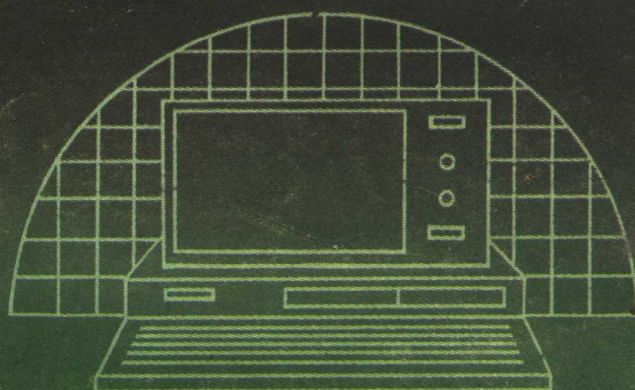


[美] L. J. 戈尔茨坦 著

IBM PC

高级BASIC及编程技巧



海 洋 出 版 社

76

IBM PC
高级BASIC及编程技巧

[美] L.J. 戈尔茨坦 著

周宝兴 陈耀明

李 光 武振东

梁祖威

译

校

海 洋 出 版 社

1986 年 · 北京

内 容 简 介

本书是为基本了解普通 BASIC 语言而希望进一步提高的读者编写的中级读物,实用性极强。书中讨论的主要内容有:设计大型程序、绘图、专门的输入程序、文件、二进制和十六进制算术运算、存储器管理、通用的字符集、打印机图形、通信、光笔及操纵杆等附件、BASIC 编译程序、BASIC 中的机器语言子程序、BASIC 2.00 以及 DOS 2.00 等,其中包括许多实用程序,可供用户使用。本书适于 IBM PC 及兼容机的所有用户阅读。

Advanced BASIC and Beyond for the IBM PC

Larrg Joel Goldstein

Robert J. Brady Co., 1984:

责任编辑:刘莉蕾

责任校对:刘兴昌

IBM PC 高级 BASIC 及编程技巧

[美] L.J 戈尔茨坦 著

周宝兴 陈耀明

李 光 武振东

梁祖威

译

校

海 洋 出 版 社 出 版 (北京市复兴门外大街1号)

新华书店北京发行所发行 牛栏山一中印刷厂印刷

· 开本: 787×1092 1/16 印张: 18^{1/4} 字数: 350 千字

1986 年 5 月第一版

1986 年 5 月第一次印刷

印数: 6000

统一书号: 17193·0611

定价: 4.00 元

译者序

美国国际商用机器公司生产的个人计算机 IBM PC 是近年来广泛出现在国际市场的一种微型计算机。它产量大，应用广，在世界许多国家都拥有较多的用户。我国引进的 IBM PC 的数量也很多，同时国内研制生产的长城 0520 和 DS-PC 等机器也与其兼容，因此，在我国 IBM PC 及兼容机的用户也是很多的。但是，目前国内公开出版发行有关 IBM PC 方面的书籍还较少，不能满足广大读者的要求。为此，海洋出版社组织编译了这套《IBM PC 丛书》，其中包括《高级 BASIC 及编程技巧》《接口技术》《科学计算的 BASIC 程序设计》《工程技术用 BASIC》《图形程序设计》《VISICAL 电子表格软件的应用》等，本书是其中之一。

IBM PC 虽然是一种个人计算机，但其性能远远超出了早期个人计算机代表性产品 TRS-80 和 APPLE 等。IBM PC 采用了 Intel 公司生产的 16 bit 微处理器 8088 作 CPU，并配有存储能力为 320 KB 的软磁盘驱动器或 10 MB 的温式硬磁盘驱动器。这种微计算机的应用软件及接口硬件品种繁多，便于在计算、管理、数据采集和分析、工业控制等各个领域中的应用。

本书主要介绍 IBM PC 的主要高级语言之一——高级 BASIC 的特性及编程技巧。高级 BASIC 除了具有一般 BASIC 的功能之外，还有图象和音乐功能，因而使 IBM PC 的 BASIC 功能更强了。但是目前系统介绍高级 BASIC 的资料较少，本书就是为基本了解普通 BASIC 语言而希望进一步提高的读者编写的，实用性很强，因此对广大 IBM PC 及兼容机的用户是一本很有价值的参考书。

