

(美) R. E. 迈斯 著

# IBM PC

## 微型计算机绘图



海洋出版社

1-41  
/1

371.01  
115/1

IBM PC

# 微型计算机绘图

[美] R.E. 迈斯 著

陶 纯 译

黄小东 校



3208505

海洋出版社

1986年·北京

## 内 容 简 介

本书清晰详细地介绍了IBM PC绘图程序的设计方法,并提供了80多个很有价值的实例程序。其中包括经常需要的各种统计图表,如:条形图、折线图、扇形图以及动画和三维图等。

本书不仅适合从事用微型计算机绘图的人员阅读使用,而且适于IBM PC的程序设计人员,特别是编制计算机图形软件的人员阅读。

JS377/05

责任编辑:刘莉蕾

责任校对:刘兴昌

## Microcomputer graphics for the IBM PC

Roy E. Myers

1984

### IBM PC 微型计算机绘图

[美] R.E. 迈斯 著

陶笃纯 译

黄小东 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 海洋出版社印刷厂印刷  
开本: 787×1092 1/16 印张: 13<sup>1</sup>/<sub>2</sub> 字数: 300千字  
1986年3月第一版 1986年3月第一次印刷  
印数: 8000

---

统一书号: 17193·0747

定价: 2.50元

# 译者的话

在我国,微型计算机正在各个领域逐步得到普及。现在,有IBM PC机的单位相当多。除了用微型机进行过程控制和科学计算外,在许多应用中,还希望在微型机上画出质量较好的图,以便更直观和清晰地显示各种结果。五花八门的计算机游戏,更离不开屏幕上的绘图。这本书正好在这方面提供了很好的材料。任何有IBM PC BASIC编程基础的人,都可以通过这本书学到许多专门的绘图方法,在IBM PC机上画出适合于自己需要的相当漂亮的图。

这本书简明易懂,引人入胜,所需的数学知识和计算机基础都不多。书中提供了80多个已经调好马上可以运行的BASIC程序,从中可以学到微机绘图的各种基本规则和方法,并加深对绘图能力和限制的理解。在这些程序中,有一部分可以直接在实际工作中使用,如几种有用的统计图表,包括扇形图、条形图和折线图。各种产生动画的方法和在二维屏幕上产生三维图象的技术,也都是非常有用的。在学习这本书的时候,除了认真读书中提供的基本知识和程序例子以外,最好能按照作者的建议,多进行一些试验。这样学到的知识才会巩固和灵活运用。

本书作者R.E.迈斯是美国宾夕法尼亚州立大学数学系副教授,在微机绘图方面有丰富的经验。在此之前,他还写过一本《APPLE微型计算机绘图》的书,颇受欢迎。

本书的翻译工作是在较短的时间内完成的,再加以水平有限,错误缺点在所难免,欢迎各方面读者批评指正。

# 前 言

计算机绘图方面的工作是在五十年代开始的。虽然从那时起这项工作取得了不少重要进展，但至今能用计算机绘图的人还不多。多年来，由于设备昂贵，大多数计算机用户对计算机绘图一直无法问津。微型计算机的出现改变了这种情况。用几千美元就可买到一套微计算机绘图系统。商业和教育机构，甚至个人都发现，用计算机绘图是可以做到的，这种系统有用、有趣、功能很强又好玩。

尽管与廉价微机连用的绘图系统还远不尽如人意，但一般水平的程序员用不了多少时间就能在这种系统上作出满不错的图。比较高级的绘图系统具有微型计算机所没有的功能。为了能作出高质量的图，微机绘图程序员必须设法弥补绘图硬件和绘图语言的不足。

本书的目的，是要提供以微型计算机绘图为中心的数学方法和程序设计方法。

所有讨论都是以IBM PC为基础的。假定读者有这种计算机，用BASIC (2.0或更新的版本)编程，使用彩色监视器/绘图接口板。所有的例程序都是在IBM PC上用的，用高级BASIC语言写成；其中许多可以用其他语言改写，也可改在其他计算机上使用。

