令，指定从某个地址的存贮单元中取出数据来，在运算完成后，再将运算结果存入指定地址的内存单元中。

计算机的存贮容量，一般是以字节为单位。字节简称为“B”，所以1K字节简称为1KB， $1K = 1024$ 。而 1024×1024 个字节称为1M字节，简称为1MB（或称为1兆）。

从存贮器取出信息，该单元中原有信息不会破坏，只有向它送入新的信息后才取代了该单元中原有信息。

四、输入 / 输出设备

它是计算机和外界交流信息的设备。人们通过输入设备把数据或程序送入计算机的内存中，通过输出设备把计算机中某些信息输出出来。微机上常用的输入 / 输出设备有打印机、显示器、磁盘机等。

计算机的硬件主要由上述几部分组成。控制器、运算器和内存贮器通常总是装在一个机箱内，并且一起进行运算和操作，所以把它们合称为“主机”。而输入 / 输出设备和外存贮器则统称为“外部设备”。又因为控制器和运算器大都是单一的电子线路构成，特别是在由大规模集成电路组成的计算机中，控制器和运算器大都装配在十分紧密的结构中，甚至干脆组装在一起，同时，它们又是数据加工处理的核心部件，所以把它们合称为“中央处理器”，简称为CPU（Central Processing Unit）。

第二节 计算机的软件

对计算机来说，光有硬件是什么事情也干不了的。要用它解决各种问题，必须给它编制各种相应的程序。为了运行、管理和维护计算机所编制的各种程序的总和称为计算机的软件。开发软件的目的是为了便于用户更好地利用计算机解决各种各样的实际问题。计算机软件可分为两大类：系统软件和应用软件。

一、系统软件

由计算机设计、制造者提供的，为了使用和管理计算机的软件，统称为系统软件。系统软件包括：

1、操作系统

它控制所有在计算机上运行的程序，并管理计算机的全部资源，以便最大限度地发挥计算机系统各部分的作用。

2、各种语言的汇编、编译、解释程序。

3、监控程序、调试程序、故障检查和诊断程序。

4、程序库

它是各种标准子程序的总和。

5、数据库管理系统

计算机用于信息处理、情报检索、企业管理中时，常需处理大量的数据和各种表格。数据库管理系统的任务是对这些数据、表格进行合理的组织，迅速而广泛地检索、修改、打印输出。

二、应用软件

它是用户自己编制的解决各种实际问题的程序。应用软件还可以逐步标准化、模块化，组合成解决各种典型问题的软件包。

计算机的硬件建立了计算机的物质基础，而各种软件则扩大了计算机的功能，扩大了它的应用范围。现今所谓使用计算机，实际上就是要学会使用这些软件，通过这些软件去利用计算机解决各种实际问题。

第三节 计算机的语言

一、机器语言

计算机是按二进制原理设计的，因此它只能识别二进制数。由二进制数码 0, 1 的数字代码组成一套指令去控制计算机完成各种操作，这种二进制的指令是面向计算机的，因此得名机器指令。它是计算机最原始的指令，这些机器指令的集合称为机器语言。

一条机器指令用来控制计算机进行一种操作。如一条运算指令，告诉计算机应当进行什么运算，哪些数据参加运算，这些数据放在什么地方，运算结果又送到什么地方等等。不同的计算机有不同的指令种类和指令结构。一台计算机所拥有的指令种类和指令结构称为这台计算机的指令系统。在计算机问世的最初 10 年里，人们就是用这种与机器紧密相关的机器语言编写程序的。这样编写出来的程序直观性很差，不易阅读和修改，且编写工作繁琐枯燥，很易出错。同时，因为所编的程序对于机器有很大的依赖性，所以程序在不同机器上的通用性很差，这曾给程序移植带来不易克服的困难。

