

Visual LISP_{for} AutoCAD 2000 程序设计

●从学会到用好

陈伯雄 冯 伟 编著



机械工业出版社
China Machine Press

本书是基于 Visual LISP for AutoCAD 2000 程序设计, 作为专业设计用户, 进行专业辅助设计程序的制作和使用。书中详细讨论了 LISP 程序设计的基本方法和应用技巧, 内容包括: 全部 VLISP 和 AutoLISP 函数解释、全部 DCL 控件和属性解释、VLISP/DCL 程序设计基础知识和应用技巧、给 AutoCAD 添加用户命令、给 AutoLISP 添加用户函数、专业应用程序设计中典型问题的解决方案和例程分析、VLISP 中的集成开发环境的功能和用法解释、VLISP 中的 ActiveX 功能及其应用、VLISP 中的反应器功能及其应用、程序包的编译和调试、程序设计辅助工具。

本书按照由浅入深、循序渐进的原则编写, 力争做到初学者能够看懂, 程序设计人员也能得到启发, 是工程师自学程序设计的良师益友。此外, 本书也适合作为 ATC 高级应用培训的基础教材。

图书在版编目 (CIP) 数据

Visual LISP for AutoCAD 2000 程序设计: 从学会到用好/陈伯雄, 冯伟编著. —北京: 机械工业出版社, 2000.3

ISBN 7-111-07853-5

I. V... II. ①陈...②冯... III. 计算机辅助设计-应用程序, AutoCAD 2000 IV.TP391.72

中国版本图书馆 CIP 数据核字 (2000) 第 02580 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码: 100037)

责任编辑: 武 江

封面设计: 姚 毅 责任印制: 何全君

三河市宏达印刷厂印刷 • 新华书店北京发行所发行

2000 年 3 月第 1 版第 1 次印刷

787mm×1092mm 1/16 • 22.75 印张 • 字数 543 千字

印数 0001—5000 册

定价: 46.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010) 68993821、68326677-2527

致 读 者

尊敬的读者：

感谢您打开并翻阅这本书。参考书的价值在于能解决自己工作中的一些问题，因此而值得一买。您正在翻看的这本书就具有这样的特点。

相当多的 AutoCAD（也包括类似的其他软件）使用者，仅是将 AutoCAD 当成了电子图版。毫无疑问，CAD 的终点不是画图。做为一名工程师，有人会问您：“您是做什么工作的？”回答“我是机械工程师”或者“我是搞 CAD 的”，对方可能随口应道：“啊，您是画图的呀！”

发生这种事情，原因多是对方不熟悉工程师的设计工作，以他的观察，工程师每天就是在画来画去，可不就是个画图的。他不知道工程师的图不是随便画出来的。

既然 CAD 叫做“计算机辅助设计”，就要搞清楚设计是什么？再讨论怎样辅助。

设计，是要完成一个创新的构造方案，满足一个应用需求的过程。在这个过程中，可以借用人类已有的知识，同时又在创建新的知识。在设计中参考已有的同类设计方案是必然的动作。但是，这不是照抄，而是借用这个方案的思路，进一步加以发挥，以便满足自己在设计中的要求。

在这样的创新或者参照的过程中，要反复对比不同的方案，进行设计参数求解，进行机构动作的讨论，甚至进行局部重要结构的试制乃至台架试验，这些都是设计过程的常见组成部分。可见，绘图只是设计过程中的一种辅助手段。

计算机，是一种“因为创造而产生需求”这样一种独特的发明。尤其是“软件”出现以后，由计算机的发明而引发的需求正在极大地扩张。有人用一句话精辟地概括了计算机应用的目的：“将我们已知如何做的事情加以自动化”。这样，辅助设计的最终目标就是，我们将已知如何设计的设计自动化，其结果就像计算机控制的电喷汽车发动机一样，能够以最佳的工况运行，它比任何一个独立的司机都能更好地控制发动机的参数，使这个发动机的性能达到极限。

但是，从总体上说，CAD 永远不能替代工程师，这是因为，人类的设计能力是一种不断扩展、不断创新的能力，这种“创新”的能力是计算机永远不能达到的。因为随着对某一问题认识的深入，将有新的思维模式产生，而且永无尽头。

实际上，目前我们仍然有许多的设计过程是可以做到“已知”的，例如，按要求生成凸轮从动滚子的运动曲线，再通过一系列确定的处理生成凸轮轮廓，这个设计就可以最终“自动化”。自动化的结果，会以极快的速度、极完美的数据处理精度，自动完成这个设计。这样，工程师就是不掌握这个设计求解过程的细节，也能在计算机上完成设计。这就是 CAD 系统的主要特点：可靠的知识传递与保存手段。