本书并不打算教BASIC程序设计方法。假定读者是熟悉这种语言的，手里有IBM PC BASIC的说明书。写各个程序例子的目的，是为了说明绘图方法，而不是为了说明程序设计方法，并且，这些程序也不一定是程序的典范。高水平的程序员将会找到许多缩短执行时间和使程序更加合理的办法。所有的程序都已检查过错误，也已试过了。如果不能象所介绍的那样运行，一般是由于键入程序时出错造成的。当然，在一些应用中直接使用这些例程序也未尝不可，但用它们作为学习工具则更为合适。弄懂了各种数学方法和编程方法之后，就可以把它们用作许多类型应用的基础。

因为计算机绘图的应用多种多样，千变万化，所以力求使书中提供的例程序反映出这种多样性。在例子中包括了商业事务绘图、动画游戏，三维图象和在美学和一些实用方面的其他显示。

计算机绘图学起来涉及的面很广，也很有意思。本书是想成为这方面的一本入门书。你可以由此出发，继续从事各种专门的应用，诸如计算机辅助设计，采用汇编语言进行高速动画的程序设计，或者把BASIC绘图程序同各种商业、科学或工程应用中的数据分析程序结合起来。

在读这本书的时候，请你记住，如果边读程序边试，将会学得更多，如果把这些程序加以改进和发展，收获将更大。要多做试验！这是学习计算机和绘图语言的功能和限制的最好途径。

# 目 录

<b>第一章 计算机绘图介绍</b> .....	( 1 )
光栅和象素.....	( 1 )
星空.....	( 1 )
蹦蹦球.....	( 2 )
DRAW 语句.....	( 3 )
小妖精.....	( 3 )
太空飞船.....	( 4 )
彩色线段.....	( 5 )
曲线.....	( 6 )
螺旋圆环.....	( 6 )
动画.....	( 7 )
总结.....	( 8 )
<b>第二章 IBM PC 的绘图特点</b> .....	( 9 )
字符绘图.....	( 9 )
文本页的存储器分配.....	( 9 )
文本屏幕绘图与图象屏幕的比较.....	( 12 )
图象屏幕 2 使用存储器的情况.....	( 12 )
图象屏幕 1 使用存储器的情况.....	( 14 )
<b>第三章 IBM PC BASIC 的绘图语言</b> .....	( 16 )
SCREEN,KEY,CLS,COLOR,PSET,PRESET,LINE.....	( 16 )
相对坐标.....	( 19 )
WINDOW 语句.....	( 20 )
PAINT 语句.....	( 21 )
CIRCLE 语句.....	( 24 )
VIEW 语句.....	( 26 )
DRAW 语句.....	( 28 )
相对移动.....	( 33 )
PMAP,POINT 和 SCREEN.....	( 33 )
BSAVE 和 BLOAD.....	( 35 )
GET 和 PUT.....	( 36 )
<b>第四章 在事务绘图方面的应用</b> .....	( 40 )
直方图.....	( 40 )
增强直方图的表现力.....	( 42 )

