1.1 关于 AutoLISP

AutoLISP 是由 Autodesk 公司开发的一种 LISP 程序语言(LISP 是 List Processor 的缩写)。第一篇关于 LISP 的参考文献是由 John McCarthy 在 1960 年 4 月的《ACM 通讯》中发表的。

除了 FORTRAN 和 COBOL，大多数在六十年代早期开发出来的语言都过时了，可是 LISP 却生下来，并且已经成为人工智能(AI)的首选程序序言。AutoLISP 解释程序位于 AutoCAD 软件包中，然而 AutoCAD R2.17 及更低版本中并不包含 AutoLISP 解释程序，这样，只有通过 AutoCAD R2.18 及更高版本才可以使用 AutoLISP 语言。

AutoCAD 软件包中包含大多数用于产生图形的命令，但仍有某些命令未被提供。例如，AutoCAD 中没有在图形文本对象内绘制矩形及作全局改变的命令。通过 AutoLISP，你可以使用 AutoLISP 程序语言编制能够在图形文本对象内绘制矩形或作全局选择性改变的程序。事实上，可以用 AutoLISP 编制任何程序，或把它嵌入到菜单中，这样定制你的系统会取得更高的效率。

现在，已经有数以百计的第三方软件开发人员使用 AutoLISP 语言编制各种应用程序软件包，例如，本文作者开发了一个名为 SMLayout 的软件包，用它可以产生各种复杂几何图形的平面布局图，这些几何图形包括管道的交叉部、过渡都、圆柱、弯管接头、圆锥以及罐顶。目前社会上非常需要 AutoLISP 程序员为应用软件及客户菜单的开发提供顾问。

在本章中，我们假定读者已经熟悉了 AutoCAD 命令及 AutoCAD 的系统变量。但是，在开始学习 AutoLISP 时，却并不需要你是一位 AutoCAD 或编程专家。同时，本章还假定读者并无编程方面的知识。如果你熟悉任何一种编程语言，那么学习 AutoLISP 就会很容易。对各种函数的评细探讨以及对例题的逐步讲解会使你学起来很有兴趣。本章讨论常用的 AutoLISP 函数以及它们 S 程序编制中的应用。对于本章中未涉及的函数，请参阅 Autodesk 公司的《AutoLISP 程序员参考手册》。AutoLISP 对硬件没有任何特殊要求。如果系统能够运行 AutoCAD，那么同样也可以运行 AutoLISP。AutoLISP 程序可以使用任何文本编辑器进行编制。

1.2 数学运算

任何编程语言都提供数学函数。在 AutoLISP 中，同样提供了编程以及数学计算所需的大部分数学函数，你可以使用 AutoLISP 对数字进行加、减、乘、除运算，还可以得到以弧度表示的角度的正弦值、余弦值及反正切值等。使用 AutoLISP 还可以进行许多其他计算。这一节主要讨论 AutoLISP 程序语言支持的常用数学函数。

1.加法

格式(+ num1 num2 num3…)

此函数(+)计算加号(+)右边所有数字的和(+ num1 num2 num3…)。这些数字可以是整数或实数。如果均为整数，则和为整数；如果均为实数，则和为实数。但是如果既有整数又有实数，则和为实数。如下所示，在前两个例子中，所有数字均为整数，所以结果是整数。在第三个例子中，一个是实数(50.0)，故结果为实数。

示例：

Command:(+2 5)返回 7

Command:(+2 30 4 50)返回 86

Command:(+2 30 4 50.0)返回 86.0

2. 减法

格式(- num1 num2 num3…)

此函数(-)从第一个数中减去第二个数(num1-num2)。如果多于两个数，就用第一个数字减去其后所有数字的和 [num1-(num2+num3…)]。在下面的第一个例子中，28 减去 14 后返回 14。因为两个数均为整数，结果亦为整数。在第三个例子中 20 与 10.0 相加，并用 50 减去两数的和(30.0)，返回一个实数 20.0。

示例：

Command：(- 28 14) 返回 14

Command：(- 25 7 11)返回 7

Command：(- 50 20 10.0)返回 20.0

Command：(- 20 30)返回河 0

Command：(- 20.0 30.0)返回-10.0

3.乘法

格式(* num1 num2 num3…)

此函数(*)计算乘号右边所有数字的乘积(num1×num2×num3…)。若均为整数，它们的乘积亦为整数；若其中含有一个实数，乘积即为实数。

示例：

Command:(* 2 5) 返回 10

Command:(* 2 5 3) 返回 30

Command:(* 25 3 2.0) 返回 60.0

Command: (* 2 -5.5) 返回-11.0

Command: (* 2.0 -5.5 -2) 返回 22.0

4.除法

格式(/ num1 num2 num3…)

此函数(/)用第一个数除以第二个数。如果多于两个数，就用第一个数除以其后所有数的乘积[num1/(num2 × num3 ×…)]。在下面的第四个例子中 ,用 200 除以 5.0 与 4 的乘积[200/(5.0×4)]。

示例：

Command：(/ 30)返回 30

Command：(/ 3 2)返回 1

Command：(/3.0 2) 返回 1.5

Command：(/ 200.0 5.0 4)返回 10.0

Command：(/ 200 -5)返回-40

Command：(/ -200 -5.0)返回 40.0

1.3 增量、减量与绝对数字

1.增量数字

格式(1+ number)

此函数(l+)使数字与 1(整数)相加，返回一个增加 1 的数。在下面的第二个例子中，1 与-10.5 相加返回-9.5。

示例：

(1+ 20)返回 21

(1+ -10.5)返回-9.5

2.减量数字

格式(1- number)

此函数(1-)从数字中减去 1(整数)，并返回一个减去 1 的数。在下面的第二个例子中-10.5 减去 1 返回-11.5。

示例：

(1- 10)返回 9

(1- 10.5)返回-11.5

3.绝对数字

格式(abs num)

abs 函数返回一个数的绝对值。该数可以是整数或者实数。在下面的第二个例子中,由于-20 的绝对值为 20，故函数返回 20。

(abs 20)返回 20

(abs -20)返回 20

(abs -20.5)返回 20.5

1.4 三角函数

1.sin 函数

格式(sin angle)

sin 函数计算一个角(以弧度表示)的正弦值。在下面的第二个例子中，sin 函数计算 Pi(180 度)的正弦值并返回 0。

示例：

Command：(sin 0)返回 0.0

Command：(sin Pi)返回 0.0

Command：(sin 1.0472)返回 0.866027

2.cos 函数

格式(cos angle)

cos 函数计算一个角(以弧度表示)的余弦值。在下面的第三个例子中，cos 函数计算 Pi(180 度)的余弦值并返回-1.0。

示例：

Command：(cos 0)返回 l.0

Command：(cos 0.0)返回 1.0

Command：(cos Pi)返回-1.0

Command：(cos 1.0)返回 0.540302

3.atan 函数

格式(atan num1)

atan 函数计算数的反正切值，返回角度以弧度表示。下面的第二个 atan 函数计算 1.0 的反正切值并返回 0.785398(弧度)。

示例：

Command：(atan 0.5)返回 0.463648

Command：(atan 1.0)返回 0.785398

Command：(atan -1.0)返回-0.785398

4.具有两个参数的 atan 函数

格式(atan　num1　num2)

还可以在 atan 函数中再指定一个数。若指定了第二个数，函数将以弧度形式返回(num1/num2)的反正切值。在下面的第一个例子中，第一个数(0.5)除以第二个数(1.0)，atan 函数计算商(0.5/1=0.5)的反正切值。

示例：

Command：(atan 0.5 1.0)返回 0.463648 弧度

Command：(atan 20 3.0)返回 0.588003 弧度

Command：(atan 2.0 -3.0)返回 2.55359 弧度

Command：(atan -2.0 3.00)返回-0.5880033 弧度

Command：(atan -2.0 -3.0)返回-2.55359 弧度

Command：(atan 1.0 0.0)返回 1.5708 弧度

Command：(atan -0.5 0.0)返回-1.5708 弧度

5.angtos 函数

格式(angtos angle [made [precision]])

angtos 函数以字符串格式返回以弧度表示的角度值。字符串格式由 made 和 precision 的设置决定。

示例：

Command：(angtos 0.588003 0 4)返回”33.6901”

Command：(angtos 2.55359 0 4)返回”145.3099”

Command：(angtos 1.5708 0 4)返回”90.0000”

Command：(angtos -1.5708 0 2)返回”270.00”

注意　在(angtos angle [mode [precision]])中：

angle 是以弧度表示的角度值。

mode 是与 AutoCAD 系统变量 AUNITS 相对应的 angtos 模式。

AutoCAD 中可用模式如下：

ANGTOS 模式	编辑格式
0	十进制角度
1	度/分/秒
2	梯度
3	弧度
4	测量单位

precision 是一个整数，用于控制小数的位数，与 AutoCAD 系统变量 AUPREC 相对应。　其最

小值为 0，最大值为 4。

在上面的第一个例子中，angle 为 0.588003 弧度，mode 为 0(十进制角度)，precision 为 4(小数点后有四位)。函数返回 33.6901。

1.5 关系表达式

在程序中，通常都需要测试某些特定的条件。若条件为真，程序执行某些功能，若不为真，执行另外一些功能。例如，条件表达式(if(< X 5))，若变量 x 的值小于 5，测试结果为真。编程过程中经常要用到这种类型的测试条件。本节讨论在 AutoLISP 编程中要用到的各种关系表达式。

1.等于

格式(= atom1 atom2...)

该函数(=)检查两个元素是否相等。若相等，条件为真，函数返回 T。同样，若指定的元素不相等，条件为假，函数返回 nil。

示例：

(= 5 5)　　返回 T

(= 5　49)　　返回 nil

(= 5.5 5.5 5.5)　返回 T

(="yes" "yes")　返回 T

(="yes" "yes" "no")　返回 nil

2.不等于

格式(/= atom1 atom2...)

该函数(/=)检查两个元素是否不相等。若不相等，条件为真，函数返回 T。同样，若指定的元素相等，条件为假，函数返回 nil。

示例：

(/=50 4)　　返回 T

(/= 50　50)　　返回 nil

(/= 50 -50)　　返回 T

(/= “yes” “no”)　返回 T

3.小于

格式(< atom1 atom2...)

