

AutoLISP 及应用开发技术

陈道洁 付守默 编著

- Auto LISP 函数及其应用
- 数据处理方法与技术
- 图形处理方法与技术
- 图形标注方法与技术
- 菜单与对话框
- 程序调试方法
- CAD 应用软件举例

成都科技大学出版社



Auto LISP 及应用开发技术

陈道洁 付守默 编著

成都科技大学出版社

JS14/08

(川)新登字 015 号

责任编辑 毕腾弟

内 容 简 介

本书是在作者多年从事 Auto CAD 教学和进行工程 CAD 系统开发的基础上编写的,本书在全面系统的介绍 Auto LISP 语言的基础上,着重介绍了该语言在工程 CAD 系统中的应用,开发技术与方法以及大量实用函数和典型程序。它是国内少有的系统介绍实用 CAD 系统开发方法与技术的教学用书。

本书共分两篇十六章,第一篇 Auto LISP 程序设计语言(共十章),分类介绍 Auto LISP 系统函数及其在工程 CAD 中的应用;第二篇 工程 CAD 系统的开发方法与技术(共六章),介绍数据处理方法与技术、图形处理和图形标注方法与技术,以及菜单与对话框应用、程序调试方法与出错处理和典型零件 CAD 系统分析等。

本书由浅入深,循序渐进,图文并茂,便于教学和自学。本书内容丰富,实用性强,反映了技术发展的新水平。

本书可作为高等学校有关专业的本科生和研究生教学用书和毕业论文、机械零件设计、模具设计的参考书。亦可作为工厂、设计研究单位从事工程设计和 CAD 的工程技术人员参考书。

Auto LISP 及应用开发技术

陈道洁 付守默 编著

成都科技大学出版社出版发行

冶金部西南勘察局测绘制印厂 印刷

开本:787×1092 1/16 印张:18

1995 年 9 月第 1 版 1997 年 4 月第 2 次印刷

字数:540 千字 印数:2001—5000 册

ISBN 7-5616-3105-7/TP·137

定价:22.00 元

前 言

计算机辅助设计(Computer—Aided Design 简称 CAD)是一种人与计算机相结合各尽所长的新型设计方法,它是以计算机为主要工具处理产品设计各个阶段的知识、数据和图形信息,并与计算机辅助制造(Computer—Aided Mamufacturing 简称 CAM)为一体,以实现产品设计和制造过程的自动化。计算机绘图(Computer Graphics 简称 CG)是 CAD 的重要组成部分,也是 CAD 技术应用中最为成功的方面。在世界工业发展国家制造业中 CG, CAD, CAM 获得广泛地应用,工程图纸已几乎全部用计算机绘制,大大地缩短了产品的设计周期,提高了产品设计质量。为增强产品的市场竞争力并与国际市场接轨,我国已确立了 2000 年工程图纸计算机化的目标,并将计算机技术的应用情况作为企业评价的指标之一,因此,对于工程技术人员来说,不仅要懂得计算机操作使用,更重要的是要掌握 CG、CAD、CAM 等先进技术。.

在微机 CAD/CAM 软件领域中,美国 Auto desk 公司的 Auto CAD 软件一花独秀,具有不可否认的市场优势。Auto CAD 以其强大的图形编辑和处理功能,开放性结构以及良好的性能价格比,在激烈的竞争中以惊人的速度向前发展,其版本更新之快令人目不暇接,在国内外博得了高度的赞许和广泛的应用,在我国已家喻户晓。Auto CAD 的开放性结构不但保证了系统本身的不断扩充与完善,而且也方便了用户针对各工程领域的特点,进行实用 CAD 系统的二次开发。Auto LISP 语言是 Auto CAD 的较重要的基本编程语言,它综合了人工智能高级语言 LISP 的特性和 Auto CAD 强大的绘图编辑功能的特点,可以说它是人工智能绘图语言,因此,Auto LISP 是工程 CAD 系统,特别是绘图系统的重要开发工具之一。

学习掌握 Auto LISP 语言的目的是利用它开发各工程领域实用 CAD 系统,这除了学习语言本身外,还要在长期实践中,不断提高编程能力,掌握开发技术与方法,开发和累积实用函数和典型程序。

