

Visual LISP

程序设计 及其应用教程

编著 周 明
董仁扬

上海科学技术文献出版社

TP312/2840D

2008

Visual LISP

程序设计 及其应用教程

编著 周 明
董仁扬

上海科学技术文献出版社

图书在版编目(CIP)数据

Visual LISP程序设计及其应用教程/周明, 董仁扬编
著. —上海: 上海科学技术文献出版社, 2008. 3

ISBN 978-7-5439-3475-7

I. V ··· II. ①周··· ②董··· III. LISP表处理语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第020631号

责任编辑: 于学松

封面设计: 通 文

Visual LISP程序设计及其应用教程

编著 周 明 董仁扬

*

上海科学技术文献出版社出版发行
(上海市武康路2号 邮政编码200031)

全国新华书店经销

江苏常熟市人民印刷厂印刷

*

开本787X1092 1/16 印张18.5 字数416 000

2008年3月第1版 2008年3月第1次印刷

印数: 1-5 000

ISBN 978-7-5439-3475-7

定价: 38.00元

<http://www.sstlp.com>

前 言

计算机辅助设计(CAD)软件,大多是一些通用系统,建造了一个进行CAD的基本平台。然而设计领域是很宽广的,就拿机械设计中的工艺装备设计来说,就有刀具设计、量具设计、夹具设计和模具设计等;而刀具设计又有齿轮刀具、拉刀、铣刀、孔加工刀具等等,它们千差万别。这些专业设计的特点是:各专业有各专业的特点,且技术含量高;设计过程复杂,但每次设计都重复同一过程。这正是计算机适合做的工作,但目前很少有这种专业设计的软件供工程师们使用。有没有一种程序设计语言能让工程师自己编写自己常用的专业程序,甚至去开发自己常用设计的专业软件,以大大提高设计效果和设计质量呢?有的,Visual LISP 就是这样的语言。Visual LISP 是著名的二维 CAD 软件 AutoCAD 和三维 CAD 软件 MDT 内嵌的语言。它是人工智能语言 LISP 的一个子集,同时将 AutoCAD 和 MDT 的命令嵌入到语言中,成为编写 CAD 软件很方便的一种程序设计语言。Visual LISP 的前身是 AutoLISP,从 AutoCAD 2000 起发展成为 Visual LISP,成为有自己编程环境且能编译、打包便于开发较大软件的程序语言。它的特点是:第一,语法简单、易学易用;第二,编写绘图及相关程序简单,就像在 CAD 软件中调用命令绘图一样,这正是设计工程师的特长。

实践也证明了这一点,作者多年来一直用 AutoLISP 开发工装的专用程序和专业应用软件,例如:在 AutoCAD 上开发了设计齿轮刀具的“滚刀插齿刀 CAD 系统”软件,“成型铣刀 CAD”软件等一批刀具设计软件,在 MDT 中建立了航标夹具通用件库,编写了数控高速旋转刀具的静平衡应用程序等,取得了很好的效果。

本书是理论与实践相结合的经验总结,既全面讲述了 Visual LISP,也介绍了作者的编程经验。本书的许多程序是董仁扬老师提供的。

本书分三篇,共计十四章。

第一篇第四章。第一章编程环境,介绍 Visual LISP 编辑环境的界面和功

能,用那部分学那部分,初学者了解界面和常用的少量功能即可。

第一篇后三章,分别介绍 AutoLISP 的数据结构、常用函数及应用实例。学完本篇即可熟练编写 AutoLISP 应用程序。用它可以编写一些专用程序,可以提高设计质量和效率,减少许多重复劳动。