就目前的可能，在 PC 上进行 CAD 可以实现的目标就是：设计过程模拟和局部设计过程的自动化。在个别设计部门，也实现了典型产品的一体化自动设计。

作为 CAD 系统，通常由三个层次的软件组合而成：

（1）设计软件的底台：提供界面、环境、核心算法、数据库等基础设施。如我们现

是在 AutoCAD 2000 这个底台上进行工作。

(2) 设计支持软件：提供与设计需要相关的、比较专业的支持软件，如国标图库、通用设计工具、设计手册等等。这些软件多数是由增值软件开发商完成的。

(3) 专业设计软件：提供窄范围、大深度的专业设计自动化或者辅助系统，例如发动机装配工具设计、胶印机控制凸轮设计、组合机主轴箱设计等等。实际上，CAD 系统能否真正体现出它的存在价值，最明显的标志就是上述专业设计软件的使用。这些软件真正起到了“设计”的作用。实践证明，在某个专业性很强的设计上，它确实能够替代工程师！

上述(1)和(2)可以由一些不必很了解设计，但软件知识较多的人来写。当然，应当由现场工程师进行评测，防止写成“脱离应用实际”的设计支持软件。而(3)就只能由设计师自己来写，这样的题目几乎是软件开发商永远的“盲区”。由于专业性太强，知识结构太偏，对于工科大学毕业生，如果没有在这个行业里做过几个真正的设计，也很难听懂专业设计师的叙述，要求他完成这个程序就太勉为其难了。可见，必须找到能够被专业设计师所掌握，又能够发挥他们的长处（熟悉设计），避开他们的短处（软件知识较少）的专业设计程序开发手段。这就是本书介绍的 Visual LISP。

在 Visual LISP 的协助下，从对 AutoCAD 进行操作的功能上讨论，仅比 ARX 少了一个功能：自定义 AutoCAD 对象。从目前的情况看，Visual LISP 是 AutoCAD 中几乎所有的应用程序的“大管家”，而且目前只有 LISP 表达式，才能以添加在脚本文件、对话框和菜单之类的其它用户定制文件描述中，直接用在 AutoCAD 的命令行中响应几乎所有的命令，甚至将一套专业设计程序放在某一根图线中，随时激活这个程序。

实际上，对计算机专业知识的掌握并不是在开始进行应用程序设计时所最必要的东西，对此，可用“急用先学，立杆见影，在用字上狠下工夫”的原则，CAD 技术是一套包罗万象的综合技术，不能指望象在校学习那样按部就班地学会，总原则是在已有的专业知识基础上延伸和扩展，突破几点、联点成线、组线成网，网密到一定程度就会产生质变，成为完整的知识面。对 AutoCAD / Visual LISP 的理解越深，使用就越加准确自如，效率才会表现出来。仅按 AutoCAD 手册资料上说的去用，会有很大的局限性。

AutoCAD 是“干”会的，不是“学”会的。我的一个朋友告诉我他的体会：“不管什么程序设计语言，只要你真的玩透了，一样能够写出漂亮的东西。”我信。浅尝则止、道听途说都是必须避免的。

尊敬的读者，这本书是笔者十几年从事 AutoCAD 应用、开发和推广中所积攒的经验和体会。相信这些经验一定能够对你遇到的难题有所帮助。在这本书的启发下，从学会 Visual LISP，到用好这种工具，解决你自己设计中的几个难题，这正是本书的编写目的。

当您有了这样的体会，千万别忘了告诉我一声，让我们一起分享成功的快乐。

在本书的编写过程中，笔者得到了单适南、谭孟恩、张朝阳、孙红梅等先生的指教，笔者在此向他们及其他给笔者帮助和支持的朋友表示衷心的感谢。书中的许多内容包含了他们的智慧。

陈伯雄 冯伟
1999 年 12 月 6 日 于长春

前 言

Auto LISP 是为 AutoCAD 的专业用户进行应用开发而设计的编程语言，它起源于 50 年代晚期，是为编写人工智能程序而设计的 LISP 语言，LISP 到现在仍然是许多人工智能程序的基础。

80 年代中期，从 AutoCAD R2.1 起，内嵌了 Auto LISP 作为 AutoCAD 的应用程序主要的编程接口 (API)，成为 AutoCAD 中的第一个用户专业程序开发工具。

