

计算机 绘图原理

莫春柳 张应丽 编著

暨南大学出版社

计算机绘图原理

莫春柳 张应丽 编著

暨南大学出版社

图书在版编目 (CIP) 数据

计算机绘图原理/莫春柳 张应丽编著. —广州:
暨南大学出版社, 1999.1

ISBN 7-81029-763-5

I. 计…

II. ①莫… ②张…

III. 电子计算机—系统设计—制图

IV. TP302.4

JS180/08

出版: 暨南大学出版社

排版: 暨南大学出版社照排中心

印刷: 封开县人民印刷厂印刷

经销: 新华书店

开本: 787×1092

字数: 14 万

印张: 7

版次: 1999 年 1 月第 1 版 1999 年 3 月第 1 次印刷

印数: 1—3000 册

定价: 14.00 元

前 言

工程制图是工科院校学生的必修课。近年来由于计算机技术的发展,计算机绘图已逐步取代手工绘图。根据新形势,国家教委于1995年对高等院校工程制图课教学基本要求进行了修订,要求培养利用计算机绘图的能力,初步掌握简单图形绘图程序的编制。根据教委的新要求,经过总结多年在计算机绘图教学方面的经验,并吸收计算机绘图技术的最新知识,编写了这本书。

本书系统地介绍了计算机绘图的基本原理,并结合工程制图的要求,介绍了图形裁剪、绘制局部视图、填充剖面线、三视图的形成、轴测图和透视图的绘制等内容,同时还介绍了实用程序常用的编程方法。

书中所有程序都是在 *Quick BASIC* 环境下调试、编译的。*Quick BASIC* 是一种结构化的 *BASIC* 语言,与 *GW-BASIC*, *BASICA* 等具有很好的兼容性,读者只要掌握 *BASIC* 的基本知识,就可以读懂书中的程序,完成本书内容的学习。

本书可用作工科大专院校本科、专科非计算机专业学生的教学用书,建议安排32~35学时,其中上机时数应不少于10学时;同时本书还可以作为从事计算机绘图工作的人员的参考用书。

本书前言、第三章、第四章、第五章、第六章、附录1和附录2由莫春柳编写,第一章,第二章由张应丽编写。编写过程中,曾参考过一些有关计算机绘图和 *Quick BASIC* 语言的教材和书籍,在此向给过我们帮助的同行人表示感谢。

由于作者水平有限,书中难免有错误和不足,恳请读者批评指正。

目 录

第一章 绪 论	(1)
1.1 计算机绘图的概况	(1)
1.2 微机绘图系统	(2)
第二章 图形语句	(5)
2.1 屏幕参数设置语句	(5)
2.2 绘图语句	(7)
第三章 二维图形的几何变换	(15)
3.1 矩阵相乘的实现	(15)
3.2 几何变换	(16)
3.3 应用举例	(21)
3.4 绘图子程序的绘制	(23)
第四章 平面图形的构成	(26)
4.1 不规则多边形的生成	(26)
4.2 平面图形的裁剪	(30)
4.3 平面图形填充剖面线	(37)
4.4 实现动画的基本方法	(41)
第五章 平面曲线	(46)
5.1 圆、圆弧	(46)
5.2 圆弧连接	(49)
5.3 规则曲线	(55)
5.4 实验曲线	(59)
第六章 三维立体的变换及隐藏线消除	(69)
6.1 立体的几何变换	(69)
6.2 矩阵变换的应用	(73)
6.3 图形的表示与数据结构	(78)
6.4 凸多面体的隐线消除	(81)
附录一 Quick Basic 编程环境简介	(91)
附录二	(97)

第一章 绪论

1.1 计算机绘图的概况

一、计算机图形学的发展

计算机是一种能高速、精确地进行大量数字处理和运算的工具，当人们探讨数字与图形之间的内在关系后，利用计算机进行图、数之间转换，这就为制图工作的计算机化提供理论基础。