第二篇第六章。学完就能开发专业应用软件。

第五章,介绍对象处理,选择集处理等函数和对话框函数。

第六章,介绍如何编写对话框,在专业设计程序中主要用于参数输入。

第七章,专业软件离不开各种数据库及它们的检索功能,本章介绍图形数据库和非图形数据库的创建和检索技术。

第八章,专业软件也离不开操作界面——菜单,本章只介绍如何创建下拉菜单。

第九章,介绍两个较大型的例子,说明应用软件开发的一些技术。

第十章,介绍在三维机械设计软件 MDT 中如何编写 AutoLISP 程序。

第三篇四章。介绍 Visual LISP 扩展函数、反应器函数,以及 Visual LISP 工程 and 应用程序包的编译。给进一步提高编程技术的人员使用。

编 者

目 录

CONTENTS

第 一 篇

◇第一章 Visual LISP 的编程环境	3
1.1 Visual LISP 集成开发环境的界面	3
1.2 输入和修改程序代码	4
1.3 语法检查和语法检查工具	7
1.4 程序调试和调试工具	8
1.5 利用控制台窗口调试程序	16
◇第二章 AutoLISP 的数据与程序	18
2.1 AutoLISP 的数据结构和数据类型	18
2.2 AutoLISP 数据的值	22
2.3 AutoLISP 程序	22
◇第三章 AutoLISP 的常用函数	28
3.1 数学计算函数	28
3.2 几何函数、查询函数和命令函数	31
3.3 字符串函数	35
3.4 表操作函数	37
3.5 符号操作函数	43
3.6 转换函数	47
3.7 条件、循环和控制函数	50
3.8 自定义函数和命令	53
3.9 循环与递归	57
3.10 用户输入函数	58
3.11 文件处理函数	62
3.12 显示控制函数	65
◇第四章 AutoLISP 程序设计实例	69
4.1 定义内部函数	69
4.2 数组与用表表示数组	72
4.3 向量与矩阵计算函数	74

4.4 点的坐标及点的变换.....	81
4.5 实用函数.....	86
4.6 标注函数.....	89

第 二 篇

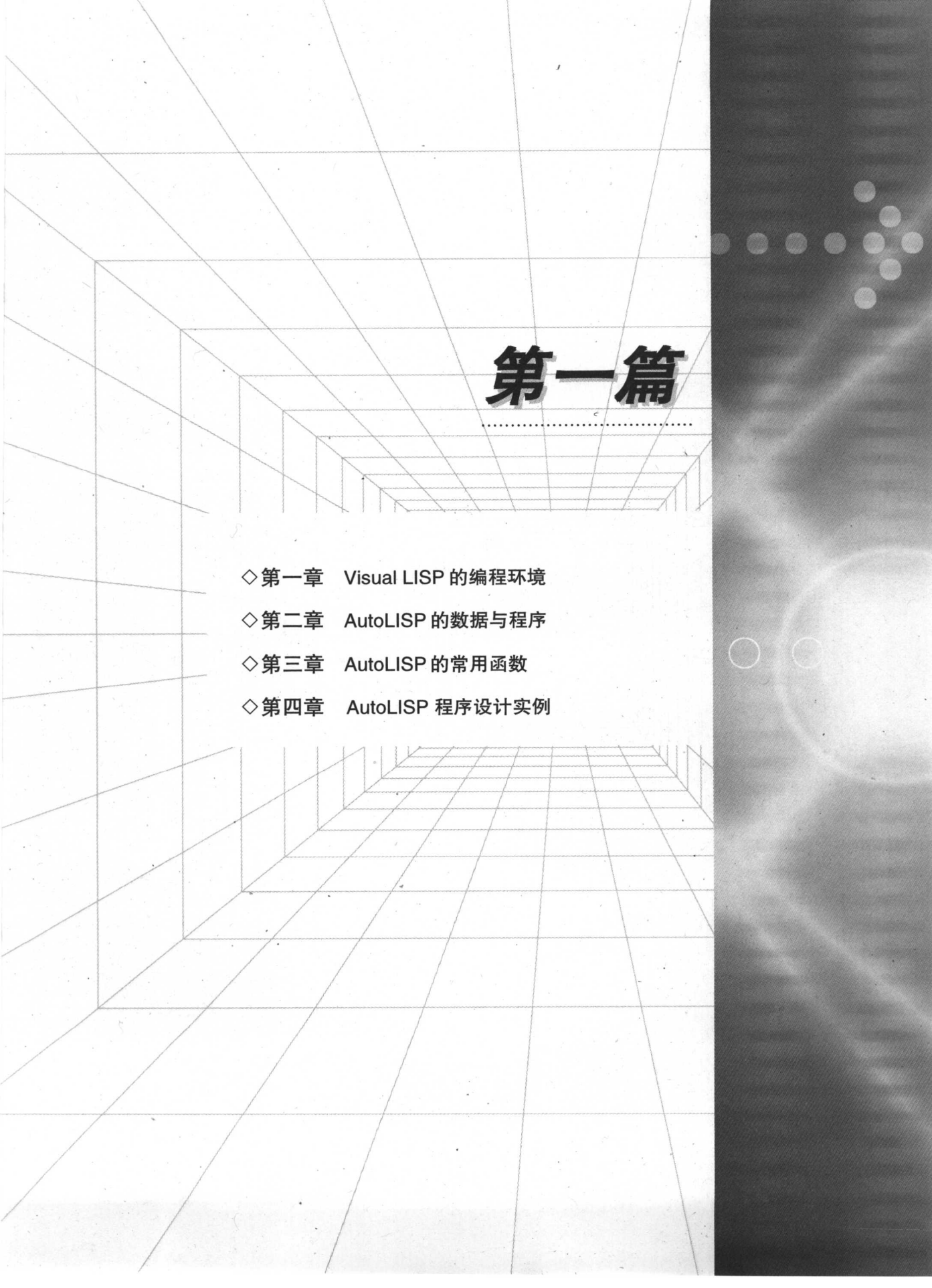
◇第五章 Visual LISP 函数补充.....	97
5.1 对象处理函数.....	97
5.2 符号表和词典处理函数	101
5.3 选择集处理函数	103
5.4 内存管理函数	107
5.5 应用程序管理函数、设备访问函数.....	108
5.6 VLX 名称空间函数	110
5.7 名称空间通讯函数	112
5.8 Windows 注册表函数	112
5.9 可编程对话框函数	113
5.10 错误处理函数.....	117
◇第六章 创建对话框.....	119
6.1 对话框语言的控件和属性	119
6.2 对话框设计	130
6.3 对话框的 AutoLISP 驱动程序	134
6.4 对话框设计实例及其驱动程序	143
◇第七章 图形数据库与非图形数据库.....	150
7.1 图形数据库	150
7.2 非图形数据库的方法及检索	157
7.3 数据文件及检索方法	163
7.4 用电子表格数据快速绘制曲线	168
◇第八章 自定义菜单.....	170
8.1 菜单文件	170
8.2 自定义用户界面	173