该函数(<)检查第一个元素(atoml)是否小于第 H 个元素(atomZ)。若为真，函数返回 T，否则返回 nil。

示例：

(<　3 5)返回 T

(< 5 3 4)返回 nil

(< “x” “y”)返回 T

4.小于等于

格式(< = atom1 atom2…)

该函数(<=)检查第一个元素(atom1)是否小于等于第二个元素(atom2),若是，函数返回 T，否则返回 nil。

示例：

(< = 10 15)返回 T

(< = “c” “n”)返回 nil

(< = 2.0 0)返回 T

5.大于

格式(> atom1 atom2…)

该函数(>)检查第一个元素(atom1) 是否大于第二个元素(atom2)。若是，函数返回 T，否则返回 nil。在下面第一个例子中，15 大于 10，因此，关系表达式为真，且函数返回 T。在第二个例子中，10 大于 9，但 9 并不大于其后的 9，因此函数返回 nil。

示例：

(> 15 10)返回 T

(>10 9 9)返回 nil

(>” c” “n”)返回 T

6.大于等于

格式(> = atom1 atom2…)

该函数(>=)检查第一个元素(atom1)的值是否大于等于第二个元素(atom2)。若是，函数返回 T，否则返回 nil。在下面第一个例子中，78 大于但木等于 50，因此，函数返回 T。

示例：

(> = 78 50) 返回 T

(> = “x” “y”) 返回 nil

1.6 defun、setq、getpoint 与 Command 函数

1.defun 函数

defun 函数用于在 AutoLISP 程序中定义函数，其格式为：

(defun name [argument])

其中

name……函数名

argument……参数列表

示例：

(defun ADNUM()，定义了一个函数 ADNUM，此函数天参数，亦无局部变量(Local symbols)。这就意味着程序中用到所有变量均为全局变量。全局变量的值在程序结束时不会丢失。

(defun ADNUM (ab c)，定义了一个含有三个参数 a、 n 和 c 的函数 ADNUM。变量 a、 n、 c 从程序外部获取它们的值。

(defun ADNUM(/a n)，定义了一个含有两个局部变量 a 和 n 的函数 ADNUM。局部变量在程序的执行期间保留其值，而且只能在它所在的程序中使用。

(defun C：ADNUM()，在函数名前加上 C：后，此函数就可以通过在 AutoCAD 的 Command：提示符后输入其函数名来执行。如果没有使用 C：，函数名则必须置于圆括号中。

注意

AutoLISP 包含一些内置函数 ,不要使用其中的任一名称作为函数名或变量名 ,以下是一些 AutoLISP 内置函数的保留名称列表。

Abs ads alloc and angle angtos append apply atom ascii assoc atan

Atof atoi distance equal fix float if length list load member nil

Open or pi read repeat reverse set type while

2.setq 函数

setq 函数用于给变量赋值，其格式如下：

(setq name value [Name value] ...)

其中

Name……变量名

value……赋予变量的值

赋予变量的值可以是任何表达式(数字表达式，字符串表达式或既含有字母又含有数字的表达式)。若

该值为字符串，其长度不可超过 100 个字符。

Command：(setq x 12)

Command：(setq x 6.5)

Command：(setq x 8.5 y 12)

在最后一个表达式中，8.5 被赋予变量 X，12 被赋予变量 Y。

Command：(setq answer”YES”)

这个表达式中，字符串值”YES”被赋给变量 answer。

setq 函数还可用于与其他表达式联合为变量赋值。下面的例子 setq 函数被用来为不同的变量赋值。

(setq pt1((getPoint”Enter start Point：))

(setq angl(getangle”Enter Included angle：”))

(setq answer(geststring”Enter YES or NO：”))

注意

不要给 AutoLISP 使用的一些内置函数名及符号赋值。下面的函数是有效的，但由于保留符号 Pi 及 angle 将被重新定义，因此不要使用。

(setq Pi 3.0)

(setq angle…)

3.getpoint 函数

getpoint 函数暂停程序的运行，允许用户输入一个点的 X、Y 坐标或 X、Y、Z 坐标。该点的坐标可以由键盘或使用屏幕光标输入。getpoint 函数的格式为：

(getPoint [Point] [Prompt])

其中 point……输入一个点，或选择一个点

prompt……将显示在屏幕上的提示

示例：

(setq Pt1(getpoint))

(setq Pt1(getPoint”Enter starting Point》

注意

不能输入其他的 AutoLISP 例程名来响应 getpoint 函数。二维或三维的点应考虑定义在当前用户坐标系统(UCS)下。

4.Command 函数

Command 函数用于在 AutoLISP 程序内部执行标准的 AutoCAD 命令。AutoCAD 命令名及命令选项必须置于双引号内。Command 函数的格式为：

(Command “Commandname”)

其中 Command……AutoLISP 函数

Commandname……AutoCAD 命令

示例：

(Command”line” Pt1 Pt2” “)

“line” ……AutoCAD LINE 命令

Pt1……第一点

Pt2……第二点

“ “……用于返回

注意

在 AutoCAD R12 之前的版本中，不能使用 Command 函数执行 AutoCAD 的 PLOT 命令。 例如，(Command”plot”…)是无效表达式。在 AutoCAD 2000、 R14 和 R13 中，才可以通过 Command 函数使用 plot 命令(Command”plot”…)。

Command 函数不能使用 AutoCAD 的 DTEXT 或 TEXT 命令输入数据。（ 可以用 Command 函数发出 DTEXT 及 TEXT 命令，还可以输入文本高度及旋转角度，但却不能在 DTEXT 或 TEXT 命令提示文本输入时输入文本)。

不能通过 Command 函数使用 AutoLISP 的输入函数。这些输入函数为 getpoint、getangle、getstring 及 getint。例如，(Command”getPoint…)和(Command”getangle…)均为无效函数。如果程序中包含这样的函数，在其被装入时就会显示一条错误信息。

例 1

编写一个程序，该程序将提示用户选择三角形的三个顶点，并通过它们绘出如图 12.1 所示的三角形。

多数程序都包含三个基本组成部分，即输入、输出及处理过程。其中处理过程的功能为根据给定的输入来产生预期的输出(见图 12-2)。

编写程序前，必须确认这三部分。

本例中，程序的输入为三个点的坐标，期望的输出为一个三角形。用以生成该三角形的处理过程为：由 P1 到 P2、由 P2 到 P3、到 P3 到 P1 各画一条直线。弄清这三部分就会使编程过程更清晰。

处理过程对于程序的成功起着很重要的作用。有时它很简单，有时却包含复杂的计算。如果程序包含大量运算，就应该把它分成若干个程序(甚至是子程序)，并按逻辑的顺序和系统的顺序安排好它们。同

时请记住，程序需要随时修改，也很有可能被其他程序员修改。因此，应尽可能使程序清晰、明了，以便其他程序员了解程序在其执行过程中的不同阶段在做些什么。如果可能，请给出草图，并且说明要点。

输入	输出
P1 点的位置	
P2 点的位置	三角形 P1，P2，P3
P3 点的位置	

处理过程

从 P1 到 P2 画线

从 P2 到 P3 画线

从 P3 到 P1 画线

下面的文件是例 1 的 AuotLISP 程序清单。右边的行号只为方便引用，并不是程序的一部分。

```
；this program will prompt you to enter three points                    1
；of a triangle from the keyboard ,or select three points               2
；by using the screen cursor .P1,P2,P3 are triangle corners.            3
                                                                        4
(defun： C：triang1()                                                   5
  (setq P1(getPoint" \n Enter first Point of triangle：" ))             6
  (setq P2(getPoint" \n Enter second Point of triangle："               7
  (setq P3(getPoint" \n Enter third Point of triangle：" ))             8
  (Command" line" P1 P2 P3" C" )                                        9
)                                                                       10
```

说明

第 1-3 行

前三行为注释行，用于描述程序中的函数。这几行很重要因为有它们，编辑程序会变得简单一些。可以在任何必要的时候使用注释。所有的注释行都必须以分号(；)开头，当程序装入时这些行会被忽略。

第 4 行：行为空行，用于分隔程序与注释部分。空行还可以用来分隔程序的不同模块。这样便于区分程序的不同部分。空行对程序没有影响。

第 5 行：(defun C: triang1()

本行中，defun 为一个 AutoLISP 函数，它定义了函数 TRIANG1。TRIANG1 为该函数的函数名。由于此函数名前带有 C：，因此该函数可以像 AutoCAD 命令一样被执行。若没有 C：，TRIANG1 命令只能置于圆括号中执行(TRIANG1)。此函数带有三个全局变量(P1,P2 ，P3)。第一次编写 AutoLISP 程序时，

保持变量为全局变量是个好习惯。这是因为装入并运行程序后，可以通过在 AutoCAD 命令提示行中输人感叹号(！)并在其后输入变量名来检查变量的值(Command：！P1)。一旦程序通过测试并运行正常，就应该使它们成为局部变量(defun c：TRIANG1(/P1 P2 P3)

第 6 行：(setq P1(getpoint”\n Enter first Point of triangle：”))

本行中，getpoint 函数暂停程序的运行，允许用户输入三角形的第一个点。提示信息 Enter first Point of triangle 显示在屏幕的提示区内。可以通过键盘输入该点的坐标，也可以用屏幕光标选择该点。随后 setq 函数将这些坐标赋予变量 P1。\ n 的作用是回车 ,其后的表达式将被打印在下一行上(“n”代表”newline”)

第 7 行和第 8 行：(setq P2(getpoint” \ n Enter second Point of triangle：”))及(setq P3(getpoint” \ n Enter third Point of triangle：”))

这两行提示用户输入三角形的第二个顶点和第三个顶点，随后把这些坐标赋予 P2 和 P3。 \ n 的作用是回车，因此输入提示显示在下一行中。

第 9 行：(Command”line” P1 P2 P3”C”)

本行中，Command 函数用来输入 AutoCAD 的 line 命令，然后从 P1 到 P2，P2 到 P3 各画一条直线。”C”(表示”close”选项)把最后一点 P3 与第一点 P1 连接起来。所有的 AutoCAD 命令及选项在 AutoLISP 程序中使用时都必须置于双引号内。变量 P1、P2. P3 之间用空格分隔。

第 10 行

本行仅包含一个用于表明函数 TRIANG1 定义完成的右括号。该括号也可以写在上一行中。把它单独放在一行是一个好习惯，因为这样做任何程序员都可以很容易的确定定义已结束。然而某些程序中，同一程序内的多个定义及模块需要明确区分开。括号及空行有助于明确定义或程序段的起始和结束。

1.7 装入一个 AutoLISP 程序

一般来说与一个 AutoLISP 程序相关的名称有两个：程序文件名和函数名。例如，TRIANG1.LSP 是一个文件名，而不是函数名。所有的 AutoLISP 文件名均以.LSP 为扩展名。一个 AutoLISP 文件可以包含一个或多个函数定义。例如，例 1 中的 TRIANG1 是一个函数名。要执行一个函数，必须装入定义该函数的 AutoLISP 程序文件。在图形编辑器中使用如下命令装入一个 AutoLISP 文件。

Command：(load" [path] filename")

其中 Command.....AutoCAD 命令提示行

load.....装入一个 AutoLISP 程序文件

[path]filename.....AutoLISP 程序文件的路径及名称

AutoLISP 文件名及可选的路径名必须置于双引号内。load 及 filename 必须放在括号中。若没有括号，AutoCAD 将试图装人一个图形或文本字体文件，而不是 AutoLISP 文件。load 同 filename 间的空格也可省略。如果 AutoCAD 成功的装入了该文件，函数名将会显示在屏幕的 Command 提示区内。要执行该程序，在 AutoCAD 的 Command：提示行中键人函数的名称，并按下 Enter 键，(Command：TRIANG1)。如果程序中函数名前没有 C：，可以通过将函数名置于括号中的形式运行该程序：

Command：TRIANG1 或 Command：(TRIANG1)

注意

定义待装入的 AutoLISP 程序的路径时应使用斜杠。例如，如果 AutoLISP 文件 triang 存放在 C 驱动器的 LISP 子目录中，使用如下命令装入该文件。还可以使用双反斜 杠(\\)代替斜杠。command：(load" c:/lisp/triang")或 command：(load" c:\\lisp\\triang")

还可以通过使用标准的 windows 拖放技术装人一个应用程序。要装人一个 LISP 程序，一种方法是在 Windows 的资源管理器中选中该文件，并拖放到 AutoCAD 的图形窗口中，选中的程序将自动装入。还有一种装入 AutoLISP 程序的方法，就是使用 load /unload application 对话框。

