

AUTOLISP (2.62 版)

程序设计及应用

霍新民 王国泉 汪琪美

H

中国科学院希望高级电脑技术公司

AUTOLISP (2.62 版)

程序设计及应用

霍新民 王国泉 汪琪美

中国科学院希望高级电脑技术公司

一九八九年四月

内 容 提 要

本书全面系统地介绍AUTOLISP 2.62版语言和程序设计的基本方法。全书共十章，介绍AUTOLISP语言的基础知识、顺序结构、分支结构、循环结构、子程序结构、输入输出功能和图形数据库的存取功能。书末的七个附录详细介绍了系统变量，DXF文件，内存管理以及错误信息表等内容，每章后均附有习题。

本书由浅入深，通俗易懂，内容详尽，例题丰富，重点突出，对所有的概念和函数均有例子加以说明。对AUTOCAD和AUTOLISP中比较难懂的问题重点进行了讲解，

本书可作大专院校CAD课程的教材，也可供工程技术人员和大专院校师生开发CAD系统时参考。

AUTOLISP (2.62版)

程序设计及应用

霍新民 王国泉 汪琪美

*

中国科学院希望高级电脑技术公司出版

中国人民解放军四二二九厂印刷

开本787×1092毫米 1/16 字数506.5千字

1989年4月第一版 1989年4月第一次印刷

北京市新闻出版局准印证号：891198

前 言

谈起微机CAD系统,恐怕AUTOCAD是目前国内最为流行的了。究其原因,可罗列若干,然最主要的是其结构的开放性。AUTOCAD为用户留出了广阔的用武之地,提供了许多进行二次开发的工具,论其功能,AUTOLISP是最强的一个。鉴于此,人们希望了解AUTOLISP的程序结构,掌握AUTOLISP程序设计的方法,进而开发出适合本专业特点的应用CAD系统。然而,打开AUTOLISP程序员参考手册,除了屈指可数的几个例子之外,剩下的就是按字典顺序排列起来的函数功能简介,使人望之却步,看后不得要领。

由于科研与教学的需要,我们只好来啃这一不易咀嚼的涩果。查阅有关资料,阅读现成例子,对照参考手册,在计算机上慢慢摸索,历两年多时间,方对AUTOLISP语言有点粗浅的理解,积累了一些资料,和关心或从事CAD工作的同志切磋交流。

本书共有十章和七个附录,系统而详细地介绍了AUTOLISP 2.62版的全部功能。结合初学者的特点,沿用其它高级语言教材的编写顺序,编入了大量的AUTOLISP程序,使读者从例子中深化对概念的理解,领悟程序设计的方法。每章后面附有习题,以供读者练习。

由于我们对AUTOLISP语言理解尚浅,加之手头资料的匮乏,错误疏漏之处一定不少,恳请读者不吝赐教。

编 者

一九八九年一月于北京

AUTOLISP程序设计及应用

供开发和应用CAD系统的专用者参考

本资料主要介绍

- LISP程序 • 分支结构 • 循环结构 • 表 •
- 字符串操作 • 输入与输出 • 自定义函数 •
- 图形数据库存取函数 • 系统变量 • 出错信与出错处理 • DXF文件 •
- 符号表与实体数据表 • ASCII表 •
- AUTOCAD2.62命令表 • AUTOLISP内存管理。