◇第九章 参数化图形程序设计举例.....	176
9.1 绘制台阶轴	176
9.2 剃前滚刀的可视化设计	182
9.3 可视化程序设计程序	187
◇第十章 三维机械设计软件 MDT 中 AutoLISP 的编程	197
10.1 在 MDT 中如何使用 AutoLISP	197
10.2 MDT 中的 AutoLISP 函数	199
10.3 MDT AutoLISP 函数的变量类型	200
10.4 应用实例.....	201

第 三 篇

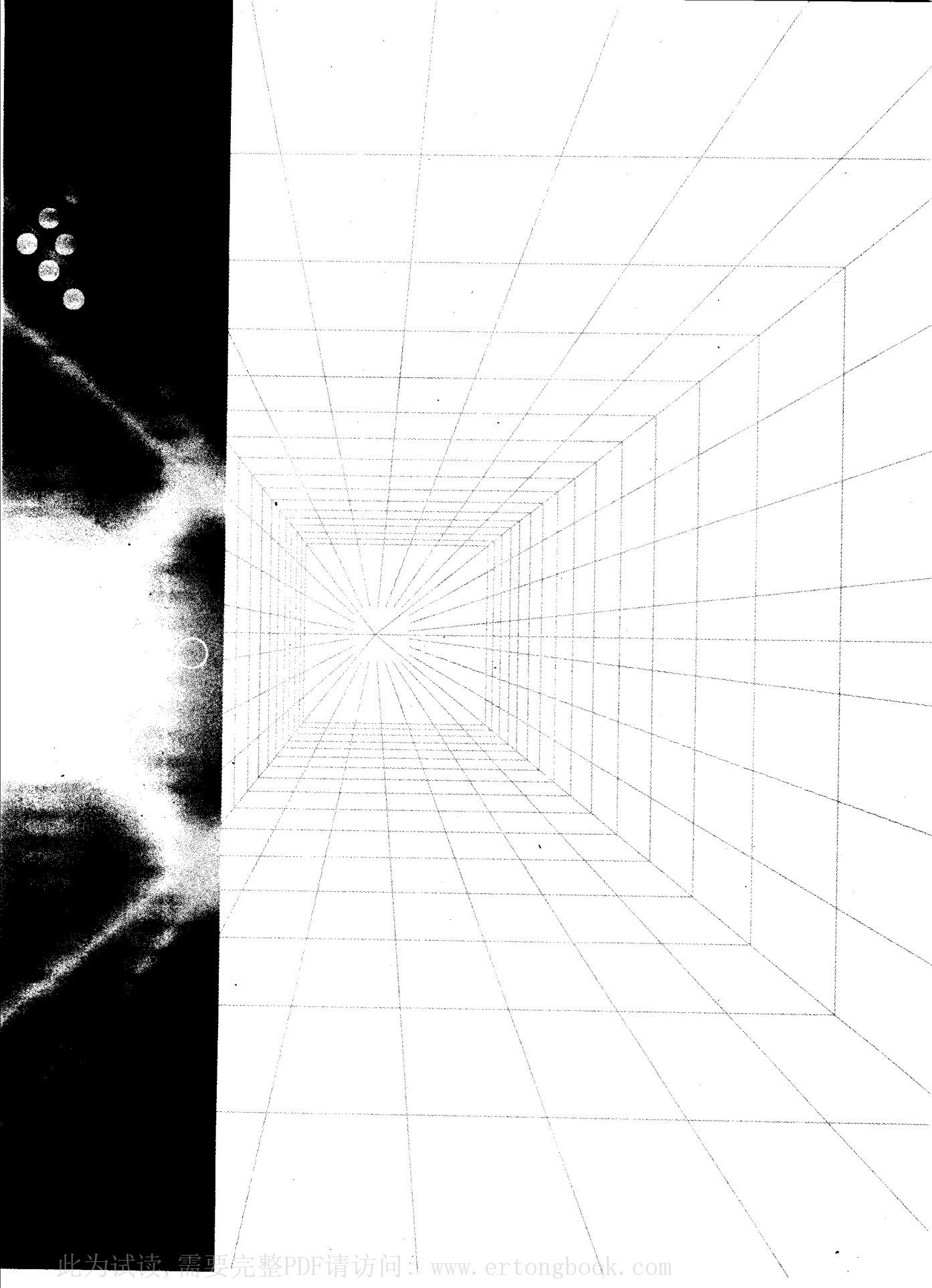
◇第十一章 Visual LISP 扩展函数	215
11.1 ActiveX 数据转换函数	215
11.2 ActiveX 方法调用函数	219
11.3 ActiveX 对象处理函数	219
11.4 特性处理函数.....	221
11.5 曲线测量函数.....	221
11.6 词典处理函数.....	223
11.7 对象处理函数.....	224
11.8 ActiveX 集合操作函数	226
11.9 其他.....	227
◇第十二章 反应器函数.....	228
12.1 反应器的概念.....	228
12.2 反应器函数.....	230
12.3 使用 Visual LISP 反应器	245
◇第十三章 Visual LISP 工程	248
13.1 管理多个 LISP 文件	248
13.2 定义工程.....	249
13.3 操作现有工程.....	252

◇第十四章 应用程序包的编译.....	256
14.1 编译和链接程序.....	256
14.2 创建应用程序模块——应用程序包.....	258
14.3 针对多文档环境的设计方法.....	263
附录 A ASCII 码转换表.....	269
附录 B DXF 组码	273