该对话框可通过在 tools 菜单中选择 load applications 或在 AutoCAD 命令提示行中输入 appload 将其显示。

load/unload application 对话框

可使用 load/unload application 对话框(见图 12 习)装入 LSP、 VLX、 FAS. VBA、DBX 及 ObjectARX 应用程序。 VBA、DBX 及 ObjectARX 文件被选中时会立即装入；LSP、VLX 及 FAS 文件在 load/unload application 对话框关闭时装人。该对话框的顶部列出了选中目录中的文件。文件的类型可以通过在 file of type 编辑框中输入(.lsp)或通过在下拉列表框中选择来改变。选择一个文件并点击 LOAD 按钮可以装入该文件。以下为 load/unload application 对话框其他特性的描述：

1.load

Load 按钮可用于装人或重新装入选中的文件，文件可以从文件列表框， Load Application 标签或 History List 标签中选取。 ObjectARX 文件不能重新装入，必须先卸载该 ObjectARX 文件，然后再次装入。

2.Load Application 标签

选择 Load Application 标签后， AutoCAD 会显示出当前已装人的应用程序。可以向该列表中添加文件，方法是从文件列表框中拖动文件名到 Load Applications 列表中。

History List 标签

选择 History List 标签后， AutoCAD 会显示出以前通过选中 Add to History 复选框装入的文件列表。如果未选中该复选框，拖放 History List 中的文件会装入该文件，但木将其添加到 History List 中。

3.add to History

选中 Add to History 复选框后，拖放 History List 中的文件会使其自动添加到 History List 中。

4.unload

Unload 按钮在选中 Loaded Applications 标签时出现。要卸载一个应用程序，在 Loaded Applications 文件列表中选中该文件名，然后选择 unload 按钮。未注册卸载的 lisp 文件及 ObjectARX 文件不能卸载。

5. Remove

Remove 按钮在选中 History List 标签时出现。要从 History List 中移除一个文件，选中该文件并选择 Remove 按钮。

6. Startup Suite

每次 AutoCAD 启动时都会自动装人 Startup Suite 中的文件。选中 Startup Suite 后， AutoCAD 会显示出包含一个文件列表的 Startup Suite 对话框。可以选择 Add 按钮向该列表中添加文件，还可以从文件列表框中拖放文件到 Startup suit1 中。要从 History List 中添加文件，右去该文件。

练习 1

编写一个在两点间画线的 AutoLISP 程序(见图 12-4)。该程序必须提示用户输入两点的 X、Y 坐标。

1.8 getcorner、getdist 与 setvar 函数

1.getcorner 函数

getcorner 函数暂停程序的运行，等候用户输入一个点的坐标。可以用键盘或使用屏幕光标将其输入。该函数需要一个基点，在屏幕上移动屏幕光标时将根据该基点显示矩形。

Getcorner 函数格式为：

(getcorner Point [Prompt])

其中 Point ..... 基点

prompt ..... 显不在屏幕上的提示信息

示例：(getcorner pt1)

(setq pt2(getcorner pt1))

(setq pt2(getcorner pt1" Enter second Point))

注意

基点及响应 getcorner 函数所选择的点均是关于当前 UCS 定位的。

若选择的是带有 X,Y,Z 坐标的 3D 点，FZ 坐标将被忽略。该点假定当前高度为其 Z 坐标。

2.getdist 函数

getdist 函数暂停程序的运行，等候用户输入距离，随后以实数形式返回该距离。getdist 函数格式为：

(getdist [Point] [Prompt])

其中 Point.....距离的第一点坐标

Prompt.....须在屏幕上显示的提示信息

示例：(getdist)

(setq dist(getdist))

(setq dist(getdist pt1))

(setq dist(getdist" Enter distance"》

(setq dist(getdist pt1" Enter second Point for distance"))

可以在屏幕上选择两个点来输入距离。例如，若赋值语句为(setq dist(getdist))，可以输入数字或选择两个点 ;若赋值语句为(setq dist(getdist pt1)) ,这里的第一点(pt1)已经定义 ,只须选择第二点。getdist 函数总是返回以实数表示的距离。例如，如果当前设置为 architecture 并且以 architecture 单位输入了距离，getdist 函数将以实数形式返回该距离。

3.setvar 函数

setvar 函数用于给 AutoCAD 系统变量赋值。系统变量名必须置于双引号中。

setvar 函数格式为：

(setvar” variable-name value)

其中 variable name.....AutoCAD 系统变量

value.....赋予系统变量的值

示例：(setvar” cmdecho” 0)

(setvar” dimscale” 1.5)

(setvar “Ltscale” 0.5)

(setvar” dimcen” 0.25)

例 2

编写一个 AutoLISP 程序 ,在给定的两条线间通过输入倒角角度及倒角距离生成一个倒角。AutoCAD 使用赋予系统变量 ChamferA 和 ChamferB 的值生成该倒角。当选择了 AutoCAD 的 Chamfer 命令后，第一个倒角及第二个倒角的距离被自动赋予系统变量 ChamferA 及 ChamferB。随后 Chamfer 命令使用这些值生成一个倒角。然而，在多数工程图中，人们更喜欢通过输入倒角长度及倒角角度的方式来生成倒角，如图 12-5 所示。

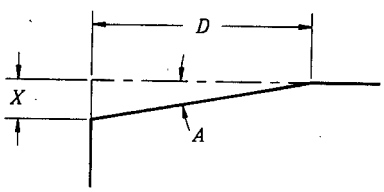

图12-5 倾角为A，距离为D的倒角

输入	输出
第一个倒角距离(D)	任意两条选中直线间的倒角
倒角角度(A)	
处理过程	计算过程
l.计算第二个倒角的距离	x/d=tan a
2.将这些值赋予系统变量 ChamferA 和 ChamferB	x=d*(tan a)
3.使用 AutoCAD 的 Chamfer 命令生成倒角。	=d* [(sin a)/(cos a)]

下面的文件是例 2 的程序清单。右边的行号只为方便引用，并不是文件的一部分。

;This program generates a chamfer by entering 1

```
;the chamfer angle and the chamfer distance                    2
;                                                              3
(defun c：chamfer(/ d a)                                      4
(setvar”cmdecho”0)                                            5
(graphscr)                                                    6
(setq  d(getdist”\n Enter chamfer distance:”))                7
(setq a(getangle”\n Enter chamfer angle：”))                  8
(setvar”chamfera”d)                                           9
(setvar”chamferb”（d(/sin a (cos a))))                        10
(Command”chamfer”)                                            11
(setvar”cmdecho”l)                                            12
(princ)                                                       13
)                                                             14
```

说明

cmdecho 系统变量 :控制 AutoLISP 的 command 函数运行时 AutoCAD 是否回显提示和输入。

第 7 行：(setq d(getdist”\n Enter chamfer distance:”))

getdist 函数暂停程序的运行，等候用户输入倒角距离，随后 setq 函数将该值赋予变量 d。

第 8 行：(setq a(getangle”\n Enter chamfer angle：”))

getangle 函数暂停程序的运行，等候用户输入倒角角度，随后 setq 函数将该值赋予变量 a。

第 9 行： setvar”chamfera’ d)

setvar 函数将变量 d 的值赋予 AutoCAD 系统变量 chamfera。

第 10 行： (setvar”chamferb”（d(/sin a (cos a))))

setvar 函数将从表达式(*d(/(sin a)(cos a)))中取得的值赋予 AutoCAD 系统变量 chamferb。

第 11 行：(Command”chamfer)

Command 函数使用 AutoCAD CHAMFER 命令生成倒角。

练习 2

编写一个 AutoLISP 程序，生成图 12-6 所示的图形。该程序应该提示用户输入 P1 点和 P2 点以及 D1 和 D2 的直径。

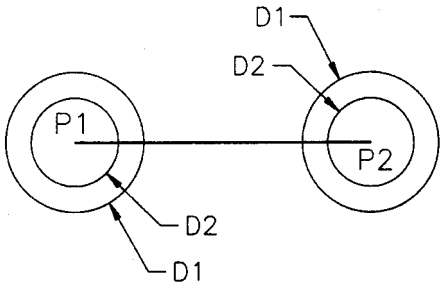

图12-6 带有连线的同心圆

1.9 List 函数

List 函数用于定义一个二维或三维点。若表达式中不包含任何变量及未定义项，则该函数还可以使用单引号(’)命名。

示例(setq x(List 2.5 3.56)) 返回 2.5，3.56

(setq x’(2.5 3.56)) 返回 2.5，3.56

1.10 car、cdr 与 cadr 函数

1.car 函数

car 函数返回一个表中的第一个元素。若表中不包含任何元素,函数返回 nil。

car 函数格式为：(car List)

其中 car……返回第一个元素

list……元素列表

示例：(car ‘(2.5 3.56)) 返回 2.5

(car ‘(x y z)) 返回 X

(car ‘((15 20) 56)返回(15 20)

(car ‘())返回 nil

其中的单引号表示～个表。

2.cdr 函数

cdr 函数返回一个移去了表的第一个元素后的列表,其格式为(cdr List)

其中 cdr ……返回第一个元素被移去的列表

list……元素列表

示例：(cdr ‘(2.5 3.56)) 返回(3.56)

(cdr ‘(x,y,z) 返回(y,z)

(cdr ‘((15 20)56) 返回(56)

(cdr ‘()) 返回 nil

3.cadr 函数

cadr 函数执行两个操作，cdr 和 car，返回列表中第二个元素。cdr 函数移去了第一个元素。car 函数返回新表中的第一个元素。、cadr 函数的格式为：(cadr List)

其中 cadr……执行两个操作(car(cdr ‘(x y z)))

List……元素列表

示例：(cadr ‘(2 3)) 返回 3

(cadr ‘(2 3 56)) 返回 3

(cadr ‘(x y z)) 返回 y

(cadr ‘((15 20) 56 24))返回 56

这些例子中， cadr 执行两个操作

(cadr '(x y z))=(car(cdr '(x y z))

=(car '(y z)) 返回 y

注意

除 car、cdr 和 cadr 函数外，还有其他几个函数用于选取列表中的不同元素。下面是这些函数的列表，其中函数 f 由列表 '((x y f)z w)组成。

(setq f '((x y)z w))

(caar f)=(car(car n) 返回 x

(cdar f)=(cdr(car f)) 返回(y)

(cadar f)=(car(cdr(car f))) 返回 y

(cddr f)=(cdr(cdr f) 返回(w)

(caddr f)=(car(cdr(cdr f))) 返回 W

1.11 graphscr、textscr、princ 与 terpri 函数

1.graphscr 函数

若系统只有一个屏幕，graphscr 函数将文本窗口转换为图形窗口，若系统有两个屏幕，该函数将被忽略。

2.textscr 函数

若系统只有一个屏幕，textscr 函数将图形窗口转换为文本窗口，若系统有两个屏幕，该函数将被忽略。

3.princ 函数

Princ 函数打印(或显示)变量的值。若变量位于双引号中，该函数打印(或显示)双引号内的表达式。

princ 函数的格式为：(Princ [变量或表达式])

示例：(princ)在屏幕上打印一个空行

(princa)在屏幕上打印出变量 a 的值

(princ”Welcome”)在屏幕上打印 Welcome

4.terpri 函数

terpri 函数用于在屏幕上打印一个新行，与\n 作用相同。该函数用于打印紧跟着 terpri 函数的一行。

示例：(setq p1(getPoint”Enter first Point：”))(terpri)

(setq p2(getPoint”Enter second Point：”))

第一行(Enter first Point：)将显示在屏幕的命令提示区中。 terpri 函数产生了一个回车，因此第二行(Enter second Point：)将显示在新的一行中，也就是第一行下面的一行。若没有 terpri 函数，这两行将显示在同一行中(Enter first Point：Enter second Point：)