本书的第一至四章由周宝兴翻译，第五至七章由陈耀明翻译，第八至十章由李光翻译，第十一至十三章及附录由武振东翻译。全书经梁祖威校订。由于译校者水平所限，难免有错误和不妥之处，欢迎读者批评指正。

译校者

1984年12月

作者序言

如果你已经熟悉了 IBM PC BASIC 的基本部分,并希望学习它的“高级性能”,那么这本书对你是很适合的。本书中,我们在 BASIC 的初等书籍和关于通信或汇编语言程序设计等高级读物之间的缺口上架起了一座桥。读完这本书,你就能对 PC BASIC 的所有方面运用自如。

本书独到之处是它的实例研究。我们指导你完成一个大型的实例研究程序的整个设计,包括绘图、主机接口、数据输入和验证以及文件的使用。对此程序的讨论是从第二章中对程序设计的讨论开始的,并在全书中交织进行。你将看到怎样建立超过教科书习题范围的程序。此外,对实例研究的扩充作了各种建议,供你试验。

讨论的题目有:设计大型程序、绘图、专门的输入程序、文件、二进制和十六进制算术运算、存储器管理、通用的字符集、打印机图形、通信、游戏操纵杆、光笔、BASIC 编译程序、BASIC 中的机器语言子程序、BASIC 2.00 以及 DOS 2.00。

本书含有许多有趣的和有用的程序。特别有用的是专门形式的输入子程序,你可以把它用在你自己的程序中。

为了使这本书成为真正的教学工具,在讨论到关键性的问题时都给出了测验你理解程度的习题。这些问题的答案在每一节的末尾。

我衷心感谢复审者对于本书插图图样提出的有价值的建议。感谢 Robert J. Brady 公司的工作人员,感谢他们在编印和发行我的计算机书籍中的良好合作。特别感谢 Paula Huber 的高效、精心的手稿编辑工作,以及我的朋友 Harry Gaines 校长和 David Culverwell 总编的大力支持。

[美] L.J. 戈尔茨坦

1983. 10. 5.

目 录

第一章 前言	(1)
1.1 本书的目的.....	(1)
1.2 本书概述.....	(1)
1.3 必要的预备知识.....	(2)
第二章 大型程序的设计和开发	(4)
2.1 设计“条线图生成程序”.....	(4)
第三章 PC 的图形和声音功能概述	(7)
3.1 直线图形.....	(7)
3.2 在文本方式画条线图(适用于所有的 PC 用户)	(13)
3.3 颜色和图形方式.....	(18)
3.4 直线、矩形和圆.....	(22)
3.5 计算机美术.....	(29)
3.6 画圆饼式统计图.....	(32)
3.7 PAINT 和 DRAW	(34)
3.8 保存和重新调用图形映象.....	(42)
3.9 VIEW 和 WINDOW	(45)
3.10 条线图——实例研究 I	(49)
3.11 PC 的声音和音乐	(52)
第四章 输入的专门方法	(57)
4.1 为什么需要输入程序.....	(57)
4.2 INKEY\$ 变量	(58)
4.3 功能键和事件捕获.....	(59)
4.4 扩充 ASCII 码	(64)
4.5 输入字符.....	(68)
4.6 输入字符串和数字.....	(72)
4.7 条线图——实例研究 II	(77)
第五章 计算机和文件	(84)
5.1 文件.....	(84)
5.2 顺序文件.....	(85)
5.3 顺序文件的其他有关事项.....	(91)
5.4 随机存取文件.....	(94)
5.5 随机存取文件应用例.....	(100)
5.6 分类方法.....	(105)

