

同济大学教材、学术著作出版基金委员会资助

测量工程 LISP 程序设计

程效军 顾孝烈 顾振雄 编著

LISP PROGRAMMING
FOR SURVEYING
ENGINEERING



同济大学出版社
TONGJI UNIVERSITY PRESS

测量工程 LISP 程序设计

程效军 顾孝烈 顾振雄 编著



同济大学出版社
TONGJI UNIVERSITY PRESS

内 容 提 要

本书共六章,内容为针对测绘工程利用 LISP 语言编程对 AutoCAD 二次开发,进行计算和绘图。第一章对 LISP 语言作简明扼要的介绍;第二章介绍测绘工程的计算和绘图中常用的 LISP 函数;第三章介绍 LISP 程序设计中的各种流程控制;第四章介绍应用 DCL 语言编制对话框;第五章为地形测量 LISP 程序设计,内容有地形图图式、线型、线形符号、地物符号和面填充符号的自定义函数设计、地形图展点连线、等高线绘制、建筑三维图测绘等的程序设计;第六章介绍工程测量中若干 LISP 程序设计,包括空间前方交会、平面平整度测定、球面多点求球心、抛物面方程回归计算、缓和曲线测设数据计算、光电测距仪参数测定等程序设计。

本书可作为高等院校测绘工程、地理信息系统、土木工程等专业的本科生及研究生的教材和工程勘察单位工程技术人员的参考书。

图书在版编目(CIP)数据

测量工程 LISP 程序设计/程效军,顾孝烈,顾振雄

编著. —上海:同济大学出版社,2012.3

ISBN 978-7-5608-4744-3

I. ①工… II. ①程…②顾…③顾… III. ①工程
测量—计算机辅助设计—应用软件,AutoLISP—程序
设计 IV. ①TB22-39

中国版本图书馆 CIP 数据核字(2011)第 271063 号

测量工程 LISP 程序设计

程效军 顾孝烈 顾振雄 编著

责任编辑 杨宁霞 责任校对 徐春莲 封面设计 陈益平

出版发行 同济大学出版社 www.tongjipress.com.cn

(地址:上海市四平路 1239 号 邮编:200092 电话:021-65985622)

经 销 全国各地新华书店

印 刷 同济大学印刷厂

开 本 787mm×960mm 1/16

印 张 18.25

印 数 1—3 100

字 数 362 000

版 次 2012 年 3 月第 1 版 2012 年 3 月第 1 次印刷

书 号 ISBN 978-7-5608-4744-3

定 价 39.00 元

本书若有印装质量问题,请向本社发行部调换 版权所有 侵权必究

前 言

AutoCAD 是美国 AutoDesk 公司推出的通用绘图软件系统。随着计算机技术的飞速发展,AutoCAD 绘图软件的使用也愈益广泛,其丰富的绘图功能和良好的用户界面,在测绘工程领域中已得到了普遍的应用,特别是该软件提供的编程工具和接口技术,为用户开发应用该软件系统创造了十分有利的条件。

AutoCAD 提供的开发工具主要有:VL(Visual LISP)、VBA(Visual BASIC Application)和 Object ARX(AutoCAD Runtime Extension),本书介绍 Visual LISP 在测量工程的数字化成图和工程测量中计算和绘图的程序设计和应用。

本书第一章对 LISP 语言作简明扼要的介绍;由于 LISP 函数十分丰富,为了专业化的学习,第二章将测量工程的计算和绘图中常用的 LISP 函数按其功能作分类介绍(省略其不常用部分),并以测量工程的内容作为函数应用实例,便于快速查阅和运用这些函数;第三章介绍 LISP 程序设计中的各种流程控制,且均以测量工程的计算内容作为应用举例,例如法方程式的解算等;第四章对应用 DCL(Dialog Control Language)语言编制对话框作扼要的介绍,作为程序中数据输入的工具;第五章为地形测量 LISP 程序设计,内容有地形图图式设计、线型设计、线形符号、地物符号和面填充符号的自定义函数设计、地形图展点连线程序设计、等高线绘制程序设计、建筑三维图测绘程序设计等;第六章介绍工程测量中若干内容的 LISP 程序设计,包括空间前方交会、平面平整度测定、空间平面多点求圆心、球面多点求球心、抛物面方程回归计算、缓和曲线测设数据计算、光电测距仪参数测定回归计算、H 变换解方程式以及平面控制网误差椭圆绘制等程序设计。