二、汇编语言

为了克服机器语言的读写困难，人们创造了汇编语言。它是一种符号语言，它的指令形式与机器指令一样，只是操作码和地址码采用一些便于人们理解和记忆的符号来表示。所以仍然属于一种面向机器的语言。与机器语言相比，虽然在直观性上有一些改进，但编写程序繁琐、枯燥、工作量大、容易出错、不易修改、没有通用性，仍然是它的致命弱点。而且用汇编语言编写的程序上机运行时，需经一个已存在机器内的汇编程序翻译成机器指令，程序才能运行。

三、高级语言

为了解决上述矛盾，在 50 年代中期，人们又开始研制另外一种语言，即高级语言。

高级语言是不依赖于具体计算机的程序设计语言。它的语句能确切地描述计算机处理问题的过程或算法，所以它是面向过程或算法的语言。它类似于人们习惯使用的自然语言和数字描述语言，因而程序设计非常方便。例如，对于

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

利用 BASIC 语言可写成

$$D = A + B * C$$

其形式与原数学式子非常接近。另外，由于多数高级语言都有国际统一标准，因此编写的程序可以不作修改或稍加修改即能在任何一台配有该种语言的计算机上运行，从而大大提高了程序的通用性。

高级语言问世后，发展极快。目前世界上已投入使用的高级语言有几百种之多，但常用的有以下几种：FORTRAN、ALGOL、COBOL、BASIC、PL/1、LISP、PASCAL 等。

四、高级语言的使用过程

计算机只能识别机器语言，所以用高级语言编写的程序必须经过“翻译”，使之变成机器语言程序，机器才能执行。通常把用高级语言或汇编语言编写的程序称为源程序，经“翻译”能被机器识别的程序称为目标程序。

承担“翻译”工作的程序因高级语言的结构不同而异。如 FORTRAN、COBOL、ALGOL、PASCAL 等为编译结构，承担“翻译”工作的程序称为编译程序，而把“翻译”过程称为编译。多数 BASIC 语言为解释结构，承担“翻译”工作的程序称为解释程序。

解释程序的“翻译”过程，象自然语言的口译，是边解释边执行，即“翻译”一句，执行一句，遇有错误立即停止。如图 1-2 所示。

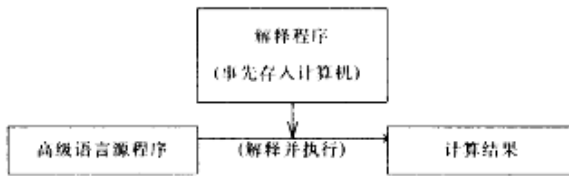


图 1-2 解释方式执行过程

编译程序的“翻译”过程，象自然语言的笔译，将一份源程序从头至尾“翻译”成机器语言，后再运行。如图 1-3 所示。

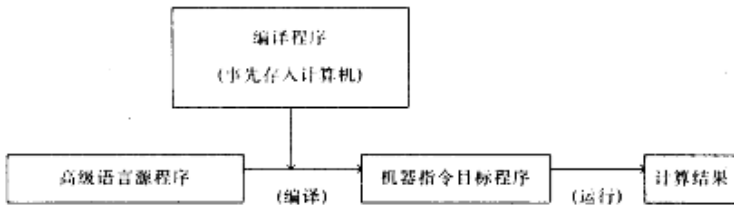


图 1-3 编译方式实现过程

解释方式占用内存较少，但执行速度较慢。而编译方式占用内存较多，但执行速度要快得多。

每一种高级语言都有与它相对应的“翻译”程序，所谓某机器上配有某种语言，就是指该计算机系统提供了这种语言的编译或解释程序。

第四节 Turbo BASIC 语言概述

我们将从四个方面简单地概括地介绍一下 Turbo BASIC 的特点。

1. 为结构化程序设计语言

随着计算机科学的发展，“结构化程序设计”的方法，因其结构清晰、易编制、易阅读和易维护，已普遍为软件工作者所接受。在较复杂的软件编制中，大量使用旧 BASIC 中的 GOTO、GOSUB、ON-GOTO 等转向语句，势必使程序的执行走向紊乱，结构缠绕不清。这是旧 BASIC 语言不适于编制较复杂软件的主要原因之一。而 Turbo BASIC 为了适应结构化程序设计的需要，在保留了以上语句的基础上，作了以下重要的扩充：