附录 C AutoCAD 2000 系统变量	278



第一篇

- ◇第一章 Visual LISP 的编程环境
- ◇第二章 AutoLISP 的数据与程序
- ◇第三章 AutoLISP 的常用函数
- ◇第四章 AutoLISP 程序设计实例



第一章 Visual LISP 的编程环境

AutoLISP 是集成在计算机辅助设计软件 AutoCAD 和 MDT 中,为扩展它们的功能和编写应用程序而设计的一种编程语言,它的起源可以追溯到 20 世纪 50 年代末的 LISP 语言。LISP 最初是为编写人工智能(AI)应用程序设计的,现在仍是许多人工智能程序的基础。LISP 之所以被选为 AutoCAD 最初的应用程序编程接口,是因为只有它才适合于工程中的非结构化设计过程。Visual LISP(简称 VLISP)对 AutoLISP 语言进行了扩展,并有一个独立的开发环境,本章介绍这个环境和它的功能,对初学者只了解界面和常用的少量功能即可。一般学习者也是用那部分学那部分。

用 VLISP 开发 AutoLISP 程序一般按以下步骤进行:明确应用程序要完成的任务以及如何完成这些任务;设计程序;编写源程序;设置程序格式以增强可读性;检查程序错误;测试和调试程序;成功后可进行编译。VLISP 提供了一个完整的集成开发环境,包括编辑器、调试器和其他工具,所提供的许多功能,使编写、修改代码以及测试和调试程序更加容易,是加速 AutoLISP 程序开发的软件工具。

1.1 Visual LISP 集成开发环境的界面

由 AutoCAD 的下拉菜单“工具”>“AutoLISP”>“Visual LISP 编辑器”将打开 VLISP 集成开发环境的界面,见图 1.1。其界面与 AutoCAD 的界面形式上类似。上边是“下拉菜单”,接下来是“工具栏”,最下面是“状态行”,中间是 3 个窗口。这 3 个窗口是:

文本编辑窗口,即文本编辑器;控制台窗口;跟踪窗口(启动时最小化显示)。图中显示了它们相应的快捷菜单,从中可以看出它们的一些功能。

下拉菜单,用户可以通过选取各个菜单项来发出 VLISP 命令。如果选中菜单上的某一项,将在屏幕底部的状态栏上显示相关命令功能的简介。菜单是与当前激活的窗口相关联的,激活不同的窗口,同一菜单内容也可能不同,且用户不能自定义菜单。

工具栏,可以快速发出 VLISP 命令。VLISP 中一共有 5 个工具栏:标准、视图、调试、工具和搜索,各自代表不同功能的 VLISP 命令组。可以通过工具栏执行部分(但不是全部)菜单命令。如果将鼠标指针指向工具栏某按钮并在其上停留几秒钟,将显示按钮代表的命令名称,且还会在状态栏上显示相关命令功能的简介。

文本编辑器是 VLISP 专用的文本编辑器,用以生成、修改和调试 VLISP 源程序。有许多专为支持 AutoLISP 编程而设计的功能,例如括号匹配、语法着色和在文字编辑器内执行 AutoLISP 表达式等。

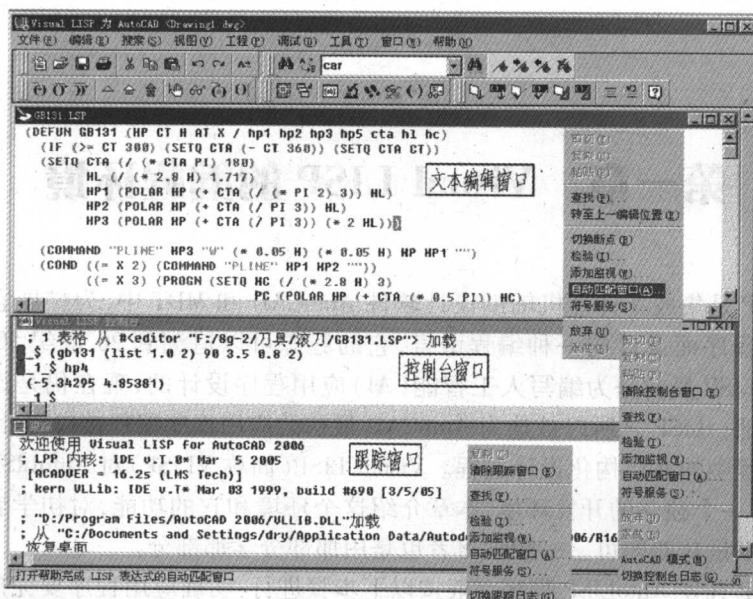


图 1.1 VLISP 集成开发环境

控制台窗口是 VLISP 主窗口中的一个独立的可滚动窗口。在控制台窗口中可以输入 AutoLISP 函数与命令,其运行方式与在 AutoCAD 命令窗口中类似,但也有所不同。

跟踪窗口,窗口显示当前 VLISP 版本信息,以及显示一些其他信息,例如,在跟踪命令模式下跟踪程序的执行过程,跟踪递归程序的迭代过程等。又如启动 VLISP 遇到错误时显示错误信息。

VLISP 还有一些其他窗口,例如跟踪 Trace 函数将打开一个新窗口并显示程序事件。虽然无法在这些输出窗口中输入文本,但可以从中复制文本,并将其粘贴到编辑器或控制台窗口中。

下面叙述它们在创建、编辑和调试 LISP 程序中非常有用的一些功能。再次强调用到那些功能,学习那些功能。

1.2 输入和修改程序代码

输入和修改程序主要使用 VLISP 文本编辑器并将 AutoLISP 代码保存到文件中。文本编辑器是 VLISP 的一个基本组成部分,它有许多专为支持 AutoLISP 编程而设计的功能,例如括号匹配、语法着色和在文字编辑器内执行 AutoLISP 表达式等,它非常易于使用。如果有使用 Windows 的经验就很容易掌握它。

1.2.1 移动、选择和备份文本

除了使用标准的 Windows 剪切、复制和粘贴功能外,VLISP 文字编辑器还允许用户将文本从文字编辑器窗口的某个位置拖到另一个位置。拖动文本的步骤:选取要移动的文本;光标移到选取文本内的任意位置,按下并不要松开鼠标左键;将文本拖到新位