例 3

编写一个程序，提示用户输入一个矩形的两个对角的坐标，然后在屏幕上画出该矩形，如图 12-7 所示。

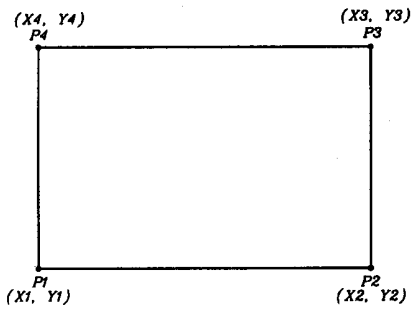

图12-7 矩形

输入	处理过程
P1 点的坐标	1.计算 PZ 点和 P4 点的坐标
P3 点的坐标	2.画出下列直线
	P1 到 P2 的直线
	P2 到 P3 的直线
	P3 到 P4 的直线
	P4 到 P1 的直线

P2 和 P4 两点的 X，Y 坐标可以通过 car 及 cadr 函数算出。car 函数从给定的列表中选取 X 坐标，cadr 函数选取 Y 坐标。

P2 点的 X 坐标:	P2 点的 Y 坐标:
X2=X3	Y2=Y1
X2=car(X3 Y3)	Y2=cadr(X1 Y1)
X2=car P3	Y2=cadr P1

P4 点的 X 坐标:	P4 点的 Y 坐标：
X4=X1	Y4=Y3
X4=car(X1 Y1)	Y4=cadr(X3 Y3)
X4=car P1	Y4=cadr P3

故，P2 点和 P4 点为：

P2=(list(car P3)(cadr P1))

P4=(List(car P1)(cadr P3))

下面的文件是例 3 的程序清单。

```
(defun c：rect1(/p1 p2 p3 p4)
(graphscr)
```

```
(setvar”cmdecho”0)
(prompt”rect1 command draws a rectangle”)(terpri)
(setq p1(getpoint ” Enter first corner”)(terpri)
(setq p3(getpoint ” Enter opposite corner”)(terpri)
(setq p2(list(car p3)(cadr p1)))
(setq p4(list(car p1)(cadr p3)))
(command ” line”p1 p2 p3 p4 “c”)
(setval “cmdecho” 1)
(Princ)
)
```

说明

第 1 行：(defun c：rect1(/p1 p2 p3 p4)

defun 函数定义了函数 rect1。

第 2 行：(graphscr)

如果当前屏幕恰好是文本屏幕，该函数将文本屏幕转换为图形屏幕。否则，对显示屏幕无影响。

第 3 行：(setvar”cmdecho”0)

函数 setvar 将 0 赋予 AutoCAD 系统变量 cmdecho，即关闭了回显。如果 cmdecho 被关闭，AutoCAD 的命令提示就不会显示在屏幕的命令提示区中。

第 4 行：(prompt”rect1 command draws a rectangle”)(terpri)

prompt 函数将显示双引号中的信息(“rect1 command draws a rectangle”)。函数 terpri 产生一个回车，因此下一行文本会打印在单独一行上。

第 5 行：(setq p1(getpoint ” Enter first corner”)(terpri)

getpoint 函数暂停程序的运行，等候用户输入一个点(该矩形的第一个角)，随后 setq 函数将该值赋予变量 P1。

第 6 行：(setq p3(getpoint ” Enter opposite corner”)(terpri)

getpoint 函数暂停程序的运行，等候用户输入一个点(该短形的对角)，随后 setq 函数将该值赋予变量 P3。

第 7 行：(setq p2(list(car p3)(cadr p1)))

cadr 函数选取 P1 点的 Y 坐标，car 函数选取 P3 点的 X 坐标。setq 函数将这两个值组成的列表赋予变量 P2。

第 8 行：(setq p4(list(car p1)(cadr p3)))

cadr 函数选取 P3 点的 Y 坐标，car 函数选取 P1 点的 X 坐标。setq 函数将这两个值组成的列表赋予变量 P4。

第 9 行：(command ＂line＂p1 p2 p3 p4 ＂c＂)

Command 函数使用 AutoCAD 的 line 命令在点 P1，P2，P3 和 P4 间画线。C(close)将最后一点 P4 与第一点 P1 连接起来。

第 10 行：(setval ＂cmdecho＂ 1)

setvar 函数将 1 赋予 AutoCAD 系统变量 cmdecho，即打开了回显。

第 11 行：(princ)

princ 函数在屏幕上打印一个空行。若没有这一行， AutoCAD 将打印出最后一个表达式的值。该值对程序毫无影响，但却可能令人费解。princ 函数用来防止在命令提示区显示该表达式的值。

第 12 行：该右括号表明完成函数 rect1 的定义，并且程序结束。

注意

在这个程序中，定义一个矩形的两个角后，该矩形被生成。当移动屏幕光标输入第 二点时，该矩形不会被拖动。然而，可以使用 getcorner 函数来拖动该矩形，如以下程序清单所示：

```
(defun c：rect2(/p1 p2 p3 p4)
(graphscr)
(setvar＂cmdecho＂0)
(prompt＂rect2 command draws a rectangle＂)(terpri)
(setq p1(getpoint＂Enter first corner＂)(terpri)
(setq p3(getcorner p1＂Enter opposite corner＂)(terpri)
(setq p2(list(car p3)(cadr p1)))
(setq p4(list(car p1)(cadr p3)))
(command ＂line＂p1 p2 p3 p4 ＂c＂)
(setval ＂cmdecho＂ 1)
(Princ)
)
```

1.12 getangle 与 getorient 函数

1.getangle 函数

函数 getangle 暂停程序的运行，等候用户输入一个角度，随后将该角度值以弧度的形式返回。

getangle 函数的格式为：(getangle [Point] [prompt])

其中　　Point.....角的第一点

Prompt.....德要显示在屏幕上的提示信息

示例：(getangle)

(setq ang(getangle))

(setq ang(getangle pt1))　pt1 为一个已定义的点

(setq ang(getangle ” Enter taper angle”))

(setq ang(getangle pt1 “Enter second Point of angle”)

角度设置会对所输入的角产生影响。可以通过使用 AutoCAD 的 units 命令或改变 AutoCAD 系统变量 angbase 和 angdir 的值来改变角度设置。下面是测量角度的默认设置。

角度是关于正 X 轴(3 点钟位置)来测量的。该设置的值保存在 AutoCAD 的系统变量 Angbase 中。

如果角度以逆时针方向测量，则该角为正；如果以顺时针方向测量则该角为负。该设置的值保存在 AutoCAD 的系统变量 angdir 中。

如果角度采用默认设置(见图 12-8a)，对于 135 度的角，getangle 函数将返回 2.35619 弧度。

示例

(setq ang(getangle” Enter angle”))　一个 135 度的角将返回 2.35619

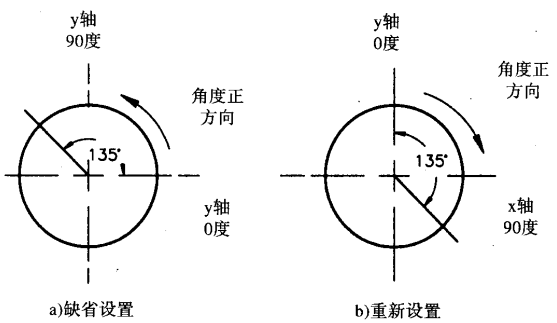

图12-8　角度设置

图 12-8n 为新的角度设置。其中 Y 轴为 0 度且角度以顺时针方向测量时为正。对于 135 度的角 getangle 函数将返回 3.92699。函数 getangle 忽略系统变量 angdir 中的方向设置，根据设置在系统变

量 angbase(见图 12-9)中的角度基准以逆时针方向计算角度。

示例：

(setq ang(getangle" Enter angle"))返回 3.92699

2.getorient 函数

getorient 函数暂停程序的运行，等候用户输入一个角度,随后将该角度值以弧度的形式返回。

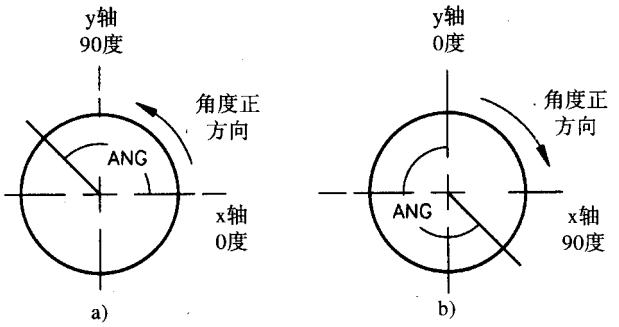

图12-9 角度测量

getorient 函数的格式为：(getorient [Point] [Prompt])

其中 Point.....该角的第一点

prompt.....需要显示在屏幕上的提示信息

示例：

(getorient)

(setq ang(getorient))

(setq ang(getorient pt1))

(setq ang(getorient "Enter taper angle"))

(setq ang(getorient" Enter second Point of angle"))

函数 getorient 与函数 getangle 非常类似，都是将角度值以弧度形式返回。然而 getorient 函数通常忽略 angbase 及 angdir 的设置，以正 X 轴(3 点钟位置)及逆时针方向测量角度。如果未改变设置，如图 12-10a(angdir 及 angbase 的默认设置)，对于一个 135 度的角，函数 getorient 将返回 2.35619 弧度。如果改变了设置，如图 12-10n，对于一个 135 度的角，函数 getorient 将返回 5.49778 弧度。尽管设置变成以正 Y 轴及顺时针方向测量角度，getorient 函数还是忽略新的设置并以正 X 轴及逆时针方向测量角度。

注意

对于 getangle 及 getorient 函数，可以通过键盘或在屏幕上选择两点来输入角度。如果赋值表达

式为(setq ang(getorient　pt1))，且其中的 pt1 点已定义，程序会提示用户输入第二点。可以在屏幕上选择一个点或输入第二点的坐标。

180 度与 Pi(3.14159)弧度是相等的。要将角度转换为弧度，使用下面的关系：

弧度数=(Pi×角度数)/180

1. 13　getint、getreal、getstring 及 getval 函数

1.getint 函数

getint 函数暂停程序的运行，等候用户输入一个整数.即使输入一个整数,该函数仍返回一个整数。

getint 函数的格式为：(getint [prompt])

其中 prompt……可选项.希望显示在屏幕上的提示信息。

示例：

(getint)

(setq numX(getint))

(setq numX(getint “Enter number of rows：”))

(setq numX(\n getint “Enter numner ofrows： “))

2.getreal 函数

getreal 函数暂停程序的运行，等候用户输入一个实数。即使输入一个整数，该函数仍返回一个实数。

Getreal 函数的格式为：(getreal[prompt])

其中　　prompt……可选项，希望显示在屏幕上的提示信息。

示例：

(getreal)

(setq realnumx(getreal))

(setq realnumx(getreal” Enter num1：”))

(setq realnumx(getreal” \ n Enter num2：”))

3.getstring 函数

getstring 函数暂停程序的运行，等候用户输入一个字符串值。即使只输入数字，该函数仍返回一个字符串。

getstring 函数的格式为：(getstring [Prompt])

其中　prompt……可选项，希望显示在屏幕上的提示信息

示例：

(getstring)

(setq answer(getstring))

(setq answer(getstring　“Enter Y for yes， N for no：”))

(setq answer(getstring” \n Enter Y for yes， N for no：”))

注意

该字符串的最大长度为 132 个字符。若长度超过 132 个字符，超出的部分将被忽略.

