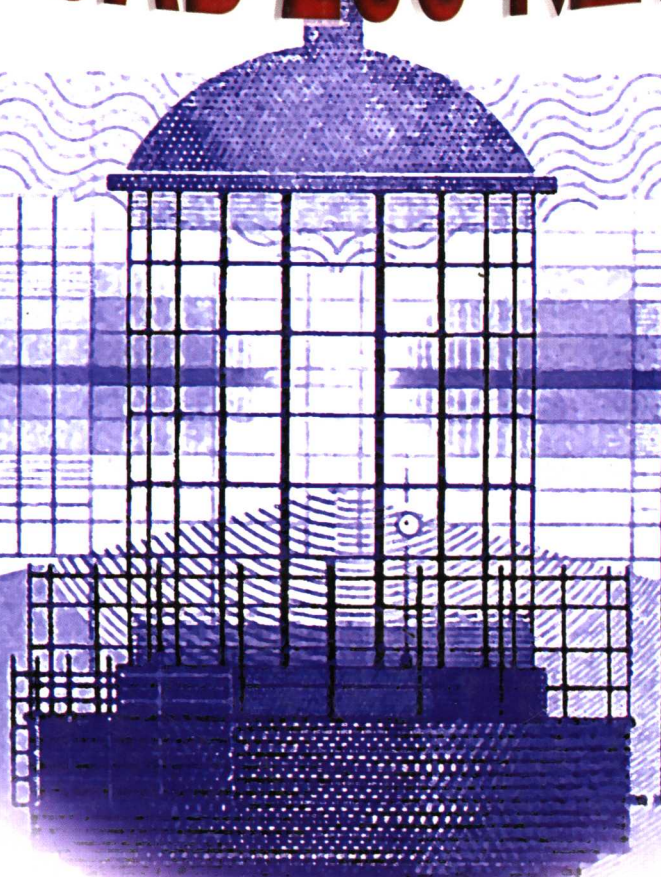




用Visual LISP开发 AutoCAD 2004应用程序



北京希望电子出版社 总策划
黎新懿 赵景亮 编写

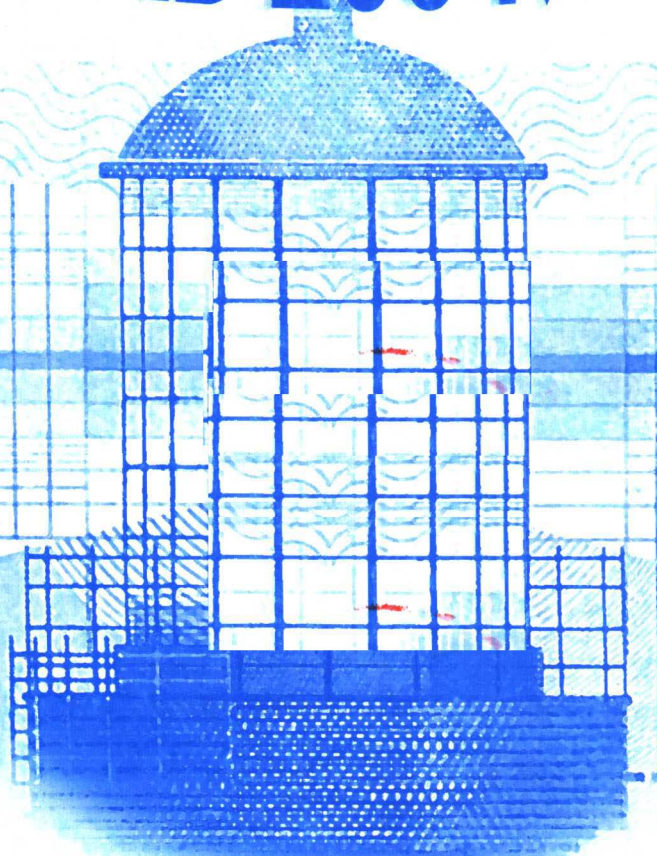


科学出版社

www.sciencep.com



用Visual LISP开发 AutoCAD 2004应用程序



北京希望电子出版社 总策划
黎新懿 赵景亮 编写

 科学出版社
www.sciencep.com

内 容 简 介

在众多的 AutoCAD 开发工具中, AutoLISP 是使用最简单、应用最广泛、使用者最多的一种开发工具, 因此一直受到 AutoCAD 开发人员的喜爱。

本书共由 14 章内容和 4 个附录构成。其中前 4 章的内容: Visual LISP 简介; AutoLISP 程序设计语言; AutoLISP 与 AutoCAD 的通信; 对实体和设备的访问。第 5 章到第 9 章的内容有: Visual LISP 环境简介; 应用 Visual LISP 开发程序; 应用 Visual LISP 进行程序调试; 编译应用程序; 程序维护。第 10 章到第 14 章介绍了: 在 AutoLISP 中使用 ActiveX 对象; 将反应器附着到 AutoCAD 图形上; 设计对话框; 对话框管理; 可编程对话框参考。附录部分的内容为: AutoLISP 函数; AutoLISP 错误代码; DXF 参考; AutoCAD 系统变量。

本书面向初中级用户、各类工程设计人员, 也可作为大专院校机械及相关专业师生或社会培训班的教材。本书内容丰富、实例典型、大部分章节后附带了习题与解答, 富有启发性。

书中部分实例文件请从 <http://www.b-xr.com> 免费下载压缩包 4331.zip。

需要本书或技术支持的读者, 请与北京中关村 083 信箱(邮编 100080)发行部联系, 电话: 010-62528991, 62524940, 62521921, 62521724, 82610344, 82675588(总机) 传真: 010-62520573, E-mail: yanmc@bhp.com.cn