订购：北京8721信箱

电话：2562329，2561058

地址：海淀影剧院北侧

乘车：320、332路在海淀黄庄站下车

目 录

第一章 绪论	(1)
§ 1.1 AUTOLISP语言是开发专业CAD系统的理想工具	(1)
§ 1.2 AUTOLISP语言的程序结构	(2)
§ 1.3 AUTOLISP程序的编辑、装入和运行	(4)
§ 1.4 LOAD函数	(6)
§ 1.5 辅助调试工具——TRACE函数和UNTRACE函数	(7)
习题	(10)
第二章 AUTOLISP语言的基本成份	(12)
§ 2.1 常数	(12)
§ 2.2 变量	(14)
§ 2.3 算术运算函数	(15)
§ 2.4 类型转换函数	(20)
§ 2.5 表达式	(21)
§ 2.6 COMMAND函数	(22)
§ 2.7 与表达式求值有关的函数	(24)
一、AUTOLISP的求值器	(24)
二、QUOTE函数禁止对表达式求值	(25)
三、EVAL函数进行再求值	(26)
习题	(27)
第三章 简单的AUTOLISP程序	(29)
§ 3.1 赋值函数SETQ	(29)
§ 3.2 结果输出函数PRINT	(30)
§ 3.3 简单的程序	(31)
§ 3.4 几何函数	(32)
一、求两点间距离的函数DISTANCE	(32)
二、求交点坐标函数INTERS	(32)
三、求向量的方向角函数ANGLE	(34)
四、求下一点坐标的函数POLAR	(34)
五、特征点函数OSNAP	(35)
六、点的坐标值	(35)
§ 3.5 交互式输入函数	(36)
一、整数输入函数GETINT	(37)
二、实数输入函数GETREAL	(37)

三、字符串输入函数GETSTRING.....	(37)
四、点输入函数GETPOINT.....	(37)
五、角度输入函数GETANGLE	(38)
六、方位输入函数GETORIENT.....	(39)
七、距离输入函数GETDIST.....	(42)
八、窗口输入函数GETCORNER.....	(43)
九、输入控制函数INITGET.....	(43)
十、只接受关键字的字符串输入函数GETKEYWORD.....	(46)
§ 3.6 SET函数与SETQ函数.....	(50)
习题.....	(51)
第四章 分支结构	(52)
§ 4.1 关系运算函数.....	(52)
一、小于函数.....	(52)
二、大于函数.....	(52)
三、小于等于函数.....	(53)
四、大于等于函数.....	(53)
五、等于函数.....	(53)
六、不等于函数.....	(53)
§ 4.2 逻辑运算函数.....	(53)
一、逻辑与函数AND.....	(53)
二、逻辑或函数OR.....	(54)
三、逻辑非函数NOT.....	(54)
§ 4.3 两分支结构.....	(54)
一、IF函数.....	(55)
二、几点说明.....	(55)
三、举例.....	(57)
§ 4.4 多分支结构.....	(59)
一、COND函数的格式.....	(59)
二、功能.....	(60)
三、几点说明.....	(60)
四、举例.....	(61)
§ 4.5 位方式逻辑函数.....	(64)
一、按位与函数LOGAND.....	(64)
二、按位或函数LOGIOR.....	(65)
三、按位求反函数~.....	(65)
四、通用位操作函数BOOLE.....	(65)
五、逻辑移位函数LSH.....	(67)
§ 4.6 其它测试函数.....	(67)
一、原子测试函数ATOM.....	(68)

二、表测试函数LISTP.....	(68)
三、测试原子是否有值的函数BOUNDP.....	(69)
四、测试表达式是否有值的函数NULL.....	(69)
五、测试表达式是否为数值型的函数NUMBERP.....	(69)
六、测试表达式是否为负数的函数MINUSP.....	(70)
七、测试表达式的值是否为零的函数ZEROP.....	(70)
八、EQ函数与EQUAL函数.....	(70)
九、类型检测函数TYPE	(71)
习题.....	(72)
第五章 循环结构	(76)
§ 5.1 WHILE 函数.....	(76)
一、WHILE函数的格式.....	(77)
二、WHILE函数的执行过程.....	(77)
三、几点说明.....	(78)
§ 5.2 多重循环.....	(80)
§ 5.3 REPEAT函数.....	(82)
§ 5.4 程序举例	(83)
习题.....	(91)
第六章 表	(95)
§ 6.1 表的概念.....	(95)
§ 6.2 表的操作.....	(96)
一、LIST函数.....	(96)
二、LENGTH函数.....	(97)
三、CONS函数.....	(97)
四、NTH函数.....	(97)
五、LAST函数.....	(98)
六、CAR函数.....	(98)
七、CDR函数.....	(99)
八、CAR和CDR派生的组合函数.....	(99)
九、REVERSE函数	(100)
十、MEMBER函数.....	(101)
十一、APPEND函数.....	(102)
十二、SUBST函数.....	(103)
§ 6.3 其它与表有关的函数.....	(103)
一、APPLY函数.....	(104)
二、MAPCAR函数.....	(105)
三、FOREACH函数.....	(106)
§ 6.4 关联表.....	(108)
一、关联表的概念.....	(108)