本书根据作者多年从事 Auto CAD 教学和实用 CAD 系统(如在 Auto CAD R10.0 基础上主持开发的 MECAD 通用机器零件设计与绘图系统,GTS 绘图工具系统,塑料注射模标准模架图形系统和在 Auto CAD R12.0 基础上开发的冷冲模 CAD 系统等)的研制与开发的实践,系统地介绍实用 CAD 系统的开发技术与方法、以及大量实用函数、典型程序和 CAD 开发实例、以提高读者二次开发实用 CAD 系统的能力。本书中介绍的 Auto LISP 语言 and 开发技术是以 Auto CAD R10.0 为基础并照顾 Auto CAD R12.0 的发展。本书介绍的实用函数和典型程序都适用于 10.0,11.0 和 12.0 版本。

本书共分两篇十六章,第一篇(前十章)介绍 Auto LISP 的基本函数,输入、输出函数,绘

图功能,分支结构,循环结构,用户函数的定义,字符串处理函数,图形数据库编辑函数和数据文件等以及它们在工程 CAD 系统开发中的应用;第二篇(后六章)介绍工程 CAD 系统的开发方法与技术,其中包括数据处理方法与技术,图形处理方法与技术,尺寸标注与工程标注方法与技术,菜单与对话框的应用,程序调试方法与出错处理,典型实用 CAD 系统分析,最后备有较详细的系统变量,Auto LISP 内部函数,Auto CAD 命令,用 Auto LISP 的 Command 函数调用 Auto CAD 命令的格式以及出错信息等六个附录。

本书的主要特点是:

1. 在内容组织上遵循了由浅入深、循序渐进的学习规律,全书深入浅出、通俗易懂,便于教学和读者自学;

2. 在介绍 Auto LISP 语言的基础上侧重于实用 CAD 系统开发技术的介绍。书中介绍的大量实用函数、典型程序和实用 CAD 系统开发实例,在二次开发中均可直接引用;

3. 本书介绍的 Auto LISP 函数均采用调用格式、功能、求值结果及注意事项的格式,十分醒目,便于查阅。在附录中全面而科学地规纳了 Auto CAD 的命令,Auto LISP 内部函数,用 Command 函数调用 Auto CAD 命令的格式等,查阅十分方便。本书是一本较好的 Auto LISP 语言的教科书,也是采用 Auto LISP 语言二次开发实用 CAD 系统的参考书。

本书由四川联合大学(原四川大学、成都科技大学)制造工程学院陈道洁教授和付守默教授编著,由四川联合大学博士生导师殷国富教授主审。

本书在编写过程中得到四川联合大学各级领导,从事 CAD 教学的老师的大力支持,研究生梁哲和文亚为本书附录的编排和参与界面技术的编写作了大量工作,特此表示感谢。

由于作者水平有限,时间仓促,错误难免,敬请读者批评指正。

编 者

一九九五年四月

目 录

第一篇 Auto LISP 程序设计语言

第一章 结论	1
1.1 Auto LISP 语言是专业 CAD 开发的重要工具	1
1.2 Auto LISP 的数据类型	2
1.2.1 原子	3
1.2.2 表和点对	4
1.3 Auto LISP 的程序结构	5
1.4 Auto LISP 的求值过程	5
1.5 Auto LISP 程序的编辑、装入和运行	6
1.5.1 Auto LISP 程序的编辑	6
1.5.2 Auto LISP 的装入和运行	7
第二章 Auto LISP 的基本函数	9
2.1 数值函数	9
2.1.1 算术运算函数	10
2.1.2 三角函数	14
2.1.3 数的类型转换函数	15
2.1.4 应用举例	15
2.2 赋值函数	17
2.2.1 赋值函数 SETQ	17
2.2.2 赋值函数 SET	18
2.3 求值与禁止求值函数	19
2.3.1 禁止求值函数 QUOTE	19
2.3.2 再求值函数 EVAL	20
2.4 表处理函数	20
2.4.1 取表部分内容函数	20
2.4.2 表的构造与修改函数	22
2.4.3 联接表的检索与修改	25
2.4.4 测量表的长度函数 LENGTH	25
2.5 综合举例	25
第三章 交互式输入函数和屏幕输出函数	27
3.1 交互式输入函数	27
3.1.1 GET 族输入函数	27
3.1.2 其它输入函数	35