计算机绘图的研究开始于 50 年代，早在 1950 年，美国麻省理工学院在风一号计算机上配备了一台类似示波器上所用的阴极射线管来显示一简单图形，计算机绘图作为一门学科——计算机图形学开始形成。这个阶段由于设备的限制，输出设备仅具有简单的显示功能，被称为被动式计算机图形学阶段。

1958 年美国 GerBer 公司根据数控机的原理生产了世界上第一台平台式绘图机，改变了电子计算机只能输出数字、文字、符号的状况，而能直接输出图形了。到 50 年代末期美国麻省理工学院林肯实验室开发的 SAGE 空中防御系统第一次使用了具有指挥和控制功能的 CRT 显示器，使用者可以在屏幕上用光笔指示出被确定的目标，从而宣告了交互式计算机图形学阶段的开始。1960 年美国 Calcomp 公司研制了第一台滚筒式绘图机，1973 年，XYNETICS 平面电机型绘图机问世，奠定了绘图机的三大结构形式。60 年代末期出现的存贮管式图形显示器和 70 年代中期出现的光栅扫描图形显示器及各种输入设备，为这一时期交互图形学的大发展提供了物质基础。1963 年美国麻省理工学院伊凡萨泽兰 (Ivansottherland) 博士，发表了论文《SKETCHPAD：一个人机通信的图形形成》，正式奠定了交互式图形学方式。随着人工智能技术与计算机图形学结合，将进入计算机图形学的第三阶段——智能计算机图形学。

二、计算机绘图的应用

由于图形学及计算机设备的发展，图形处理技术也不断发展，从二维线框模型，发展到三维表面模型、三维实体模型、复杂的光照模型以及运动模拟等。加之交互式的绘图系统的出现与发展，引起工业产品的设计与制造的革命性变革，出现了计算机辅助设计 (CAD)、计算机辅助制造 (CAM)、计算机辅助工程 (CAE) 以及集设计、制造、管理于一体的 CIMS 系统，被广泛地应用于许多行业：

在机械行业中，用来设计绘制零件图、轴测体结构装配图、处理造型图以及传统系统图、电气系统图等；

在建筑行业中，用来设计绘制施工图、结构图和透视图；

在汽车、飞机和船舶工业中，可用于设计外形模线及透视图等；

在地质测量方面，绘制地形图、等高线图；

在电子工业方面，绘制集成电路掩膜图；

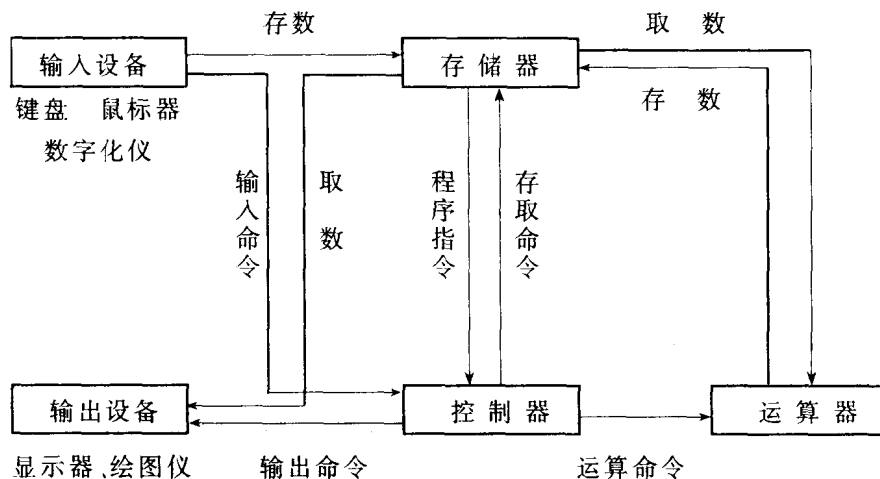
在气象、海洋研究方面，绘制天气图、潮流、潮汐图；

在经济、统计方面，绘制各种统计图表；

还可以在科研工作中进行过程模拟，在各种媒体上进行艺术动画宣传等。

1.2 微机绘图系统

通用的微型计算机的设备和具有图形输入和输出功能的图形设备，安装上相应的绘图软件组成一个微机绘图系统。其硬件设备相互之间的连接关系如下图示：



下面从硬件和软件两方面，介绍一下微机图形系统的工作原理：

一、硬件

1. 中央处理器 (CPU)