4.getvar 函数

getvar 函数允许检索一个 AutoCAD 系统变量的值。其格式为：(getvar”variable”)

其中　　variable……AutoCAD 系统变量名

示例：

(getvar)

(getvar”dimcen”)返回 0.09

(getvar”ltscale”)返回 1.0

(getvar”limmax”)返回 12.00, 9.00

(getvar”limmin”)返回 0.00,0.00

注意

AutoCAD 系统变量名必须置于双引号中。

一条 getvar 语句只能检索一个变量的值。要检索多个系统变量的值，需要对每个变量使用单独的 getvar 语句。

1.14 polar 与 sqrt 函数

1.polar 函数

Polar 函数利用相对于给定点的角度及距离定义一个点(见图 12-11)。该角度以弧度表示，逆时针方向为正(假定 angbase 及 angdir 均为默认设置)。

polar 函数的格式为：(polar point angle distance)

其中 Point……参照点

angle……对于参照点的角度

distance……与参照点间的距离

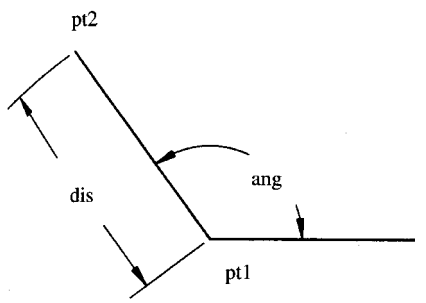

图12-11 使用polar函数定义点

示例：

(Polar pt1 ang dis)

(setq pt2(polar pt1 ang dis))

(setq pt2(polar '(2.0 3.25)ang dis))

2.Sqrt 函数

sqrt 函数计算一个数的平方根，且其返回值总为实数。sqrt 函数的格式为：(sqrt number)

其中 number……待求平方根的数(实数或整数)

如图 12-12 所示，应用 sqrt 函数：(setq hyp(aqrt(+(* base base)("ht ht))))

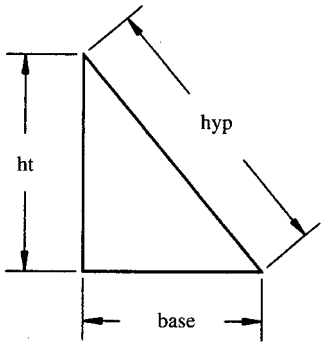

图12-12 sqrt函数的应用

示例：

(sqrt 144) 返回 12.0

(sqrt 144.0) 返回 12.0

(setq x(sqrt 57.25)) 返回 7 .566373

(setq x(sqrt(* 25 36.5))) 返回 30.207615

(setq x(sqrt(/ 7.5(cos 0.75)))) 返回 3.2016035

(setq hyp(sqrt(+(* base base)(* ht ht))))

例 4

编写一个 AutoLISP 程序，该程序可以画出一个等边三角形及其内切圆(见图 12-13)。该程序还应提示用户输入圆的半径及圆心。

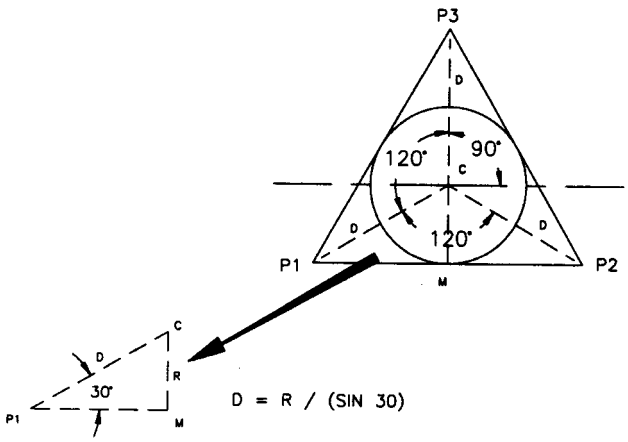

图12-13 等边三角形及其内切圆

下面的文件为例 4 的程序清单。

(defun dtr(a)

```
   (*a( Pi 180.0))
)
(〔defun c：trgcir(/ r c d p1 p2 p3)
(setvar”cmdecho”0)
(graphscr)
   (setq r(getdist”\n Enter circle radius：”)
   (setq c(getPoint” \ n Enter center of circle：”))
   (setq d(/r(sin(dtr 30))))
   (setq P1(Polar c(dtr 210)d))
   (setq P2(Polar c(dtr 330)d))
   (set P3(Polar c(dtr 90) d))
   (command”circle” c r)
   (Command”line”  p1  p2  p3”c”)
   (setval”cmdecho”1)
   (princ)
    )
```

练习 3

编写一个 AutoLISP 程序，画出一个等腰三角形 P1，P2，P3，三角形的底边(P1，P2)与正 X 轴之间有一个夹角 N(见图 12-14)。该程序还应提示用户输入起点 P1，L1 的长度及角 A 和角 N。

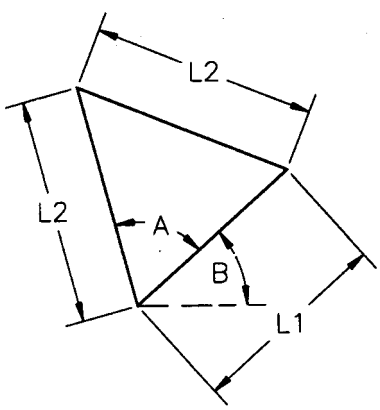

图12-14 带倾角的等腰三角形

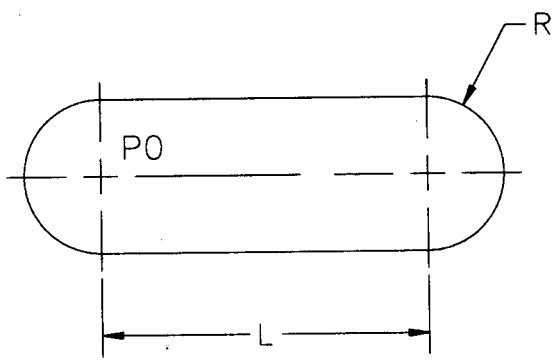

图12-15 长度为L, 半径为R的键槽

练习 4

编写一个程序，画出一个带中心线的键槽。该程序还应提示用户输入键槽的长度，宽度及中心线的

图层名(见图 12-15)。

1.15 itoa、rtos、strcase 及 prompt 函数

1.itoa 函数

itoa 函数将一个整数转换成字符串，并将该字符串返回。

itoa 函数的格式为:(itoa number)

其中 number……待转换为字符串的整数

示例：

(itoa 89)返回”89”

(itoa -356)返回”-356”

(setq intnum 7)

(itoa intnum)

(setq intnum 345)

(setq intstrg(itoa intnum))

2.rtos 函数

rtos 函数将一个实数转换成字符串,并将该字符串返回。

rtos 函数的格式为：(rtos realnum)

其中 alnum……待转换为字符串的实数

示例：

(rtos 50.6)返回”50.6”

(rtos -30.0)返回”-30.0”

(setq realstrg(rtos 5.25))返回”5.25”

(setq realnum 75.25)

(setq realstrg(rtos realnum))返回”75.25”

rtos 函数还可以带有 mode 和 precision 参数。带有 mode 和 precision 参数的 rtos 函数格式为：

(rtos realnum [mode] [precision])

其中 realnum……实数

mode……单位模式，如 decimal， scientific

precision……数字精度

3．strcase 函数

strcase 函数将字符串中的字符转换为大写形式或小写形式。

其格式为：(strcase string [true])

其中 string.....待转换大小写的字符串

true.....若其值不为 nil，则所有字符转换为小写形式

true 是可选项。若省略或其值为 nil，字符串将转换为大写形式，若其值不为 nil，字符串将转换为小写形式。

示例：

(strcase" welcome Home") 返回" WELCOME HOME"

(setq t 0)

(strcase" Welcome Home" t) 返回" welcome home

(setq answer(strcase(getstring" Enter Yes or No：")))

4．prompt 函数

prompt 函数用于在屏幕的命令提示区内显示消息,该消息的内容必须置于双引号中。Prompt 函数的格式为：(prompt message)

其中 message..... . . 要显示在屏幕上的消息

示例：

(prompt" Enter circle diameter：")

(setq d(getdist(prompt" Enter circle diameter：")))

注意 在双屏幕系统中，prompt 函数在两个屏幕上均显示消息。

例 5

编写一个程序，画出半径分别为 rl 和 r2 的两个圆，用来表示间距为 d 的两个皮带轮。连接两圆圆心的直线与 X 轴成一个夹角，如图 12-16 所示。

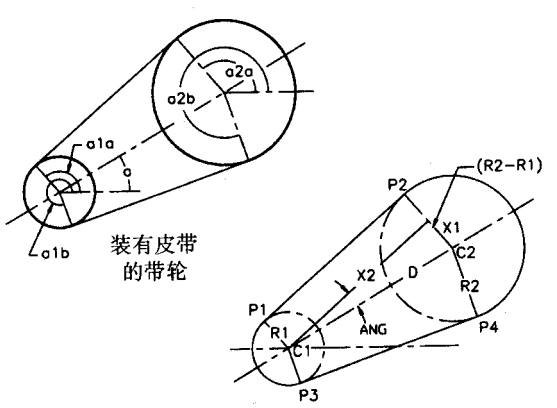

图12-16 带公切线的两个圆

输入	输出
小圆的半径 rl	半径为 rl 的小圆
大圆的半径 r2	半径为 r2 的大圆
圆间距离 d	两圆的公切线
中心线倾角 a	
小圆的圆心 c1	

处理过程

1．计算 X1、X2 的长度

2．计算角度 ang

3．根据 c1 点确定 c2 点

4．确定点 P1，P2，P3，P4

5．以 c1 为圆心，r1 为半径画出小圆

6．以 c2 为圆心，r2 为半径画出大圆

7．画出 P1 到 P2，P3 到 P4 的直线

计算过程

X1=r2-r1

X2=sqrt [d^2-(r2-r1)^2]

tan ang=X1/X2

ang=atan(x1/x2)

ala=90+a+ang

alb=270+a ang

a2a=90+a+ang

a2b=270+a ang

下面的文件为例 5 的 AutoLISP 程序清单；

```
(defun dtr(a)
(* a (/ pi 180.0))
  )
(defun c：belt(/r1 r2 d a c1 xl x2 c2 pl p2 p3 p4)
(setVar”cmdecho”0)
(graphscr)
```

```
(setq r1(getdist"  \n Enter radius of small pulley : "  ))
(setq r2(getdist"  \n Enter radius of large pulley : "  ))
 (setq d(getdist"  \n Enter distance between pulleys:" )
(setq a(getangle"  \n Enter angle of pulleys:" )
 (setq c1(getpoint"  \n Enter center of small pulleys:" )
 (setq x1(-r2 r1))
  (setq x2(sqrt(-(* d d)(*(-r2 r1)(-r2 r1)))))
  (setq   ang(atan (/   xl   x2)))
 (setq c2(polar (c1 a d))
 (setq pl(polar c1(+ ang   a (dtr 90)) rl))
  (setq p3(polar c1(-(+ a dtr(270)) ang)rl))
 (setq p2(Polar c2(+ ang a dtr(90)) r2))
  (Setq p4(polar C2(-(+ a (dtr 270) Ang)r2))

  (command"  circle"   c1   p3)
  (command"  circle"   c2   p2)
  (command"  line"     pl   p2"   ")
  (command"  line"     p3   p4"   ")
   (setvar"  cmdecho   "1)
  (Princ))
```