Auto LISP 是符号~函数式语言，数据和函数都用符号表达，这就使程序设计极其灵活。计算机应用开发的要点是：**将我们已知如何做的事情加以自动化**；而教会 AutoCAD 做一件事的主要的方法就是 Auto LISP 程序设计，因此，如果你是要做辅助设计，而不仅仅是计算机画图，Auto LISP 程序设计就是 AutoCAD 用户所必须掌握的基本技能。其主要特点是：

1) Auto LISP 比较容易学会和使用。笔者自学 Auto LISP 程序设计和教学过程中，充分体会到了这一特点。一个应用程序设计知识很少的人，经过几十个课时的教学和上百个小时的练习，就可以相当好地掌握和使用了。

2) Auto LISP 是表类型数据处理语言，擅长处理各种结构的数据表。数据类型是相当随意的，尤其善于处理不同长度和组织结构的数据类型；你可以按需求及最佳结构，设计和使用自定义的结构类型数据，甚至是几十万个元的巨大的结构类型数组。

3) Auto LISP 是用前缀表示来写程序，即先说做什么，再说对谁作。

4) Auto LISP 没有语句、过程等这类一般程序设计语言中通常的概念，Auto LISP 程序是由一个或多个、顺序排列或多层嵌套结构的“函数调用表”表达式组合而成的。

5) 擅长描述人机交互操作的过程。对于各种用户输入的接收、错误识别与恢复等方面的优秀功能（尤其是在 DCL 程序加入后），怕是很少有别的语言能与之相比。

6) 可以在 Auto LISP 程序中引用 AutoCAD 的几乎所有的功能，进行多图档下的图形处理的自动化，尤其是较深层的应用——对象数据库、对象反应器等等的操作。

7) Auto LISP 源程序可以用任何一种文本文件制作功能产生，比如 EDIT、Dana 等，但是，在 Visual LISP 集成开发环境中，将有更好的操作功能。程序文件的扩展名应当是 *.LSP。编好的 Auto LISP 源程序，可以在【命令:】提示下用 (Load “程序名”) 调入源程序，之后以再用程序所描述的不同的方式运行（立即执行、内部调用、外部调用）。

自 AutoCAD R14 开始出现了 Visual LISP (VLISP)，这是 Autodesk 公司为了增强 Auto LISP 程序开发能力而设计的软件工具。

VLISP 的集成开发环境提供了许多功能，使得 Auto LISP 在程序设计中，编写、修改和调试程序变得更加容易。

VLISP 为 Auto LISP 增加了许多新函数，大大扩展了 Auto LISP 的能力。在对 AutoCAD 进行专业程序设计的应用范围内，已经比较接近 ARX 程序的能力，但是，程序的风险大大低于 ARX，对编程者计算机专业知识的要求也大大低于 ARX。

VLISP 增加了 Auto LISP 应用程序的工程管理和程序包编译功能。使 Auto LISP 效率

和保密性得到很大提高。

可见，在今后相当长的一段时间内，被 VLISP 增强的 Auto LISP 是专业用户进行应用程序设计的最佳手段。

目 录

致读者

前言

第 1 章 Visual LISP 概况	1
1.1 启动 Visual LISP	1
1.2 Visual LISP 界面概要	1
1.2.1 菜单区	2
1.2.2 工具栏区	2
1.2.3 控制台窗口	3
1.2.4 状态栏	3
1.2.5 跟踪窗口	3
1.2.6 文本编辑器	3
1.2.7 Visual LISP 的其他窗口	4
1.3 加载和运行 Auto LISP 程序	4
1.3.1 打开一个 Auto LISP 程序	4
1.3.2 加载并运行程序片断	5
1.3.3 加载并运行整个程序	5
1.4 使用扩展的 Auto LISP 函数	6
1.5 关闭 Visual LISP	6
第 2 章 使用 Visual LISP 文本编辑器	7
2.1 文本编辑器概要	7
2.2 文本编辑器快捷菜单	7
2.3 文本编辑器的快捷键	8
2.3.1 修改文本	8
2.3.2 选定文本	8
2.3.3 浏览	9
2.3.4 格式化	9
2.3.5 移动和复制	10
2.4 查找和替换	11
2.4.1 查找和替换	11
2.4.2 书签	12
2.5 语法结构分色和匹配	13
2.5.1 语法分色	13
2.5.2 匹配	14
2.5.3 使用自动匹配搜寻结果	16