CPU 是计算机的核心部件，它主要由控制器和运算器组成，其中控制器用于控制计算机各部件按预定的步骤（程序指令）自动地进行指令的操作，根据 CPU 的字长可以把 CPU 分为 8 位，16 位，32 位和 64 位，CPU 的字长与数据传输总线的宽度相匹配，决定计算机数据传输的流量及处理信息的能力。

CPU 的内部结构和工作的时钟频率，决定 CPU 执行指令的速度，时钟频率越高，CPU 执行指令速度越快。

2. 存储器

存储器分为内存储器 and 外存储器两部分，简称内存和外存。

计算机内存分随机存储器 (RAM) 和只读存储器 (ROM) 两种。在计算机主机中，它们直接与运算器、控制器联系，容量虽小，但运行速度快。我们输入的数据和程序一般存放在 RAM 内，一旦断电，在 RAM 中的信息将全部消失。因此，需要保留的数据和程序，应及时存入外存储器。

内存的存取速度快，容量可根据用户的需要配置。多为 4M，8M，16M 等，最大编

址范围可达到 64KM。

计算机外存储器，随着软件的集成化和要求处理图形的需要，为克服内存容量有限且数据易失的缺点，微机一般都配有容量更大，可以保存数据的外部存储器。计算机外存储器与内存储器一样，分为可读写与只读两类；可读写存储器常用的有软磁盘驱动器和硬磁盘驱动器，普通 3 寸软盘的容量为 1.44M，有一种称为 ZIP 的软磁盘驱动器其容量达 100M。硬磁盘驱动器，目前多使用 3 寸高容量磁盘。

可读写光盘（俗称 MO）常用的容量有 230M、640M、1.3G 等。

只读外存储器常用的有 CD-ROM 已从两倍速发展到 24 倍速甚至更高倍速。

3. 显示器

显示器用来显示计算机内部信息，是人机交流的窗口。为不影响主机的数据处理能力，显示器通常是作为外围设备而独立存在的，它通过标准的串行接口与主机连接。目前使用最多的是阴极射线管（CRT）显示器，采用光栅扫描的工作方式。

光栅扫描有逐行扫描和隔行扫描两种方式。隔行扫描在不提高水平扫描速度的条件下，增加扫描的线数；但隔行扫描由于线路复杂，画面易闪烁，所以一般很少采用隔行扫描方式。

分辨率是衡量显示器性能的一个重要指标，通常所说的分辨率是指整个屏幕可显示的最大象素数目，象素是指屏幕画面上的每一个点。

显示器要配有图形显示卡（适配器），在图形软件的支持下才能显示图形，下面介绍几种较常用的图形显示卡：

（1）CGA 彩色图形显示卡：

它有三种图形显示方式：

方式 0：320×200 彩色（四色），对应象素 64000 点

方式 1：320×200 单色，对应象素 64000 点

方式 2：640×200 单色，对应象素 128000 点

显示缓冲区（VRAM）为 16KB，存储一帧信息，以方式 0 工作时，每一个象素点 2 个二进制位，所以可显示出四种颜色；以方式 1 工作时，每个点占用一个二进制位。