本书主要内容为利用 Visual LISP 语言对 AutoCAD 的二次开发功能用于测绘工程的程序设计,为作者历年来在教学、科研和生产的实践中研制、积累、整理与归纳而成。所编程序对计算与绘图并重,内容新颖,方法具体,说明详尽,便于阅读,举例实用,利于自学入门和深造提高。所介绍程序严密完整,切实可靠,加载于 AutoCAD 2004 及以后版本均可运行。一些程序附有详细数例,可进行演示和验证。

由于计算科技发展迅速,尤其是“计算机程序的优化是无止境的”,而作者的学识水平和实践经验有限,书中内容难免有不恰当之处,尚祈读者批评指正。

编著者

2012 年 1 月

目 录

前言

1 LISP 语言概述	(1)
1.1 LISP 语言的简单介绍	(1)
1.2 LISP 语言的数据类型	(4)
1.2.1 整型	(4)
1.2.2 实型	(4)
1.2.3 字符串	(4)
1.2.4 表	(4)
1.2.5 函数	(4)
1.2.6 图元名	(5)
1.2.7 选择集	(5)
1.3 LISP 语言的变量	(5)
1.3.1 变量的标识符	(5)
1.3.2 变量的数据类型	(5)
1.3.3 预定义标识符	(5)
1.3.4 显示变量的值	(6)
1.4 LISP 语言的表达式	(6)
1.4.1 表达式的构成	(6)
1.4.2 表达式的求值规则	(6)
2 LISP 函数	(8)
2.1 数值计算函数	(8)
2.1.1 和函数	(8)
2.1.2 差函数	(8)
2.1.3 乘函数	(9)
2.1.4 除函数	(9)
2.1.5 加一函数	(9)
2.1.6 减一函数	(9)
2.1.7 求余函数	(9)
2.1.8 取整函数	(9)
2.1.9 绝对值函数	(9)

2.1.10	幂函数	(9)
2.1.11	e 幂函数	(9)
2.1.12	平方根函数	(9)
2.1.13	正弦函数	(9)
2.1.14	余弦函数	(9)
2.1.15	反正切函数	(9)
2.1.16	对数函数	(10)
2.1.17	最大值函数	(10)
2.1.18	最小值函数	(10)
2.1.19	转实型数函数	(11)
2.2	赋值函数	(11)
2.3	交互输入函数	(12)
2.3.1	整数输入	(12)
2.3.2	实数输入	(12)
2.3.3	字符输入	(12)
2.3.4	点输入	(12)
2.3.5	文件选取	(12)
2.4	几何运算函数	(13)
2.4.1	方位角运算	(13)
2.4.2	距离运算	(13)
2.4.3	极坐标法运算	(13)
2.4.4	交点运算	(14)
2.5	关系运算函数	(14)
2.5.1	等于	(14)
2.5.2	不等于	(14)
2.5.3	小于	(15)
2.5.4	小于等于	(15)
2.5.5	大于	(15)
2.5.6	大于等于	(15)
2.5.7	相等	(16)
2.6	逻辑运算函数	(16)
2.6.1	与函数	(16)
2.6.2	或函数	(16)
2.6.3	非函数	(16)
2.7	数据类型转换函数	(17)
2.7.1	角度转换弧度	(17)
2.7.2	弧度转换角度	(17)