	多重直方图.....	( 43 )
	折线图.....	( 45 )
	扇形图.....	( 48 )
	PUT和 GET 的一种应用.....	( 49 )
<b>第五章</b>	<b>动画.....</b>	<b>( 52 )</b>
	用于文本屏幕动画的字符绘图.....	( 52 )
	游戏用的字符绘图.....	( 54 )
	翻页.....	( 56 )
	图象屏幕上的字符绘图.....	( 58 )
	用户设计字符.....	( 59 )
	用LINE语句的动画 .....	( 63 )
	用DRAW 语句 的动画.....	( 64 )
	用于动画的GET,PUT,LINE语句及各种 字 符 .....	( 65 )
<b>第六章</b>	<b>计算机绘画的软件工具.....</b>	<b>( 68 )</b>
	改变屏幕比例.....	( 68 )
	剪除.....	( 69 )
	一种剪除方法.....	( 69 )
	变换.....	( 72 )
	变换的矩阵表示.....	( 74 )
	恒等.....	( 75 )
	改变比例.....	( 76 )
	映象.....	( 76 )
	切变.....	( 77 )
	旋转.....	( 78 )
	齐次坐标.....	( 81 )
	平移.....	( 81 )
	复合变换.....	( 81 )
<b>第七章</b>	<b>解析几何与计算机绘图.....</b>	<b>( 88 )</b>
	坐标轴.....	( 88 )
	直线.....	( 90 )
	圆.....	( 91 )
	直角坐标法.....	( 91 )
	用角度增量来产生点.....	( 92 )
	不用 WINDOW 语句时的圆.....	( 93 )
	点的递推产生.....	( 94 )
	椭圆.....	( 95 )
	用CIRCLE语句画椭圆.....	( 97 )
	旋转和平移.....	( 98 )
	更多的变换.....	( 99 )

	参数方程.....	( 100 )
	抛物线.....	( 102 )
	双曲线.....	( 103 )
	极坐标曲线.....	( 104 )
	$Y=F(X)$ 形式的曲线.....	( 105 )
<b>第八章</b>	<b>三维绘图基础</b> .....	( 106 )
	坐标系.....	( 106 )
	变换.....	( 109 )
	改变比例.....	( 109 )
	旋转.....	( 110 )
	平移.....	( 111 )
	映象.....	( 111 )
	顺序变换.....	( 111 )
	矩阵的逆.....	( 112 )
	变换一个坐标系.....	( 113 )
	投影.....	( 114 )
	三维物体的二维图象.....	( 114 )
	编程序.....	( 119 )
	观察点和透视.....	( 120 )
	图象的变换.....	( 123 )
	$Z=F(X, Y)$ 形式的曲面.....	( 126 )
<b>第九章</b>	<b>遮线和遮面子程序</b> .....	( 128 )
	面的朝向.....	( 129 )
	定向矢量.....	( 129 )
	可见性.....	( 130 )
	举例.....	( 130 )
	顶点数组.....	( 131 )
	面数组.....	( 132 )
	法线数组.....	( 134 )
	可见性检验.....	( 135 )
	边数组.....	( 136 )
	画图.....	( 137 )
	两个物体; 删除.....	( 140 )
	用颜色涂掉.....	( 142 )
	两个物体; 对分搜索.....	( 144 )
	两个物体; 任意观察点.....	( 146 )
	操作的理论.....	( 146 )
	顶点数组.....	( 148 )
	面数组.....	( 150 )



	法线数组.....	( 151 )
	边数组.....	( 151 )
	物体的优先级.....	( 153 )
	优先级 1 的边.....	( 154 )
	优先级 2 的边.....	( 154 )
	剪裁被部分遮住的边.....	( 155 )
	边端点的可见性状态.....	( 156 )
	具有 $Z=F(X,Y)$ 形式的曲面的遮线消除法.....	( 161 )
<b>附录1</b>	<b>二进制, 十进制和十六进制</b> .....	( 166 )
	二进制.....	( 166 )
	十六进制.....	( 167 )
	十六进制与二进制的关系.....	( 168 )
<b>附录2</b>	<b>存储器寻址</b> .....	( 170 )
	BASIC 数据段.....	( 171 )
<b>附录3</b>	<b>矢量</b> .....	( 172 )
	矢量的标量积.....	( 172 )
	直线的矢量表示.....	( 173 )
	三维空间中的矢量.....	( 174 )
	矢量的点乘积.....	( 174 )
	矢量的叉乘积.....	( 175 )
	半平面.....	( 176 )
<b>附录4</b>	<b>矩阵</b> .....	( 178 )
	矩阵乘积.....	( 178 )
<b>附录5</b>	<b>角的弧度量度</b> .....	( 180 )
<b>附录6</b>	<b>附加程序清单</b> .....	( 181 )
	IBM PC系统图.....	( 181 )
	三维曲面.....	( 185 )
	曲面/平面.....	( 186 )
	三维直方图.....	( 189 )
	收入/费用/盈余折线图.....	( 189 )
	房屋.....	( 189 )
	图 7.1.....	( 192 )
<b>附录7</b>	<b>IBM PC BASIC 绘图语句一览表</b> .....	( 195 )
	《IBM PC 丛书》简介.....	( 198 )