（2）EGA 增强型彩色图形显示卡

它除与 CGA 兼容外，增加以下显示方式

方式 3：320×200 彩色（16 色）

方式 4：640×200 彩色（16 色）

方式 5：640×350 单色

方式 6：640×350 彩色（16 色）

以方式 3、4 工作时，VRAM 容量为 4×16KB；以方式 5 工作时，VRAM 容量为 2×16KB；以方式 6 工作时，VRAM 容量 4×16KB。

（3）VGA（Video Graphics Array）卡

它除与 EGA 兼容外，增加以下显示方式

方式 7：640×480 黑白二色

方式 8：640×480 彩色 16 色

方式 9：320×200 彩色 256 色

VGA 的 VRAM 需 $4 \times 64\text{KB}$ 的容量。

(4) TVGA, PVGA 均是 Super VGA 产品, 与 VGA 兼容且有更高的分辨率和更多的色彩选择, 其最大分辨率为 1024×768 , 256 色。VRAM 容量为 $12 \times 64\text{KB}$ 。

4. 键盘

键盘是很常用的输入设备。下面介绍几个 DOS 的专用组合键和程序编辑常用键:

< Ctrl > + < Alt > + < Del > : 使系统热启动。

< Pause > : 暂停当前操作, 按任意键后继续运行。

< Ctrl > + < Break > : 中断当前进行的操作。

< Home > : 使光标移到当前行的句首。

< End > : 使光标移到光标所在行的末尾。

< Esc > : 退出或废除当前行, 等待新的输入。

< Del > : 删除当前光标位置上的字符, 使后继字符向左移一位。

< Ins > : 设置或解除字符插入方式。

< ← > : 删除当前光标位置前的字符。

< Space Bar > : 空格键

二、软件

计算机软件分为两大类, 系统软件和应用软件。系统软件包括各种语言和它们的解释、编译程序; 系统的监控管理、调试、故障检查和诊断程序。常用的操作系统有 MS-DOS、Unix、Windows 等。应用软件是计算机用户编制的用于解决各种实际问题的程序。将应用软件标准化模块化, 形成解决某个领域多种问题的应用程序组合, 这就称为软件包。现在被广泛应用的 AutoCAD、3D Studio 等, 就是典型的应用软件。

本书后续章节将要介绍 Quick Basic 编程环境。QBASIC 属于系统软件, 本书中其他章节介绍的程序段是以 QBASIC 作支撑软件, 在 QBASIC 环境下编译、调试、运行的。

第二章 图形语句

2.1 屏幕参数设置语句

一、SCREEN 语句

功能：根据配置的显示器和适配卡，设置屏幕的显示方式以及相应的参数，为计算机绘图提供一种图形显示的环境

格式：SCREEN [方式] [, 开关] [, 页号 1] [, 页号 2]

说明：方式——显示方式代码。可取值为 0, 1, 2, 7, 8, 9, 10, 11, 12, 13。

开关——选择彩色显示和单色显示的开关。取值范围为 0 ~ 255。当该值为 0 时，选择彩色显示，非零时选择单色显示。该开关只有在方式 0 下起作用。

页号 1——指出绘图语句要写入的屏幕页号，不同的适配卡有不同的页数值，如 IBM 视频图形数组适配器 (VGA) 在方式 0 下，可以有 4 或者 8 页。

页号 2——指出要显示的屏幕页号。

下面介绍几种常用代码对应的屏幕显示方式：

SCREEN 0——文本方式。根据显示器不同的分辨率适配卡，对 8×8 字符可显示的文本大小为 40×25, 40×43, 40×50, 80×43 或 80×50 等。

SCREEN 1——中分辨率绘图方式，320×200 个像素点。有 4 种颜色属性，设定显示方式与 CGA 彩卡显示方式 0 对应。

SCREEN 2——高分辨率绘图方式 640×200 个像素点，图形只有两种颜色属性，设定的显示方式与 CGA 彩卡显示方式 2 对应。

SCREEN 12——特高分辨率绘图方式 640×480 个像素点，有 16 种颜色属性，设定的显示方式与 VGA 彩卡显示方式 8 对应。

二、COLOR 语句

功能：选择图形显示的颜色。

格式：在不同的屏幕显示方式下 COLOR 语句的格式不同。

SCREEN 0 及 SCREEN 7 至 SCREEN 10 各种方式下：

COLOR [前景色] [, 背景色]