2.5.4	按历史匹配	17
2.5.5	按系统匹配	17
2.6	程序代码格式化	18
2.6.1	Visual LISP 程序格式样式	18
2.6.2	Visual LISP 的注释	19
2.7	语法检查	20
2.7.1	检查括号匹配	20
2.7.2	用语法分色检查拼写错误	20
2.7.3	使用检查命令检查其他语法错误	20
第 3 章	调试程序	22
3.1	Visual LISP 调试功能总论	22
3.2	学习调试器使用方法	22
3.2.1	调试准备	22
3.2.2	试验跟踪和分步调试过程	23
3.2.3	试验动态跟踪调试	24
3.3	Visual LISP 调试功能	25
3.3.1	调试功能概况	25
3.3.2	VLISP 中程序中中断的基本概念	28
3.3.3	关于可继续中断循环	28
3.3.4	关于进入可继续中断循环之后的调试方法:	30
3.3.5	关于不可继续中断循环	30
3.3.6	断点操作	30
3.4	数据检验工具	32
3.4.1	打开和关闭跟踪日志文件	33
3.4.2	使用监视窗口	33
3.4.3	使用跟踪堆栈窗口	35
3.4.4	使用符号服务对话框	40
3.4.5	使用检验窗口	41
3.5	访问 AutoCAD 图形对象	45
第 4 章	AutoLISP 程序设计	49
4.1	AutoLISP 的应用特色	49
4.1.1	AutoLISP 比较容易学会和使用	49
4.1.2	AutoLISP 适于给专业用户作开发工具	50
4.1.3	AutoLISP 程序语法特点	52
4.1.4	AutoLISP 程序与 AutoCAD 应用开发	52
4.2	编写 AutoLISP 专业用户功能的一般过程	53
4.2.1	专业应用程序设计的一般过程	54

4.2.2	专业应用程序设计的质量要求	55
4.2.3	专业应用程序设计的辅助手段	56
4.3	在命令行使用 AutoLISP 表达式	58
4.3.1	建立和引用 AutoLISP 变量, 响应命令提示	58
4.3.2	用 AutoLISP 表达式响应命令提示	58
4.4	在其他定制文件中使用 AutoLISP 表达式	60
4.4.1	在脚本文件中使用 LISP 表达式	60
4.4.2	在菜单文件中使用 LISP 表达式	60
4.4.3	ACAD.LSP、*.MNL、和 ACADDOC.LSP	61
4.4.4	将 AutoLISP 表达式放在某个图形对象中	62
4.5	给 AutoCAD 添加用户命令	62
4.5.1	用户自定义函数之一	63
4.5.2	(VL-Command) (Command) 函数	63
4.5.3	在程序中引用 AutoCAD 命令时的注意事项	64
4.5.4	实例 I	65
4.5.5	实例 II	66
4.5.6	怎样描述关于点位的响应	67
4.5.7	响应“选择对象:”的提示	69
4.6	怎样在 AutoLISP 程序中, 组织和使用对象选择集	70
4.6.1	选择集过滤器的使用实例	70
4.6.2	在程序中, 对象选择集组建方法	72
4.7	给 AutoLISP 添加用户函数	74
4.7.1	Defun 函数的再认识	74
4.7.2	形式参数和实际参数	75
4.7.3	自定义函数的必要性	75
4.7.4	制造自定义函数的返回值	76
4.7.5	自定义函数的变量表写法	77
4.8	AutoLISP 程序可能的运行模式	78
4.9	在程序中使用 DCL	81
4.9.1	概论	81
4.9.2	DCL 程序的语法	82
4.9.3	对话框的使用	82
4.9.4	察看对话框	83
4.9.5	制作第一个 DCL	83
4.9.6	使用图像控件	85
4.9.7	DCL 支持程序	89
4.9.8	DCL 的检验和出错处理	89
4.9.9	DCL 程序的简单控制和使用	93
4.9.10	用 DCL 查设计参数表	94