二、关联表的检索·····	(108)
三、关联表中元素的代换·····	(108)
四、CONS函数的另一个重要功能·····	(109)
§6.5 程序举例·····	(111)
习题·····	(119)
第七章 字符串操作 ·····	(121)
§7.1 字符常数·····	(121)
§7.2 字符串操作函数·····	(122)
一、字符串链接函数 STRCAT·····	(122)
二、求字符串长度函数 STRLEN·····	(122)
三、取子串的函数 SUBSTR·····	(122)
四、由ASCII码转换成字符的函数 CHR·····	(123)
五、求一个字符的ASCII码的函数 ASCII·····	(124)
六、大小写转换函数 STRCASE·····	(124)
§7.3 字符的比较·····	(125)
§7.4 字符型数据和其它类型数据的转换函数·····	(126)
一、数字串转换成实数的函数 ATOF·····	(126)
二、数字串转换成整数的函数 ATOI·····	(127)
三、整数转换为整数串的函数 ITOA·····	(127)
四、实数转换为实数串的函数 RTOS·····	(127)
五、角度转换为字符串的函数 ANGLOS·····	(123)
六、READ函数·····	(129)
§7.5 程序举例·····	(129)
习题·····	(135)
第八章 输入与输出 ·····	(137)
§8.1 常用的输出函数·····	(137)
一、PRINC函数·····	(137)
二、PRINT函数·····	(139)
三、PRINI函数·····	(140)
四、WRITE—LINE函数·····	(140)
五、WRITE—CHAR函数·····	(143)
六、PROMPT函数·····	(143)
七、TERPRI函数·····	(144)
§8.2 文件的存取·····	(145)
一、OPEN函数·····	(145)
二、CLOSE函数·····	(147)
三、文件的写操作·····	(147)
四、文件的读操作·····	(147)
五、举例·····	(148)

§ 8.3 其它输入输出函数	(150)
一、TEXTSCR 函数	(150)
二、GRAPHICS 函数	(150)
三、GRCLEAR 函数	(150)
四、GRDRAW 函数	(151)
五、GRTEXT 函数	(153)
六、GRREAD 函数	(154)
七、REDRAW 函数	(155)
八、菜单调用函数MENUCMD	(156)
§ 8.4 程序举例	(159)
习题	(165)
第九章 自定义函数	(169)
§ 9.1 函数的定义	(171)
一、用户函数定义语句DEFUN	(171)
二、DEFUN函数的值	(172)
三、用户函数值的数据类型	(173)
四、用户函数哑元的类型	(174)
五、全局变量与局部变量	(174)
§ 9.2 函数的副作用	(174)
一、全局变量	(174)
二、局部变量	(175)
三、函数的副作用	(176)
§ 9.3 函数的调用	(177)
一、调用规则	(177)
二、举例	(178)
三、特殊用户函数名	(186)
§ 9.4 λ 函数	(187)
一、 λ 函数的定义	(187)
二、 λ 函数的调用	(187)
习题	(189)
第十章 图形数据库存取函数	(192)
§ 10.1 选择集操作函数	(192)
一、选择集构造函数 SSGET	(193)
二、选择集长度函数 SSLENGTH	(194)
三、检索实体名函数 SSNAME	(194)
四、向选择集中加入新实体的函数 SSADD	(194)
五、从选择集中移出实体的函数 SSDEL	(195)
六、测试某实体是否在选择集中的函数 SSMEMB	(196)
§ 10.2 实体名操作函数	(197)