置;松开鼠标按钮。若是移动并复制文本,不同的是在松开鼠标按钮之前按下 CTRL 键。

双击鼠标左键选取文本是一个很有用的方法。如果光标正好在某开括号之前,VLISP 将选取光标以后直到相匹配闭括号之前的所有文本;如果光标紧跟在某闭括号之后,VLISP 将选取光标之前直到相匹配开括号之后的所有文本。另外,选取文本后按 CTRL + SHIFT + [键将选取文本扩展到与已选取文本前面相匹配的左括号;按 CTRL + SHIFT +] 键将选取文本扩展到与已选取文本后面相匹配的右括号。这样,在输入代码的初期就少犯括号不匹配的错误。

还可以使用“改写”模式来插入文字,按 INSERT 键可以切换“改写”模式的开关状态。在“改写”模式下光标形状由垂直变为水平。

“打开”文件时将产生一个扩展名为“.lp”的备份文件,在“文件”菜单上选择“重新打开”,即可将编辑的文件恢复成它原有内容而放弃所做的所有修改。

1.2.2 搜索和文本书签

编辑 VLISP 程序常常需要搜索,搜索包含“查找”和“替换”,“替换”可以将查找到的文本替换为用户指定的文本字符串,它们在工具栏上的按钮为 。弹出如图 1.2 和图 1.3 的对话框。

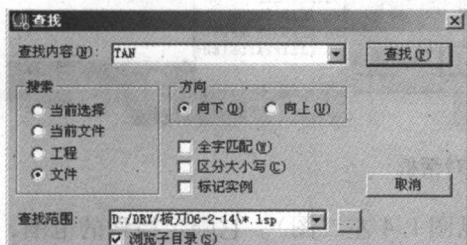


图 1.2 查找对话框

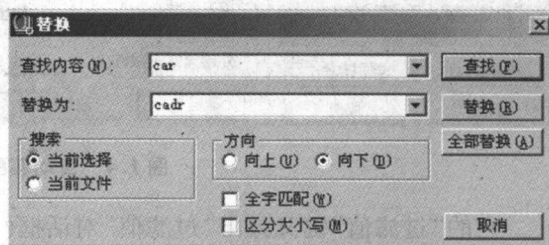


图 1.3 替换对话框

它提供了非常强大的文本查找功能,从如下选项可以看出:

“当前选择”仅查找文字编辑器窗口中亮显的文本。

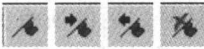
“当前文件”在活动编辑器窗口中的整个文件中查找。

“工程”选中该选项后,VLISP 将提示指定要搜索的 VLISP 工程名。它将搜索指定工程中所有的文件并在弹出的“输出窗口”中显示所有匹配。关于工程参见第十三章。

“文件”如果选中了该选项,VLISP 将让您指定要搜索文本所在的 Windows 目录(文件夹),即指定搜索范围,还可以让 VLISP 搜索该目录的所有子目录。VLISP 将搜索所有文件并在弹出的“输出窗口”中显示所有匹配字符串。

“标记实例”如果选中该选项,在每个找到的文本处都将加上“书签”。这样以后您就可以快速找到这些代码所在的位置。图 1.1“控制台窗口”左边两个竖立的长方形就是“书签”。

“书签”功能允许用户在每个窗口中添加多达 32 个,帮助用户更方便地浏览 VLISP

文字编辑器窗口中的文件。设置了 32 个后,再添加新的书签将删除最早的书签。按工具栏上的“切换书签”按钮,也能添加书签到光标所在的位置,再按“切换书签”按钮则删除。按“上一个书签”按钮和按“下一个书签”按钮,就可将光标跳到书签环中的上一个或下一个书签处。按“清除所有书签”按钮,将清除活动窗口中的所有书签。在工具栏上,这 4 个按钮分别是: .