2.7.3	字符串转换实数	(18)
2.7.4	字符串转换整数	(18)
2.7.5	整数转换实数	(18)
2.7.6	整数转换字符串	(18)
2.7.7	实数转换字符串	(18)
2.8	字符串处理函数	(18)
2.8.1	合并字符串	(19)
2.8.2	读取字符串	(19)
2.8.3	读取子字符串	(19)
2.9	表处理函数	(19)
2.9.1	列表	(19)
2.9.2	并表	(20)
2.9.3	构造表	(20)
2.9.4	倒表	(20)
2.9.5	求表长	(20)
2.9.6	取表首	(20)
2.9.7	取表余	(21)
2.9.8	取表中任意元素	(21)
2.9.9	取表尾	(21)
2.9.10	换表元	(21)
2.9.11	表元循环处理	(22)
2.10	流程控制函数	(22)
2.10.1	条件判断	(22)
2.10.2	持续	(23)
2.10.3	多条件判断	(23)
2.10.4	定量循环	(24)
2.10.5	条件循环	(24)
2.11	文件操作函数	(24)
2.11.1	文件加载	(24)
2.11.2	文件打开和操作	(25)
2.11.3	文件读取	(25)
2.11.4	文件写入	(26)
2.11.5	文件关闭	(26)
2.12	输出函数	(26)
2.13	图元操作函数	(27)
2.13.1	依次命名图元	(27)
2.13.2	最后图元命名	(27)

2.13.3	获取图元表	(28)
2.13.4	建立图元表	(28)
2.14	选择集函数	(29)
2.14.1	选择集创建	(29)
2.14.2	选择集添加	(29)
2.14.3	选择集删减	(30)
2.14.4	选择集返回	(30)
2.14.5	选择集长度	(30)
2.15	CAD 命令函数	(30)
2.15.1	几何作图命令	(30)
2.15.2	图形修饰命令	(38)
2.15.3	图案填充命令	(46)
2.15.4	图层命令	(46)
2.15.5	剪辑板命令	(48)
2.15.6	视图命令	(48)
2.15.7	坐标变换命令	(49)
2.15.8	系统变量命令	(50)
2.16	自定义函数	(51)
2.16.1	自定义函数概述	(51)
2.16.2	自定义函数实例	(51)
3	LISP 流程控制	(60)
3.1	条件流程控制	(60)
3.1.1	条件分支控制流程	(60)
3.1.2	多条件分支控制流程	(61)
3.1.3	条件表达式中的持续函数	(62)
3.2	循环流程控制	(63)
3.2.1	定量循环	(63)
3.2.2	条件循环	(63)
3.2.3	多重循环	(64)
3.3	子程序流程控制	(73)
4	LISP 程序的对话框设计	(75)
4.1	对话框概述	(75)
4.2	对话框的控件	(76)
4.2.1	基本控件	(76)
4.2.2	集控件	(76)

4.2.3	单选集控	(77)
4.2.4	确定取消组件	(77)
4.3	对话框控制语言	(77)
4.3.1	控件的定义	(77)
4.3.2	控件的引用	(78)
4.3.3	控件的属性	(78)
4.3.4	对话框设计	(79)
4.4	对话框操作	(81)
4.4.1	对话框操作函数	(81)
4.4.2	对话框驱动程序	(84)
5	地形测量 LISP 程序设计	(88)
5.1	地形测量数字成图与地形点编码	(88)
5.1.1	数字成图中 CAD 的应用与二次开发	(88)
5.1.2	地形点编码	(88)
5.2	地形绘图图式符号库设计	(92)
5.2.1	图式符号库的设计	(92)
5.2.2	图式符号库的建立与应用	(93)
5.3	线型设计	(96)
5.3.1	简单线型的定义	(96)
5.3.2	复杂线型的定义	(97)
5.3.3	地形绘图线型文件的建立与应用	(98)
5.4	线形符号绘制	(101)
5.4.1	直线围墙符号绘制	(102)
5.4.2	圆弧形围墙绘制	(103)
5.4.3	直线铁路绘制	(105)
5.4.4	圆弧形铁路绘制	(106)
5.4.5	直线路堤绘制	(108)
5.4.6	圆弧形路堤绘制	(109)
5.5	矩形房屋及台阶符号绘制	(110)
5.5.1	矩形房屋符号绘制	(111)
5.5.2	台阶符号绘制	(112)
5.6	自定义函数的调用	(114)
5.6.1	临时加载调用自定义函数	(114)
5.6.2	编制菜单调用自定义函数	(115)
5.7	地形绘图面填充符号绘制	(117)
5.7.1	地形绘图面填充符号设计	(117)