图书在版编目(CIP)数据

用 Visual LISP 开发 AutoCAD 2004 应用程序 / 黎新懿,
赵景亮编写. —北京: 科学出版社, 2003

ISBN 7-03-012358-1

I. 用... II. ①黎...②赵... III. ①计算机辅助设计—应用软件, AutoCAD 2004 ②LISP 表处理语言—软件开发 IV. TP391.72

中国版本图书馆 CIP 数据核字(2003)第 097320 号

责任编辑: 周艳

/ 责任校对: 但明天

责任印刷: 双青

/ 封面设计: 梁运丽

科学出版社 出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003 年 12 月第一版

开本: 787×1092 1/16

2003 年 12 月第一次印刷

印张: 23 1/2

印数: 1—4000

字数: 542412

定价: 36.00 元

前 言

在众多的 AutoCAD 开发工具中, AutoLISP 是使用最简单、应用最广泛、使用者最多的一种开发工具,它的显著特点就是简单易学,可以直接使用,因此一直受到 AutoCAD 开发人员的喜爱。

但是, AutoLISP 只是一种解释性语言,而且不能加密,所以只能开发一些小的应用程序,无法成为开发商业软件的平台。为了解决这个问题, Autodesk 公司在 AutoCAD R14 中推出了 Visual LISP 测试版本,又在 AutoCAD 2000 中推出了正式版本。经过不断的发展, Visual LISP 不断增强了自身的程序开发和调试功能,从而真正成为了一个功能强大的开发平台。

Visual LISP 作为一个集成开发环境,提供了以下功能。

- 一个完整的 AutoLISP 集成开发环境,具有透明的加载编译技术。
- 强大的语法检查器,可以识别 AutoLISP 语法是否错误,以及调用 AutoLISP 函数时参数的输入是否正确。
- 提供一个文本编译器,可以改善程序的运行速度和提供安全、高效的交互平台。
- 特别为 AutoLISP 语言提供了源程序调试器。通过应用强有力的中断处理,可以为程序的调试提供最大的灵活性。这个工具也支持在一个窗口中单步运行 AutoLISP 源程序,而且实时地在 AutoCAD 的图形窗口中显示程序的运行结果。
- 支持 AutoLISP 和 DCL 语言的文本文件中的彩色代码显示功能,并支持 AutoLISP 语法的其他功能。
- 提供一个 AutoLISP 格式化器,可以设置程序代码的书写结构,使程序具有一个更容易读懂的格式。
- 综合的检验器和监视器实用工具,用户可以非常方便地在浏览和修改数据结构时获取变量和表达式的值。这些特征可以用来检验 AutoLISP 数据和 AutoCAD 图形实体。
- 为 AutoLISP 函数提供了上下文相关的帮助系统。
- 强有力的符号名查找和自动匹配特征。
- 项目管理系统,可以使维护多文件应用程序变得非常容易。
- 提供打包功能,可以把编译过的 AutoLISP 文件打包成为一个 ARX 模块。
- 桌面的存储与恢复功能,可以从其他 Visual LISP 进程中保护和再利用 Windows 环境资源。
- 智能的系统控制台使 AutoLISP 用户开发应用程序时的方便性和高效性达到了一个新的水平。Visual LISP 控制台的基本功能和 AutoCAD 文本屏幕的功能是相适应的,它还提供了一些方便的交互功能,比如历史记录滚动和完全输入的行编辑功能。

Visual LISP 包含自己的窗口和菜单,这些窗口和菜单与 AutoCAD 有显著的不同。但是, Visual LISP 不是一个独立的软件,它不能独立运行于 AutoCAD 之外。无论什么时候

都必须先启动 AutoCAD，然后才能利用 Visual LISP 进行工作。

在用户安装 AutoCAD 2004 时，如果选择了典型安装或者全部安装，那么 Visual LISP 就会被自动安装到当前计算机中。值得注意的是，典型安装并不会安装 Visual LISP 的教程部分，而本书的“程序维护”一章要用到教程，此时可以再次运行 AutoCAD 的安装程序添加此部分。

本书由黎新懿、赵景亮、弓丹志编写，其中黎新懿编写第 1、2 章，赵景亮编写第 3、4 章，弓丹志编写其余各章，另外参加收集和整理工作的还有于伟、刘昕、周旋、宁可、王青、高新波、王大智、富晓静、龚瑞男、周剑、徐大丰、张龙、倪晚成。