# 第一章 计算机绘图介绍

计算机绘图这一术语，对不同的人有不同的意义。有些人会想到电子游戏，另一些人则可能想到训练飞行员用的飞机模拟器。对建筑师来说，计算机绘图为建筑物的设计提供了一种作计划和显示的手段。而一个工程师则会把计算机绘图用于设计和分析电子线路或者桥梁。

计算机在绘图方面的用途是多种多样的。本书只打算介绍在廉价微型机上产生图象所必需的数学方法和程序设计方法。用 IBM PC 作为讨论的基础，所有例子都用高级 BASIC 语言写成。在这一章中，我们将介绍一些程序例子，这些程序既可说明计算机的能力，又能说明在以后各章中要用到的程序设计方法。

**注意：**在这一章中，不少程序例子用了一些要在后面才讲到的方法，所以暂时不必去深究编程细节，只要欣赏这些图就行了。

## 光栅和象素

IBM PC 是用光栅显示来绘图的。这就是说，显示屏由许多水平光栅线组成，而每条光栅线则由许多称为象素的点所构成。在 IBM PC 的高分辨绘图方式，有 200 条光栅线，每条线包含 640 个象素。每条光栅线都编了号，从 0 到 199；每条线中的象素也编了号，从 0 到 639。所以，每个象素由两个数确定，即光栅线号和线中的象素号。通过把一些象素点亮，把另一些象素灭掉，显示屏上就产生出了图象。

在进行光栅扫描中，计算机绘图语言必须具有点亮特定象素（即图象点）的能力。在 IBM PC BASIC 中，是由 PSET 和 PRESET 语句完成这个任务的。PSET (X, Y) 将点亮光栅行 Y 中 X 位置的象素。例如，PSET (0, 0) 将点亮屏幕左上角的点；而 PSET (639, 199) 则点亮高分辨图象屏幕右下角的点。PRESET (X, Y) 将把指定的点灭掉。

## 星空

尽管大多数图象不会只用到很少几个点，但我们要讲的第一个程序例子只用 PSET 语句来产生图象。

程序 P1-1：星空

```
10 'PROGRAM P1-1 (SPACE)
20 'plots randomly selected points
100 '=====
110 RANDOMIZE TIMER
120 KEY OFF;SCREEN 2;CLS
```

```

130 FOR I=1 TO 100
140 X=640*RND,Y=200*RND
150 PSET(X,Y)
160 NEXT I

```

在以上的星空例子中，把星星画在第140行随机选定的位置上，每次运行程序就画出一幅新的星空图景。

注意：如果你的BASIC版本没有TIMER功能，只要在程序中去掉110行。它仍能工作，但每次运行程序将得到相同的星星图案。

## 蹦蹦球

第二个程序例子还是主要以PSET语句为基础，模拟一个蹦蹦球。

程序P1-2：蹦蹦球

```

10 'PROGRAM P1-2 (BOUNCING BALL)
20 'Simulates a ball bouncing off floor and walls
100 '=====
110 KEY OFF;SCREEN 2;CLS
120 DX=6; X=7, Y=0: A=1: V=0
130 LINE(0,0)-(0,191)
140 LINE-(639,191)
150 LINE-(639,0)
160 PSET(X,Y);PSET(X+1,Y)
170 IF Y=190 THEN V=-V;IF V=0 THEN V=-19
180 IF X=637 OR X=1 THEN DX=-DX
190 V=V+A
200 PRESET(X,Y);PRESET(X+1,Y)
210 Y=Y+V;X=X+DX
220 GOTO 160