一、从图形数据库中得到实体名的函数 ENTNEXT.....	(197)
二、从图形数据库中得到最后一个主实体名的函数 ENTLAST.....	(198)
三、得到指定实体名的函数 ENTSEL.....	(199)
§ 10.3 实体数据函数.....	(199)
一、删除实体的函数 ENTDEL.....	(200)
二、得到实体定义数据的函数 ENTGET.....	(201)
三、实体数据表更新的函数 ENTMOD.....	(202)
四、实体再生函数 ENTUPD.....	(203)
五、程序举例.....	(203)
§ 10.4 符号表访问函数.....	(209)
一、TBLNEXT函数.....	(209)
二、TBLSEARCH函数.....	(210)
习题.....	(238)
附录A 系统变量	(239)
A.1 概 述.....	(239)
A.2 AutoCAD2.62版的系统变量综述.....	(240)
A.3 系统变量的使用说明.....	(245)
一、与系统环境有关的变量.....	(245)
二、与UNITS命令有关的变量.....	(247)
三、与绘图命令有关的变量.....	(249)
四、与编辑查询命令有关的变量.....	(250)
五、与显示控制命令有关的变量.....	(251)
六、与辅助绘图命令有关的变量.....	(252)
七、控制尺寸标注的变量.....	(254)
附录B 出错信息与出错处理	(257)
B.1 出错信息表.....	(257)
B.2 内部出错处理函数.....	(259)
B.3 用户自定义出错处理函数.....	(259)
附录C DXF文件	(262)
C.1 概 述.....	(262)
一、DXF文件的总体结构.....	(262)
二、组码与组值.....	(263)
C.2 标题节.....	(264)
C.3 表节.....	(265)
一、线型表.....	(267)
二、层表.....	(267)
三、字体表.....	(268)
四、视图表.....	(269)
C.4 块节.....	(269)

一、用户块.....	(270)
二、图案块.....	(270)
三、尺寸块.....	(270)
C.5 实体节.....	(271)
附录D 符号表与实体数据表	(277)
D.1 层表中每层的数据定义表.....	(278)
D.2 线型表中每种线型的数据定义表.....	(279)
D.3 字体表中每个字体的数据定义表.....	(280)
D.4 视图表中每个视图的数据定义表.....	(282)
D.5 块表中每个块的数据定义表.....	(283)
D.6 实体数据表.....	(289)
附录E ASCII (美国标准信息交换码) 表	(312)
附录F AutoCAD2.62命令表	(314)
F.1实用命令.....	(314)
F.2 实体绘制命令.....	(314)
F.3 层、颜色和线型命令.....	(315)
F.4 块与属性的命令.....	(315)
F.5 显示控制命令.....	(315)
F.6 编辑查询命令.....	(316)
F.7 辅助绘图工具.....	(317)
F.8 其它命令.....	(317)
附录G AUTOLISP的内存理管	(318)
G.1 正确设置堆栈空间.....	(318)
一、堆与栈.....	(318)
二、堆栈空间的设置.....	(318)
G.2 结点空间的回收.....	(319)
G.3 清除所有的用户函数和变量.....	(320)
G.4 虚拟函数分页功能.....	(321)
参考书目	(322)

第一章 绪 论

§ 1.1 AutoLISP语言是开发专业CAD系统的理想工具

计算机辅助绘图和设计 (CADD) 大约出现于60年代初期, 直到80年代初期才得以广泛应用。此时, AutoCAD一类微机CAD系统相继问世。它们把以前在大型计算机和小型计算机上才能实现的高级技术, 奉献给广大的微机用户。促进了CAD技术的广泛应用。微机CAD系统能够完成大型机CAD系统百分之八十的工作, 而代价仅为大型机CAD系统的百分之十。这就是通常所说的CAD的80/10规律。

美国AutoDESK公司研制的AutoCAD系统自问世以来, 即以惊人的速度向前发展, 其版本更新的神速真令人有点目不暇接, 在国内外得到了广泛的赞许和应用。AutoCAD在激烈的微机CAD系统的竞争中, 之所以能得以生存和发展, 除了其它一些因素之外, 其自身的优点仍然是其成功的基础。

没有任何一个CAD软件系统是对所有人和所有情况都是适用的, 所谓“通用”是相对的。AutoCAD是一个通用的软件, 这只是说, 它为各种专业应用CAD提供了必要的尽可能多的图形编辑功能。要使一个通用的CAD系统适合自己专业的实际需要, 必须进行二次开发, 或者说, 把通用的CAD系统个性化。为此, 几乎所有先进的CAD系统都为用户进行二次开发提供了可能, 其中最重要的是编程语言。