5.7.2	地形绘图面填充符号库的建立与应用	(121)
5.8	地形点展点与连线	(122)
5.8.1	地形点展点与连线程序设计	(122)
5.8.2	地形点展点与连线程序应用	(124)
5.9	等高线地形图绘制	(126)
5.9.1	等高线地形图自动绘制原理	(126)
5.9.2	等高线地形图自动绘制程序	(130)
5.9.3	等高线地形图自动绘制程序应用	(142)
5.10	建筑三维图测绘的 LISP 程序设计	(144)
5.10.1	应用于城市规划的城市地形三维模型	(144)
5.10.2	三维建筑图测绘	(145)
5.10.3	建筑轴线拟合和建立建筑坐标系	(150)
5.10.4	建筑物实体的构建	(152)
5.10.5	建筑三维模型的建立	(176)
6	工程测量 LISP 测序设计	(180)
6.1	空间前方交会计算程序设计	(180)
6.1.1	空间前方交会原理	(180)
6.1.2	空间前方交会计算公式	(180)
6.1.3	空间前方交会计算程序设计	(182)
6.1.4	空间前方交会算例	(185)
6.2	空间平面平整度测定计算程序设计	(187)
6.2.1	平面方程式参数和平整度	(187)
6.2.2	平面平整度测定计算程序	(189)
6.2.3	平面平整度测定算例	(192)
6.3	测定空间圆周上多点计算圆心坐标程序设计	(193)
6.3.1	平面上按圆周点计算圆心点坐标公式	(193)
6.3.2	空间平面法向量计算公式	(194)
6.3.3	空间坐标变换计算公式	(196)
6.3.4	观测平面圆周上多点拟合圆心坐标计算公式	(196)
6.3.5	测定空间圆周上多点计算圆心坐标程序设计	(197)
6.3.6	测定空间圆周上多点计算圆心坐标算例	(203)
6.4	测定球面上多点计算球心坐标程序设计	(204)
6.4.1	测定球面上四点坐标计算球心坐标和球半径的公式	(204)
6.4.2	测定球面上多点坐标平差计算球心坐标和球半径的公式	(206)
6.4.3	测定球面上多点平差计算球心坐标程序设计	(208)
6.4.4	测定球面上多点计算球心坐标算例	(214)

6.5 工程结构物的抛物面方程式回归计算程序设计	(215)
6.5.1 工程结构物抛物面方程式回归计算概述	(215)
6.5.2 抛物面方程式回归计算公式	(216)
6.5.3 抛物面方程式回归计算程序设计	(217)
6.5.4 抛物面方程式回归计算算例	(225)
6.6 缓和曲线测设数据计算程序设计	(228)
6.6.1 缓和曲线元素和主点坐标计算公式	(228)
6.6.2 缓和曲线和圆曲线细部点坐标计算公式	(231)
6.6.3 缓和曲线测设数据计算程序设计	(231)
6.6.4 缓和曲线测设数据计算算例	(241)
6.7 光电测距仪周期误差测定计算程序设计	(243)
6.7.1 光电测距仪周期误差测定计算公式	(243)
6.7.2 光电测距仪周期误差测定平差计算程序设计	(246)
6.7.3 光电测距仪周期误差测定算例	(250)
6.8 光电测距仪加常数和乘常数测定计算程序设计	(251)
6.8.1 光电测距仪加常数和乘常数测定计算公式	(251)
6.8.2 光电测距仪加常数和乘常数测定计算程序设计	(253)
6.8.3 光电测距仪加常数和乘常数测定算例	(256)
6.9 用 H 变换解观测值方程式程序设计	(258)
6.9.1 H 变换的理论公式	(258)
6.9.2 H 变换的实用公式	(259)
6.9.3 用 H 变换解观测值方程式程序	(260)
6.9.4 用 H 变换解观测值方程式算例	(267)
6.10 根据数据文件绘制平面控制网及点位误差椭圆	(270)
6.10.1 绘制平面控制网图和误差椭圆的数据文件及绘图方法	(270)
6.10.2 根据数据文件绘制平面控制网及误差椭圆程序	(270)
6.10.3 绘制平面控制网和误差椭圆实例	(274)
参考文献	(277)