(1) 充实了循环结构和判断结构。在循环结构中，除了 FOR/NEXT 的定次数循环外，还可以用 DO/LOOP 的不定次数条件循环。WHILE 与 UNTIL 条件子句，既可以出现在 DO 语句上，又可以出现在 LOOP 语句上，十分灵活。在判断结构中，增加了 IF 块语句。即 IF-THEN-ELSE 可用于多语句块，且可以嵌套。此外，还有 SELECT CASE 语句可供多走

向的判断选择。

(2) 增加了多行函数和过程。旧 BASIC 的子程序是一种 GOSUB/RETURN 的功能块，用 GOSUB 语句调用某个子程序，在子程序中遇到 RETURN 语句时再返回到主程序去执行。这种子程序有两个缺点：

a、它的入口处与普通语句没有任何区别，很容易意外地“掉进”子程序去工作（即不是由 GOSUB 进入子程序）。

b、它的变量都是全局变量，子程序中的变量很容易和主程序中的变量混淆，很难用这种子程序设计出模块化的程序。

Turbo BASIC 的函数和过程克服了这种子程序的缺点，提供了真正的递归调用，参数传递，访问局部变量、静态变量和全局变量的能力。函数定义和过程定义可放在程序中的任何位置，决不会未经调用而意外地“掉进”去执行。因此，使用多行函数和过程很容易设计出模块化的程序。

(3) 行号可以省略。

以上这些新增加的功能有利于编写结构化、模块化的程序，有利于完成复杂的工作，它使得程序更容易编写，更容易调试，更容易阅读，从而也更容易维护。

2. 实用性好

(1) Turbo BASIC 是编译性语言，所以程序的运行速度快。

(2) 自动支持 8087 协处理器。同一个编译好的程序，既可以在有 8087 的机器上运行（速度快），又可以在无 8087 的机器上运行。当在有 8087 的机器上运行时，对浮点数据的处理速度 Turbo BASIC 大大地快于 PC 编译 BASIC。

(3) 调试程序快。它将程序的编辑、编译、运行和追踪等工作融为一个整体，均在内存中进行，各种工作之间的转换非常方便。运行的总是编译后的程序，大大缩短了调试程序的时间。

(4) 可充分利用计算机的全部内存。旧 BASIC 版本只能使用 64KB 内存，不能扩充，因此在应用上受到限制，不能运行大的程序。而现在微机内存容量增加很多，如 IBM-PC 微机一般已扩充到 512KB、640KB，但内存的 64KB 以外的多余容量 BASIC 程序无法利用。Turbo BASIC 可以利用全部内存，就为解决复杂问题、运行大的程序提供了可能。

(5) 采用 IEEE（美国电气和电子工程师协会）标准的浮点格式，而不是 Microsoft 公司专用的浮点格式，因此双精度数的取值范围在 10^{-307} 与 10^{308} 之间。

(6) 数值型数据的精度高。内部超越函数的值是双精度浮点数，具有长达 16 位的数字；增加了一种长整型数据类型，可处理长达 10 位的整数。

(7) 对数组的使用更加灵活。可同时说明数组的上界和下界；可使用动态数组；每个数组可长达 64K。

(8) 增强了图形功能。可用图案填充封闭区域，可在图形屏幕上任意设置窗口，可在窗口内重新定义图形坐标系，很容易画出笛卡尔坐标系中的函数图形。

(9) 增加了一种二进制格式的数据文件，有利于与其它语言进行数据交换。

(10) 调用汇编语言的子程序更加方便。在 Turbo BASIC 中，可以调用 DOS 和 BIOS 的软中断；可把机器代码放在过程中调用，也可以调用磁盘上独立的 .COM 文件；还可以通过绝对地址访问汇编语言子程序。