作者

2003 年秋

目 录

第1章 Visual LISP 简介..... 1	4.10 习题与解答..... 82
1.1 AutoLISP 简介.....1	第5章 Visual LISP 环境简介..... 84
1.2 Visual LISP 简介.....7	5.1 Visual LISP 概述..... 84
1.3 本章小结.....8	5.2 Visual LISP 环境界面..... 85
1.4 习题与解答.....8	5.3 加载并运行 AutoLISP 应用程序..... 90
第2章 AutoLISP 程序设计语言..... 9	5.4 退出 Visual LISP 环境..... 92
2.1 数据类型.....9	5.5 本章小结..... 92
2.2 表达式.....14	第6章 应用 Visual LISP 开发程序..... 93
2.3 数据存储结构.....15	6.1 系统控制台..... 93
2.4 AutoLISP 语言的控制流程.....16	6.2 应用文本编辑器..... 98
2.5 函数与函数控制.....20	6.3 应用控制台和文本编辑器辅助功能..... 104
2.6 程序文件的使用.....22	6.4 格式化代码..... 106
2.7 出错处理.....23	6.5 语法错误检查.....113
2.8 AutoLISP 应用举例.....25	6.6 本章小结.....114
2.9 本章小结.....29	6.7 习题与解答.....114
2.10 习题与解答.....29	第7章 应用 Visual LISP 进行程序调试.... 117
第3章 AutoLISP 与 AutoCAD 的通信..... 33	7.1 Visual LISP 程序调试功能简介.....117
3.1 查询和命令.....33	7.2 调试程序示例.....117
3.2 显示控制.....36	7.3 应用 Visual LISP 程序调试功能..... 121
3.3 获取用户输入.....37	7.4 应用数据检验工具..... 124
3.4 几何工具.....38	7.5 本章小结..... 131
3.5 数据转换.....39	7.6 习题与解答..... 132
3.6 文件控制.....48	第8章 编译应用程序..... 134
3.7 设备访问和控制.....51	8.1 编译和链接程序..... 134
3.8 本章小结.....51	8.2 创建应用程序模块..... 138
3.9 习题与解答.....51	8.3 针对多文档环境的设计方法..... 144
第4章 对实体和设备的访问..... 53	8.4 本章小结..... 149
4.1 选择集介绍.....53	第9章 程序维护..... 150
4.2 选择集控制函数.....54	9.1 管理多个 Visual LISP 文件..... 150
4.3 对象操作.....60	9.2 定义工程..... 151
4.4 老式多段线和细多段线.....71	9.3 对现有工程进行操作..... 158
4.5 非图形对象操作.....72	9.4 优化应用程序代码..... 160
4.6 扩展图元数据 xdata.....73	9.5 本章小结..... 162
4.7 Xrecord 对象.....79	第10章 在 AutoLISP 中使用 ActiveX 对象.. 163
4.8 访问符号表和词典.....79	10.1 概述..... 163
4.9 本章小结.....82	10.2 AutoCAD 对象模型..... 164

10.3 访问 AutoCAD 对象.....	165	附录 A AutoLISP 函数.....	258
10.4 应用 Visual LISP 函数实现 ActiveX 方法.....	169	A.1 赋值、求值与禁止求值函数.....	258
10.5 应用 ActiveX 与其他应用程序交互.....	183	A.2 数值计算函数.....	259
10.6 本章小结.....	188	A.3 关系运算函数.....	261
10.7 习题与解答.....	189	A.4 逻辑运算函数.....	263
第 11 章 将反应器附着到 AutoCAD 图形上 ...	191	A.5 几何运算函数.....	264
11.1 反应器类型和事件基本概念.....	191	A.6 判断函数.....	266
11.2 定义回调函数.....	193	A.7 流程控制函数.....	268
11.3 创建反应器.....	195	A.8 表处理函数.....	269
11.4 在多重名称空间中使用反应器.....	196	A.9 字符串处理函数.....	271
11.5 查询、修改和删除反应器.....	197	A.10 数据类型转换函数.....	273
11.6 临时反应器和永久反应器.....	200	A.11 交互输入函数.....	275
11.7 本章小结.....	201	A.12 有关文件操作的函数.....	277
第 12 章 设计对话框	202	A.13 打印输出函数.....	280
12.1 对话框部件.....	202	A.14 有关函数的函数.....	281
12.2 用 DCL 定义对话框.....	203	A.15 错误处理函数.....	285
12.3 用 Visual LISP 显示对话框.....	206	A.16 与 AutoCAD 直接相关的函数.....	286
12.4 调整对话框的布局.....	207	A.17 实体操作函数.....	288
12.5 设计指南.....	211	A.18 符号表操作函数.....	291
12.6 本章小结.....	214	A.19 扩展数据操作函数.....	293
12.7 习题与解答.....	214	A.20 内存管理函数.....	294
第 13 章 对话框管理	219	A.21 对话框操作函数.....	295
13.1 使用 AutoLISP 程序控制对话框.....	219	A.22 控制环境变量的函数.....	299
13.2 动作表达式和回调.....	221	A.23 词典操作函数.....	299
13.3 控件处理.....	223	A.24 应用程序操作函数.....	301
13.4 嵌套对话框.....	226	A.25 Visual LISP 扩展函数.....	303
13.5 隐藏对话框.....	226	A.26 有关 ActiveX 的函数.....	317
13.6 列表框和弹出式列表.....	229	附录 B AutoLISP 错误代码	323
13.7 图像控件和按钮.....	232	附录 C DXF 参考	326
13.8 特定应用数据.....	234	C.1 ASCII DXF 文件.....	326
13.9 DCL 错误处理.....	235	C.2 二进制 DXF 文件.....	346
13.10 本章小结.....	236	C.3 幻灯片文件.....	347
13.11 习题与解答.....	236	C.4 幻灯片库文件格式.....	349
第 14 章 可编程对话框参考	242	附录 D AutoCAD 系统变量	350
14.1 控件属性.....	242	D.1 A 系统变量.....	350
14.2 DCL 属性目录.....	243	D.2 B 系统变量.....	353
14.3 DCL 控件目录.....	247	D.3 C 系统变量.....	354
14.4 本章小结.....	257	D.4 D 系统变量.....	359
		D.5 Q 系统变量.....	365

第 1 章 Visual LISP 简介

本章将向读者介绍 AutoLISP 和 Visual LISP 的基本概念、功能、工作方式以及相互之间的区别。Visual LISP 代表了新一代的 LISP 语言，它在 AutoLISP 的基础上增加了更多的功能。本章的最后，是关于 AutoLISP 语言使用的简单习题和示例。通过本章的学习和章节末尾的练习，读者将能够得到对 AutoLISP 语言的基本认识。

本章重点：

- AutoLISP 基本概念
- Visual LISP 基本概念与功能介绍

1.1 AutoLISP 简介

LISP(List Processing Language)是人工智能领域中广泛采用的一种程序设计语言。主要用于人工智能、机器人、专家系统、博弈、定理证明等领域。LISP 也被称为符号式语言，因为它处理的对象是符号表达式。LISP 语言的程序和数据都以符号表达式的形式来表示，即一个 LISP 程序可以把另一个 LISP 程序作为它的数据来处理。