说明

第 1 行：(defun dtr(a)

本行中，defun 函数定义了一个用于将角度转换为弧度的函数 dtr(a)。

第 2 行： (* a (/ pi 180.0))

(/Pi 180)将 Pi 的值除以 180，得到的结果再与角 a 相乘(180 度等于 Pi 弧度)。

第 4 行：(defun c：belt(/r1 r2 d a c1 xl x2 c2 pl p2 p3 p4)

本行中，defun 函数定义了一个函数。c：belt,该函数用于生成带公切线的两个圆。

第 12 行：(setq x1(-r2 r1))

本行中，setq 函数将 r2-r1 的值赋予变量 xl。

第 13 行：((setq x2(sqrt(-(* d d)(*(-r2 r1)(-r2 r1)))))

本行中，(- r2 rl)用 r2 减去 r1，(*(- r2 r1)(r2   r1))计算(- r2 r1)的平方。

(sqrt(-(* d d)(*(-r2 r1)(-r2 r1))))则计算该差的平方根，Setq x2 该表达式的结果赋予变量 x2

第 14 行：(setq ang(atan(/ x1 x2)))

本行中 ,(atan(/xl x2)以算(/ x1　x2)结果的反正切。Setq ang 将以弧度表示的角度值赋予变量 ang．

第 15 行：(setq　cZ(polar c1 a d))

本行中，(polar c1 a d)使用 polar 函数确定 c2 点的位置，该位置是由相对于 c1 点的距离 d 及与正 X 轴的夹角 a 来确定的。

第 16 行：(setq pl(polar c1(+ ang a(dtr 90))rl))

本行中，(polar c1(+ ang a(dtr 90))rl)确定 pl 点位置，该位置是由相对于 c1 点的距离 r1 及与正 X 轴的夹角(+ ang a(dtr 90))来确定的。

第 20 行：(command”circle” c1 p3)

本行中，command 函数使用 AutoCAD 的 CIRCLE 命令，以 c1 为圆心，P3 点定义的距离为半径画圆。

第 22 行：(command”line” pl P2” ”)

本行中 ,command 函数使用 AutoCAD 的 LINE 命令从 P1 到 P2 画一条直线。结尾的一对双引号(“”)表示返回，用于终止 LINE＃令。

练习 5

编写一个 AlltOLISP 程序，画出两个圆及它们的内公切线，如图 12-17 所示。该程序还应提示用户输入圆的直径及两个圆的中心距。

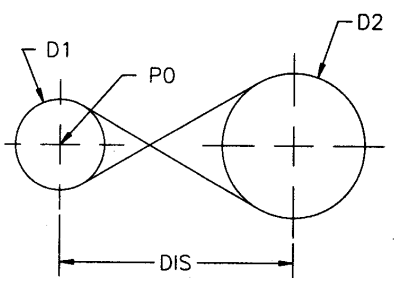

图12-17　带有内公切线的两个圆

1.16 流程图

流程图以图形方式来表示算法，可用于对问题进行系统的分析。特别是当问题中包含条件语句时更有助于人们的理解。流程图是由在程序中代表特定功能的标准符号构成的。例如，矩形用于表示程序执行时发生的过程。每个块都通过表示操作结果的直线连接起来。图 12-18 给出了可在流程图中使用的标准符号。

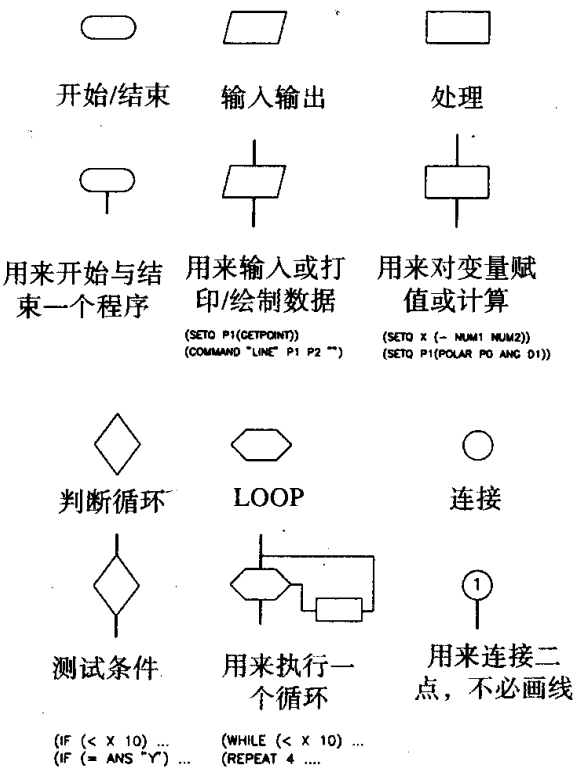

图12-18 流程图符号

1.17 条件函数

本章前面讨论的关系函数用于建立两个元素间的关系。例如，(< X Y)描述了一个操作的测试条件。为了有目的的使用这些函数，就必须用到条件函数。例如，(if(< x y)(setq z(- y x))(setq z(- x y)))描述了当该条件为真(T)和假(nil)两种情况时将采取的动作。若条件为真，z=y-x,若不为真，z=x- y。因此，对于包括 AutoLISP 在内的任何编程语言来说，条件函数都是非常重要的。

1．if 函数

如果指定的条件返回”真”，if 函数求解第一个表达式(then)；若指定的条件返回”假”，则求解第二个表达式(else)。if 函数的格式为：(if condition then [else])

其中 condition……指定的条件语句

then……条件返回 T 时求解的表达式

else……条件返回 nil 时求解的表达式

示例：

```
(if(= 7 7)(“true”))……返回”true”
(if(= 5 7)(“true”)(“false”)……返回”false
(setq ans “yes”)
(if(= ans”yes”)(“yes”)(“no”)……返回”yes”
(setq numl 8)
(setq num2 10)
(if(> num1 num2)
(setq x(- num1 num2))
(setq x(- num2 num1))
)
```

例 6

编写一个 AutoLISP 程序，从一个大数中减去一个小数。程序应提示用户输入这两个数。

输入　　输出

数字(num1)　　X=num1-num2 或

数字(num2)　　X=num2-num1

处理过程

若 num1＞num2，则 X=num1-num2

若 num1＜num2，则 X=num2- num1

图 12-20 中的流程图使用标准流程图符号描述编写该程序的过程.

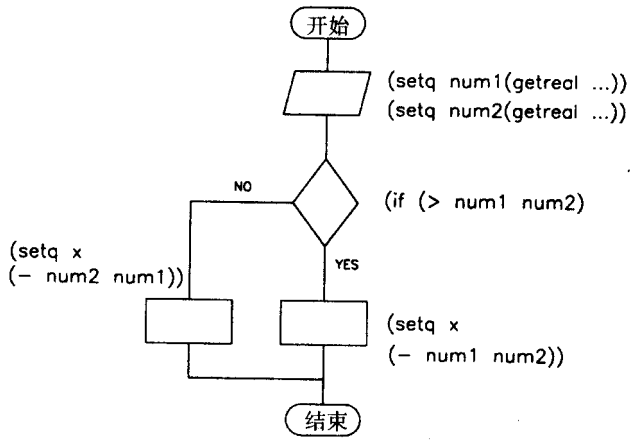

图12-20 例6的流程图

```
defun c：subnum()
 (setvar" cmdecho" O)
 (setq num1(getreal" \n Enter first number：" ))
(setq num2(getreal" \n Enter second number：" ))
(if(＞ num1 num2)
    (setq x(- num1 num2))
    (setq x(- num2 num1))
      )
 (setvar" cmdecho" 1)
 (princ)
   )
```

说明

第 5 行：(if(＞ num1 num2)

本行中,if 函数求解测试表达式(> num1 num2)。若条件为真,返回 T;若不为真,返回 nil。

第 6 行：(setq x(- num1 num2))

若测试表达式(if(＞ num1 num2))返回 T，该表达式被求解。变量 num1 的值减去 num2 的结果被赋予变量 x。

第 7 行：(setq x(- num2 num1))

若测试表达式(if(＞ num1 num2)返回 nil，该表达式被求解。变量 num2 的值减去 num1 的结果被赋予变量 X。

第 8 行：)

该右括号表明完成对函数的定义。

例 7

编写一个 AutoLISP 程序使两个数相乘或相除，并且还应提示用户输入乘法或除法的选择。若没有输入正确的选择，程序将显示出恰当的消息。

程序清单:

```
(defun c：mdnum()
    (setval”cmdcho”0)
    (setq num1(getreal”\ n Enter  first number：”))
    (setq num2(getreal”\n Enter second number：))
    (prompt”Do you want to multiply or divide． Enter M or D：”)
    (setq ans(strcase(getstring)))
    (if(= ans”M”)
      (setq x(“num1 num2))
    )
    (if(= ans”D”)
      (setq x (/ num1 num2))
    )
    (if(and (/= ans”D”)(/= ans”M”))
      (prompt”sorry!wrong entry, Try again”)
    )
    (setvar”cmdecho” l)
    (Princ))
```

2．progn 函数

progn 函数用于与 if 函数搭配，以求解多个表达式。其格式为：

(prong 表达式表达式)

当测试条件返回”真”时，证函数只能求解一个表达式。progn 函数可用于与 if 函数搭配以求解多个表达式。

示例：

```
(if(= ans”yes”)
```

```
(pronn
(setq x(sin ang))
(setq y(cos ang))
(setq tanang(lx y))
))
```

3．While 函数

while 函数求解一个测试条件，若该条件为真(表达式不返回 nil)，while 语句后边的操作将反复执行，直到该测试条件返回 nil。

While 函数的格式为：(While testexpression　Operations)

其中 textexpression……测试某条件的表达式

operations……到测试条件返回 nil 为止，将要执行的操作

示例：

```
(while(= ans “yes”)
(setq x(+ x 1)
(setq ans(getstring “Enter yes or no:”))
)
(while (< n 3)
(setq x(+ x 10))
(setq n(1+n))
)
```

例 8

编写一个 AutoLISP 程序，求解给定数字的 n 次幂。该幂为一个整数。函数还应提示用户输入一个数及其幂数。

输入　　输出

数字 X 的 n 次幂　　结果 X^n

处理过程

l．令 t=l，c=1

2．将 t*X 的结果赋予变量 t

3．重复该过程直到计数器 C 小于或等于 n

下面的文件为例 8 的程序清单。

```
(defun c：nPower()
    (setvar”cmdecho”0)
    (setq x(getreal” \ n Enter a number：”))
    (setq n(getlnt” \ n Enter Nth power-lnteger number：”))
    (setq t 1)(setq c 1)
    (While( < = C n)
    (setq t(* t x))
    (setq c(l+ c))
    )
    (setvar”cmdecho”  l)
    (princ t)
```