3. 有良好、方便的工作方式

PC BASIC 分为解释 BASIC 和编译 BASIC。解释 BASIC 运行速度慢，不适宜于编制较大的程序。而 PC 编译 BASIC 生成的可执行文件运行速度虽然快，但生成可执行文件的过程却比较费事，要经过编辑-编译-连接-运行这样一系列的过程。其中要涉及三个专用软件，还要建立几种中间文件才能得到最终的执行文件，操作麻烦且占用大量的存贮容量。如果在上述过程中发现错误，就需重复这种过程，十分费事费时。

Turbo BASIC 克服了上述的 PC BASIC 的缺点，它完全抛弃了解释方式，将编辑、编译、运行、追踪等工作融为一个整体，由一个名为 TB.EXE 的系统文件提供了完整的工作环境，它包含编辑器、编译器和运行时的程序库，它工作时全部在内存中进行，避免了费时的访盘工作，且各种工作之间的转换非常方便，编译速度可达每分钟上千行。因此大大缩短了调试程序的时间，提高了工作效率，这是美国 Borland 公司推出的 Turbo 系列软件区别于其它软件的重要特征之一。

在上述 TB.EXE 提供的工作环境下，用 Edit 命令可输入、编辑源程序，用 RUN 命令可自动地先编译后执行程序，发现错误，光标就自动地停在源程序的错误之处等待修改，同时显示关于错误的说明，多数错误的原因从这个说明中就可搞清楚，只有少数需要查阅 Turbo BASIC 使用手册中的详细说明。修改后用 RUN 命令可立即重新编译、执行程序。因此，这个软件对用户来说使用非常方便。

另外，Turbo BASIC 把编辑程序时使用的命令和操作使用的命令从语言系统中分离出去，编辑工作由编辑器完成，操作命令和环境参数均以菜单方式提供。屏幕上可同时提供四个窗口：Edit（编辑窗口），Trace（追踪窗口），RUN（运行窗口），message（信息窗口）。通过这四个窗口，可分别显示各种信息。其中 Edit 和 RUN 两个窗口都可放大到全屏幕。这样，不仅概念更加清楚，而且使用更加方便。

调试好的程序既可以用源码文件（.BAS）的形式存入磁盘，也可以用可执行代码（.EXE）的形式存入磁盘。这对应用软件的开发和利用无疑将会带来极大的方便。

4、与 PC BASIC 的兼容性好

PC BASIC 的程序格式以及绝大多数函数、语句均在 Turbo BASIC 中有效，且在使用行号或标号的情况下，仍可使用 PC BASIC 的转向语句，如 GOTO 和 GOSUB 等。因此，大量的原有的 PC BASIC 程序不会作废，而且一般情况下可直接在 Turbo BASIC 中调试运行。

综上所述，Turbo BASIC 既保持了与原 PC BASIC 的兼容性及易学易用的特点，又作了重大的扩充，使之成为实用性强、标准化高、能解决较复杂问题的结构化程序设计语言。

习题一

- 1-1 计算机的硬件主要包括哪几部分？
- 1-2 计算机的软件主要包括哪几部分？
- 1-3 什么是机器语言？
- 1-4 什么是高级语言？
- 1-5 试述高级语言的使用过程。
- 1-6 什么是汇编程序、解释程序、编译程序？
- 1-7 Turbo BASIC 语言的主要特点是什么？

第二章 Turbo BASIC 语言基础

学习程序设计语言的最好方法是自己动手编写程序，并能在机器上运行和调试这些程序。为此，我们必须了解 Turbo BASIC 程序的基本结构，程序中可以使用的基本符号、数据类型、运算符及表达式等。这一章我们将分别对这些问题加以说明。

第一节 Turbo BASIC 程序的基本结构

Turbo BASIC 程序是按照 Turbo BASIC 语言的规定编写的程序，它是如何构成的呢？让我们先看一个简单的例子。下面的程序是要计算两个数的和与积，并把结果打印出来。

```
' Calculating sum and product of A,B.
```

```
LET A=100  
LET B=150  
LET S=A+B  
LET P=A*B  
PRINT "A+B=";S, "A*B=";P  
END
```

```
RUN
```

```
A+B= 250          A*B= 15000
```