1 LISP 语言概述

1.1 LISP 语言的简单介绍

LISP(List Processing Language)是一种表处理方式的程序设计语言,由于处理的对象是符号表达式(Symbolic Expression),因此也称为符号式语言(Symbolic Language)。LISP 于 20 世纪 50 年代由 John McCarthy 创立,在几十年的发展过程中产生许多版本,广泛用于人工智能、机器人、专家系统等领域,其中 Common LISP 是近年由美国的几所大学和工业界的研究人员共同研发且比较完善的版本。

美国 AutoDesk 公司在 20 世纪 80 年代采用与 Common LISP 最相近的语法,且针对 AutoCAD 增加许多功能创立 AutoLISP 语言,使用户能对 AutoCAD 进行二次开发,便于工程计算与绘图有机结合,扩大数据的表处理功能和图形的编辑功能。AutoLISP 是一种以解释方式运行于 AutoCAD 内部的程序设计语言,是一个通用的计算机辅助设计的绘图系统软件,具有完备的数学运算功能和调用 CAD 的绘图功能,可以使设计、计算和绘图融为一体,因此广泛用于机械、建筑、土木工程、测绘工程等领域。对于测绘工程而言,多用于测量作业中的数字地形测量的机助成图以及工程测量中的计算和图形绘制等。

AutoLISP 语言是一种比较特殊的“表结构语言”,又称为“表处理语言”(List Processing Language),LISP 代表“表处理语言程序编制”(List Processing Language Programming)。LISP 语言的基本形式为括号中包含若干个元素,形同一个“表”。LISP 表有两种标准表,即“函数表”和“引用表”(数据表),其中绝大部分为函数表。其内容为:

(函数 [参数 1] [参数 2] ...)

一对开括号和闭括号组成一个“表”,开头必须为一个“函数”,然后为若干个“参数”,总称为“元素”。元素与元素之间(函数与参数之间、参数与参数之间)至少用一个空格分开,表中参数的有无或数量,由函数的性质规定。例如,一个最简单的标准表如下,在括号中包含一个余弦函数(cos)和一个参数(π),两者之间有一个空格,组成一个求角度的余弦值的表达式:

(cos PI) — 返回值为“-1”

这种表相当于一个求值表达式 $\cos(\pi)$,是 LISP 程序的基本表达形式。表中的参数也可以是另一个表,称为表的“嵌套结构”。例如,正弦函数(sin)表中的参数为一个除函数(/)的表:

(sin (/ PI 2)) — 返回值为“1”

除函数是里层的表(/ PI 2),PI 为被除数,2 为除数;正弦函数(sin ...)是外层的

表。表达式运算的先后顺序是先对里层的表求值,得到“ $\pi/2$ ”;再对外层的表求值,得到“1”。

AutoLISP 提供了大量运算功能全面的函数(称为“内部函数”),供用户编程使用。AutoLISP 的“函数”扩大了数学中函数的内涵,除了可以作数学运算的函数以外,举凡文件操作、数值类型转换、CAD 绘图命令等均可作为“函数”。因此,学习 AutoLISP 首要的是了解各种 LISP 函数的功能和使用方法。

编制 AutoLISP 程序实际上是对函数的调用,或者说函数是 AutoLISP 语言处理数据的基本工具。因此,AutoLISP 程序是由一个或多个依次排列或多层嵌套的表组合而成。执行 AutoLISP 程序就是调用一些函数,这些函数可以再调用其他一些函数,也就是在对各个函数求值过程中实现函数的功能,进而实现程序的计算和绘图功能。

自行开发的 AutoLISP 程序也可以认为是 AutoCAD 软件中增加的“自定义函数”。而自定义函数也可以调用其他的自定义函数,被调用的自定义函数即相当于一个子程序,子程序之间又可以互相调用。因此 AutoLISP 程序也可以说是内部函数和自定义函数的集合。