例 9

编写一个 AUtOLISP 程序，生成一个带孔的法兰盘(见图 12-24)。程序还应提示用户输入该法兰盘的圆心、直径、孔径、孔数及起始角。

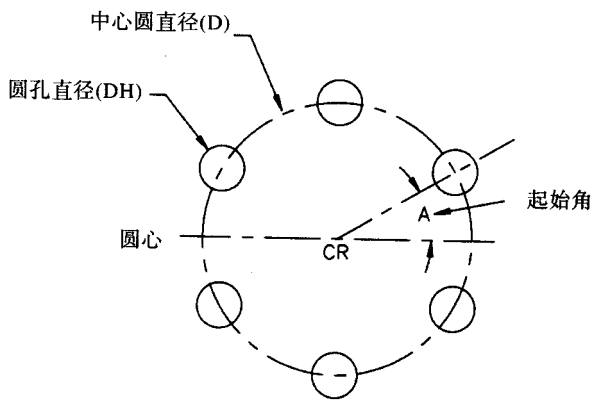

图12-24 六孔法兰盘

```
(defun  c：bc1()
(graphscr)
(setvar”cmdecho”0)
  (setq cr(getpoint”\n Enter center of Bolt-Circle:”))
    (setq d(getdist”\n Dia of Bolt-Circle：”))
    (setq n(getint”\n Number of holes In Bolt-Circle：”))
    (setq a(getangle” \ n Enter start angle：”))
    (setq dh(getdist” \ n Enter diameter of hole：”))
```

```
    (setq inc(/(* 2 pi)n))
    (setq ang 0)
    (set r(/ dh 2))
    (While( <  ang(* 2 pi))
    (setq P1(Polar cr(+ a inc)(d 2)))
    (command”circle”    p1 r)
    (setq a(+ a inc))
    (setq ang(+ ang inc))
      )
    (setvar”cmdecho”    l)
  (Princ)
  )
```

4．repeat 函数

repeat 函数根据函数中指定的次数 n 反复求解表达式(见图 12-25)，变量 n 必须为整数。rePeat 函数的格式为：rePeat n

其中　n ……n 为定义该表达式求解次数的整数

示例：

```
(repeat 5
(setq x(+ x 10))
)
```

例 10

编写一个 AutoLISP 程序 ,生成给定数目的同心圆。程序还应提示用户输入圆心,起始角及半径增量(见图 12-26)。下面是例 10 的 AutoLISP 程序清单。

```
(defun c：concir()
(graphscr)
(Setvar”cmdecho”0)
(setq c(getpoint ”\n Enter center point of circle:”))
(setq n(getint ”\n Enter number of circle:”))
(setq r(getdist ”\n Enter radius of first circle:”))
(setq d(getdist ”\n Enter radius increment:”))
```

```
(repeat n
(Command" circle" c r)
(setq r(+ r d))
(setvar" cmdecho" 1)
(Princ)
)
```

注意

AutoCAD 允许每次启动时自动装入指定的 AutoLISP 程序。例如，如果你正在处理一个项目且已经装入了一个 AutoLISP 程序，那么当你创建另一张图时，该程序会自动装入。可以通过将该文件名添加至 Load/Unload Application 对话框中的 Startup Suite 中来使该特性生效。要获得更多信息，参见本章前面讨论的 LoadIUload Applicahon 对话框。

例 11

编写一个程序生成一个过渡部的平面布局图，并标出尺寸。该过渡部及其未标注的平面布局如图 12-27 所示。

下面的文件为例 11 的 AUtOLISP 程序清单。程序不必一定用小写字母书写，也可以用大写字母或大小写字母混合来书写程序。

```
(defun c：TRANA(/)
(graphscr)
(setvar" cmdechd' O)
(setq L(getdist "\n Enter length of bottom rectangle:"))
(setq w(getdist "\n Enter width of bottom rectangle:"))
(setq h(getdist "\n Enter height of transition:"))
(setq L1(getdist "\n Enter length of top rectangle:"))
(setq w1(getdist "\n Enter width of top rectangle:"))

(Setq x1 (/(- w w1)2))
(setq y1(/(- L L1)2))
(setq dl(Sqrt(+(* h h)(* xl xl))))
(setq d2(sqrt(+(* dl dl)(* yi yi))))
(setq s1(/(-L L1)2))
```

```
(setq pl(sqrt(-(* d2 d2)(* s1 s1))))
(setq s2(/(- w w1)2))
(setq p2(sqrt(-(* d2 d2)(* s2 s2))))

(setq t1(+ L1 s1))
(setq t2(+ 1 w))
(setq t3(+ 1 s2 w1))
(setq t4(+ 1 s2))
(setq pt1(list 0 0))
(setq pt2(list s1 p1))
(setq pt3(list t1 p1))
(setq pt4(list 1 0))
(setq pt5(list t4 p2))
(setq pt6(list t3 p2))
(setq pt7(list t2 0))
(command" layer" "make" "ccto" "c" "l" "ccto" " ")
(command" line" ptl pt2 pt3 pt4 pt5 pt6 pt7" c" )
(setq sf (/(+ 1 w)12)
(setvar" dimscale" sf)
(setq c1(list 0(- 0(* 0.75 Sf)))
(setq c7(list(- 0(* 0.75 sf)) 0))
(setq C8(list(-l(* 0.75 Sf)) 0))
(command" layer" ' 'make" "cctd" "c" "2" "cctd" " ")
(command" dim" "hor" pt1 pt2 of" " "base" pt3" " "base" pt4" " "exit"
(command" dim" "hor" pt4 pts of" " "base" pt6" " "base" pt7" " "exit"
(Command" dim" "vert" ptl pt2 pt2" " "exit" )
(Command" dim" "vert" pt4 pt5 pt5" " "exit" )
(command" dim" "aligned" ptl pt2 c7" " "exit" )
(command" dim" "aligned" pt4 pt5 c8" " "exit" )
(setvar" cmdecho" l)
(princ)
```

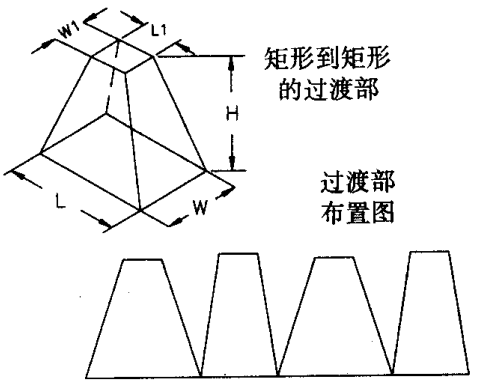

图12-27 过渡部及其平面布局

例 12

编写一个 AutoLISP 程序生成如图 12-28 所示的圆锥平面布局图，并且对其进行标注。

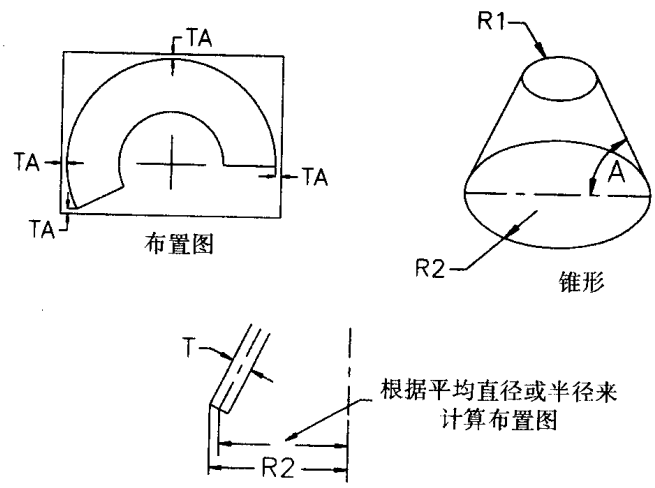

图12-28 圆椎及其平面布局

下面的文件为例 12 的 AutoLISP 程序清单：

```
;DTR function changes degrees to radiuns
(defun DTR(a)
(* Pi (/180.0 pi))
)
;RTD function changes radians to degrees
(defun rtd(a)
(* a(/ 180.0 pi))
```

```
)
(defun tan(a)
(/(sin a)(cos a))
)
(defun c：cone-lp(/)
(graphscr)
(setvar" cmdecho" 0)
(setq r2 (getdist "\n enter outer radius at larger end:")
(setq r1 (getdist "\n enter inner radius at smaller end:")
(setq t (getdist "\n enter sheet thickness:")
;this part of the program calculates various paramemeters
;needed in calculating the strip layout
(setq x0 0)
(setq y0 0)
(setq sf (/r2 3))
(setvar "dimscale " sf)
(setq ar a)
(setq tx(/(* t(sin ar))2))
(setq rx2(- r2 tx))
(setq rx1(+ r1 tx))
(setq w(*(* 2 pi)(cos ar)))
(setq  rl1(/ rx1(cos  ar)))
(setq rl2(/ rx2(cos ar)))

;this Part Of the program calculates the x-coordinate
;of the points
  (setq xl(+ x0 rl1)
      x3(+ x0 rl2)
      x2(- xO(* rll(cos( - pi w))))
      x4(- x0(* rl2(cos(- pi w))))
;this part of the program calculates the y-coodlnate
```

```
;of the points
(setq y1   y0
      y3 y0
      y2(+ y0(* rll(sin(- pi w))))
      y4(+ y0(* rl2(sin(- pi w))))
)

(setq p0(list x0 y0)
      P1(list   X1 yl)
      P2(list X2 y2)
      p3(list X3 y3)
      p4(list X4 y4)
      )
(command" layer"  "mak"  "ccto"  "c"  "1"  "ccto"  " ")
(command" arc" pl" c" p0 p2)
(command" arc" p3' 'c" p0 p4)
(command" line" pl p3"  ")
(command" line" p2 p4"  ")

(setq f1(/r2 24))
(set f2(/ r2 2))
(setq d1(list(+ x3 f2)y3))
(setq d2(list x0(- y0 f2)))

(command" layer"  "make"  "cctd"  "c"  "2"  "cctd"  " ")
(setvar" dimtih" 0)
(command" dim"  "hor" p0                                                    pl
d2"  " "baseline" p3"  " "baseline" p2"  " "baseline" p4"  " "exit" )
(command" dim"  "vert" p0 p2 dl"  " "baseline" p4"  " "exit" )
(setVar" dimscale" 1)
(setVar" cmdecho" 1)
```

(Princ)

)

复习题

1．求解下列 AutoLISP 函数

Command：(+ 2 30 5 50)　　Command：(+ 2 30 4 55.0)

(- 20 40)　(- 30.0 40.0)　(* 72 5 3 2.0)　(* 7 -5.5)

(/ 299 -5)　(/ -200 -9.0)　(l- 99)　(l- -18.5)

(abs -90)　(abs -27.5)　(sin pi)　(sin l.5)

(cos Pi)　(cos 1.2)　(atan l.1 0.0)　(atan -0.4　0.0)

(angtos 1.5708 0 5) (angtos -1.5708 0 3)　(< "X"　"y")　(>= 80 90 79)

2．setq 函数用于给　　赋值

3．　函数暂停程序的运行，允许用户输入一个点的 X、Y 坐标或 X、Y、Z 坐标。

4　　函数用于在 AutoLISP 程序内执行标准的 AutoCAD 命令。

5．一个 AutoLISP 表达式中的 AutoCAD 命令名及命令选项必须置于双引号中。(F/T)

6．Getdist 函数暂停程序的运行，等候用户输入　　并以实数形式返回该距离。

7．　　函数为 AutoCAD 系统变量赋值。系统变量名必须置于　　中。

8.cadr 函数执行两个操作，　　和　　，来返回列表中第二个元素。

9．　函数类似于\n 一样在屏幕上打印一个新行。

10．　　函数暂停程序的运行，等候用户输入角度，并以弧度形式返回该角度值。

11．　　函数总是以正 X 轴及逆时针方向测量角度。

12．____函数暂停程序的运行，等候用户输入一个整数。即使输入一个实数，该函数仍返回一个整数。

13．　函数允许检索 AutoCAD 系统变量的值。

14．　　函数根据与给定点的角度和距离来定义点。

15．　　函数计算一个数的平方根，并且总是返回一个实数。

16．　　函数将一个实数转换为字符串，并将该字符串返回。

17．if 函数求解测试表达式(> num1 num2)。若条件为真返回 _ ，不为真返回 _ 。

18．　　函数可以与 if 函数搭配以求解多个表达式。

19 .While 函数求解测试表达式 若条件为真(表达式不返回 nil) ,While 语句后的操作将　　直到测试表达式返回　　为止。

20．repeat 函数按照函数中指定的次数 n 反复求解表达式。变量 n 必须为实数。(T/F)