5.7 BASIC 的文件命令	(110)
第六章 存储器管理	(114)
6.1 二进制和十六进制数	(114)
6.2 比特、字节和存储器	(119)
6.3 存储器中的数据存储	(123)
6.4 字节运算	(127)
6.5 字节运算的应用例	(130)
6.6 高级图形技巧	(137)
6.7 图形方式绘制字符	(140)
6.8 条线图——实例研究Ⅲ	(142)
6.9 IBM PC 字符集的扩充	(146)
附录 A: 几个有用的存储器单元	(146)
附录 B: BASIC 如何存储单精度和双精度数	(147)
第七章 条线图生成程序	(151)
7.1 初始化和控制程序	(151)
7.2 错误陷阱	(154)
7.3 条线图生成程序	(156)
7.4 设计大程序的高级技巧	(170)
第八章 IBM/EPSON 打印机	(173)
8.1 打印原理	(173)
8.2 打印机命令	(177)
8.3 打印机的图形功能	(180)
8.4 屏幕图形转储	(186)
8.5 回到我们研究的课题	(189)
第九章 计算机通讯	(190)
9.1 计算机通讯原理	(190)
9.2 通讯文件	(193)
9.3 为通讯设立陷阱	(201)
第十章 PC 其他附件的使用	(203)
10.1 光笔的使用	(203)
10.2 操纵杆的使用	(206)
10.3 绘图仪的使用	(210)
第十一章 BASIC 编译程序	(213)
11.1 为什么要对程序编译	(213)
11.2 程序编译	(213)
11.3 编译程序的参数	(224)
11.4 编译程序和解释程序之间的差别	(225)
11.5 编译的错误信息	(226)
11.6 重新回到我们研究的课题	(227)

第十二章 在 BASIC 中使用汇编语言子程序	(228)
12.1 8088 微处理器	(228)
12.2 为什么要使用汇编语言子程序?	(229)
12.3 调用机器语言子程序.....	(231)
12.4 编写机器语言子程序.....	(232)
12.5 使用汇编程序.....	(235)
12.6 把子程序放入存储器中.....	(237)
12.7 结论.....	(241)
第十三章 DOS 2.00 的一些特点	(242)
13.1 路径、目录和子目录.....	(242)
13.2 DOS 2.00 的修改述评	(247)
13.3 DOS 2.00 中的新命令	(250)
13.4 增强的批文件的能力.....	(255)
13.5 输入和输出的重新定向.....	(260)
13.6 BASIC 2.00	(262)
部分习题答案.....	(263)
有关本书程序软磁盘上的文件的使用.....	(272)
BASIC 错误信息	(274)
DOS 2.00 命令一览表	(276)
PC BASIC 参考指南	(277)
《IBM PC 丛书》简介	(284)

第一章 前 言

1.1 本书的目的

我的许多朋友、家庭成员和学生都曾选择过一本较好的入门书，并从中学习了关于 IBM PC 的 BASIC 程序设计。

但是，在学完这种书之后，他们通常问我：“我接着读些什么？我怎样才能学到更多的关于 IBM PC 的 BASIC 程序设计知识？”我理解他们的问题。IBM PC BASIC 是一种非常丰富的语言，苛求在一本书里包括所有想要知道的东西是不现实的。PC BASIC 毕竟含有 150 条以上的命令、语句和函数，可买到的书籍似乎都立足于 BASIC 的最基本部分，因此，也没有什么书值得推荐的。

我对文献中的这种欠缺思考了一年多以后，决定写这本书。我的目的是讨论 BASIC 程序设计中的所有“高级”方面，以及一些巧妙使用 BASIC 的窍门。

在与朋友和学生的交谈中，我逐渐地了解到，所有的 BASIC 初级读物都还有另一个严重的缺陷：它们几乎都是只介绍各种 BASIC 语句和命令的句法，而根本不讨论比较复杂的程序设计和开发。鉴于这种情况，我决定把程序设计和开发这个题目作为本书的主要题目之一。

没有任何东西可以替代实践经验，因此，我决定用实例来讲述程序设计和开发。在此实例中，我们将建立一个叫做“统计用的条线图生成程序”(THE BAR CHART GENERATOR)的程序。这个程序比较长，也相当复杂，并且可以启发我们去开发一些在程序设计中非常有用的子程序。

1.2 本书概述

在第二章，我们讲述“条线图生成程序”的要求，并阐明所需要的一些主要子程序，然后回到对高级 BASIC 的研究。在讨论中，我们导出了对于“条线图生成程序”非常有用的概念。当我们遇到这些概念时，就建立各种需要的子程序。当我们讨论到第七章时，我们已准备好将这些子程序汇总到一个程序中去了。