3.2 屏幕输出函数	35
3.2.1 用于屏幕和文件的输出函数	36
3.2.2 只用于屏幕的输出函数	38
3.3 综合举例	38
第四章 Auto LISP 的绘图功能	42
4.1 COMMAMD 函数	42
4.2 图形处理函数	44
4.2.1 求相对角度的函数 ANGLE	44
4.2.2 求两点间距离的函数 DISTANCE	45
4.2.3 求极坐标函数 POLAR	45
4.2.4 求二直线交点坐标的函数 INTERS	46
4.2.5 目标捕捉函数 OSNAP	46
4.3 屏幕操作函数	48
4.3.1 文本图形屏幕转换函数 GRAPHSCR 和 TEXTSCR	48
4.3.2 重画屏幕图形函数 REDRAW	48
4.3.3 清图形屏幕函数 GRCLEAR	49
4.3.4 向屏幕文本显示区写文字的函数 GRTEXT	49
4.4 存取 Auto CAD 系统变量的函数	49
4.4.1 获得系统变量值的函数 GETVAR	49
4.4.2 设置系统变量值的函数 SETVAR	50
4.5 综合举例	50
第五章 分支结构	55
5.1 关系运算函数	55
5.2 逻辑运算函数	56
5.2.1 逻辑“与”函数 AND	56
5.2.2 逻辑“或”函数 OR	56
5.2.3 逻辑“非”函数 NOT	56
5.3 二分支条件函数 IF	57
5.4 多分支条件函数 COND	58
5.5 其它测试函数	60
5.5.1 判断数的性质的函数	60
5.5.2 判断数据类型的函数	61
5.5.3 等值函数	62
5.6 顺序控制函数 PROGN	63
5.7 综合举例	63
第六章 循环结构	66
6.1 WHILE 函数	66
6.2 REPEAT 函数	67
6.3 其它的循环函数	68
6.3.1 FOREACH 函数	68
6.3.2 MAPCAR 函数	69

6.3.3 APPLY 函数	70
6.4 综合举例	70
第七章 函数的定义	75
7.1 自定义函数	75
7.1.1 定义用户函数的函数 DEFUN	75
7.1.2 函数的调用	76
7.1.3 函数的副作用	76
7.1.4 使用 DEFUN 的注意事项	78
7.1.5 自定义函数在开发专业 CAD 软件中的应用	78
7.1.6 自定义函数中全局变量和局部变量的设置	80
7.2 增加和修改 Auto CAD 命令	82
7.2.1 增加 Auto CAD 命令	82
7.2.2 修改 Auto CAD 命令	83
7.2.3 恢复 Auto CAD 命令	83
7.3 综合举例	84
第八章 字符串处理函数	89
8.1 字符与 ASCII 码互换函数	89
8.1.1 字符转换成 ASCII 码函数 ASCII	89
8.1.2 ASCII 码转换成字符函数 CHR	89
8.2 测量字符串长度函数 STRLEN	90
8.3 数串互换函数	90
8.3.1 整数转换成整数串的函数 ITOA	90
8.3.2 数字串转换成整数函数 ATOI	90
8.3.3 数字串转换成实型数函数 ATOF	91
8.3.4 数转换成实数串的函数 RTOS	91
8.3.5 角度转换为字符串函数 ANGLOS	92
8.4 字符中的连接函数 STRCAT	93
8.5 字符串大小写转换函数 STRCASE	93
8.6 字符串截取函数 SUBSTR	93
8.7 字符串转换成表或原子的函数 READ	94
8.8 字符串处理函数在实用机械 CAD 中的应用	94
8.9 综合举例	95
第九章 图形数据库编辑函数及其应用	97
9.1 实体的基本概念	97
9.1.1 实体	97
9.1.2 实体名	98
9.1.3 选择集	98
9.2 选择集构造函数 SSGET	98
9.2.1 直接从图形屏幕上选定	98
9.2.2 SSGET 过滤器	99
9.2.3 应用 SSGET 函数的注意事项	100