AutoCAD向人们提供了许多二次开发的工具。如菜单、图案、线型、样板图、数据交换文件 (DXF), 宏命令文件 (SCR) 和 AutoLISP 语言。AutoCAD充分开放的结构, 是其成功的主要原因。它不仅为用户提供了丰富的绘图和编辑功能, 而且为用户留出了发挥自己聪明才智的广阔的用武之地。AutoLISP语言是主要的二次开发工具。

LISP语言是一个古老的人工智能语言。自1960年由J. McCarthy提出已有近三十年的历史, 是迄今在人工智能学科领域中应用最广泛的一种程序设计语言。不论是早期的人工智能程序, 还是近期著名的人工智能系统, 大部分是用LISP语言写成的。人们认为, LISP语言曾武装了一代人工智能科学家。虽然近年来也出现了一些新的人工智能语言, 但就其应用的广泛性和普遍性来看, LISP仍然是最主要的人工智能程序设计语言。

和其它许多高级语言一样, LISP在其发展过程中产生了许多方言——MacLISP、InterLISP、ZetaLISP和CommonLISP。

AutoLISP采用了和CommonLISP最接近的语法和习惯约定, 可以说, 它是CommonLISP的一个很小的子集。

总的来看, AutoLISP具有如下特点:

1. 它是一个嵌入式语言。它不是运行在操作系统下, 而是运行于AutoCAD的环境下, 或者说, 它嵌入在AutoCAD之中。
2. 采用了LISP语言的语法。
3. 吸收了LISP语言的主要语句 (或称函数), 具有LISP语言的语法结构。

4. 加入了与图形处理有关的语句。

5. 可直接调用几乎全部的AutoCAD命令。

不难看出, AutoLISP语言既具有一般高级语言的基本结构和功能, 又具有一般高级语言所没有的强大的图形处理能力。因此, AutoLISP程序既可完成通常的科学计算和数据处理, 又能直接调用AutoCAD的命令。这两者的有机结合, 使之成为专业CAD系统的强有力的开发工具。

§1.2 AutoLISP语言的程序结构

一例胜千言, 让我们先从下面的几个例子, 对AutoLISP程序的概貌有一大致了解, 读者不要急于弄清程序的细节。

【例一】编写求解具有两实根的一元二次方程 $ax^2+bx+c=0$ 的程序。

FORTRAN程序	AutoLISP程序
C CALCULATE ROOTS	, CALCULATE ROOTS
READ (*,100)A,B,C	(SETQ A (GETREAL)); enter A, B, C
100 FORMAT(3F10.0)	(SETQ B (GETREAL) C (GETREAL))
D= B*B-4.0*A*C	(SETQ D (- (* B B)(* 4.0 A C)))
X1=(-B+SQRT(D))/(2.0*A)	(SETQ X1 (/ (+ (- B) (SQRT D))(* 2.0 A)))
X2=(-B-SQRT(D))/(2.0*A)	(SETQ X2 (/ (- (- B)(SQRT D))(* 2.0 A)))
PRINT *, X1, X2	(PRINT X1) (PRINT X2)
END	

从上面的FORTRAN程序和AutoLISP程序的对照中, 我们可以看出AutoLISP程序具有和FORTRAN等高级语言一样的数值计算能力, 只不过语句格式和表达式的书写格式不同罢了。

【例二】在图形屏幕上画圆心在(5, 5), 半径为5的圆, 并在其中画上45度的阴影线。

使用AutoCAD命令, 对话过程如下:

Command: CIRCLE

3 P/ 2 P/TTR/ <Center point> : 5, 5

Diameter/ <Radius> : 5

Command: HATCH

Pattern (? or name/U,style) : U

Angle for crosshatch lines <0> : 45

Spacing between lines <1.000> : 1

Double hatch area? <N> N

Select objects: L

1 found

Select objects:

实现相同功能的AutoLISP程序如下:

```
(DEFUN C:CH ( )  
  (COMMAND "CIRCLE" "5,5" 5)  
  (COMMAND "HATCH" "_U" 45 1 "N" "L" "")  
)
```