SCREEN 1 方式下：

COLOR [背景色] [, 调色板代码]

SCREEN 12 和 SCREEN 13 方式下：

COLOR [前景色]

说明：前景色——当前属性下字符或图形的显示颜色

背景色——当前属性下屏幕背景的颜色

调色板——当前颜色的属性。

下面列出在常用的几种显示方式下，COLOR 语句中各种参数的取值范围

SCREEN 0：前景色取值范围是 0~31，背景色取值范围是 0~7。

SCREEN 1：背景色取值范围是 0~15，调色板号取值范围为 0~255。决定颜色的两种属性，即有两块调色板，奇数为一种属性，偶数为另一种属性，每块调色板下有 4 个颜色代码。

SCREEN 2：不能使用 COLOR 语句，否则出错。

SCREEN 12：前景色取值范围为 0~15，代表一个颜色属性码。

在 SCREEN 1 的显示方式下，背景色数值代码与不同调色板下颜色代码所对应的颜色，如表 2.1 和表 2.2 所示。

表 2.1 背景色数值代码

代码	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
颜色	黑	蓝	绿	青	红	品红	棕	深灰	浅灰	浅蓝	浅绿	浅青	淡红	淡品红	黄	白

表 2.2 不同调色板下颜色代码

色码	偶数调色板	奇数调色板
0	与背景色相同	与背景色相同
1	绿	青
2	红	品红
3	黄	白

三、WINDOW 语句

功能：重新定义屏幕坐标系

格式：WINDOW [[SCREEN] (x1, y1) - (x2, y2)]

说明：SCREEN——任选项。有此选项，屏幕坐标系原点在左上角，X 轴由左至右为正向，Y 轴从上至下为正向，无此选项，原点在左下角，X 轴正向不变，Y 轴从下至上为正向。

(x1, y1) - (x2, y2)——重新定义的屏幕左上角，右下角（有 SCREEN 选项时）或左下角，右上角的坐标。

在 WINDOW 语句后使用的绘图语句的坐标，都是相对重新定义的坐标系而言的。

四、VIEW 语句

功能：定义图形显示的边界，即一个由指定坐标确定的矩形取景框

格式：VIEW [[SCREEN] (x1, y1) - (x2, y2) [, [颜色] [, 边框色]]]

说明：SCREEN——任选项。选择该项时，把当前定义的 WINDOW 区域内的图形映射到整个屏幕上，但只显示位于取景框内的图形，此时各点坐标是屏幕坐标系下的绝对坐标；不选该项时，将 WINDOW 区域内的图形全部映射到取景框内，此时各点坐标是相对于取景框的相对坐标。

$(x1, y1) - (x2, y2)$ ——为视口即取景框的左上角和右下角的坐标，它们的取值必须小于屏幕最大坐标值。

颜色——指出矩形取景框要填充的颜色代码，若缺省，则不填充。

边框色——指出绘制矩形取景框边界的颜色代码，若缺省，则不加边框线。

五、LOCATE 语句

功能：在不同的屏幕显示方式下，确定光标的位置。

格式：LOCATE 行号，列号

说明：行号——确定字符所在行的位置，在 SCREEN 1 和 SCREEN 2 的屏幕显示方式下，取值范围为 1 ~ 25，在 SCREEN 12 的显示方式下，取值范围为 1 ~ 30。

列号——确定字符所在列的位置，在 SCREEN 1 的显示方式下，取值为 1 ~ 40，在 SCREEN 2 和 SCREEN 12 的显示方式下，取值范围为 1 ~ 80。

六、CLS 语句

功能：清除屏幕上除原程序外的所有字符和图形。可以作为语句写在程序中，也可作为命令使用。

2.2 绘图语句

一、PSET 语句

功能：在屏幕上点亮指定的坐标点。

格式：PSET [step] (x, y) [, 颜色]