练习 6

编写一个 AutoLISP 程序，以 C1 为圆心，D1，D2,D3 为直径画出三个同心圆(见图 12-29)。程序还应提示用户输入圆心 C1 的坐标及圆的直径 D1，D2,D3o

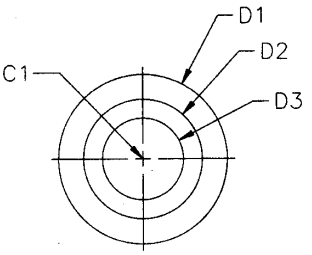

图12-29 直径分别为D1、D2、D3的三个同心圆

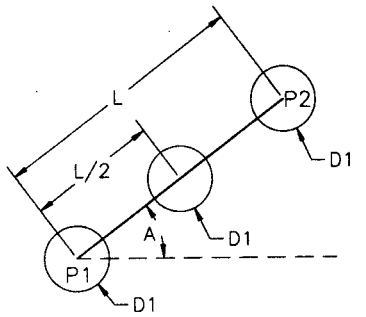

图12-30 圆和与X轴夹角为A的直线

练习 7

编写一个 AutoLISP 程序，从 P1 点到 P2 点画一条直线(见图 12-30)。直线 P1 P2 与正 X 轴的夹角为 A，P1 点与 P2 点间的距离为 L，圆的直径为 D1(D1=L/4)。

练习 8

编写一个 AutoLISP 程序，画出一个等腰三角形 P1 P2 P3(见图 12-31)。程序还应提示用户输入起点 P1，长度 L1 及内角 A。

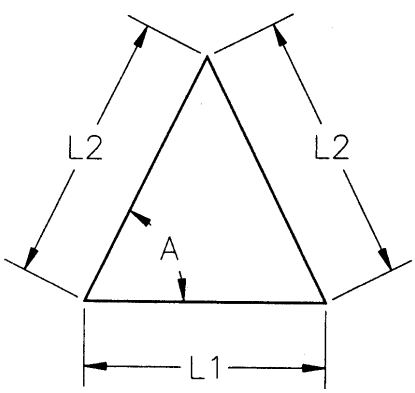

图12-31 等腰三角形

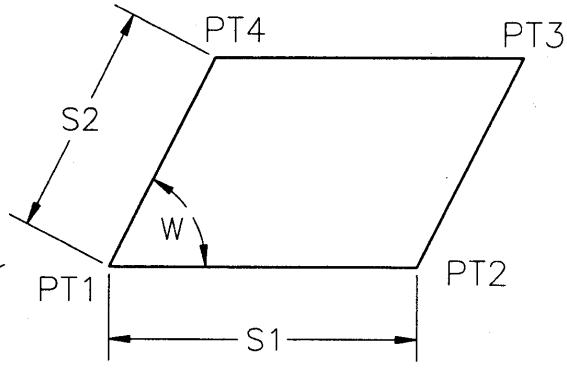

图12-32 边长为S1、S2,夹角为W的平行四边形

练习 9

编写一个 AutoLISP 程序，画一个边长为 S1、S2，夹角为 W 的平行四边形，如图 12-32 所示程序还应提示用户输入起点 PT1，长度 S1、S2 及夹角 W。

练习 10

编写一个 AutoLISP 程序 ,画出一个边长为 A 的正方形 ,以及该正方形的内切圆 ,如图 12-33 所示，正方形的底边与正 X 轴的夹角为 ANG。程序还应提示用户输入起点 P1，长度 S 及角 ANG。

练习 11

编写一个 AutoLISP 程序，画出一个等边三角形及其外接圆(见图 12-34)。程序应提示用户输入圆的半径及圆心。

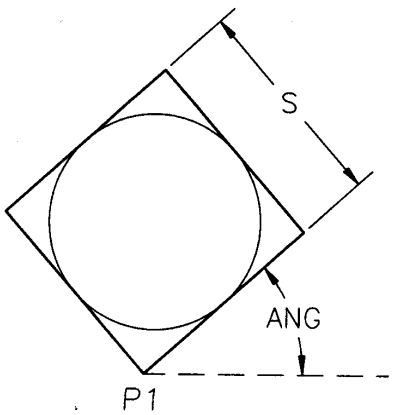

图12-33 边长为S, 倾角为ANG的正方形

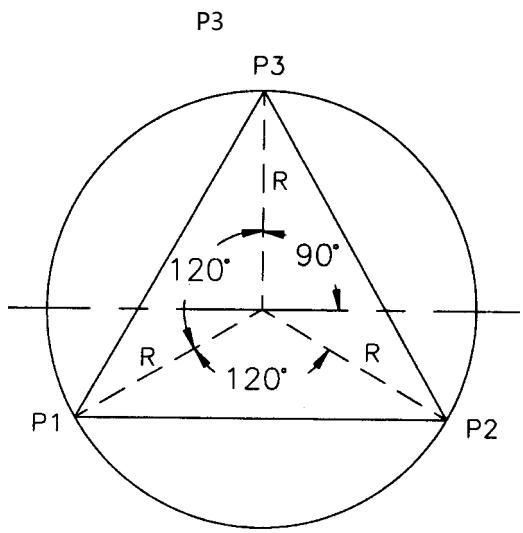

图12-34 圆内等边三角形

练习 12

编写一个 AutoLISP 程序 ,删除包含于上限(limmax)及下限(limmin)间的所有实体。使用 AutoCAD 的 SETVAR 及 ERASE 命令删除这些实体。

练习 13

编写一个 AutoLISP 程序，画出两个圆及它们的公切线，如图 12-35 所示，程序还应提示用户输入圆的直径及两圆的中心距。

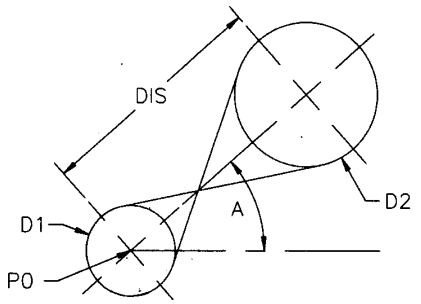

图12-35 带有公切线的圆，其中心线倾角为A

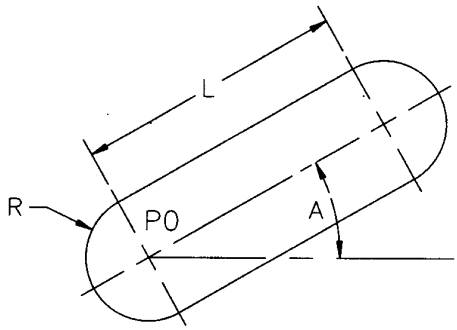

图12-36 长度为L,半径为R的键槽

练习 14

编写一个 AutoLISP 程序，画出一个带有中心线的键槽。该程序应提示用户输入键槽的长度,宽度及中心线的图层名(见图 12-36)。

练习 15

编写一个 AutoLISP 程序，画出一条直线，并生成给定数目(N 条)的与其平行的直线。

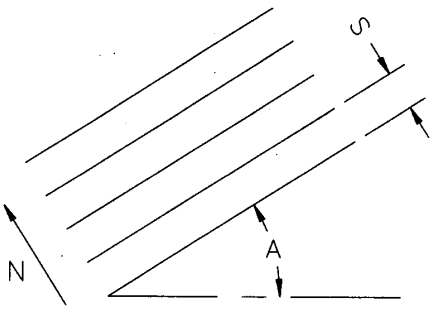

图12-37 N条间距为S的平行线

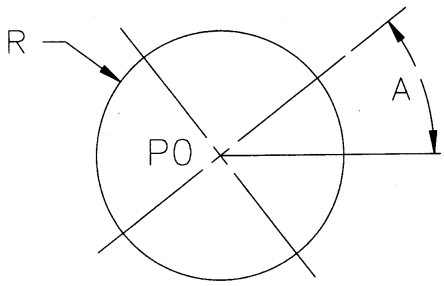

图12-38 中心线倾角为A的圆

练习 16

编写一个 AutoLISP 程序，画出一个带中心线的圆。该程序应提示用户输入圆的直径，圆心及中心线倾角，如图 12-38 所示。

练习 17

编写一个 AutoLISP 程序，画一个键槽。该程序应提示用户输入该键槽的宽度、深度、角度及起始点。如图 12-39 所示。

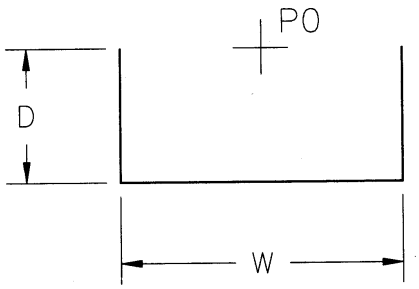

图12-39 宽度为W,深度为D的键槽

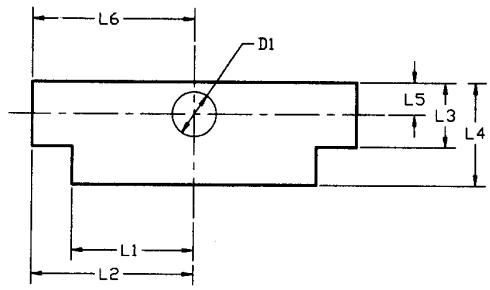

图12-40 练习18用图

练习 18

编写一个 AutoLISP 程序，画出如图 12-40 所示的带中心线以及尺寸标注的图形。假设 L5=D1，L3=1.5*D1,L6=10*D1,L1=L6-D1,L4=L3+D1。

练习 19

编写一个 AutoLISP 程序，画出如图 12-41 所示的带键槽的轮毂。该程序应提示用户输入 P0(轮毂中心或轴心)的值、D1(轴径)、D2(轮毂外径)、W(键宽)以及 H(键高)。程序应将中心线画在 Center 层(绿色)，将尺寸 T 和 W 画在 Dim 层(紫红色)

练习 20

编写一个 AutoLISP 程序，画出如图 12-42 所示的套筒的两个视图。该程序应提示用户输入起点 P0，长度 L1、L2 以及套筒直径 ID、OD、HD。前视图与侧视图之间的距离为 DIS(DIS=l.25*HD)。程序应将隐藏线画在 HID 层，将中心线画在 CEN 层。中心线超出实体边界线 0.75 个单位。

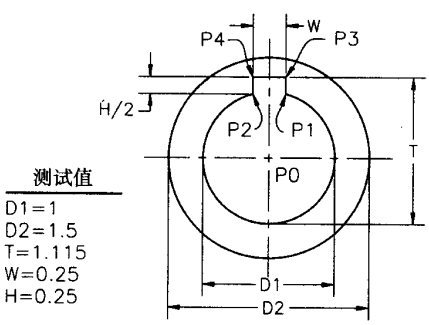

图12-41 练习19用图

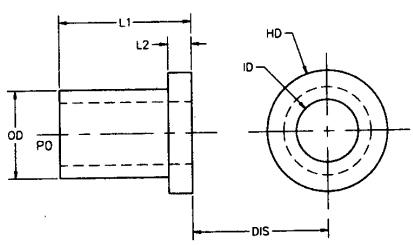

图12-42 套筒的两个视图