对于这个例子来说,去掉第一行和最后一行也是可以的。此例说明,AutoLISP程序可以调用AutoCAD的几乎全部命令,这个程序也可称为AutoCAD命令的批处理文件。

〔例三〕画一个长方体和一个圆柱体,要求圆柱的底面座在长方体的顶面之上,且圆柱体底面圆心与长方体顶面的中心重合,圆柱底面半径为长方体宽度的四分之一。长方体的长 L ,宽 W ,高 H_1 ,圆柱的高 H_2 以及长方体左下角的位置 P_0 由用户临时确定。

AutoLISP程序如下:

```
(DEFUN C:F2 ( )  
  (SETQ P0 (GETPOINT "\nLower left corner: "))  
  (SETQ L (GETDIST P0 "\nLength: "))  
  (SETQ W (GETDIST P0 "\nwidth: "))  
  (SETQ H1 (GETDIST P0 "\nH1: "))  
  (SETQ H2 (GETDIST P0 "\nH2: "))  
  (SETQ R (/ w 4.0))  
  (SETQ P1 (POLAR P0 0.0 L))  
  (SETQ P2 (POLAR P1 (/ PI 2.0) W))  
  (SETQ P3 (POLAR P2 PI L))  
  (SETQ PC (INTERS P0 P2 P1 P3))  
  (COMMAND "ELEA" 0.0 H1)  
  (COMMAND "PLINE" P0 P1 P2 P3 "C")  
  (COMMAND "ELEA" H1 H2)  
  (COMMAND "CIRCLE" PC R)  
  (COMMAND "ZOOM" "E")  
  (COMMAND "VPOINT" "1,-1,1")  
  (COMMAND "HIDE")  
)
```

这个例子是AutoLISP计算功能和图形功能的结合。读者可以把上述三个例子输入,然后运行它,借以体会AutoLISP程序的功能。

从上述例子中可以看出,AutoLISP程序具有如下结构特点:

一、AutoLISP程序由表组成。这也就是LISP语言又称表处理语言的原因。所谓表,就是用圆括号括起来的用空格分隔的项形成的数据结构形式。由于表中的每一项又可以为表,因此,表可以嵌套任意层。表是AutoLISP语言的基本形式,因此在本书中对“函数”,“表”,“表达式”和“语句”不加严格区别。例如,例一中计算 b^2-4ac 并将其赋值给D的语句

```
(SETQ D (- (* B B) (* 4.0 A C)))
```

就是一个表，它有三项SETQ, D和 $(- (* B B) (* 4.0 A C))$ ；第三个表项是一个表，有三项 $-$, $(* B B)$ 和 $(* 4.0 A C)$ ，而后两项又是表，一个有三项 $*$, B和B，一个有四项 $*$, 4.0, A和C。对于任何一个表来说，相邻两项之间一般用一个或多个空格分开，以便计算机能分清表项，不致产生误解（当然，有时，两项间不留空格，计算机也能分清表项。如相邻两表项中，有一个是带括号的表常数，带双撇号的字符串等）。计算机把表的第一项认作函数名（表示其功能的操作码），其余各项认作该函数的变元。例如， $(* 4.0 A C)$ ，表的第一项“ $*$ ”，计算机认作函数名，表示此表执行累乘运算，其余三项认作变元，即累乘函数操作的对象。因此， $(* 4.0 A C)$ 即求三个数的连乘积。而写成 $(* 4.0 AC)$ ，计算机则把 $* 4.0$ 看成一个函数名，其操作对象为AC，因为没有名为 $* 4.0$ 的函数使程序出错而中断运行。这是AutoLISP程序设计中常见最顽固的错误，不少初学者常常需要很长时间，经过多次碰壁之后才会改正。

一个AutoLISP程序或一个用户定义的函数（或称子程序）也是一个表。如例二的程序就是一个表。

二、为改善程序的可读性，可在程序的任何地方加入注释内容。注释的标志是分号。计算机把从分号开始到本行（指逻辑行）末尾的全部内容认作注释。计算机对注释内容并不解释执行。