上面的程序共有 8 行。

第 1 与第 2 行的第 1 个字符是单引号 (')，表示这两行是注释行。第 1 行注释的内容是求两数的和与积，第 2 行为空白。

第 3 行至第 6 行为赋值语句。第 3 行和第 4 行是把 100 和 150 分别赋值给变量 A 和 B，而第 5 行和第 6 行是把运算后的结果分别赋值给变量 S 和 P。

第 7 行为输出语句。

第 8 行为终止语句，表示程序结束。

RUN 下面为本程序的运行结果。

上面所涉及到的语句，以后都要详细介绍，这里先粗略说明一下。

通常把这种利用 Turbo BASIC 语言编制的程序称为 Turbo BASIC 语言源程序 (Source Program)。关于 Turbo BASIC 语言源程序有以下一些规定：

(一) Turbo BASIC 源程序是由简单语句和控制结构组成的，一般有若干行。每行最多可有 249 个字符。但为了易于阅读，一行中一般不放 80 个以上的字符。当要输入 80 多个字符的一个语句时，可以把它拆成几行，在每行末尾 (除最后一行外) 用下划线“_”作为续行符，Turbo BASIC 就知道下一行与这行是同一个语句。作为续行的例子，我们把前面程序中的 PRINT 语句拆成两行：

```
PRINT "A+B=";S,_  
      "A*B=";P
```

Turbo BASIC 中的简单语句可占一行、或多个语句共占一行，只要总长度不超过 249 个字符，此时两个语句之间用冒号“:”分隔。例如可把前面程序中的第 3 行与第 4 行的两个语句写在同一行中：

```
LET A = 100 : LET B = 150
```

其中的冒号就起了分隔语句的作用。

至于控制结构，由语句或语句组给出，因此可以占用多行，这将在以后有关章节再作介绍。

(二) Turbo BASIC 的简单语句一般可分为四个部分：

1、语句标号

在 PC BASIC 中，每行必须有一个行号，行号是放在程序行的最左端，如：

```
30 LET A = 100
```

这里 30 就表示行号。它可以是从 1 至 5 位的十进制整数，必须在 0—65535 的范围内取值。没有行号的程序被作为出错处理。

而在 Turbo BASIC 中，认为没有必要在程序中每行都保留一个行号，行号可以省略，而提倡用标号，以提高程序的可读性。

标号必须以字母开头，以冒号结束，中间可有字母或数字，标号必须单独占一行（可以带注释）。它所标识的语句要紧跟在下一行。

标号和行号可以作为转移语句的目标，也可以用于追踪程序时的显示输出，在转移语句（GOTO 或 GOSUB）中提到标号时不带冒号。

2、关键字

关键字是用来区别语句的类型，它通知计算机执行一定的操作及如何处理跟在关键字后边的数据。如：

```
LET A = 100
```

LET 就是关键字，是赋值的意思，通知计算机把 100 这个数送到 A 中（或者说，把 100 赋值给变量 A）。

3、语句体

语句中跟在关键字后边的部分叫做语句体，是关键字要执行的内容。例如：

```
60 LET P = A * B
```

行号 关键字 语句体

LET 是关键字，它指挥计算机将变量 A 与 B 相乘，其结果送到 P 中。P = A * B 就是 LET 要执行的内容——语句体。

又如：

```
70 PRINT "A+B=",S,"A * B=";P
```

行号 关键字 语句体

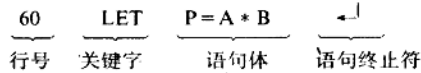
PRINT 是关键字，它指示计算机进行输出，PRINT 后面就是要输出的内容。

4、语句终止符

语句终止符是表示一个程序行的结束，它通常是使用键盘上的一个特定的键来完成它的功能，这个键我们一般称它为回车键。回车键因机而异，如有的用 ENTER 键表示回车键；有的用 ↵ 键；有的用 RETURN 键表示等。但不管用什么键来表示，它们的功能是一样的，都