4.9.11	图像按钮的控制和使用	96
4.9.12	DCL 结束的控制方法	98
4.9.13	自定义图像按钮集	99
4.9.14	列表盒与用户对有限参数的选择输入	100
4.9.15	列表盒内容设置的几个技术要点	102
4.9.16	不确定值的输入问题	106
4.10	对象数据库的访问与程序设计中的 CAGD	112
4.10.1	传统的数据库访问方法	112
4.10.2	找到要分析对象的图元名的方法	115
4.10.3	对象数据的使用	116
4.10.4	用 ActiveX 数据结构访问数据库	118
4.10.5	在专业设计程序中的 CAGD	120
第 5 章	AutoLISP 程序设计技巧集锦	125
5.1	理解和正确使用 AutoLISP 的条件函数	125
5.2	求两条直线段的交点	127
5.3	使用对象描述双元表	130
5.4	利用绘图时建立的条件进行尺寸标注	132
5.5	(HandEnt) 函数和永久型选择集	133
5.6	在 AutoLISP 中定义和使用多维数组	134
5.7	使用外部参数文件	136
5.7.1	其他应用程序生成的参数文件	136
5.7.2	AutoLISP 程序专用的外部数据文件	137
5.7.3	读入外挂数据文件的结束处理	138
5.8	设计手册查找自动化	138
5.9	建立和使用非图形数据	140
5.9.1	属性格式的非图形数据结构	141
5.9.2	扩展的对象数据库	143
5.9.3	图形词典	148
5.10	自定义出错处理	151
5.10.1	设置公用的自定义出错处理功能	151
5.10.2	在自定义函数中, 函数实参的检验和出错退出功能	153
5.10.3	在函数的定义中, 就地处理出错	155
5.11	参数化的、柔性的图形库	156
5.12	在 Get 族函数中使用关键字	158
5.13	怎样编写设计过程模拟程序	160
5.14	正确使用 AutoCAD 的系统变量	164
5.15	在程序中使用 Fillet 功能	165
5.16	怎样在一个程序中生成多个图形文件	166

5.17	根据图档自动加载应用程序	167
5.18	自定义工具栏图标丢失问题	169
5.19	Read 函数的灵活使用	170
5.20	Eval 函数的用法	172
5.21	Set 函数的用法	177
5.22	SsGet 函数的用法	177
第 6 章	工程—应用程序集的管理和编译	179
6.1	管理多个程序文件	179
6.1.1	Visual LISP 的工程	179
6.1.2	工程中涉及到的文件类型	179
6.2	使用 Visual LISP 工程	180
6.2.1	创建一个新的工程	181
6.2.2	确定工程中 AutoLISP 源程序	182
6.2.3	调入文件的次序	182
6.2.4	选择编译器选项	183
6.3	工程文件的操作	185
6.3.1	在工程中查找源文件中的内容	187
6.3.2	将工程放在应用程序包中	187
第 7 章	应用程序包的编译	189
7.1	程序包的编译和链接	189
7.1.1	VLISP 的编译器	190
7.1.2	用(VLisp-Compile)函数编译程序	190
7.1.3	加载和运行已编译程序	191
7.1.4	编译后的函数调用问题	192
7.2	制作应用程序包	192
7.2.1	生成一个新的应用程序包	193
7.2.2	应用程序包的更新和修改	196
7.3	多图档下的程序设计	198
7.3.1	变量空间	198
7.3.2	使函数在全部图档中有效	199
7.3.3	应用程序的独立变量空间	199
7.3.4	使具有独立变量空间的非 C 函数对外部引用有效	200
7.3.5	在其他 VLX 程序中引用独立变量空间的函数	201
7.3.6	在独立变量空间的 VLX 中使用图档变量空间中的变量	202
7.3.7	使图档变量空间中的变量在所有图档中都有效	202
7.3.8	在各个变量空间之间进行数据共享	202
7.3.9	在多图档环境中的自定义错误处理	203

第 8 章 ActiveX 的使用	204
8.1 ActiveX 对象概论	204
8.2 AutoCAD 的对象模型构架	204
8.2.1 对象的属性	204
8.2.2 对象的方法	205
8.3 ActiveX 对象访问概论	205
8.3.1 访问应用程序对象	205
8.3.2 生成和访问图形对象	206
8.4 用 AutoLISP 函数使用 ActiveX 方法	208
8.4.1 查找所需要的函数	209
8.4.2 确定函数的参数	209
8.4.3 将数据转换为 ActiveX 类型	210
8.4.4 使用变体类型数据	211
8.4.5 使用安全数组	212
8.4.6 转换其他 AutoLISP 数据类型, 为 ActiveX 方法准备数据	215
8.4.7 实例分析	215
8.4.8 确定方法或属性是否适用于某对象	217
8.4.9 使用集合(Collections)对象	217
8.4.10 释放对象和空出内存	219
8.4.11 转换对象的引用	220
8.4.12 ActiveX 方法出错的处理	221
8.5 ActiveX 与其他应用程序的交互	224
8.5.1 引入一个类型库	224
8.5.2 与应用程序建立联系	226
8.5.3 一个应用程序实例	227
8.5.4 不引入类型库使用 ActiveX 方法	229
8.5.5 访问和更新图形对象属性	230
8.5.6 使用参数带回返回值的 ActiveX 方法	232
第 9 章 反应器的使用	234
9.1 反应器	235
9.1.1 反应器类型	235
9.1.2 反应器的回调事件	236
9.1.3 反应器的回调函数	237
9.2 生成反应器	238
9.2.1 生成对象反应器	239
9.2.2 给反应器添加应用数据	240
9.2.3 在多图档中使用反应器	241
9.2.4 查询、修改和删除反应器	242