这个实例是贯穿全书的一个议题，但是，并不是每一节都与它有关。例如，第三章是一个关于在 PC 上产生图形和声音的专门的讨论。然而，尚未到该章的结束，此实例就开始起作用了。

第四章是对专门形式输入的详细讨论。我们将学习到专门的程序员怎样从“防弹”方法进行输入。我们还要建立专门形式的输入子程序，你可以把它们用于你自己的程序之中。

绝大多数的程序都使用 PC 的文件功能，因此，在第五章内，我们将讨论顺序存取和随机存取的文件，并将建立一个作为应用程序的列表管理程序，研究一些排序算法。

为了建立“条线图生成程序”，我们还需要具有一些二进制和十六进制算术运算的知识。我们将学习第六章中需要了解的知识，然后研究 RAM 的内部组织、存储器地址、准备替换的字符集，以及许多其他的有趣题目。

在第七章，我们收集了“条线图生成程序”的各程序段，并把它们装配到一个工作程序中去。第八章讨论 IBM/EPSON 打印机的优点，用二进制数的知识和打印机绘图命令编写一个图形屏幕打印子程序。

在第九章，我们学习用 RS 232-C 接口进行计算机通信。为了说明这些命令，我们将建立一个允许两台 IBM PC 互相通讯的通讯程序。

第十章着重讨论在 PC BASIC 情况下使用的游戏操纵杆、光笔和绘图仪。

在第十一章中，你将会遇到 BASIC 编译程序。如果你对 BASIC* 感到不太满意，那么可以用 BASIC 编译程序去编译你的程序，以达到机器语言所具有的速度。

在第十二章，我们说明在 BASIC 中怎样使用机器语言子程序。这一章为你开始学习汇编语言程序设计提供了一个跳板。

在本书中，我们还讨论了 2.00 版所提供的 BASIC 扩充，特别是 LINE 和 PAINT 中的修改，以及新的语句 VIEW 和 WINDOW。但是，在第十三章以前我们一直回避 DOS 2.00，直到第十三章才全面介绍它。

1.3 必要的预备知识

假设你已经熟悉了 IBM PC BASIC 的初步的程序设计，即熟悉 L.J. 戈尔茨坦和 M. 戈尔茨坦所著《IBM PC 程序设计与应用》(IBM PC-Programming and Applications, Robert Brady 公司, 1982)一书中讲述的内容。

在今后我们将以“GG”代表这本书。

明确地说，假设你已懂得 IBM PC 的 DOS(1.1 版)和基本 BASIC(1.1 版)，这包括以下内容：

DOS 1.1 命令

CHKDSK	COMP	COPY
DATE	DIR	DISKCOMP
DISKCOPY	ERASE	FORMAT
MODE	RENAME	SYS
TIME	TYPE	

BASIC 1.1 语句、函数和命令

ABS	INSTR	REM
ASC	INT	RENUM
ATN	KILL	RESTORE
AUTO	LEFT \$	RIGHT \$
CHAIN	LEN	RND
CHR \$	LET	RUN

* 指解释程序。——译者注

CLEAR	LINE INPUT	SAVE
CLOSE	LIST	SGN
CLS	LLIST	SIN
CONT	LOAD	SPACE \$
COS	LOCATE	SPC
DATA	LOG	SQR
DATE \$	LPRINT	STOP
DEFFN	LPRINT USING	STR \$
DEFtype	MERGE	SWAP
DELETE	MID \$	SYSTEM
DIM	NAME	TAB
EDIT	NEW	TAN
END	ON...GOSUB	TIME \$
EXP	ON...GOTO	TRON
FIX	OPTION BASE	TROFF
FOR...NEXT	PRINT	VAL
GOSUB...RETURN	PRINT USING	WHILE...WEND
GOTO	RANDOMIZE	WIDTH
IF	READ	WRITE
INPUT		

如果用户已经忘记了(或根本不了解)其中的某些语句,请参考相应的 IBM 手册 (DOS 或 BASIC 参考手册)或者象“GG”这样的 PC BASIC 初级读物。