在 AutoLISP 程序中,又可以用 AutoCAD 的绘图命令函数(command ...)调用 AutoCAD 几乎所有的绘图功能,即两者具有良好的交互性(interaction)。例如,在地形图机助成图中,既可直接利用 AutoCAD 绘图命令,绘制具有固定形式的地形图图式,建立图式符号库,连接地形点之间的线条,等等;也可用 AutoLISP 语言编制的专用程序来完成计算和绘图任务,例如,原始观测数据向绘图数据的转换,地形点展绘与初步连线,各种地物图形和等高线的绘制等。在工程测量中,某些需要提供计算与图形成果的项目,利用 AutoLISP 完整的计算与绘图相结合的功能,可以获得理想的效果。

以下介绍一个简单的自定义函数,其功能为在 CAD 屏幕上捕捉两个平面点位 P_1, P_2 , 求这两点间的边长 D 和方位角 A (角度以弧度表示),并在两点间画一条直线:

```
(defun c:Pol( ) ;自定义函数(defun)名“Pol”,作为 CAD 添加一个命令(c:)
  (setq P1 (getpoint "\n P1:")) P2 (getpoint "\n P2:"))
  ; 在屏幕用光标捕获 P1,P2 点
  (setq D (distance P1 P2)) ; 调用距离函数(distance)计算两点间平距
  (setq A (angle P1 P2)) ; 调用角度函数(angle)计算两点间方位角
  (print "Dist=") (princ D) ; 命令行显示平距
  (print "Azim=") (princ A) ; 命令行显示方位角
  (command "line" P1 P2 " ")
  ; 调用 CAD 命令中绘直线函数,两点间用直线连接
  (princ) ; 退出程序
) ; 结束自定义函数
```

以上 LISP 程序中“;”后为注释,目的是为了了解程序,与程序的运行无关。以上程序可以在 CAD 屏幕的命令行中输入并试运行,输入时不应输入注释,程序中的各个表不要换行,仅需保留必要的空格,最后以“回车”结束。如果输入无误,则命令行显示程序名“pol”。运行程序前,先在 CAD 屏幕画若干个二维点位,然后在命令行键入 pol 命令,程序即提示捕获点位 P1,P2,分别捕获后,命令行即显示距离(Dist=)和方位角(Azim=)的数值,屏幕上可见 P1,P2 两点间的连线。

LISP 表的另一种形式为纯粹的数表,称为“引用表”(数据表)。即放在一对圆括号中的元素的有序集合,且表中的元素没有“函数”而只有“参数”,其形式为:

([参数 1] [参数 2]...)

引用表一般作为储存数据的数组。在测绘工程的计算与绘图中,引用表的一个重要应用为代表计算或绘图中以坐标为参数的“点位”。例如,代表一个二维点位坐标或三维点位坐标的引用表为:

$(x_i \ y_i)$ 或 $(x_i \ y_i \ z_i)$

表中各个元素为有序集合(相当于数组),即每个元素都有一个序号,从左向右第 1 个元素序号为 0,第 2 个元素序号为 1,第 i 个元素序号为 $i-1$ 。

下列表的嵌套结构代表一组三维点位的坐标(相当于一个二维数组):

$((x_1 \ y_1 \ z_1) (x_2 \ y_2 \ z_2) \dots)$

由于数组对测量作业的计算与绘图有特别重要的作用,因此引用表在测绘工程中的用处很多。AutoLISP 提供一系列“表处理函数”,就是对引用表的建立、扩充、修改、矩阵运算等进行操作。

AutoLISP 语言在测绘工程中典型的应用之一是参数化绘图的程序设计。例如,在大比例尺地形绘图作业中要画一条铁路,按照《图式》的规定是相当形象化的:两条按轨距的平行线,沿平行线在垂直方向每隔 10mm 画一象征枕木的粗线条符号;利用 AutoLISP 程序绘图时的必要参数是轨距和直线段的两个端点点位,或圆弧曲线段的起点、中间点和终点点位;按规定的符号和必要的参数,再调用 CAD 绘图命令函数,就可以编制一个画铁路符号的自定义函数。