说明：(x, y)——点的坐标值。

step——任选项，选择该项时，(x, y) 所给出的是相对坐标；不选择该项时，(x, y) 给出的是在屏幕坐标系下的绝对坐标。

颜色——指定亮点的颜色。缺省时为当前颜色。

二、PRESET 语句

功能：在屏幕上点亮指定的坐标点。当颜色缺省时，亮点颜色为背景颜色

格式：PRESET [step] (x, y) [, 颜色]

说明：PRESET 语句 PSET 语句功能是一样的，区别在于当颜色缺省时，PSET 用当前的前景色画点，而 PRESET 用背景色画点，此时 PRESET 的作用是抹点。

下例说明上述两语句的作用：

例 1. 点在屏幕上任意位置闪烁

```
SCREEN 1
10 X=INT (319×RND); Y=INT (199×RND)
   C=INT (3×RND)
   PSET (x, y), C
   FOR DELAY=1 TO 500; NEXT
   PRESET (x, y)
   IF INKEY $ = " " THEN 10 ELSE 20
20 END
```

三、LINE 语句

功能：在屏幕上划一条直线或一个矩形，或一个填充的矩形。

格式：LINE [[step] (x1, y1)] - [step] (x2, y2) [, [颜色] [, [B [F]] [, 线型]]]

说明：(x1, y1) ——直线的起始坐标；

(x2, y2) ——直线的终止点坐标。

颜色——直线或矩形框的颜色代码，SCREEN 1 显示方式下取值为 0 ~ 3，
SCREEN 12 显示方式取值为 0 ~ 15。

step——任选项，选择该项时，表示其后的坐标是相对坐标。

B——任选项，选择该项时，以 (x1, y1), (x2, y2) 为对角顶点画一个矩形。

BF——任选项，选择该时，表示以 (x1, y1), (x2, y2) 为对角顶点画一个矩形，并以指定颜色或当前色填充该矩形。

线型——一个十六进制的数串，用来记录屏幕上象素点的亮暗情况，以画出点划线，虚线等不同线型的线。缺省线型为实线。

确定线型数串的方法：

屏幕上各象素的亮暗情况是由二进制数来决定的，0 表示某点不亮，1 表示该点亮。取相邻的 16 个象素点为一个线型单元，如要画一段点划线，象素点亮或暗的排列应该为

1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

以一位十六进制数记录 4 位二进制数，上述象素点亮暗情况排列可用四位十六进制数记录，记为 &HFFCC。

按上述方法，通过改变二进数的排列可以构造所需的线型。

下面几个例子是 LINE 语句的不同用法。

例 2. 用 LINE 语句画一个三角形，三个顶点坐标为 (40, 50), (40, 150), (200, 150)。

```
SCREEN 1; COLOR 0, 1
```

```
LINE (40, 50) - (40, 150), 1 ( 或 LINE (40, 50) - STEP (0, 100), 1)
```

```

LINE - (200, 150), 1 ( 或 LINE - STEP (160, 0), 1)
LINE - (40, 50), 1 ( 或 LINE - STEP (-160, -100), 1)
END

```

例 3. 画一个点划线方框, 顶点坐标为 (80, 60) - (280, 160)

```

SCREEN 1: COLOR 0, 1
LINE (80, 60) - (280, 160), 1, B, &HFFCC
END

```

例 4. 变化的彩色方框

```

SCREEN 12
FOR I=1 TO 20
CLS: C = INT (RND * 15)
x1 = INT (RND * 639): y1 = INT (RND * 479)
x2 = INT (RND * 639): y2 = INT (RND * 479)
LINE (x1, y1) - (x2, y2), C, BF
FOR DELAY = 1 TO 500: NEXT
NEXT I
END

```

四、DRAW 语句

功能: 根据字符串表达式的定义在屏幕上画图

格式: DRAW 字符串表达式

说明: 字符串表达式——由绘图子命令组合在一起的字符序列。

绘图子命令有多种, 可分为两类:

1. 光标移动命令:

(1) 前缀命令: 出现于其他命令的首部, 以指定光标移动的性质。

B——表示其后的光标移动命令只移动光标不画线。

N——表示其后的光标移动命令, 移动光标画线后光标又返回起始位置。

(2) 移动命令: 指出光标移动的方向和距离:

U [n] ——向上移动 n 个单位 D [n] ——向下移动 n 个单位

L [n] ——向左移动 n 个单位 R [n] ——向右移动 n 个单位

E [n] ——向右上移动 n 个单位 F [n] ——向右下移动 n 个单位

G [n] ——向左下移动 n 个单位 H [n] ——向左上移动 n 个单位

Mx, y——移动画线命令, 若 x, y 前有 ± 号, 则表示相对坐标, 无符号则表示绝对坐标。

上列移动命令中, 若 n 缺省, 则为 1; 另光标移动 n 个单位是相对光标当前位置而言的。

2. 角度、颜色比例命令

An——设置旋转角度, n 的取值范围为 0 ~ 3, 0 = 0, 1 = 90, 2 = 180, 3 = 270

TAn——设置旋转角度，n 取值范围为 $-360 \sim 360$ 。取正数为逆时针旋转，负数为顺时针旋转。

Cn——设置颜色，n 为颜色代码，缺省时为当前颜色。

Sn——设置比例因子，n 取值为 $1 \sim 255$ 。比例因子与移动命令中的移动单位值 n 相乘，即为实际移动的距离。无 Sn 命令时，比例因子自动取为 4，即 $n=4$ 时，一个单位就是一个光点位置。

x 串表达式——子串命令，允许在串表达式内再用 x 带出另一个串表达式。]]

例 5. 画一颗五角星图案

SCREEN 12

DRAW "BM300, 100"

DRAW "C1; S12; BM - 2, 0; M + 7, 17; M - 17, - 12; M + 20, 0; M - 17, 12; M + 7, - 17"

END

五、CIRCLE 语句

功能：根据给定的参数画圆或椭圆。

格式：CIRCLE [STEP] (x, y), 半径 [, [颜色] [, [起始角, 终止角] [, 比率]]]

说明：STEP——任选择；选择该项则指明其后的 (x, y) 是相对坐标。

(x, y) ——圆心坐标。

半径——所画圆的半径，或椭圆的长半轴。

颜色——所画圆或椭圆的颜色。

起始角——所画圆弧段的起始角度，取值范围为 $-2\pi \sim 2\pi$

终止角——所画圆弧段的终止角度，取值范围为 $-2\pi \sim 2\pi$

当起始角或终止角为负值时，该语句的作用是将圆弧两端与圆心画线连接，构成扇形。

比率——Y 轴半径与 X 轴半径的比值；画圆时用来调整因分辨率不同造成圆的失真，在不同的屏幕显示方式下比率值自动进行调整；画椭圆时为平行 Y 轴的半轴与平行 X 轴的半轴的比率。

若利用 CIRCLE 语句画圆，在 SCREEN 1 时，比率值为 $5/6$ ，SCREEN 2 时比率值为 $5/12$ ，SCREEN 12 时，比率值为 $3/4$ 。

若利用 CIRCLE 语句绘制椭圆时，当比率值大于 1，则椭圆长轴平行于 Y 轴；小于 1，则长轴平行于 X 轴。

例 6. 画一些由椭圆、扇形组成的图案，并比较 VIEW 语句有无 SCREEN 选项的区别。

SCREEN 12: PI = 3.14159

WINDOW SCREEN (0, 0) - (640, 480): GOSUB c

10 IF INKEY\$ = " " THEN 10

CLS

VIEW SCREEN (10, 10) - (300, 180),, 5: GOSUB c

```

VIEW (300, 200) - (600, 440), , 6: GOSUB c
END
c:  CIRCLE (160, 100), 80, 1, , , 3
    CIRCLE (80, 100), 80, 2, , , 3/10
    CIRCLE (320, 200), 200, 3, -PI, -PI/2
    CIRCLE STEP (-100, -42), 100
RETURN