第二章 大型程序的设计和开发

在 BASIC 程序设计的初级读物中, 你已学习了最基本的 BASIC 语句的句法以及如何把这些语句编制到一个程序中去。但是, 在这些书中, 说明性的例程序通常都比较短, 并且它们的逻辑都十分简单。然而, 在本书中, 我们的主要目的之一是提供建立大型程序的一些诀窍, 因此, 在我们讨论任何 BASIC 语句之前, 让我们先概述一下“条线图生成程序”(以后我们将多次用到这个实例), 以突出这个目的。

2.1 设计“条线图生成程序”

在开发一个大型程序时, 仔细地进行程序设计是很重要的。对于你的程序要干什么? 将产生什么输出? 从哪儿输入? 都必须有明确的概念。

BASIC 中的一个主要缺点是它允许你不经思索或计划就坐下来编写程序(我敢断定你必定有不少有益的想法!)。如果你编写一个小的程序, 则不进行设计也能获得成功。但是, 当程序需要大量不同的子程序的相互联系、产生不同的输出和影响各程序变量时, 程序设计就变得十分必要了。

现在, 让我们简单讲述一下“条线图生成程序”的设计过程。

请看图 2.1 中的图形。这是一个典型的条线图。你应注意该图的以下特点。该图以图形描述了三组数据, 一组条线对应每一组数据。我们用条线上打阴影的方法来区分其中不同的数据组。

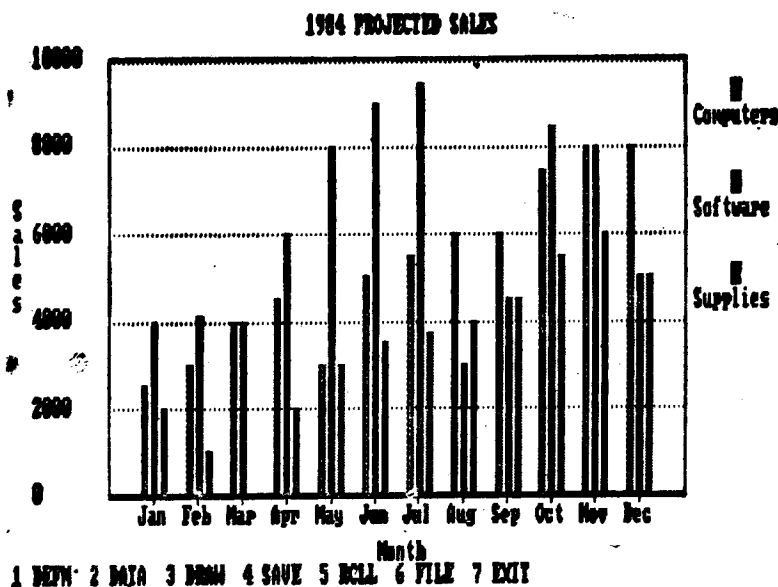


图 2.1 典型的条线图

条线都被描绘在一个坐标系中，这是一个矩形的方框，它的边被标上了阅读该图所必需的信息。坐标系的水平底边叫做 x 轴，在此轴的下面标上了描述各条线的月份。

坐标系的垂直左边线叫做 y 轴。沿 y 轴方向是数字刻度，它可使你确定条线的数字高度。

作为一个完整的图表， x 轴和 y 轴都标有标题。注意， y 轴的标题是垂直地排列在刻度左边的，它在坐标系垂直向的中心位置。 x 轴的标题在坐标系下边的中央、 x 轴刻度之下。条线图的标题在坐标系上方的中央。

三组数据的每一个也有标题，它们显示在坐标系的右边。每个标题的上边是一个方块，它是对应特定数据组的条线阴影形状。

以上我们粗略地浏览了条线图。现在我们可以明确地说：我们的目的是要建立一个叫做“条线图生成程序”的程序，这个程序将根据用户提供的数据显示出条线图。

在我们能够达到目的以前，将需要学习大量的关于程序设计的知识。但是，让我们从程序开发的第一步——程序设计开始吧。

为了设计程序，我们首先应列出程序要执行的具体功能清单。以下就是我们要设计的程序必须具有的功能的清单：

1. 在屏幕上画条线图。
2. 把条线图对应的数据存到磁盘上，这样，以后可以重新绘出条线图来。
3. 从磁盘上重新调用条线图数据。
4. 从磁盘数据文件中读出数字数据项。
5. 编辑条线图数据以进行修改。