LISP 在它 10 多年的发展过程中产生了多种版本，比如 MacLISP, InterLISP, ZetaLISP, CommonLISP 等。其中 CommonLISP 是近几年美国的几所大学（比如 MIT, STANFORD, CMU, RUTGERS 等）和工业界（如 Bell 实验室、DEC 公司、HP 公司等）的人工智能研究人员协同推出的，它概括了 MacLISP、InterLISP、ZetaLISP 等版本的特点，因而功能较强且拥有其他版本的一些优点，是目前 LISP 语言比较完整的版本。

AutoLISP 语言是嵌套在 AutoCAD 内部、将 LISP 语言与 AutoCAD 有机结合的产物。使用 AutoLISP 可以直接调用几乎所有的 AutoCAD 命令。AutoLISP 语言既具备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图形处理功能，是当今世界上 CAD 软件中被广泛采用的语言之一。

美国 Autodesk 公司在 AutoCAD 内部嵌入 AutoLISP 语言的目的是使用户充分利用 AutoCAD 进行二次开发，实现直接增加和修改 AutoCAD 命令，随意扩大图形编辑功能，建立图形库和数据库并对当前图形进行直接访问和修改，开发 CAD 软件包等。

AutoLISP 语言最典型的应用之一就是实现参数化绘图程序设计，包括尺寸驱动程序和鼠标拖动程序等。尺寸驱动是指通过改变实体标注的尺寸值来实现图形的自动修改，鼠标拖动即利用 AutoLISP 语言提供的(GRREAD[<track>]) 函数，让用户直接读取 AutoCAD 的输入设备（通常为鼠标），任选项追踪光标移动存在且为真时，调整所需的参数值而达到自动改变屏幕图形的属性。到目前为止，大多数参数化程序都是针对二维平面图编制的。实际上，立体图同样可以实现参数化绘图。另一个 AutoLISP 的典型应用就是驱动利用 AutoCAD 提供的 PDB 模板构成 DCL(Dialog Control Language)文件创建自己的对话框。

自从 AutoLISP 嵌入 AutoCAD 之后，使仅仅作为交互式图形编辑软件的 AutoCAD 编程能真正进行计算机辅助设计、绘图的 CAD 软件。由于 LISP 灵活多变、易于学习和使用，

因而使 AutoCAD 成为功能很强大的工具性软件。

AutoLISP 具有如下的特点:

- AutoLISP 语言是在普通 LISP 语言基础上,扩充了许多适用于 CAD 应用的特殊功能而形成的,是一种仅能以解释方式运行于 AutoCAD 内部的解释型程序设计语言。
- AutoLISP 语言中的一切成分都是以函数的形式给出的,它没有语句概念和其他语法结构。执行 AutoLISP 程序就是执行一些函数,再调用其他函数。
- AutoLISP 把数据和程序统一表达为表结构,即 S-表达式,因此可以把程序当作数据来处理,也可以把数据当作程序来执行。
- AutoLISP 语言中的程序运行过程就是对函数求值的过程,是在对函数求值的过程中实现函数的功能。
- AutoLISP 语言的主要控制结构是采用递归方式。递归方式的使用,使得程序设计简单易懂。

1.1.1 AutoLISP 的程序结构

与一般的程序语言不同的是,AutoLISP 语言没有“语句”这一概念,AutoLISP 程序是由一个或者一系列按照特定顺序排列的标准表组成的。那么什么是所谓的标准表呢?先来看几个标准表的例子。

例 1-1

```
(setq x 10.0)
(setq y 10.0)
(setq z (+ x y))
(setq a "HelloWorld!")
(command "line" "0,0" "10,10" "")
(command "circle" "100,100" "50")
```

读者不妨在 AutoCAD 2004 环境中试着运行以上的几个例子。比如,在命令行中 command:提示符下输入:

```
(setq x 10.0)
```

如图 1-1 所示。

按回车,就能看到程序运行的结果:

10.0

这也就是程序运行完毕之后变量 x 的值。

类似地,其余几条语句运行之后的结果分别为:

10.0

20.0 (也就是 x 和 y 相加之后的值)

"HelloWorld!";

在坐标分别为(0,0) 和(10,10)的 2 个点之间画一条线段;