9.3 选择集操作函数	100
9.3.1 测量选择集长度函数 SSLENGTH	100
9.3.2 实体名检索函数 SSNAME	101
9.3.3 向选择集中加入新实体函数 SSADD	101
9.3.4 从选择集中移出实体的函数 SSDEL	102
9.3.5 测试实体是否在选择集中函数 SSMEMB	102
9.4 实体名和选择集与 Auto CAD 一起使用	102
9.5 实体名操作函数	103
9.5.1 实体名搜索函数 ENTNEXT	103
9.5.2 获得最后一个主实体名函数 ENTLAST	104
9.5.3 选择任一实体函数 ENTSEL	104
9.6 实体名和选择集在开发机械 CAD 中的实际应用	105
9.7 实体数据函数	106
9.7.1 获得实体定义数据函数 ENTGET	106
9.7.2 修改实体在图形数据库中定义函数 ENTMOD	110
9.7.3 更新复杂实体屏幕图象函数 ENTUPD	111
9.7.4 删除或恢复实体定义函数 ENTDEL	112
9.8 实体数据函数在机械 CAD 中的应用	113
9.8.1 获得主实体名的途径	113
9.8.2 获得子实体名的方法	114
9.8.3 获得实体数据并对其进行修改的方法	114
9.9 综合举例	115
第十章 数据文件	119
10.1 打开文件的 OPEN 函数	119
10.2 关闭文件函数 CLOSE	120
10.3 输入、输出函数	121
10.3.1 输出函数	121
10.3.2 输入函数	123
10.4 数据文件在开发机械 CAD 系统中的应用	123
10.5 综合举例	125

第二篇 工程 CAD 系统的开发方法与技术

第十一章 数据处理方法与技术	127
11.1 第一种检索法—直接检索法	127
11.1.1 数据存贮	128
11.1.2 数据检索	128
11.2 第二种检索法—专门检索函数检索法	130
11.2.1 数表的分类	130
11.2.2 数表数据的存贮	131
11.2.3 数据检索函数	132

11.3 第三种检索法—数据文件存贮检索法	134
11.3.1 XXX·TXT 数据文件的格式	134
11.3.2 检索函数	135
11.4 综合举例	136
第十二章 图形处理方法与技术	141
12.1 绘图环境初始化	141
12.2 绘图坐标点的设置与计算	141
12.2.1 绘图基点设置	142
12.2.2 绘图坐标点的设置	142
12.2.3 绘图坐标点的计算	144
12.3 绘图参数值的输入与传输	148
12.3.1 绘图参数值的输入	148
12.3.2 绘图参数值的传输	148
12.4 绘图方法	152
12.4.1 单一变参图形程序设计法	152
12.4.2 用自定义函数建立子图变参图形库方法	154
12.4.3 用表建立子图变参图形库方法	156
12.5 图形修改方法	158
12.5.1 对图素的特性及线型的修改和字高的修改	158
12.5.2 图形的开窗消隐	159
12.5.3 图形修改	159
12.5.4 图形位置的修改	159
第十三章 图形标注方法与技术	162
13.1 图形尺寸与公差标注	162
13.1.1 图形标注初始化	162
13.1.2 尺寸标注函数的开发	162
13.1.3 图形尺寸标注方法	167
13.2 工程标注	170
13.2.1 工程标注函数的开发	170
13.2.2 工程标注方法	173
13.3 在图上书写文字的方法	175
13.3.1 文字串连续书写的函数	175
13.3.2 填写标题栏的编程方法	176
13.3.3 书写技术要求的编程方法	177
13.3.4 填写技术参数表的编程方法	178
第十四章 界面技术	179
14.1 菜单及其开发与应用	179
14.1.1 Auto CAD 菜单的描述	179
14.1.2 菜单的构造	179
14.2 对话框及其应用	186
14.2.1 对话框和 DCL 语言简介	186