在我们拟订出上述必要的功能清单以后，再看看程序还要求输入什么。请看图 2.1，图中的每一个可见的要素都对应着一个输入：

1. 图题
2. x 轴和 y 轴标题
3. x 轴和 y 轴刻度
4. 条线
 - a. 规定几组反映不同数据序列的条线。
 - b. 不同数据序列的条线需要用不同的阴影来区分。
 - c. 每一组条线需要一个识别的标题。

在研究上述功能清单时，我们要问：我们应该容许多少组条线？应该容许多少根条线？我们考虑最多为三组条线，每组最多 20 根条线。用这些数目，我们就可以在一帧屏幕上显示全部数据。这将使程序的数据输入部分比较容易。这些数目无论如何都是足有余的，因为没有什么方法可以在屏幕上一次容纳 $3 \times 20 = 60$ 根条线。否则条线将会重叠起来。但是，也存在一些应用程序，你可以用它们显示单个数据序列的 20 根条线或三组数据序列，每组 12 根条线。我们考虑的数目是足够大的，足以适应这些选择。

我们已经知道了程序将干什么和要求输入什么。现在，让我们设想一下程序将怎样工作。我的意思不是叫你此时坐下来开始编写程序，而是应该问一下自己：用户将怎样使用这个程序？可以作这样的想象：用户正坐在计算机旁，这时你问你自己：他（她）要用这个程序干什么？我在考虑这个问题时认为，用户正在从列出的各种操作的菜单上

进行选择，这菜单是按功能键表显示的。这就指导我们对对应功能键来定义一系列操作：

- F 1——定义条线图参数
- F 2——输入 x 轴标度和数字数据
- F 3——根据当前的数据画条线图
- F 4——存储条线图
- F 5——重新调用条线图
- F 6——由数据文件输入数据
- F 7——退出

至此，我们开始看到了显露出的程序结构。它必须有七个主要的程序，七个功能键的每一个对应一个。请注意必须具有的功能 5（编辑功能）没有作为一个程序出现其中。如我们将要见到的那样，要进行编辑是很容易的，只要直接进入数据输入程序 F 1 和 F 2 就可以了。

除了这七个基本的程序以外，我们还需要一个控制程序，它使我们通过按下相应的功能键就可以在各功能中进行选择。我们的程序还需要大量的数组和许多变量，它们都要求有特定的初始值。用叫做初始化程序的一部分特定程序来处理所有的这些定义是很方便的。

根据上述讨论，我们现在可以草拟出我们的程序了：

```
100  'Initialization
900  'Control Routine
1000 'Define Bar Chart Parameters
2000 'Input Data
3000 'Draw Chart
4000 'Save Chart
5000 'Recall Chart
6000 'Read Data File
7000 'Exit
```

这就是程序设计！我们已保留了大量的空间，以便填入各程序行。我们知道，这七个基本程序还需要一些别的程序。然而，上述方案将成为我们的指南。

为了编写出上述程序设计中所需的各个程序，现在我们要深入学习 BASIC。

第三章 PC 的图形和声音功能概述

为了建立“条线图生成程序”，我们必须学习 PC BASIC 的图形和声音语句。这就是本章的目的。

3.1 直线图形

当你一开始启动 BASIC 时，屏幕处于文本方式。在文本方式中，IBM PC 的视频显示器可以显示 25 行，每行为 40 或 80 个字符。这最多给出了 25×80 即 2000 个可能的字符位置，这些不同的字符位置将屏幕分成一个个小小的矩形。图 3.1 给出了行宽度为 80 字符的细分情况。