以坐标为(100,100)的点为圆心, 50 为半径画了一个圆。

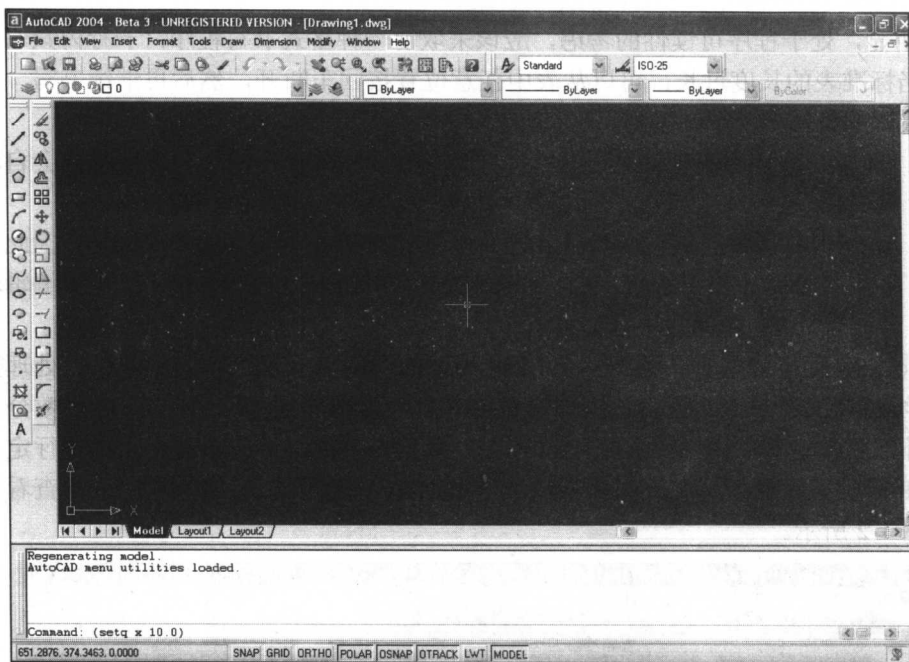


图 1-1 在 command:命令提示符下输入 AutoLISP 语句

到这里可以得到标准表的基本概念：标准表由左右 2 个括号组成，其功能由括号中的代码实现；标准表的第 1 个元素（如 `setq`、`+`、`command`）均为操作符或者函数名，其余的元素都是相应函数的参数。其一般格式为：

(函数名 [<参数 1>] [<参数 2>] [<参数 n>])

在这里需要强调的是，标准表中可能包含着别的表，这是允许的，就像上面的 `(setq z (+ x y))` 一样。

将例 1-1 中的 6 个标准表组合在一起，就可以看作是一个 AutoLISP 程序了。这个程序所实现的功能就是其中每个标准表所实现功能的总和。同样地，只要将希望程序实现的功能不断地分解，分别用不同的标准表来实现，然后再组合在一起，就能实现读者所期望的任何功能。

AutoLISP 程序的书写格式有着如下的特点。

- AutoLISP 程序是由标准表组成的，因此程序中所有的括号都必须左右匹配。
- AutoLISP 程序阅读函数时，按照从左到右的规则进行。
- 操作符（也可以理解称为函数名，比如赋值函数，`+` 相加函数等等）必须放在标准表中第 1 个元素的位置。标准表中的函数与参数、参数与参数之间至少要用一个空格来分开。
- 2 个标准表之间和表内的多余空格和回车都是不需要的，因此一个表可以占多行，一行内也可以写下很多表。比如，例 1-1 中的程序可以写成下面的形式。

```
(setq x 10.0) (setq y 10.0) (setq z (+ x y)) (setq a "HelloWorld!") (command "line" "0,0" "10,10"
"") (command "circle" "100,100" "50")
```

很明显，处于程序可读性的考虑，应该采取前一种格式书写程序，一行内只写一个标准表。当标准表的长度过长，可以从表中任意位置按回车断开，然后再继续书写，以保持程序源代码的整洁。

- AutoLISP 程序中使用分号 “;” 作注释。AutoLISP 执行程序的时候会自动跳过每一行中分号以后的部分。注释可以放在程序中的任何地方。
- AutoLISP 程序的扩展名为(.lsp)，并以 ASCII 码文本文件的形式表达，因此可以使用 Windows 操作系统自带的 notepad 或者 UltrEdit-32 text editor 等文本编辑软件打开已有的.lsp 文件并进行编辑。

AutoLISP 程序就是对一个个 AutoLISP 函数的调用。函数是 AutoLISP 语言处理数据的工具，学习掌握 AutoLISP 语言，核心就是要掌握 AutoLISP 函数。AutoLISP 函数分为内部函数和外部函数。其中内部函数是 AutoCAD 系统自带的，外部函数是用户自行定义的。AutoLISP 提供了大量的系统内部函数，读者可以在 AutoCAD 2004 中按 F1 键查看帮助文档，如图 1-2 所示。

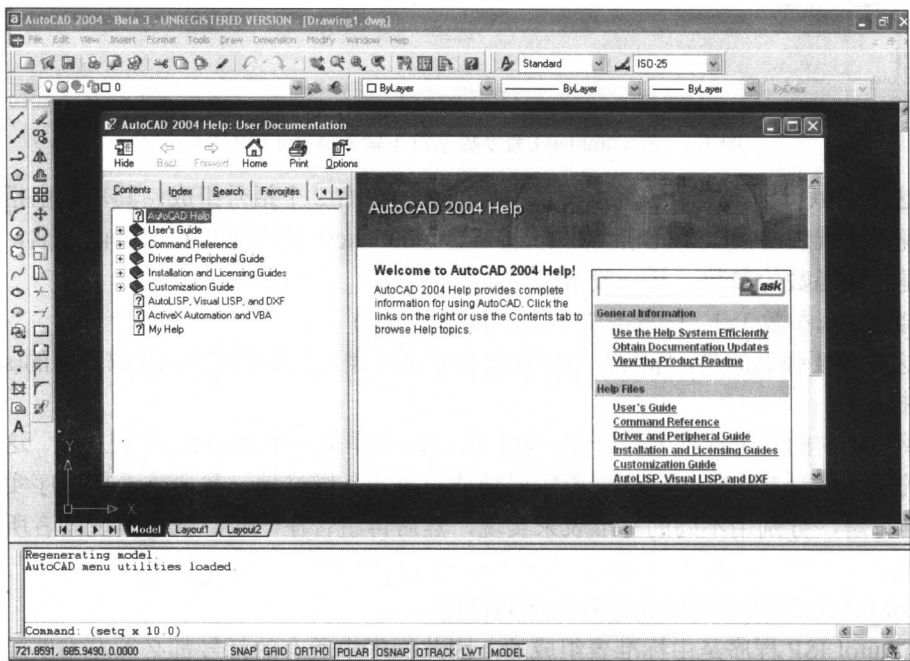


图 1-2 按 F1 键查看 AutoCAD 2004 帮助文件

在 Customization Guide → AutoLISP,Visual LISP,and DXF 中查看 AutoLISP Reference ，里面包含了所有 AutoLISP 所提供的系统内部函数，按照英文字母 A~Z 的顺序排列。如图 1-3 所示。