9.3	临时的、或永久的反应器	244
9.4	对象反应器支持程序的自动加载	245
9.5	反应器应用注意事项	246
第 10 章	附录	248
10.1	AutoLISP 数据类型	248
10.2	AutoLISP 基本函数	250
10.2.1	应用程序管理函数	250
10.2.2	数学计算函数	252
10.2.3	等式和条件函数	255
10.2.4	出错处理函数	258
10.2.5	函数处理函数	260
10.2.6	表操作函数	263
10.2.7	字符串处理函数	268
10.2.8	符号操作函数	271
10.3	AutoLISP 实用函数	275
10.3.1	转换函数	275
10.3.2	设备访问函数	278
10.3.3	显示控制函数	279
10.3.4	文件操作函数	281
10.3.5	几何函数	284
10.3.6	命令与查询函数	285
10.3.7	用户输入函数	289
10.4	AutoLISP 选择集和符号表函数	292
10.4.1	扩展的数据库操作函数	292
10.4.2	对象操作函数	292
10.4.3	选择集操作函数	295
10.4.4	符号表操作函数	299
10.5	AutoLISP 内存管理函数	302
10.6	Visual LISP 的扩展函数	302
10.6.1	ActiveX 集合操作函数	302
10.6.2	ActiveX 数据转换函数	303
10.6.3	ActiveX 方法调用函数	307
10.6.4	ActiveX 对象处理函数	308
10.6.5	ActiveX 属性处理函数	310
10.6.6	应用程序对象处理函数	311
10.6.7	Visual LISP 反应器函数	312
10.6.8	VLX 变量空间函数	328
10.6.9	VLX 变量空间通信函数	329

10.6.10	系统注册表函数	329
10.7	DCL 操作函数	330
10.7.1	对话框打开与关闭控制函数	330
10.7.2	控件和属性操作函数	331
10.7.3	列表类控件操作函数	332
10.7.4	图像控件操作函数	332
10.8	DCL 预定义控件	333
10.8.1	DCL 控件的预定义属性	333
10.8.2	DCL 控件的自定义属性	337
10.8.3	DCL 的预定义动作控件	337
10.8.4	DCL 的预定义控件簇	340
10.8.5	DCL 的预定义修饰和提示性控件	341
10.8.6	DCL 的预定对话框结束和出错处理控件	343
10.9	AutoLISP 程序分析器	343
10.9.1	用分析器分析程序的操作过程	344
10.9.2	程序分析	346
10.9.3	主对话框中的有关功能	346
10.9.4	选项对话框中的有关功能	346
10.9.5	一些说明	347
10.9.6	程序分析规则文件	348

第 1 章 Visual LISP 概况

1.1 启动 Visual LISP

Visual LISP (VLISP) 集成开发环境是在单独的窗口中运行的, 用户必须启动 VLISP, 才能在它的集成开发环境 VLIDE (Visual LISP interactive development environment) 中工作。启动 Visual LISP 的步骤:

- 1) 启动 AutoCAD。
- 2) 从 AutoCAD 菜单中选择【工具】|【Auto LISP】|【Visual LISP 编辑器】, 或在命令提示处输入命令: VLISP 或者 VLIDE。

1.2 Visual LISP 界面概要

启动 VLISP 后显示如图 1-1 所示的界面:

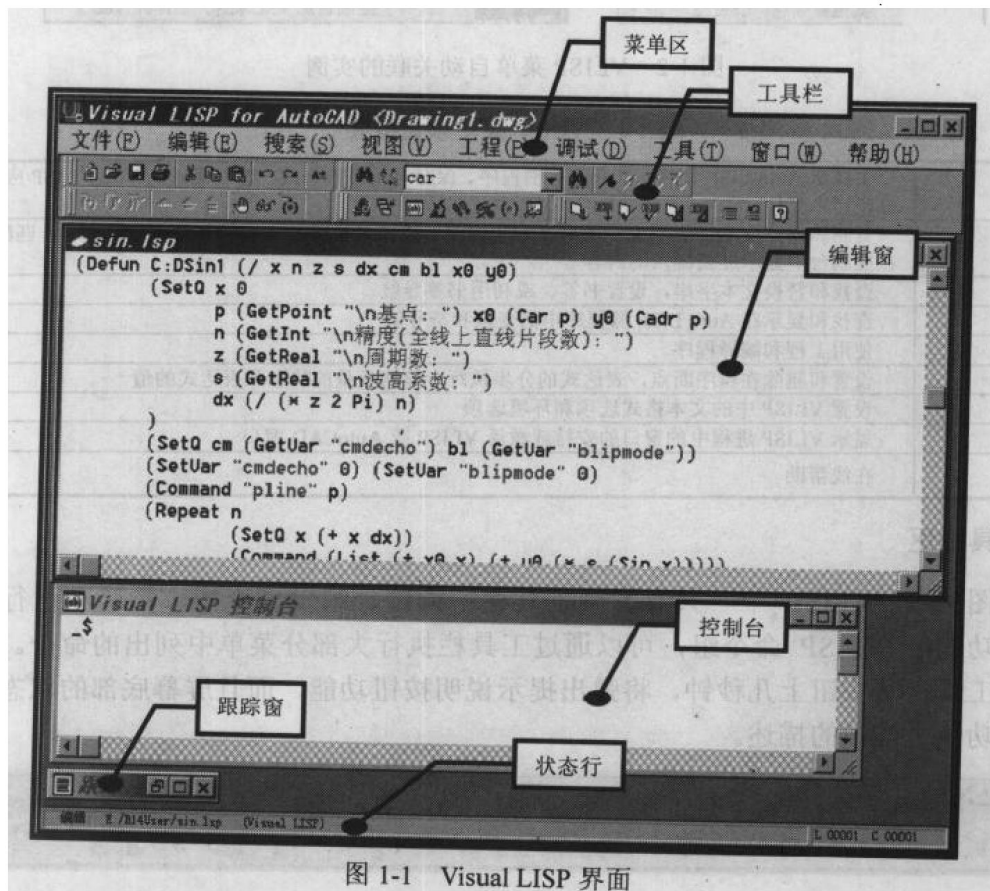


图 1-1 Visual LISP 界面

1.2.1 菜单区

菜单的操作和表达规则与 AutoCAD 相类似。选择了菜单上的某项，将在状态栏上显示相关功能的简短解释。与 AutoCAD 菜单不同的是，VLISP 的菜单是与当前的操作自动关联的智能化菜单，随着当前窗口和操作的变化，同一个菜单中的内容可能不同。VLISP 菜单不能由用户定制。

参见图 1-2，在编辑窗口下，点取【编辑(E)】菜单（左图）；再在控制台窗口下点取同一个【编辑(E)】菜单（右图），可见两者在【括号匹配】菜单项之后的内容有所不同。各菜单的功能参见表 1-1。