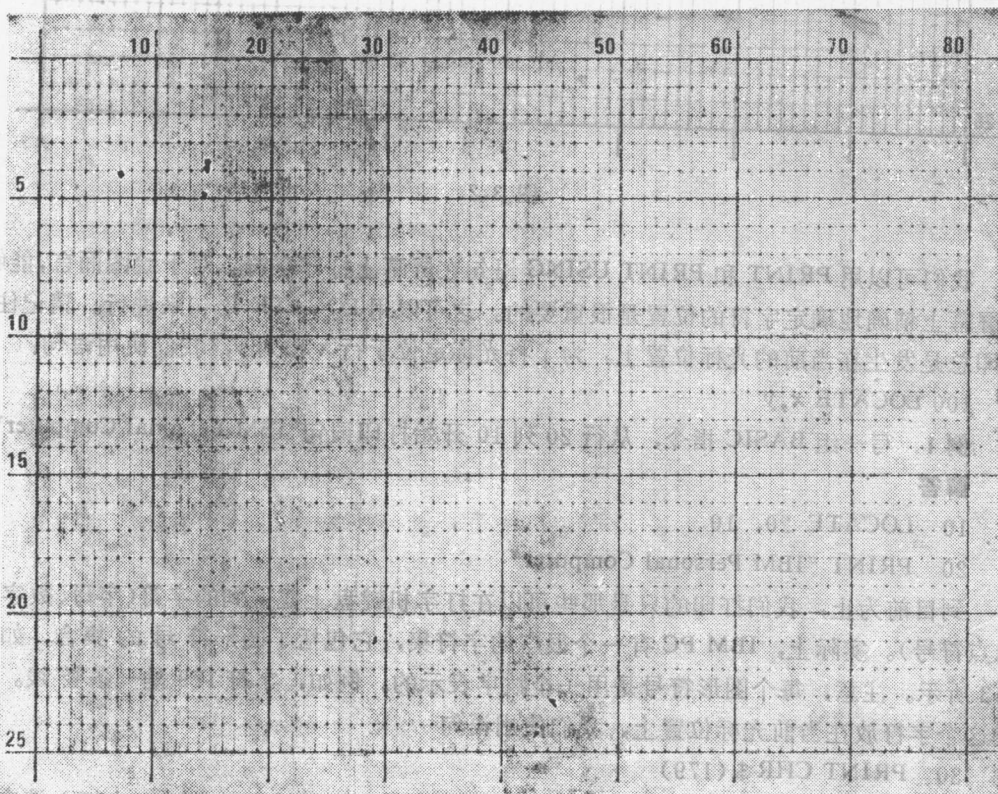


图 3.1 文本方式的屏幕划分

屏幕上被划分成的那些小方块是按行和列排列的。我们将行编号成 1 到 25，行 1 在屏幕的顶端，而行 25 在屏幕的底部。列被编号成 1 到 80。列 1 在最左边，列 80 在最右边。屏幕上的每一个矩形是用表示行和列的一对数字区分的。例如，我们在图 3.2 中显示了第 12 行第 16 列上的矩形。