14.2.2 对话框的构成	189
14.2.3 部件属性探讨	194
14.2.4 对话框控制语言	194
14.2.5 对话框驱动程序	195
14.2.6 对话框在开发冷冲模 CAD 系统中的应用	199
第十五章 程序的调试与出错处理	204
15.1 Auto LISP 程序调试的一般方法	204
15.1.1 Auto LISP 程序调试方法	204
15.1.2 设置断点打印变量值的方法	205
15.1.3 设置自定义函数分段调试法	206
15.1.4 单步执行调试法	207
15.2 Auto LISP 编程中常出错误分析	208
15.2.1 语法错误	208
15.2.2 逻辑错误	210
15.3 用户定义 ERROR 函数	211
第十六章 典型零件 CAD 系统的分析	213
16.1 系统的总体结构及流程	213
16.1.1 系统的功能与流程	213
16.1.2 系统模块划分	213
16.1.3 系统的主菜单	213
16.2 数据输入与修改	217
16.2.1 原始数据输入	217
16.2.2 数据修改	218
16.3 设计计算	218
16.3.1 设计计算	218
16.3.2 设计结果显示及打印	228
16.4 结构选型模块	230
16.5 图形绘制与标注	231
16.6 图形输出	231
附录 A Auto CAD R12.0 系统变量一览表	233
附录 B Auto LISP 系统函数一览表	244
附录 C Auto CAD 命令简表	252
附录 D 用 COMMAND 函数调用常用 Auto CAD 命令格式	265
附录 E 标准出错信息	271
附录 F ASCII(美国标准信息交换码)表	275
主要参考文献	278

第一篇 Auto LISP 程序设计语言

第一章 绪 论

1.1 Auto LISP 语言是专业 CAD 开发的重要工具

LISP (LIST Processing Language) 是一种计算机的表处理语言, 是迄今为止人工智能学科领域广泛应用的一种程序设计语言。而 Auto LISP 语言是一种嵌在 Auto CAD 内部的 LISP 编程语言, 它是 LISP 语言和 Auto CAD 的有机结合的产物。它采用了与 LISP 语言中的 Common LISP 最相近的语法和习惯约定, 同时它又针对 Auto CAD 增加了许多新的功能, 如增加了与图形处理有关的函数, 它可以直接调用几乎全部 Auto CAD 命令。因此它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功能。它是当今世界上计算机辅助设计和绘图最广泛采用的语言。

美国 Auto desk 公司在 Auto CAD 中嵌入 Auto LISP 编程语言的目的是为了使用户充分利用 Auto CAD 进行二次开发。利用 Auto LISP 可以直接增加和修改 Auto CAD 命令, 可以实现对当前图形数据库的直接访问和修改, 可随意扩大图形编辑修改功能, 并结合各国标准建立大量标准件、非标准件的图形库和数据库, 而且可以结合专业开发各种实用的 CAD 软件系统, 如我们用 Auto LISP 语言开发了“MECAD 机械零件 CAD 软件包”、“GTS 图形开发工具”、“模架图形库”、“冷冲模 CAD”等应用软件。总之 Auto LISP 语言是开发专业 CAD 的重要工具。

Auto LISP 开发 Auto CAD 的一个典型的也是最重要的应用就是实现参数化绘图程序设计。工程上要绘制一个几何图形, 必须要给出充分而必要的尺寸, 这些尺寸就是决定该几何图形形状和大小的绘图参数。参数化绘图程序就是根据这些可变参数编写出可生成相应图形的程序。

例如图 1.1 所示为一矩形, 它的可变绘图参数有两个, 即 L 和 W。这两个参数就是决定该矩形形状和尺寸的必要而充分的参数, 它们是变量, 只要确定了这两个可变参数的具体数值, 该图形即确定了。

以下就是根据这两个可变参数和基点编写的 Auto LISP 程序:

```
(defun BOX (bp L w / sp)
  (command "Pline"
    bp
    (setq sp (polar bp 0 L))
```

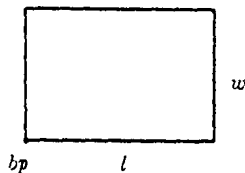


图1.1 参数绘图

```

(setq sp (polar sp (/ pi 2.0) w))
(setq sp (polar sp pi L))
"c"
)
)

```

分别赋予形参 bp , l , w 以具体数值时调用该函数,回车后即可在屏幕上的点“30,20”处绘出该图形,如图1.1所示,其调用命令为:

```
command: (BOX '(30 20) 28 16)
```

参数绘图程序设计又称变参图形程序设计,在开发实用CAD程序系统中应用极广,它是计算机辅助设计绘图最重要的方法。图1.2所示,即为程序设计自动生成的轴的零件工作图。