```

六、PAINT 语句

功能：用指定的图案或颜色填充封闭的图形区域。

下面按语句的不同功用介绍该语句的格式

1. 填色

格式：PAINT (x, y) [, 色码], 边色

说明：(x, y) ——要填充区域内任意一点的坐标值。

色码——不同屏幕方式下有效的颜色代码值，指出要填充的区域的颜色，缺省时为当前色。

边色——指出要填充的图形区域的边框颜色。

例 7. 在两个椭圆的共有区域和扇形区域内填上指定颜色。

```

SCREEN 12
CIRCLE (160, 100), 80, 2, , , 3
CIRCLE (180, 100), 80, 2, , , 3/10
PAINT (160, 100), 4, 2
CIRCLE (320, 200), 200, 3, -PI, -PI/2
PAINT (330, 210), 5, 3
END

```

2. 图案填充

格式：PAINT (x, y), 串表达式

说明：串表达式——用 CHR\$ (图形码) 代表一种图案，最多可用 64 种图案构成组合图形，串表达式可写为 CHR\$ (图形码 1) + CHR\$ (图形码 2) + ... + CHR\$ (图形码 n), $n \leq 64$ 。

图形码由二位十六进制数表示。例如，要绘制锯齿形的填充图案，可按下列方法编写串表达式，用两位二进制数控制一光点，当二进制数为 11 时光点亮；为 00 时光点暗。以八位二进制数表示一个基本图案，锯齿形图案由七个基本图案构成：

11 00 00 00		C0
00 11 00 00		30
00 00 11 00		0C
00 00 00 11	转换为	03
00 00 11 00	十六进制数	0C
00 11 00 00		30

锯齿形图案的串表达式可写为:

```
CHR$ (&HC0) + CHR$ (&H30) + CHR$ (&H0C) + CHR$ (&H03) +
CHR$ (&H0C) + CHR$ (&H30) + CHR$ (&HC0)
```

例 8. 用锯齿形图案填充椭圆区域

```
SCREEN 1: COLOR 0, 0
CIRCLE (100, 100), 50, 1, 0, 3/10
CIRCLE (100, 100), 80, 1, 0, 3/10
TIN$ = CHR$ (&HC0) + CHR$ (&H30) + CHR$ (&H0C) + CHR$
$ (&H03) + CHR$ (&H0C)
TIM$ = CHR$ (&H30) + CHR$ (&HC0)
PAINT (160, 100), TIN$ + TIM$ . !
END
```

七、GET 语句

功能: 把指定屏幕范围内的图形存入指定的数组中。

格式: GET (x1, y1) - (x2, y2), 数组名 [下标]

说明: (x1, y1) - (x2, y2) —— 屏幕上某矩形区域的对角点坐标, 该区域应完全包含要存储的图形

数组名——可以是任一类型的数值数组, 用于存储图形的数值。

下标——确定数组存储空间的大小, 要求足以容下整个图形信息。

数组的大小要由矩形区域来决定, 下列给出在不同屏幕显示方式下计算数组下标的公式。

在 SCREEN 1 方式下:

$$N = (4 + \text{INT}((x2 - x1) * 2 + 7) / 8) * ((y2 - y1) + 1)$$

在 SCREEN 2 方式下:

$$N = (4 + \text{INT}((x2 - x1) + 7) / 8) * ((y2 - y1) + 1)$$

在 SCREEN 12 方式下:

$$N = 4 * (4 + \text{INT}((x2 - x1) + 7) / 8) * 4 * ((y2 - y1) + 1)$$

八、PUT 语句

功能: 把 GET 语句存储的图象显示在屏幕上

格式: PUT (x, y), 数组名, 显示方式。

说明: (x, y) —— 显示存储图象窗口的左上角坐标。

数组名——存储图象的数组, 与存图时所用的数组同名。

显示方式——要显示的图象和屏幕上原有图象之间的相互作用方式, 有五种方式, 见表 2.3。