表示一个程序行的结束。

最后把简单语句的基本结构用图表示如下：



这里还应说明的是：语句终止符的作用是通过打入一个回车键来实现的，在屏幕上并没有任何显示。在书写程序时，各语句行的末尾也不必特别注明语句终止符，但不能认为那里没有语句终止符，这是需要注意的。

(三) 对一个 Turbo BASIC 程序来说，虽然 END 语句不是必须的，但使用它是一种良好的习惯，它的作用是终止程序的执行。

(四) 程序中可任意使用空行、空格和注释，以增加程序的易读性，它们对程序的运行毫无影响。

(五) 关于在程序中添加注释的方法

Turbo BASIC 的注释方法有两种：

1、使用单引号

这种方法是在行的开头或结尾处放上一个单引号，而后写上注释的内容，它可以是任何文字符号。例如：

```
' This is a program.
```

```
LET A=10      'Assign "10" to A.  
END
```

但是在 DATA 语句后不能用单引号引出注释，此时如需要可使用下面介绍的 REM 语句。

2、使用 REM 语句

REM 是 REMARK (注释) 的缩写，但关键字只能使用 REM。在 REM 后写上注释的内容，它同样可以是任意的字符串。REM 语句可以单独占一行，也可以放在一行的最后。例如：

```
REM Example  
LET B=15      :REM Assign "15" to B.  
PRINT B  
END
```

当 REM 语句放在一行的最后时，关键字 REM 前面应放上冒号，以示与前面的语句分开。

还应说明：程序在编译和执行时不理睬注释的内容，在目标程序中注释部分也不占用空间。

(六) Turbo BASIC 语句可以分成可执行语句和不可执行语句两类。可执行语句要引起机器的某种操作，如 PRINT、LET 语句就是可执行语句；不可执行语句通常是一些说明性的语句或定义性的语句，如注释语句就是不可执行语句。当 Turbo BASIC 看到不可执行语句时，不引起任何操作；但不可执行语句也是程序的组成部分，它们起到提供信息等作用。

第二节 Turbo BASIC 的字符集和保留字

一、字符集

Turbo BASIC 语言允许出现的字符包括字母表的字符、数字字符和一些专用字符，这些字符组合在一起，就构成了 Turbo BASIC 语言的字符集。在这个字符集中的全部字符都是 Turbo BASIC 能够识别的字符。如果在编制程序的时候，使用了字符集以外的其它字符，就会出现程序错误。因此，必须严格遵循。

字母字符：大写和小写的全部英文字母

数字字符：0~9 的数字

专用字符：见表 2-1

表 2-1 专用字符

字符	名称
	空格 (实为空白, 在键盘上按空格键)
=	等号或赋值号
+	加号或连接符号
-	减号或负号
*	星号或乘号
/	斜线或除号
\	反斜线或整除符号
^	乘幂符号
(左括号
)	右括号
%	百分号或整型数字说明符
#	数字符号或双精度型数字说明符
!	感叹号或单精度型数字说明符
\$	美元符号或字符串类型说明符或伪语句的前缀
&	表示 and 的字符或长整型数字说明符
,	逗号
.	句号或小数点
'	单引号
:	分号
;	冒号或语句间的分隔符
?	问号或 PRINT 的代替符
<	小于
>	大于
"	双引号或字符串分隔符
[]	方括号 (只用于数组)
_	下划线或续行符

还有一些专用字符，它们虽然在 Turbo BASIC 中没有什么特殊意义，但它们也是 Turbo BASIC 的字符，并且可以被显示或打印。读者如需要可查阅所用机器的有关手册，这里就不赘述了。

最后还应强调的是：在 Turbo BASIC 源程序中，绝对不允许出现字符集以外的符号，如 α 、 β 、 γ 、 π 、 ω 等，也不许把字符书写含糊，以免出错。因此，必须注意以下字母和数字的