```

行160得到一个“球”。为能把球看得清楚些，并排画两个点。行130—150画出“墙”和“地板”，球从这两者上面弹回来。当球运动时，水平运动的速度不变，为DX；垂直运动由可变速度V及固定的加速度A所控制。球的“运动”通过画一个点〔行160中的PSET(X, Y)〕，然后再将它抹掉〔行200中的PRESET(X, Y)〕的办法来模拟。改变球的位置(X, Y)（行210），再在新的位置上画球。球从墙上和地板上蹦回来由行170和180所控制。为了维持运动，当球蹦到最低点时（Y=190且V=0），行170将球加速。

行130—150的那些LINE语句，当然是画连结指定点的线段。LINE(0,0)一(0,191)从屏幕左上角向下画一条直线。LINE一(639,191)从刚才最后画到的那一点开始（这里是(0,191)）画一条线到点(639,191)。同样，LINE一(639,0)是画一条连结点(639,191)和(639,0)的线。LINE语句还有其他一些选择项。有几个将在下面的程序中加以说明。

PRESET和PSET的作用也许没有LINE那样明显。目前我们可以说，PSET是用现在的前景色（没有加以指定；缺省色为白色）在规定位置上画一个点。PRESET(X, Y)实际上也是在(X, Y)处画一个点，不过是用背景色来画。当然，用背景色画一个点的作用就是抹掉这个点。

### DRAW语句

用PSET, PRESET和LINE语句就可画出多种多样的图来。然而，在IBM PC BASIC中的绘图语句不只是这几个。程序例子P1-3就是利用DRAW语句得到简单图象的动画的一个例子。行150和180中的DRAW语句利用这些行中和字符串ST\$中的编码指令，字符串ST\$是在行130和140中定义的。在第三章中还要给出一些用DRAW语句的例子。

#### 程序P1-3：脉动

```
10 'PROGRAM P1-3 (PULSATE)
20 'Uses DRAW graphics
100 '=====
110 KEY OFF; SCREEN 1,0; CLS
120 SC=2; INC=2
130 ST$="BH4 R2 D8 L2 U2 R8 D2 L2"
140 ST$=ST$+"U8 R2 D2 L8 U2"
150 DRAW"BM 160, 100; C1; S=SC; XST$; "
160 OLDSC=SC; SC=SC+INC
170 IF SC=2 OR SC=40 THEN INC=-INC
180 DRAW "BM 160, 100; CO; S=OLDSC; XST$; "
190 IF INKEY$ < > "" THEN END按任何一个键，程序便结束
200 GOTO 150
```

### 小妖精

程序P1-4所画的图是以另一种方式画的，它用的是文本显示屏上的字符绘图方式。这个程序利用绘图字符（列于IBM PC BASIC说明书的附录G）画出了一个小妖精。小妖精的各个部分由在行130—190中定义的字符串变量所规定。为了画出这个小妖精来，我们把各相应部分用PRINT语句显示在屏幕上（行330—360）。

#### 注释和建议

1. 当把小妖精的各个部分显示出来时，在每一部分的前后各加两个空格（定义在S\$中）。这样，就可以产生此图象在屏幕上水平方向移动的动画。空格的作用是，在新的位置上显示出小妖精的一个部分时，同时把原来的图象抹去。