Visual LISP 是 Autodesk 公司于 1997 年推出的为加速 AutoLISP 程序开发而设计的软件开发工具,是新一代的 AutoLISP 语言,兼容以前的 AutoLISP 语言并扩展其功能。其新功能主要有:

(1) 完备的语法检查功能,能识别语法错误和函数的非法参数输入;程序中以字符的颜色区分表中元素的类别,例如,对于键入一个内部函数,自动显示为蓝色字符;如果其中有一个字符输错了,则函数的字符自动显示为黑色(自定义函数才是黑色),作为警示。

(2) 有符号名查找和自动匹配功能,例如,变量的标识符能自动地前后统一。

(3) 有功能完备的源程序调试器;预报编程中的语法错误,例如,有非法参数(不是该函数所规定的参数)、缺少参数、括号不配对等均能明确指出。

(4) 有语言格式化器,例如颜色区分、规范程序的结构等,增强了程序的可读性。

- (5) 有综合检验器和监视跟踪功能。
- (6) 有完整的文件编译器,改善程序的运行速度和安全性。

1.2 LISP 语言的数据类型

AutoLISP 的数据类型,除了一般程序设计语言的整型、实型、逻辑型、字符串等数据类型以外,还有表、函数、图元名、选择集等数据类型。

1.2.1 整型

整型数是 32 位带符号的整数,正号(+)可省略。

1.2.2 实型

实型数是带有小数点的数值,小数点前的零(0)不能省略。例如 0.618 不能写成 .618。实型数用双精度表示,并且至少有 14 位的精度。

1.2.3 字符串

字符串又称字符常数,是双引号内的字符序列,大小写字母、数字、符号、空格都有意义,例如,“BASIC”、“输入文件名”等。字符串可以为空串(“”),即不包括任何内容。

1.2.4 表

表由一对括号中包含若干元素组成,各元素之间至少置一空格,元素的排列是有序的(即不可随意调换其次序)。最外层的表中元素的个数,称为表的长度。元素的个数不限,也可以没有元素(即空表)。元素也可以是另一个表。例如:

(+ 12 13)、(setq x 0.36)——表中各有 3 个元素,其长度为 3;

(sin (/ PI 2))——表中有 2 个元素,长度为 2,第 2 个元素为另一个表;

(1.2 2.4)——纯数字的表称为数表,长度为 2,可代表一个 $x=1.2$ 、 $y=2.4$ 的二维点;

(3.6 4.8 5.2)——长度为 3,可代表一个 $x=3.6$ 、 $y=4.8$ 、 $z=5.2$ 的三维点;

((1.2 2.4) (0.3 5.1) ...)——表中元素是另一个表,代表一个包含若干二维点位的数组。

LISP 程序的一切数据输入、输出、操作、运算、绘图均通过表及其函数来完成。

1.2.5 函数

函数是 AutoLISP 语言中应用最广泛的操作运算代码,一般列为表中第一个元素。例如四则运算符(+, -, *, /)、三角函数(sin, cos)、关系符(<, >, >=)、流程控制关键字(if, while, repeat)、绘图命令(command)等,可以进行操作和运算的都称

为函数。由 AutoLISP 语言规定其功能的函数称为内部函数(或称为关键字、保留字),由用户自行编制的函数称为自定义函数(defun,即 definite function),或称为子程序或程序。自定义函数的命名应避开内部函数名。

1.2.6 图元名

图元名(entity name)是 AutoCAD 针对图形对象指定的十六进制的数字标识符。AutoLISP 通过标识符可在图形数据库中找到该图形对象(图元),以便进行访问或编辑。用户对图形对象也可另行命名自己指定的标识符,便于自己识别。

1.2.7 选择集

选择集是多个图形对象命名的集合,通过 AutoLISP 建立选择集或向指定的选择集添加或移去图形对象,还可以对其内部指定的成员进行访问或编辑。

1.3 LISP 语言的变量

1.3.1 变量的标识符

标识符(SYMBOL)作为变量或函数的命名,一般由英文字母(大、小写等价)、数字或中间加下划号(_)等字符组成,字符个数(标识符长度)不受限制,但长度大于 6 个字符时会增加储存单元。标识符的命名应避免与内部函数的名称相同,否则会改变内部函数的性质。