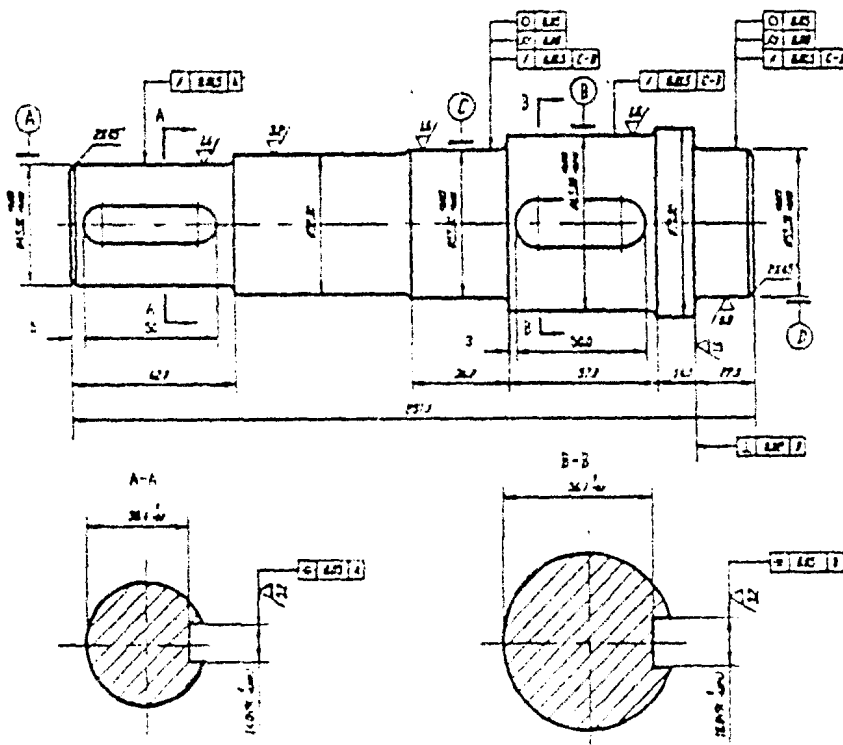


图 1.2 轴

1.2 Auto LISP 的数据类型

计算机是信息处理的自动机。信息在计算机中是用数据表示,不同语言其数据表示不同。Auto LISP 的数据类型有以下十种:

- 整型数 (INT)
- 实型数 (REAL)

- 符 号 (SYM)
- 字符串 (STR)
- 表 (及用户定义的函数) (LIST)
- 文件描述符 (FILE)
- Auto LISP 的内部函数 (SUBR)
- Auto CAD 的选择集 (PICKSET)
- Auto CAD 的实体名 (ENAME)
- 函数分页表 (PAGETB)

本章只介绍前五种数据类型,其它类型将在相应的章节中介绍。

这五种数据类型中前四种称为原子(ATOM),原子中包括数字原子(整型数和实型数)、符号原子和串原子。

所以 Auto LISP 最基本的数据是原子和表,它们又总称为符号表达式(Symbolic-expression)也称为 S-表达式。

1.2.1 原 子

1. 整 数

整数是由 0、1、2、…、9、+、- 等字符组成,在 PC-DOS/MS-DOS 系统中整数为 16 位带符号数,其范围是从 -32768~+32767,32 位机为 -2,147,483,648 到 + 2,147,483,647 之间,如整数超出此范围,计算机将提示出错信息。

2. 实型数

实型数是用双精度的浮点数表示,并且有至少 14 位精度,即整数后跟小数,如实数的数值小于 1,小数点前必须加 0,不能直接以小数点开头。否则被误认为点对而出错。实型数的范围比整型数大得多,如对于 16 位微机实型数范围约为 $-1.797693 \times 10^{308}$ 到 1.79793×10^{308} 。它不易超界,故尽量采用实型数。

实型数也可采用科学记数法表示,如 0.12×10^9 可表示为 0.12E9。

3. 符 号

符号(symbol)包括除“),“(,“ .”,“,”,””,“;”之外的任何打印字符。符号原子的长度没有限制,但尽量不超过 6 个,否则要占用额外的内存,降低运行速度。在 Auto LISP 中符号的大小写是等效的,如以下的符号原子都是合法的。