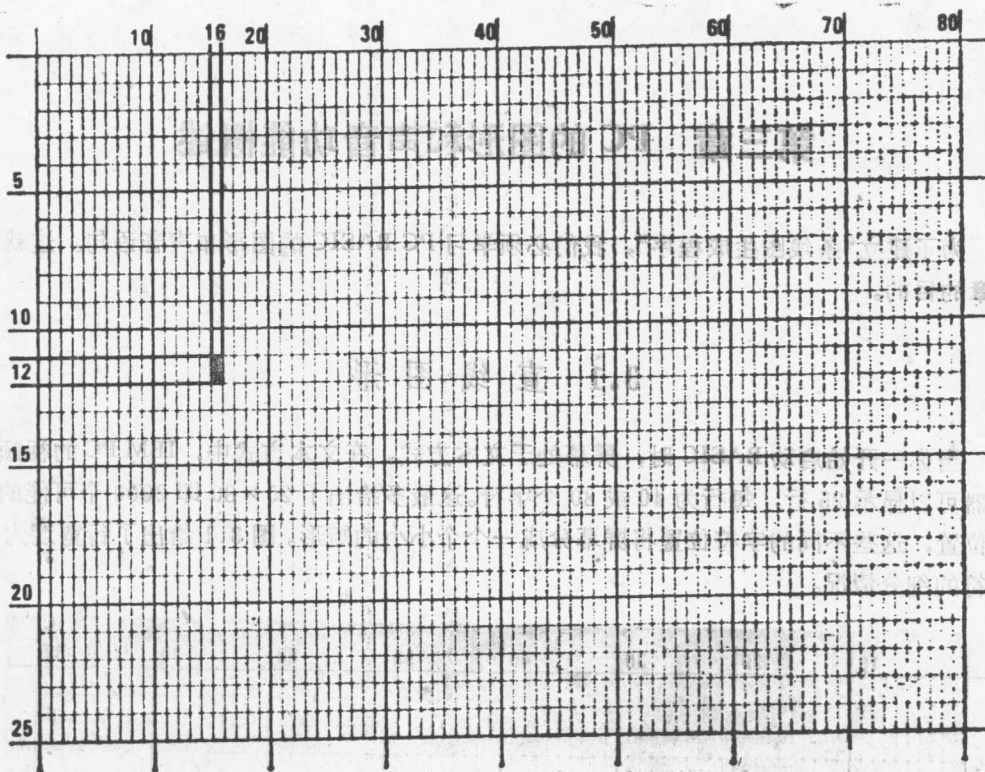


图 3.2

我们可以用 **PRINT** 和 **PRINT USING** 语句在屏幕上打印字符。对于绘图而言,能够在屏幕上精确地确定字符的位置是很重要的。这可以用 **LOCATE** 语句来进行。请记住,打印总是发生在当前的光标位置上。为了将光标定位于行 x 和列 y , 我们使用语句:

```
100 LOCATE x,y
```

例 1. 写一组 BASIC 指令, 从行 20 列 10 开始打印词句“IBM Personal Computer”

解答

```
10 LOCATE 20, 10
```

```
20 PRINT "IBM Personal Computer"
```

到目前为止, 我们打印的只是那些可以在打字机键盘上找得到的字符(字母、数字和标点符号)。实际上, IBM PC 有一个很广的字符集, 它包括了图形符号的集合, 如图 3.3 所示。注意, 每个图形符号是用一个数字表示的。例如, 字符 179 是一条竖线。为把这个字符放在当前光标位置上, 我们使用语句

```
30 PRINT CHR$(179)
```

可以把一个图形字符插入到程序行中, 方法是按住 **ALT** 键在数字键盘上打入该字符号码。这有一个好处, 即在 **PRINT** 语句中就可以看到要打印的字符。例如, 上面的语句可象下面那样出现在屏幕上:

```
30 PRINT |
```

其中, | 符号是在键盘上通过按住 **ALT** 键打入 179 输入的。下面, 我们将用 **CHR\$** 函

ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
128	Ç	166	æ	204	⌈	242	≥
129	ü	167	ø	205	≡	243	≤
130	é	168	¸	206	⌋	244	ƒ
131	â	169	┌	207	±	245	ℓ
132	ä	170	┐	208	⌋	246	÷
133	à	171	½	209	≡	247	≈
134	á	172	¼	210	≡	248	°
135	ç	173	ı	211	ℓ	249	•
136	ê	174	«	212	ℓ	250	•
137	ë	175	»	213	≡	251	√
138	è	176	⌘	214	≡	252	∞
139	ï	177	⌘	215	≡	253	²
140	î	178	⌘	216	≡	254	■
141	í	179	└	217	└	255	(空白 'FF')
142	À	180	┘	218	└		
143	Á	181	┘	219	■		
144	Ê	182	┘	220	■		
145	Æ	183	┘	221	■		
146	Ⓐ	184	┘	222	■		
147	Ⓢ	185	┘	223	■		
148	Ⓞ	186	≡	224	α		
149	Ⓟ	187	┘	225	β		
150	Ⓤ	188	ℓ	226	┐		
151	û	189	ℓ	227	π		
152	ÿ	190	ℓ	228	Σ		
153	Ö	191	┘	229	σ		
154	Ü	192	┘	230	μ		
155	¢	193	┘	231	τ		
156	£	194	┘	232	Φ		
157	₣	195	┘	233	θ		
158	₤	196	—	234	Ω		
159	₥	197	+	235	δ		
160	á	198	⌈	236	∞		
161	í	199	⌈	237	Ø		
162	ó	200	ℓ	238	€		
163	ú	201	≡	239	∩		
164	ñ	202	±	240	≡		
165	Ñ	203	≡	241	±		

图 3.3 IBM PC 图形和特殊字符