1.3.2 变量的数据类型

AutoLISP 语言不需要在变量赋值前对其进行类型(TYPE)说明(而多数计算机语言,这都是必需的),变量被赋予的值的类型即为变量的类型。同一变量在程序不同的运行阶段可以赋予不同类型的值。用 type 函数可以了解变量的类型。

1.3.3 预定义标识符

AutoLISP 对下列 4 个标识符 nil, T, Pause, PI 进行预定义,在编写 LISP 程序时可直接使用,但自定义函数名不应与之同名。

(1) nil——如果变量未赋值,其值为 nil。如果将某变量赋值为 nil,则表示取消该变量原有的值,并释放储存空间。如果逻辑变量的值为 nil,表示不成立,相当于其他语言的 false。

(2) T——作为逻辑变量的值, T 表示成立,相当于其他语言的 true。

(3) Pause——Pause 与绘图命令函数 command 配合使用,使程序暂停等候用户的输入。

(4) PI——定义为常量 π (圆周率)。

1.3.4 显示变量的值

当 AutoLISP 程序运行后,在 AutoCAD 屏幕的命令提示区,键入“!”及已赋值变量的标识符,可显示变量的值。如果显示 nil,则表示该变量尚未赋值。这在检查程序的运行情况(例如检查发生错误之处)时很有帮助。

1.4 LISP 语言的表达式

1.4.1 表达式的构成

AutoLISP 的处理对象是符号表达式,简称表达式,相当于其他语言中的语句。AutoLISP 的表达式以“表”的形式存在,包含若干元素,其格式如下:

(函数 [参数 1] [参数 2] ...)

每个表达式以左括号开始,第一个函数必须是函数名,随后是各个参数,参数的有无或多少由函数的性质规定,最后以右括号结束。每个参数也可能是另一个表达式,称为表达式的嵌套或表的嵌套,成为外层表达式和内层表达式。例如,三角形中已知一边(c)和两角(α, β),求另一边的公式:

$$b = c \frac{\sin \beta}{\sin(\alpha + \beta)}$$

AutoLISP 的表达式为具有 5 个层次的表:

(setq b (* c (/ (sin beta) (sin (+ alfa beta)))))

1.4.2 表达式的求值规则

每一个表达式必能返回一个“值”,每一个内层表达式返回的值,都能被外层表达式所使用。因此,一个多层次表达式的执行是从外到内的层层调用,而其求值过程是从内到外的层层求值,最后得到一个最外层表达式的值。例如,上列计算边长的表达式 (setq b (* c (/ (sin beta) (sin (+ alfa beta))))) 共有 5 个层次,其求值过程为:①和函数 (+ alfa beta) 求值,得到 α, β 两角之和;②正弦函数 (sin beta), (sin (+ alfa beta)) 分别求值;③除函数 (/ (sin beta) (sin (+ alfa beta))) 求值;④乘函数 (* c (/ (sin beta) (sin (+ alfa beta)))) 求值;⑤赋值函数 (setq b (* c (/ (sin beta) (sin (+ alfa beta))))) 求值,最后得到所求的三角形另一边长。

“值”的含义很广,可以是数值、数表、字符串、逻辑值和操作符等。例如导线测量中方位角推算时,如果算得的方位角值 (azim) 小于零,则应加 $360^\circ (2\pi)$,其表达式为:

(if (< azim 0.0) (setq azim (+ azim PI PI)))

其中关系运算函数 (< azim 0.0) 表示一个关系式“ $\text{azim} < 0.0$ ”,如果关系式成立,则返回逻辑值 T(真);如果关系式不成立,则返回逻辑值 nil(假)。条件判断函数 (if ...) 如果获得逻辑值 T,则向后续的赋值函数 (setq azim (+ azim PI PI)) 求值(将原有的 azim 加 2π)。