A A12 PC x-38-6 *A

在 Auto LISP 中任何符号都是有值的,即符号都要赋以一定的数值,或者说符号总是约束在一定值上。一般用赋值函数 setq 进行赋值,如

(setq x 25.0)

意思是将 25.0 赋给 x,这时 x 的当前约束值即为 25.0。一个符号在使用前如不赋以任何值,则该符号的值为 nil(空),它不占用内存空间。

要想查看一个符号的当前约束值,可在 command 提示符下键入(!)后紧跟要查看的符号,例如:

command: ! x

25.0

4. 字符串

字符串是由双引号引起来的字符组成的,如:

“ABC” “135” “Ab C” “”

字符串可以包括任何可打印的字符,字符串中字母的大小写及空格都是有意义的,字符串的最大长度为100个字符,超过100个后面的字符是无效的,若字符串中没有任何字符即为空串“”。

1.2.2 表和点对

表是指放在一对相匹配的左,右括号中的一个或多个元素的有序集合。表中的每一个元素可以是任何类型的S-表达式,既可以是数字原子、符号、字符串、也可以是表。元素与元素之间要用空格隔开,而元素与括弧之间可不用空格,因为括弧本身就是有效的分隔号。

例如:

(15 (a b) c d)

上例表中有四个元素,即15,(a b),c, d,其中第二个元素又是一个表。表是可以任意嵌套的,上例表中即嵌套了一个表(a b),表可以嵌套很多层,从外层向里依次称为0层(也称顶层)、1层、2层、…。我们所指的表中的元素是指表的顶层元素。

表中的元素是有顺序的,为便于对表中元素进行存取,每个元素都有一个序号。从左向右,第一个元素序号为0,第二个元素序号为1,第*i*个元素序号为*i*-1。

表的大小为表的长度,即表中顶层元素的个数。没有任何元素的表称为空表。空表用()或nil表示。在Auto LISP语言中nil是一个特殊的符号原子,它既是原子又是表。

表有两种基本类型:标准表和引用表。

1. 标准表是Auto LISP程序的基本结构型式,Auto LISP程序就是由标准表组成的,标准表是用于函数的调用,其中第一个元素必须是系统内部函数或用户定义的函数,其它的元素为该函数的参数,如上面提到的赋值函数的调用,即采用标准表的形式。

(setq x 25.0)

表中第一个元素setq为系统内部定义的赋值函数,x和25.0均为setq的参数。

2. 引用表 这种表第一个元素不是函数,即不作为函数调用,常作为数据处理。

引用表中一个重要应用是用来表示图中点的坐标,用来表示点的坐标时,是用实型数作为元素构成的表。

表示二维点的坐标是用两个实型数构成的表,如(20.0 30.5),第一个元素表示点的x坐标,第二个元素表示点的y坐标。表示三维点的坐标,是用三个实型数构成的表,如(20.0 82.5 1.0),其中三个元素依次表示x坐标、y坐标和z坐标。

3. 点对(dotted pair)也是一种表,表中有二个元素,二元素中间为一圆点“.”,圆点与元素之间必须用空格分开。这点要十分注意,否则容易出错。

例如:

(A . B)

就是一个点对,A和B与圆点均用空格分开,其中第一个S-表达式A为该点对的左元素

第二个S—表达式B为点对的右元素。点对亦可任意嵌套。一定要注意点对的书写格式,例如:

`(X . (B . (Y . Z)))`

为合法点对,

而 `(X . (B . (Y·Z)))`

即为非法的。

点对常用于构造联结表。

1.3 Auto LISP 的程序结构

Auto LISP 语言没有“语句”术语,相当于其它高级语言的语句,一律采用标准表的形式,Auto LISP 程序往往是由一个或一系列按顺序排列的标准表所组成。例如:

`(setq x 25.0)`

是上面提到的标准表,又可以看作是一个 Auto LISP 程序。

又如:文件名为 `pq.lsp` 的 Auto LISP 文件是由以下程序组成的:

`(setq x 25.0)`

`(setq y 12.2)`

`(+ (* x y) x)` ;x+xy

以上是由三个标准表组成的程序,每个标准表的第一元素如 `setq`, `+`, `*` 均为系统提供的函数,称为系统内部函数。`setq` 为赋值函数, `+` 为加函数, `*` 为乘积函数,其它的元素为相应函数的参数。这个程序是将 25.0 赋给 `x`, 将 12.2 赋给 `y`, 求变量 `x` 和 `y` 的值的乘积,再求此乘积与 `x` 的总和。

Auto LISP 语言的书写格式:

1. 函数必须放在表中第一元素的位置。算术运算函数 `+`, `*` 等亦放在表中第一元素位置,即放在操作数之前,而不是放在它们的中间,这与算术运算的书写格式不同,初学者可能会感到不习惯。

2. 在 Auto LISP 程序中,一行可写几个甚至多个标准表,并由回车结束此行。如:

`pq.lsp` 的程序为:

`(setq x 25.0) (setq y 12.2) (+ (* x y) x)` 回车换行。

3. 一个标准表如果很长(如多次嵌套),一行写不下,也可分几行书写。

4. Auto LISP 程序中可以使用注释。注释以一个分号(`;`)开始,并在一行末尾结束,如 `pq.lsp` 程序中;`x+xy` 即为注释。注释的作用是对程序作解释,Auto LISP 求值器总是忽略分号和在其同一行后面的任何符号。注释可放在程序中的任何地方。

5. Auto LISP 程序一般是以扩展名为“`.LSP`”的 ASC II 码文本文件的形式存储。

1.4 Auto LISP 的求值过程

如上所述,Auto LISP 程序是由一系列按顺序排列的标准表所组成的,运行 Auto LISP 程序就是依次运行每个标准表。由于 Auto LISP 语言是一种求值语言。Auto LISP 的核心是

求值,求值器将读入每行程序并对其进行计算、求值并返回求值结果,其具体求值过程如下:

1. 对表中第一项元素求值结果为函数名。表中其余元素作为该函数参数,先对各参数进行求值,再将各参数的值传给该函数进行运行,运行结束时即返回该函数的值,例如:

command: (* 2 3) 返回值 6

求值器先对上面表的第一个元素(符号“*”)求值,求值结果为 Auto LISP 的内部求积函数的名字。函数“*”的代码告诉求值器取表的其余参数分别求值,数2、3求值结果仍为数本身,再将其值传给函数运行,然后返回求积结果(6)。

2. 对表中各参数的求值,其过程因参数数据类型不同而分为以下几种情况:

(1) 若参数为数(整型数、实型数)、字符串及两个特殊原子 T 和 nil,对它们的求值结果即为该参数本身。

(2) 若参数为符号原子,则以它们的当前约束值作为求值结果。

(3) 若参数为引用表,则必须打上禁止求值的单引号,此时求值结果为引用表本身,例如:

(setq f '(/ 4.0 2)) 返回值 (/ 4.0 2)。

3. 若标准表为多层嵌套,其求值过程总是从最里层的表开始,并依次向外层求值,最后返回顶层表的求值结果,例如:

(- (* 3 2) 1) 返回值 5 ;返回顶层元素求值结果。

1.5 Auto LISP 程序的编辑、装入和运行

1.5.1 Auto LISP 程序的编辑

Auto LISP 程序是以 ASCII 码文本文件的格式来编辑的,所以,任何一种 ASCII 文本编辑器均可用来编辑 Auto LISP 的源程序。最早采用的文本编辑器是 DOS 提供的 EDLIN 行编辑程序,用 EDLIN 编辑一个文本文件的过程如下:

C>EDLIN <文件名>

New file

* I

1 * (defun f (x y)

2 * (+ (* x y) 2.0)

3 *)

4 * 按 CTRL-C 键

* E

Auto LISP 程序可以驻留在任何驱动器的任何目录之下,因此,一个完整的 Auto LISP 文件描述符应包括:磁盘驱动器名、目录名、文件名和扩展名。如驻留在 A 盘子目录 cl 下的 input 文件为 A:\cl\input.lsp。Auto LISP 文件均必须带有固定的扩展名(.lsp)。

关于 EDLIN 有关的命令可参阅 DOS 使用手册。

用 EDLIN 行编辑程序修改不方便。目前市场上已有很多全屏编辑软件可供选用,如 QE, TB 等。