充分了解并利用这些函数，是编写高质量 AutoLISP 程序的基本条件，读者应该养成经常翻阅帮助文件的良好习惯。

在学习 AutoLISP 的系统内部函数时，必须掌握以下基本内容。

- 函数调用格式：即函数名、函数要求的参数个数和类型。
- 函数功能：即该函数的作用，它对其参数如何处理。

- 函数的求值情况：即哪些参数要求值，哪些不要求值。
- 函数求值返回结果的类型。

1.1.2 AutoLISP 的内存分配

AutoLISP 的变量(整型、实型、字符串等)、内部函数和外部函数等，都只是在 AutoLISP 的编辑过程中，存储在计算机的内存中。当运行 AutoLISP 程序时，AutoCAD 需要 2 个很大的内存区域。

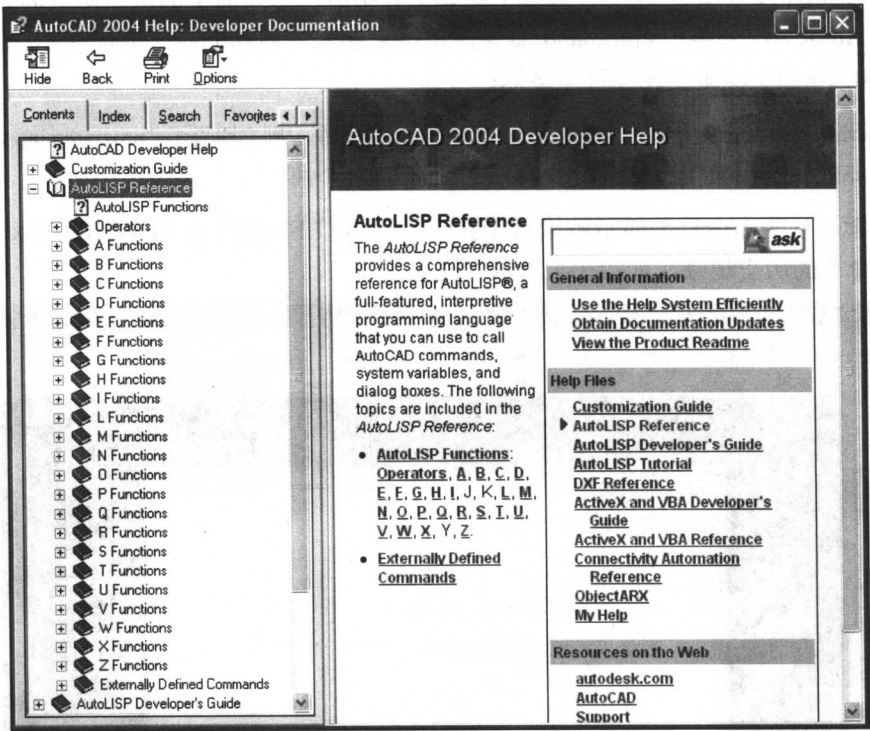


图 1-3 按字母顺序查看 AutoLISP 系统内部函数

- **Heap(堆区域)**：存储所有的函数和变量。程序使用的函数和变量越多，变量名越长，函数越复杂，该区域所占的空间就越大。
- **Stack(栈区域)**：存储函数的变量和局部结果。标准表中嵌套的表越多，函数执行的递归次数越多，该区域所占的空间就越大。

AutoLISP 隐含的堆和栈空间大小分别如下。

- Heap:25000 bytes
- Stack:20000 bytes

在 AutoCAD 下运行 AutoLISP 的时候，用户不能扩展系统的堆或栈空间。如果用户定义的函数数量太多，或者是其他的什么原因，以至用光了所有的堆空间，AutoLISP 将显示如下的信息。

Insufficient node space

并且中止当前函数的运行。如果在执行 AutoCAD 时,没有足够的内存装入 AutoLISP,则显示下列信息。

Insufficient memory-AutoLISP disabled.

只有在内存足够的情况下, AutoLISP 才能正常启动。

1.1.3 AutoLISP 程序的执行过程

对于很短的 AutoLISP 程序(只有 1 至 2 个表组成,比如简单的数值函数的运算),可以直接在 AutoCAD 环境中 command 命令提示符下输入,返回的结果立即显示在文本屏幕上。例如在 command 命令提示符下键入:

(setq string "programming with AutoLISP is great fun!")按回车,则立即显示 "programming with AutoLISP is great fun!" 也就是赋予string的值。