2. 程序P1-4也利用了一种称之为翻页的方法，使得到的动画更为连贯。为看出翻页的好处，可在行320的SCREEN之前插入一个撇号，再运行这个程序。

3. 在第五章中还要更多地用到字符绘图和翻页方法。在那里讨论动画方法时，将再

一次引用这个例子（作为程序P5-3）。

程序P1-4：小妖精

```
10 'PROGRAM P1-4 (GREMLIN)
20 'Animated character graphics
100 '=====
110 KEY OFF: SCREEN 0, 1, 0: COLOR 4, 1, 14
120 WIDTH 40: LOCATE, , 0: CLS: SCREEN, , 1: CLS
130 S$=CHR$(32)+CHR$(32)
140 A$=CHR$(8)+CHR$(8)
150 B$=CHR$(219)+CHR$(219)
160 C$(0)=CHR$(188)+CHR$(186)
170 D$(0)=CHR$(32)+CHR$(200)
180 C$(1)=CHR$(168)+CHR$(200)
190 D$(1)=CHR$(188)
200 A=2: B=34: C=1
210 '=====
220 FOR I=A TO B STEP C
230 GOSUB 320
240 FOR K=1 TO 50: NEXT K
250 NEXT I
260 FOR L=1 TO 30
270 GOSUB 320
280 NEXT L
290 SWAP A,B: C=-C 改变方向
300 GOTO 220
310 '=====
320 SCREEN, , J, 1-J
330 LOCATE 5, I: PRINT S$+A$+S$
340 LOCATE 6, I: PRINT S$+B$+S$
350 LOCATE 7, I: PRINT S$+C$(J)+S$
360 LOCATE 8, I: PRINT S$+D$(J)+S$
370 J=1-J
380 IF INKEY$( < > )="" THEN SCREEN 0, 1, 0, 0: END
390 RETURN
```

## 太空飞船

下面的这个程序画出一艘儿童画的太空飞船。

程序P1-5：太空飞船

```
10 'PROGRAM P1-5 (SPACE SHIP)
```

```

20 'Uses LINE to draw a simple shape
100 '=====
110 KEY OFF; SCREEN 2; CLS
120 LINE(250,180)-(250,50)
130 LINE-(320,20)
140 LINE-(390,50)
150 LINE-(390,180)
160 LINE(390,120)-(470,180)
170 LINE-(170,180)
180 LINE-(250,120)

```

这种程序对于理解图象屏幕的工作是很有价值的。先在图画纸上粗线条地画一下，会有助于你想出这个程序是怎么写的。

### 彩色线段

就设计而言，下面这个程序是很简单的。它在屏幕上的随机位置画彩色线段。这是我们使用中分辨率屏幕（屏幕1）的第一个程序。这种屏幕有200个光栅行，每行包含320个像素。尽管高分辨图象屏幕（屏幕2）上不能使用彩色，但在这种情况下可以变一下。

行130和140为行160所画的线段选择随机的端点。在画各个线段时，使用不同的颜色。

程序P1-6：彩色线段

```

10 'PROGRAM P1-6 (COLORED LINES)
20 'Drops random colored lines
100 '=====
110 KEY OFF; SCREEN 1, 0; CLS
120 RANDOMIZE TIMER
130 X1=320* RND; Y1=200* RND
140 X2=320* RND; Y2=200* RND
150 C=C+1
160 LINE (X1, Y1)-(X2, Y2), C MOD 4
170 IF C<400 THEN GOTO 130
180 COLOR B MOD 16; CLS
190 B=B+1; C=0
200 GOTO 130

```

现在我们来看一个设计者程序。

程序P1-7：云纹图案

```

10 'PROGRAM P1-7 (MOIRE)
20 'Draws overlapping random patterns
100 '=====

```

```

110 KEY OFF; SCREEN 1,0; CLS; A=5
120 X1=320*RND; Y1=200*RND
130 C=C+1; IF C=4 THEN C=0; P=1-P; IF P THEN
    B=B+1; IF HEN B=0
140 COLOR B,P; Y=199
150 FOR X=0 TO 319 STEP A
160 LINE (X, 199)-(X1, Y1), C; LINE-(319-X, 0), C
170 NEXT X
180 FOR Y=0 TO 199 STEP A
190 LINE (319,199-Y)-(X1, Y1), C; LINE-(0, Y), C
200 NEXT Y
210 FOR I=1 TO 500; NEXT I
220 IF INKEY$ ( ) " THEN END
230 GOTO 120