区别。

英文字母	数字
Q	0
.	1
S	5
T	7
D	0
Z	2
G	6

二、保留字

在 Turbo BASIC 语言中, 所谓字, 是由一些字符组成的有意义的字串。其中一些字对 Turbo BASIC 有专门的意义, 这些有专门意义的字就称为保留字 (或称关键字)。因此, 保留字是语言本身所固有的。Turbo BASIC 保留字包括全部的命令、语句、函数名称和运算符名称等。保留字因为已经有专门的含义, 因此, 它不能作为变量的名字。在语句行中, 应该注意把保留字与数据或语句的其它部分用空格或其它语法中允许的专门字符分隔开。也就是说, 这些保留字在语句中不能和语句的其它部分紧挨在一起, 否则就要出错, 只有隔开了, Turbo BASIC 才能识别它们。这一点是在书写程序时应该注意的。

第三节 常量、变量、数组

常量和变量以及数组是 Turbo BASIC 语言中三种类型的量, 它们既可以取数值, 也可以取字符值。

一、常量

所谓常量, 是 Turbo BASIC 在执行期间使用的实际值。Turbo BASIC 语言有三种常量: 数值常量, 字符串常量和命名常量。

(一) 数值常量 (numeric constants)

数值常量是一个数, 有正负之分, 数值中不能包含有逗号。数值常量可以有五种表示方法。

1、普通记数法 例如:

347, -520, 3.1416

2、指数记数法

指数记数法就是用指数形式表示一个数。它是表示非常大的数和非常小的数的一种简单方法。例如对于 1.03×10^9 与 7.9×10^{-6} 这两个数, 可用 Turbo BASIC 的指数记数法表示如下:

1.03E+9 (或 1.03E9), 7.9E-6

这里字母 E (或 e) 表示乘方的底数 10。

通常我们把这种以指数形式表示的正数或负数称为浮点数; 而把那些正的或负的实数, 就是包含小数点的那些数称为定点数。

浮点数通常有三个部分组成:

(1) 尾数部分, 它是一个带有正号或负号的整数或定点数。

(2) 字母 E (或 D)

(3) 指数部分, 它是一个带有正号或负号的整数, 它指的是 10 的乘幂次数。

这里字母 E 的意义是表示单精度浮点数，而 D 是表示双精度浮点数（关于单精度数和双精度数可参看下一节）。

例如：25E-2

这里 25 是尾数，-2 是指数，它表示的数为 25×10^{-2} 。如果把 this 浮点数换算为一个定点数，它的值就是 0.25。

3、十六进制整数

一个十六进制的整数最多可有四位数字，以 &H 作为十六进制整数的标志（或称为前缀）。十六进制整数的数字符号有 0~9, A, B, C, D, E, F。

例如：

&H11 表示十进制中的 17

&H2D 表示十进制中的 45 ($2 \times 16 + 13 = 45$)

而 11 和 2D 是十六进制中的两个数。

4、八进制整数

一个八进制整数最多可有六位数字，以 &O 或 &作为八进制整数的标志（前缀），八进制的数字符号有 0~8。

例如：

&O11 表示十进制中的 9

5、二进制整数

一个二进制整数最多可有十六位数字，以 &B 作为二进制数的前缀，二进制的数字符号只有 0 和 1。

例如：

&B11 表示十进制中的 3

下面用不同的数制来表示十进制中的整数 256：

256

&H100

&O400 （或&400）

&B100000000

关于数制间的转换，为节省篇幅，本书就不介绍了。

(二) 字符串常量 (*string constants*)

在本章第一节开头的例子中曾有这样一个语句：

```
PRINT "A+B=";S;"A * B=";P
```

这里的 "A+B=" 和 "A * B=" 就是两个字符串常量。它们在此例中是作为输出提示的。此例的输出结果为

```
A+B= 250    A * B= 15000
```

所以 "A+B=" 代表的内容为 A+B=；而 "A * B=" 代表的内容为 A * B=。

因此，一个字符串常量就是用双引号括住的任意符号序列。这个序列中的符号可以是字母、数字、标点符号以及其它机器可以显示或打印的专用字符。在这里双引号是字符串常量的定界符，而不是字符串的组成部分。

例如：

```
"NAME"
```