1.2.3 自动匹配和完词功能

自动匹配功能是一种查找 VLISP 符号表的工具。符号表包括 AutoLISP 阅读器能读入的所有符号,其中包括用户程序中的符号和 AutoLISP 语言所用的符号。完词功能,可以只输入词的一部分,由 VLISP 帮您完成剩余部分。它们可以帮助用户快速、准确的输入程序代码。

在工具栏上,“自动匹配”按钮是:  弹出以下对话框(见图 1.4 左图)。

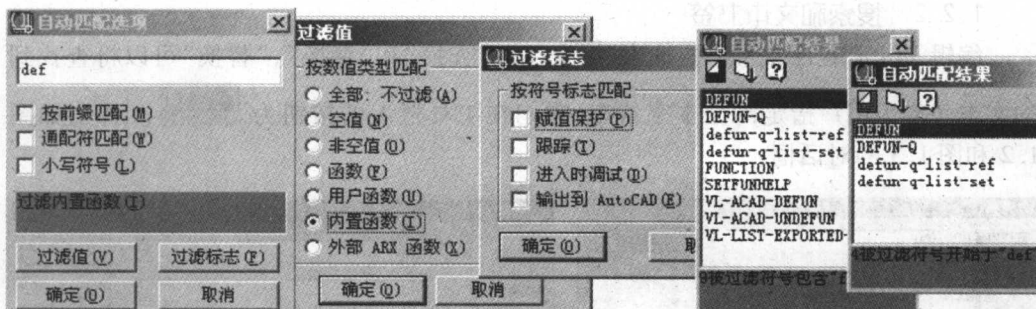


图 1.4 自动匹配对话框

它的“过滤值”键又弹出“过滤值”对话框(见图 1.4 左二图)。它限定检索的范围,例如选定“内置函数”则只在 VLISP 函数内匹配,图 1.4 右二图是自动匹配结果;又如选定“空值”则仅匹配值为 nil 的符号。它的“过滤标志”键弹出“过滤标志”对话框(见图 1.4 左三图)。它也限定检索的范围。在文本编辑器和控制台窗口都适用。



图 1.5 按系统匹配

“自动匹配选项”对话框中还有 3 个过滤选项:“按前缀匹配”;“通配符匹配”;“小写符号”进一步缩小检索的范围,图 1.4 右图就是再选“按前缀匹配”后自动匹配结果。这样在记不清函数名的时候,容易将正确的函数名找出,再把它复制到需要的地方。

完词功能有两种,即按系统匹配和按历史匹配。

“按系统匹配”(见图 1.5)在文本编辑器或控制窗口,输入不完整的表达式如“(se”,选取“se”按“Ctrl”加“Shife”加“空格”键,则匹配成第一个函数“set”;若不是需要的函数,再按“Ctrl”加“Shife”加“空格”键,则弹出显示所有的全

部函数的符号表,从中选择需要的函数例如“setq”,则不完整的函数匹配成“(setq”。

若符号表超过 15 个符号则显示前面讲的“自动匹配结果”对话框。

“按历史匹配”,曾经输入过的符号记录在符号表中。输入符号后,下拉菜单中选择“搜索”>“按历史匹配”来调用该功能,就能由后向前匹配曾经输入过的符号。

1.3 语法检查和语法检查工具

在运行程序之前检测程序的语法错误,VLISP 提供了许多功能,使我们在程序运行前就将一些错误排除掉。

1.3.1 语法分色和检查

VLISP 文本编辑器能识别内部函数、字符串、整数、实数和注释,并用不同颜色显示。默认的语法分色,其颜色设置为:内部函数和保护字符为蓝色;字符串为粉色;整数为绿色;实数为青色;注释为灰底粉字;未知项如自变量为黑色。它们可以帮助发现错误,如输入的内部函数不是蓝色,则输入有误。能分色的程序不只是 VLISP,还有 DCL、SQL 和 C++。语法分色的颜色由下拉菜单“工具”>“窗口属性”>“按语法着色...”弹出的对话框设置,一般取默认的语法分色。

还可以用 VLISP 检查命令做其他语法检查,检查命令可以检测如下错误:给函数提供的参数数目不正确;给函数提供的参数不正确;某些特殊函数(例如 lambda、setq 和 foreach 等)的语法不正确。由下拉菜单“工具”>“检查编辑器中文字”或“检查选定文字”,工具栏检查按钮为  ,后者只检查部分程序。图 1.6 是检查的结果,它弹出一个“编译输出”窗口,在其中显示检查出的错误。某些语法错误只能在程序运行时才能被检查出来,语法检查命令无法检测出这些错误。例如,在调用一个函数时,它需要整数参数,而您提供的是字符串,AutoLISP 在运行前无法检测出这种错误。所以,只有在程序运行时才能检测出这种错误。

1.3.2 程序格式化和括号匹配

程序格式化和括号匹配是另一个检查语法错误方法。

从图 1.1 文本编辑窗口中的程序可以看出,AutoLISP 程序是由众多配对的圆括号组成的,只要一对不配对,程序将出错,是最常出现的语法错误。而找出哪对括号不配对,十分伤脑筋。VLISP 文本编辑器从两方面圆满地解决了这个问题,一方面是程序的自动格式化使少犯括号不配对的错误;另一方面出了错有检查括号匹配方法。