```

当运行上面这个程序时，你将会看到互相重叠的彩色线段可能会得到非常漂亮的图案。行120为设计确定一个随机的中心。这个程序画出一个接一个的设计图，后一个重叠在前一个上，各个设计图的色彩是不同的（行130和140）。行210使程序运行到那里暂停片刻，以便在图案被覆盖之前有时间将它观察一番。

## 曲线

用点和直线段可以构成所有的图象。用一些直线段作近似就能画出曲线来。下面这个例子就说明这种画曲线的方法。

程序P1-8: 曲线

```

10 'PROGRAM P1-8(CURVE)
20 'STRAIGHT LINES - - >CURVE
100 '=====
110 KEY OFF; SCREEN 2; CLS
120 FOR Y=0 TO 199 STEP 4
130 LINE (0, Y)-(639* Y/199,199)
140 NEXT Y
150 IF INKEY$ ="" THEN 150在按键之前一直让光标在屏幕之外

```

很清楚，这个程序只画了一些直线段（行130），可是所产生的图象却近似一根曲线。

## 螺旋圆环

虽然可以用CIRCLE语句画圆（程序P1-10就是这样做的），但也有一些情况，直接计算圆上的点要更好一些。第七章中介绍几种这样做的方法。程序P1-9用其中的一个方法画一系列同心圆。算出圆上的各点之后，用颜色可变的一些短线段把这些点联起来，结果

是一个色彩艳丽的螺旋圆环。

在行120—140中定义的各个变量，用来计算圆上各点（行180）并画各个直线段（行200）。所用的颜色由行190控制。

程序P1-9：螺旋圆环

```
10 'PROGRAM P1-9 (SPIRAL BAND)
20 'Draws a colorful circular band
100 '=====
110 KEY OFF: SCREEN 1,0: CLS
120 DT= .1: C=COS(DT): S=SIN(DT)
130 SC=5/6: CX=160: CY=100
140 X=0: Y=90
150 PSET(X+CX, SC* Y+CY)
160 FOR J=1 TO 20
170 FOR I=1 TO 63
180 T=X* C-Y* S: Y=Y* C+X* S: X=T
190 CO=CO+ .5
200 LINE-(X+CX, SC* Y+CY), CO MOD 4
210 NEXT I
220 Y=Y-2
230 NEXT J
```

## 动画

在这一章最后一个例子中，我们将在中分辨图象屏幕上画一辆汽车，并让它在屏幕上横向移动。程序用了几个新的语句，也有一些是在旧的语句上作了修改。在行180, 200和220中用CIRCLE语句画汽车的车轮和圆形行李箱。PAINT语句（行190,210,230）用规定的颜色填满这些圆的内部。最后，GET语句（行260）把描绘汽车的数据存在一个数组中，PUT语句模拟动画片，把这辆汽车放在不同的新位置上。

程序P1-10：动画汽车

```
10 'PROGRAM P1-10 (ANIMATED CAR)
20 'Simple animation
100 '=====
110 PI=3.141593
120 KEY OFF: SCREEN 1,0: COLOR 1,4: CLS
130 'Draw a car.
140 LINE (25, 19)-(80,35),2,BF
150 LINE (25, 5)-(50,20),2,BF
160 LINE (3,33)-(82,35),2,BF
170 LINE (32,7)-(48,18),,BF
```



```
180 CIRCLE (25,35), 20,2,-PI/2,-PI
190 PAINT (22,32),2
200 CIRCLE (25,35),8
210 PAINT (25,35)
220 CIRCLE (70,35),8
230 PAINT (70,35)
240 'Store it.
250 DIM A(268)
260 GET (1,5)-(84,55), A
270 'Move it.
280 FOR I=1 TO 200
290 PUT (I,5), A, PSET
300 NEXT I
```

### 总结

这一章的目的是给出一些产生各种图象的简短程序例子。虽然这些例子并不是很复杂的，但有几个却用了并非一看就懂的方法。这些方法将在以后各章中解释。你往下看下去，这些程序就会清楚了，你很快就有能力改进这些程序了。