如果读者认为命令行的显示范围过于狭小,可以按 F2 键弹出 AutoCAD Text Window 查看所有命令的历史记录,如图 1-4 所示。

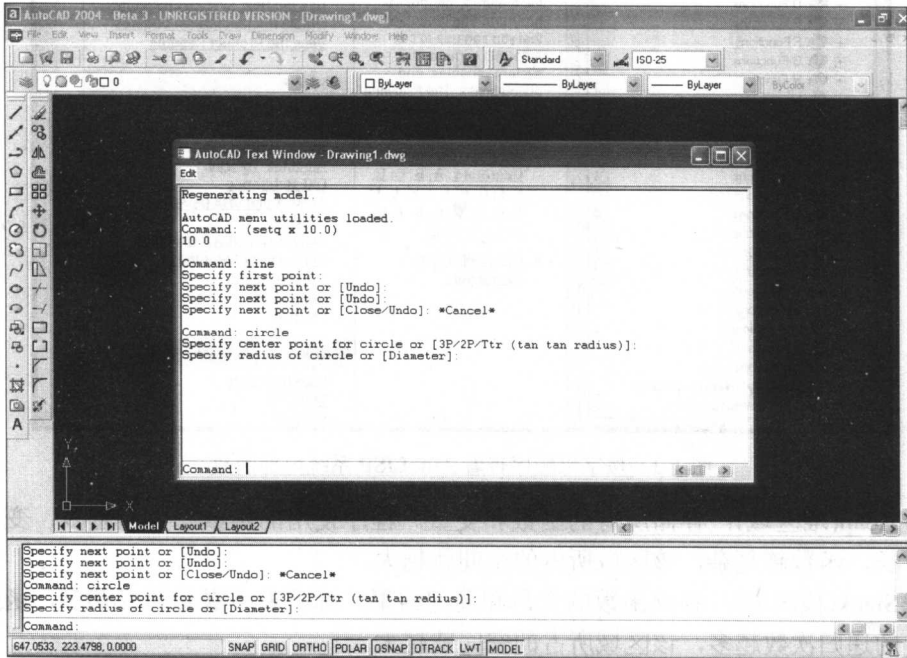


图 1-4 按 F2 弹出 AutoCAD Text Window 查看命令历史记录

对于一般的 AutoLISP 应用程序,需要采用文本编辑器进行编辑,否则一行行地输入语句,一旦出现错误,将很难处理。一般情况下,在安装了 AutoCAD 2004 之后,就可以利用软件自带的文本编辑器进行 .lsp 文件的编写和调试。如果由于某种特殊原因,使得用户的程序编制环境与 AutoCAD 环境相分离,我们就要采取其他的文本编辑器来编辑程序。在这里推荐使用 UltraEdit 9.0,这是一款相当不错的文本编辑软件,支持多种格式的文本格式。下面的例子将示范如何利用 UltraEdit 9.0 编写 .lsp 程序文件,再用 Load 函数装载并运行。

例 1-2

(1) 打开 UltraEdit 9.0, 在文本编辑环境中输入如下代码。

```
(princ "\nWelcome!")  
(defun c:welcome()  
  ( print "AutoLISP welcomes you!"))
```

(2) 将上面的程序保存为 "D:\lisp\MyAutoLISP.lsp"。

(3) 启动 AutoCAD, 在 command 命令提示符下键入:

```
(load "d:/lisp/myautolisp.lsp")
```

(注意: 这里用的是除号/, 读者可以试试看用反斜杠\会发生什么结果)
回车后, 即执行第 1 句并返回最后一个 defun 函数定义的函数名:

Welcome! C:WELCOME

程序文件装载完毕。

(4) 运行 defun 定义的函数, 在 command 命令提示符下键入: welcome
按回车, 命令行中就显示 "AutoLISP welcomes you!"

1.2 Visual LISP 简介

自 AutoCAD R14.01 开始, 出现了 Visual LISP 语言, 它是 Autodesk 公司为了增强 AutoLISP 程序开发能力而设计的软件工具。Visual LISP 为 AutoLISP 增加了许多新的函数, 大大扩展了 AutoLISP 的能力。在对 AutoCAD 进行专业设计的应用范围内, 已经比较接近 ARX 程序的能力。Visual LISP 是设计用来加速 AutoLISP 程序开发的软件工具。Visual LISP 的集成开发环境提供了很多特性, 使用户创建和修改原代码、测试和调试程序更加方便。此外, Visual LISP 还提供了用于传递 AutoLISP 中特殊应用的方法。在 Visual LISP 中, 用户可以在一个单独的环境中执行大部分必要的操作, 包括文本编辑、程序调试和与 AutoCAD 和其他应用程序的交互。

以下是 Visual LISP 集成开发环境的组成部分。

- 语法检查器: 可以识别错误的 AutoLISP 结构和内部函数调用中不合适的参数。
- 文件编译器: 可以改善程序的执行速度并提供一个安全有效的发布平台。
- 资源调试器: 允许用户在一个窗口中一步步地执行 AutoLISP 程序源代码, 并在 AutoCAD 图形窗口中显示代码执行的结果。
- 文本编辑器: 使用 AutoLISP 和 DCL 颜色编码对程序源代码进行编辑。
- AutoLISP 格式工具: 可以重新安排程序的格式, 增强程序可读性。
- 检查和观察工具: 可以方便地访问变量和表达式的值, 还可以浏览 AutoCAD 图形实体的各项属性值。
- 上下文相关菜单: 提供有关 AutoLISP 函数的信息以及对符号名的搜索。
- 项目管理系统: 使用户可以更容易地管理多文件应用程序。编译过的 AutoLISP 文件将被封装为一个单独的模块。

- 窗口的保护和恢复功能：使用户可以保护并再次使用任一 Visual LISP 情景的窗口环境。
- 智能化的控制台窗口：为 AutoLISP 用户提供了更多的便利和更高的效率。控制台的基本功能对应于 AutoCAD 文本屏幕的功能，同时还提供了一些交互特性。

Visual LISP 拥有自己的窗口和菜单，它们与 AutoCAD 的其他窗口是不同的。但是 Visual LISP 不能离开 AutoCAD 而单独运行。用户如果要使用 Visual LISP，必须首先运行 AutoCAD；当用户在 Visual LISP 集成开发环境中执行 AutoLISP 程序时，也需要与 AutoCAD 中的图形和命令窗口进行交互来相应程序的提示。

1.3 本章小结

本章向读者简单介绍了 AutoLISP 和 Visual LISP 的基本概念，包括 LISP 语言的历史和 AutoLISP、Visual LISP 的基本概念、AutoLISP 的程序结构、内存分配、执行过程和其他一些特点。同时，本章还向读者简单说明了 AutoLISP 和 Visual LISP 之间的区别。在随后的 3 章中，将详细学习 AutoLISP 语言。