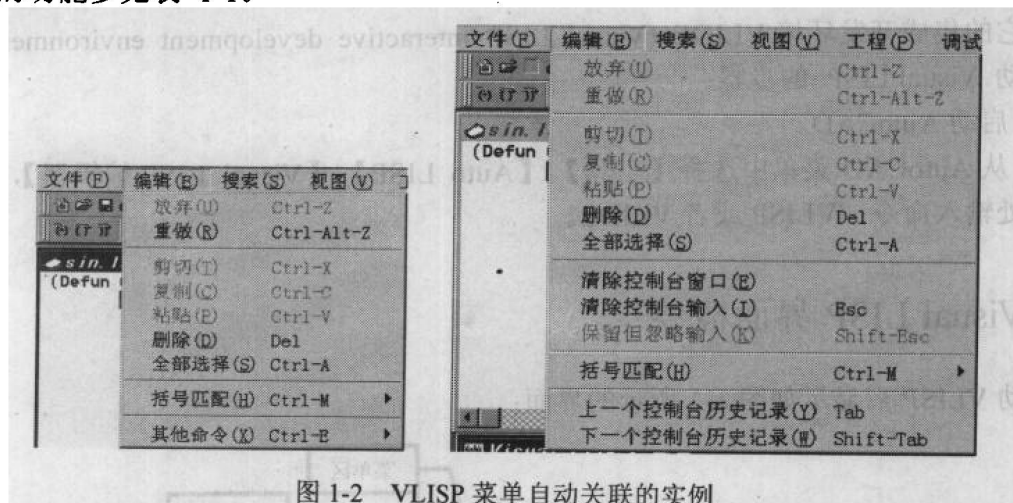


图 1-2 VLISP 菜单自动关联的实例

表 1-1 VLISP 菜单项目概要

文件	生成新 Auto LISP 程序、打开已有的程序、保存对程序的修改、建立编译的 Visual LISP 应用程序、打印程序文件
编辑	复制和粘贴文本、撤销上一次对文本的修改、选择编辑器的或控制台窗口中的文本、匹配括号、在控制台中显示过去键入的内容等
搜索	查找和替换文本字符串，设置书签、或利用书签导航
视图	查找和显示在 Auto LISP 源程序中的变量和符号值
工程	使用工程和编译程序
调试	设置和删除在程序断点，表达式的分步执行，检测变量的状态和表达式的值
工具	设置 VLISP 中的文本格式选项和环境选项
窗口	显示 VLISP 进程中的窗口的安排或激活 VLISP 或 AutoCAD 窗口
帮助	在线帮助

1.2.2 工具栏区

参见图 1-3。VLISP 中一共有五个工具栏：调试、编辑、查找、检验和运行，各自代表不同功能的 VLISP 命令组，可以通过工具栏执行大部分菜单中列出的命令。将鼠标指针停在工具栏某按钮上几秒钟，将弹出提示说明按钮功能，而且屏幕底部的状态栏上还会显示对功能更详细的描述。



图 1-3 VLISP 的工具栏

1.2.3 控制台窗口

控制台窗口是一个单独的，内容可以滚动的窗口。

在控制台窗口中，可以像在 AutoCAD 命令行那样输入 Auto LISP 表达式，也可以不用菜单或工具栏而直接在控制台窗口中发出很多 Visual LISP 命令。

在控制台窗口，可以输入和运行 Auto LISP 表达式并查看结果，这很像在 AutoCAD 命令窗口中所做的，但也有一些不同，例如，在 VLISP 中显示一个 Auto LISP 变量的当前值，只需在控制台中键入变量名，然后回车；而在 AutoCAD 命令窗口中观察变量值，就必须在变量名前加一个惊叹号。

在控制台窗口，将显示 Auto LISP 运行诊断信息和一些 Auto LISP 函数的结果。

下面是一些典型功能：

- 1) 可以执行表达式并显示表达式的返回值。
- 2) 可以输入较长的表达式，每行用 **Ctrl+Enter** 结束即可继续在下一行接着输入。
- 3) 可以复制或粘贴文本，在控制台窗口中可以使用大部分文本编辑命令。
- 4) 通过在控制台中按 **Tab** 键可以回溯到以前输入的命令。可以多次按 **Tab** 键来回溯到更早输入的命令，按 **Shift+Tab** 键则可以反向回溯命令。
- 5) 通过在控制台中按 **Tab** 键还可以实现对输入历史的关联搜索。例如，如果在输入了 (Se 之后按下 **Tab** 键，VLISP 将回溯到您最近输入的那个以 (Se 头的表达式。按 **Shift+Tab** 键则可以反向回溯。
- 6) 按 **Esc** 键可以清除在控制台提示下刚才输入的内容。
- 7) 按 **Shift+Esc** 键将跳过在控制台提示下输入的内容，出现新的控制台提示行。
- 8) 在控制台窗口的任何地方单击鼠标右键或按 **Shift+F10** 键将显示控制台弹出菜单。例如，可以用这个菜单功能复制和粘贴控制台命令行上的文本、查找文本和初始化 VLISP 调试功能。

注意：如果在控制台提示符下键入内容，回车之前又切换到了 AutoCAD 窗口，再次返回到 VLISP 窗口时，这些文本将不会保留在提示符下，而且不能用 **TAB** 回溯功能找到它们，因为这样的内容并没有被记下来。

比较遗憾的是，控制台窗口中显示的较长内容不会自动回行。例如：显示一个变量的值，内容是 1000 个元素的数组，要想看到后面的元素，只好用水平滚动条慢慢地找。

1.2.4 状态栏

位于屏幕底部的状态栏中所显示的信息因，将实时关联 VLISP 中所做的工作。

1.2.5 跟踪窗口

还会看到一个最小化的跟踪窗口。在启动时，该窗口会包含 VLISP 当前版本的信息，如果 VLISP 在启动时遇到错误，它还会包含相应的错误信息。

1.2.6 文本编辑器

用 VLISP，大部分时间是生成或修改源程序，VLISP 有专用的文本编辑器。每打开