程序自动格式化,从图 1.1 和图 1.6 文本编辑窗口中的程序已经格式化,可以看出

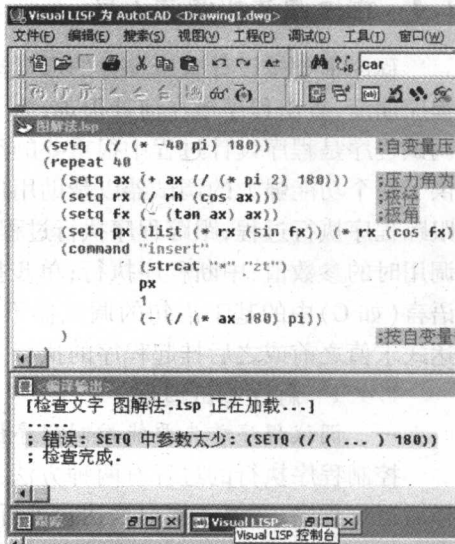


图 1.6 错误检查

程序排列规范可读性很好。如何实现?其一,手工操作,在适当位置回车则自动缩进,并使程序的配对圆括号对齐。其二,自动,由下拉菜单“工具”>“设置编辑器中代码的格式”或“设置选定代码格式”操作,工具栏操作按钮为  ,前者使整个程序自动格式化,后者只格式化部分程序。格式化的格式可自行设置,由下拉菜单“工具”>“环境选项”>“Visual LISP 格式选项…”弹出的对话框操作,一般选默认格式即可。

括号匹配,也有两个简单办法:其一是匹配;其二是选择。匹配:将光标放在左圆括号处按“Ctrl”加“)”键则向后匹配,操作后光标的位置就是配对右括号的位置;将光标放在右圆括号处按“Ctrl”加“(”键则向前匹配,操作后光标的位置就是配对左括号的位置。选择:前面已讲到将光标放在左圆括号处,双击鼠标左键,则向后选择与它配对右括号间的一段程序,当然配对的右括号也找到了;将光标放在右圆括号处,双击鼠标左键,则向前选择与它配对左括号间的一段程序,当然配对的左括号也找到了。它们也可由下拉菜单“编辑”>“括号匹配”中的选项操作。

1.4 程序调试和调试工具

程序很难按设计者所预计的那样运行。若程序运行所得的结果不对,或者引起了程序崩溃,查出程序错在何处很难,为了查错,就需要调试程序、查找并改正程序错误。调试程序是程序设计过程中很重要的环节,且是程序开发中最费时间的过程,VLISP 提供了一个功能强大的调试器以帮助用户调试程序,加速调试过程。调试器的功能包括:跟踪程序执行过程;跟踪程序执行过程中的变量值;查看表达式的求值顺序;检验函数调用时的参数值;中断程序执行;单步执行程序;检验堆栈。VLISP 中的调试器和其他语言(如 C)中的基于语句的调试器不同,它是基于表达式的。这样,调试器可以在表达式求值之前或之后挂起程序的执行。

1.4.1 程序调试过程

一、调试程序首先要能控制程序执行的过程,在控制的基础才能进行调试

控制程序执行的过程有两种方法:一是选择下拉菜单中“调试”>“立即停止”,选择后运行程序,在第一次对表达式进行求值时将使程序暂停,暂停后可以采用各种调试命令使程序继续执行;另一种方法是设置“断点”,设置断点后,遇着断点中断程序执行,中断后可以采用各种调试命令使程序继续执行。当程序暂停时,相应的 VLISP 文字编辑器窗口会亮显程序暂停时所处的当前表达式。此时控制台窗口中会出现一个中断循环标志。利用控制台窗口,可以访问和操纵断点处程序所处的环境,也可以通过“监视”窗口检查变量的值。

二、调试有两种模式:“单步调试”和“自动执行”

“单步调试”允许用户一次只执行一个或几个表达式来达到跟踪、调试程序,正在求值的表达式将亮显。“自动执行”可以让 VLISP 自动单步执行程序并计算表达式的过程,就好像不断发出执行“下一表达式”命令,这时文字编辑器窗口将亮显正在求值的表达式,而“监视”窗口中的数据也不断更新。用下拉菜单中“调试”>“自动执行”的选项来切换。