三、除字符串中的字母外，AutoLISP程序中的字母可以大写，也可以小写。计算机系统理解为大写。

四、由于两个表项之间空格个数不限，一个程序中的所有语句（即表）可以一个接着一个写下去（只要本文编辑程序允许），即一行可以写多个语句。也可在两个表，甚至两个表项之间留任意多个空格和空行，这样，我们就可把程序写成阶梯状，以改善程序的可读性。总之，AutoLISP是一种书写格式非常自由的语言。

§1.3 AutoLISP程序的编辑，装入和运行

一、程序的编辑

AutoLISP程序要输入到具有固定后缀（.LSP）的文件中，文件的后缀表示程序的类型，就如同FORTRAN程序的后缀为.FOR，BASIC程序的后缀为.BAS一样。文件中程序的编辑——程序的输入和修改，一般是用文本编辑命令EDLIN进行的。

AutoLISP程序可以驻留在任何驱动器的任何目录之下。因此，完整的文件描述符应包括磁盘驱动器名，目录名，文件名和后缀。例如，驻留在当前目录下的AutoLISP文件EXAMPLE.LSP，驻留在A盘根目录下的文件A:ABC.LSP，驻留在C盘目录/CAPS/LSP下的文件C:/CAPS/LSP/FL.LSP。

凡是在能调用EDLIN命令的地方，都可输入和修改AutoLISP程序，有三种可能：

1. 在AutoCAD之外，即进入AutoCAD之前或退出AutoCAD之后进行

C> EDLIN EXAMPLE.LSP

2. 在AutoCAD之中，用AutoCAD命令SHELL调用EDLIN命令进行

Command: SHELL

DOS Command: EDLIN EXAMPLE.LSP)

3.在AutoCAD之中,用AutoCAD命令EDIT进行

Command: EDIT)

File to EDIT: EXAMPLE.LSP)

二、程序的装入

驻留在外部存储介质上的AutoLISP程序只有装入到内存才能运行。程序的装入只能在AutoCAD当中才能进行。因为它是用AutoLISP语言的内部函数LOAD来实现的。读者欲了解LOAD函数的功能,可参考§1.4节。

装入函数LOAD把欲运行的AutoLISP程序装入内存,交给解释程序解释执行。AutoLISP解释程序放在AutoCAD覆盖文件ACADL.OVL中,要求最小内存512K,其更新信息包含在文件README.DOC之中。AutoCAD配置菜单允许用户决定,是否选用AutoLISP功能。当上述条件满足时,才能在AutoCAD中装入和运行AutoLISP程序。

装入操作可以在Command提示符下进行,例如,

Command: (Load "EXAMPIE"))

也可以把它放在菜单中或SCR文件中实现。

由于文件的内容不同,装入的结果也不相同。

1.文件中没有用户函数时,系统装入文件并自动执行其中的语句,即融装入和执行于一体。§1.2节中的例一属于这种情况。不过,这种形式的程序,运行一次即需装入一次。

2.文件只由用户函数组成,除以分号打头的注释文字外,函数体外没有其它语句。在这种情况下,装入操作只是把文件中的用户函数装入内存,并不能调用这些函数。§1.2节中的例二属于这一情况。

3.文件中既有用户函数,在函数之外又有其它语句。装入时,自动执行用户函数体外的其它语句,而把用户函数装入内存。函数体外的语句不一定集中在文件的开头,它可以散布在文件的任何地方。

需要重复说明的是,装入时,被装入的是后缀为LSP的文件名,它必须用双撇号括起来,不可带后缀LSP。例如,要装入EX.LSP,下面的格式都是错误的:

Command: (LOAD EX))

Command: (LOAD "EX.LSP"))

Command: LOAD "EX")

装入时,如果此程序文件中的括号不配对,则会提示出错信息:

n)

其中,n是一个整数,它表示缺少的右括号数,一般情况下,只要键入n个右括号,即可从这种状态下解脱出来,而出现Command提示符。有时,即使键入n个右括号仍不起作用。这是因为,此时的括号不配对是由于字符串的双撇号不配对,系统对双撇号后的右括号认为是字符串中的字符,新键入的右括号也被纳入字符串中。此时,必须先键入一个双撇号,再键入n个右括号,即可从这种状态下解脱出来。

如果出现其它出错信息,请参考附录B的错误信息表,找出错误所在,用EDLIN命令