1.4 习题与解答

习题 1 求表达式 ax^2+bx+c 的值（其中 $a=1, b=2.5, c=4, x=2.34$ ），用 AutoLISP 语言中简单的 setq 赋值函数给每个变量赋值，然后求出表达式的值。

解答：

```
(setq a 1) (setq b 2.5) (setq c 4) (setq x 2.34)
```

```
(setq result (+ (* a (* x x)) (* b x) c))
```

最后从命令行中得到结果：15.3256

习题 2 利用 AutoCAD 内部函数 rectang 绘制一个矩形。矩形的左下角坐标为(0,0)，右下角坐标为(100,100)，用 AutoLISP 语言中的 command 函数实现。

解答：

```
(command "rectang" "0,0" "100,100")
```

习题 3 设表 a 为 f (1 2 3)，将表中的第 1 个元素赋给 x，第 2 个元素赋给 y，第 3 个元素赋给 z。

（提示：按 F1 打开 AutoCAD 2004 Help，单击 Customization Guide 中的 AutoLISP, VisualLISP and DXF，打开 AutoCAD Developer Help，查看 AutoLISP Reference 中的 C Function 函数表的前 3 个：caddr, cadr, car）

解答：

```
(setq a '(1 2 3))
```

```
(setq x (car a)) (setq y (cadr a)) (setq z (caddr a))
```

第 2 章 AutoLISP 程序设计语言

本章将向读者介绍 AutoLISP 的一些基本概念以及 AutoLISP 的数据类型和一些基本功能。AutoLISP 代码是不需要编译的,用户可以模仿上一章中的程序实例,直接在 AutoCAD 命令提示符(Command:)中输入代码,并看到结果。

本章重点:

- AutoLISP 表达式和数据类型
- AutoLISP 变量
- 基本输出函数
- 符号和函数控制
- 出错处理

2.1 数据类型

AutoLISP 的数据类型主要包括: 整形数、实型数、字符串、表、选择集、实体名称、VLA 对象等等。

2.1.1 整型

整型即整数,整数由数字组成,正号“+”可以省略。AutoLISP 的整数为 32 位带符号的数字,其范围为-2147483648~2147483647。如果输入的数超过了这个范围,AutoLISP 会将整数转换为实数。然后,对 2 个有效整数进行算数运算时,如果其结果超过了这个范围,那么所得的结果将是无效的。下面这个例子展示了 AutoLISP 对于整数逸出的处理机制。

例 2-1

在命令行输入 AutoLISP 所能允许的最大整数:

```
(setq a 2147483647)
```

回车,则命令行正常显示 a 的值: 2147483647

而当输入的整数大于所能允许的最大整数,AutoLISP 将整数转化为实数:

```
(setq a 2147483648)
```

回车,命令行显示 a 的值: 2.14748e+009

当 2 个有效整数进行算数运算的结果超过了 AutoLISP 允许的范围,那么将产生一个无效的结果,例如:

```
(setq a (+ 2147483646 3))
```

回车,命令行显示 a 的值为-2147483647,这显然是错误的。

但有时这样的运算也能产生正确的结果,比如:

```
(setq a (+ 2147483648 2))
```

至于 AutoLISP 对于负数的处理办法，读者可以自行按照上面的办法进行试验，这里不再赘述。

2.1.2 实型

实型数是带小数点的数，在-1 和 1 之间的实数必须以 0 开头，例如“.5”是错误的表示法，应该写成“0.5”。实数用双精度的浮点数表示，并且至少有 14 位有效位数的精度。在这里要注意的是，Visual LISP 并不显示所有的有效数字。

实型数也可以用科学计数法表示，即数字后可以有一个 e 或者 E，其后是数的指数。例如 0.12×10^9 表示为 0.12E9，0.0000041 表示为 4.1e-6。类似于 3.1、0.23、-56.123 以及 21,000,000.0 都是合法的 AutoLISP 实型数。

2.1.3 字符串

字符串又被称为字符常数，它是由双引号引起来的字符序列。字符串中字母的大、小写和空格符都被认为是有效的字符。

字符串中字符的个数（不包括双引号）称为字符串的长度。字符串的最大长度为 100 个字符。如果字符的格数超过了 100,那么第 100 个字符以后的字符将被认为是无效的。字符串可以是空的，就是“”，这叫做空串，其长度为 0。

任何字符都可以用\nnn 的格式表示，其中反斜杠是 ASCII 码的前导标识字符，nnn 是该字符八进制的 ASCII 码。例如，字符串 ABCD 也可以表示为\101\102\103\104。由于反斜杠已经作为 ASCII 码的前导标识字符，它的\nnn 格式表示为\114；双引号已经作为字符串的定界符，它的\nnn 格式表示是\042。

一些常用的控制字符，比如反斜杠、双引号，除了可以用\nnn 的格式表示以外，还可以用“\转义字符”的格式表示为“\\”、“\”等特殊字符，见表 2-1。

表 2-1 常用控制字符的表示方法

控制字符	用“\”为前导的转义字符表示	用“\”为前导的 ASCII 码数值表示
反斜杠“\”	\\	\114
双引号“”	双引号“”	\042
Esc 键	\e	\033
回车键	\r	\015
换行	\n	\012
Tab 键	\t	\011

注意其中的字符 e、n、r、t 都必须小写。

下面的表达式中的“\n”和“\012”都起到换行的作用。

例 2-2

(setq a (getpoint "\n请用鼠标指示一个点:"))

(setq a (getpoint "\012请用鼠标指示一个点:"))

2.1.4 表

表以左括号“(”开始，以配对的右括号“)”结束，“()”里面可以是空的，可以有