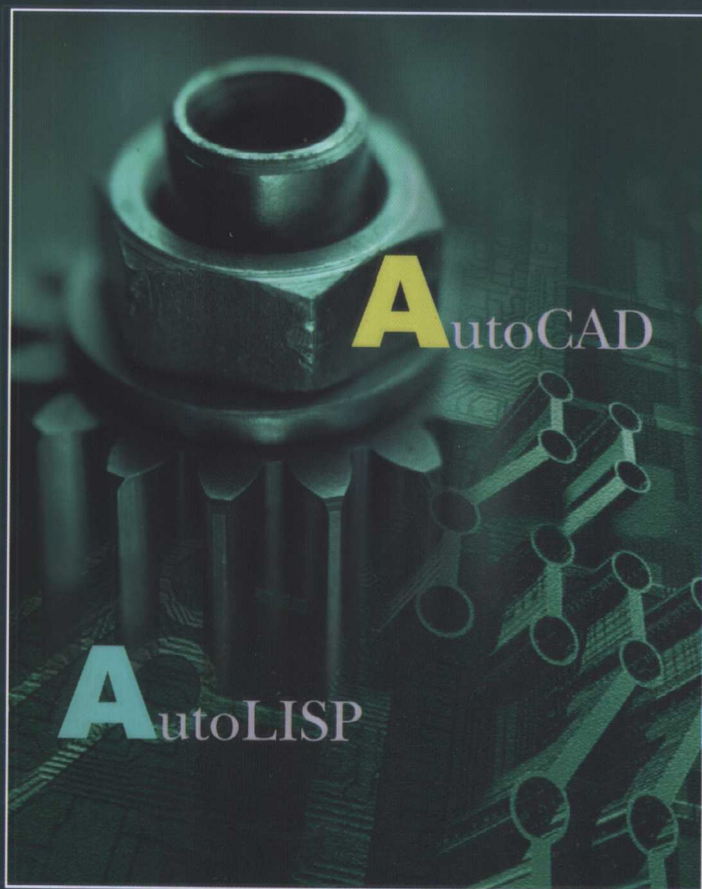


AutoCAD 2004与AutoLISP 二次开发技术

赵景亮 李志刚 等 编著



清华大学出版社

AutoCAD 2004与AutoLISP

二次开发技术

清华大学出版社 北京 2004年12月第1版第1次印刷 16开 256页 32.00元

ISBN 7-302-12400-0



本书主要介绍AutoCAD 2004二次开发技术，包括AutoCAD 2004二次开发环境、AutoCAD 2004二次开发方法、AutoCAD 2004二次开发应用等。本书可作为AutoCAD 2004二次开发人员的参考书，也可作为AutoCAD 2004二次开发课程的教材。

清华大学出版社
Tsinghua University Press
www.tup.tsinghua.edu.cn

AutoCAD 2004 与 AutoLISP 二次开发技术

赵景亮 李志刚 等 编著

清华大学出版社
北 京

内 容 简 介

本书介绍了最新版本的 AutoCAD 2004 环境下的 AutoLISP 二次开发技术。全书共分 10 章,内容涉及 AutoLISP 语言的基础知识(数据类型、程序结构、数据存储方式等)、Visual LISP 集成开发环境的使用以及编写 AutoLISP 程序的技巧,另外,还介绍了 ActiveX 对象及 AutoCAD 反应器方面的知识。

本书适用于具有一定 AutoCAD 应用经验,并且熟悉 Windows 系统操作的用户。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

AutoCAD 2004 与 AutoLISP 二次开发技术/赵景亮,李志刚等编著. —北京:清华大学出版社, 2004.4
ISBN 7-302-08178-6

I .A… II .①赵… ②李… III .计算机辅助设计—应用软件, AutoCAD 2004, AutoLISP IV .TP391.72

中国版本图书馆 CIP 数据核字(2004)第 014382 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

责任编辑: 冯志强

封面设计: 付剑飞

印 刷 者: 北京市清华园胶印厂

装 订 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 19 字数: 472 千字

版 次: 2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号: ISBN 7-302-08178-6/TP·5906

印 数: 1~4000

定 价: 28.00 元

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

前 言

AutoCAD 是美国 Autodesk 公司开发的通用计算机辅助绘图与设计软件包,具有易于掌握、使用方便、体系开放等特点。自 1982 年问世以来,已经进行了多次升级,从而使其功能日趋完善,而且已广泛应用于机械、建筑、电子、航天、土木工程、纺织、轻工等各个领域。

AutoLISP 语言是 AutoCAD 中用于二次开发的工具,已经成为计算机辅助设计领域二次开发的标准。Visual LISP(VLISP)作为帮助使用 AutoLISP 语言进行程序开发的一个集成开发环境,不仅提供了许多工具来帮助编写、调试以及编译 AutoLISP 程序,而且还提供了一些新的功能,如发布用 AutoLISP 语言编制的 ObjectARX 应用程序等。另外,AutoCAD 在版本的更新过程中也不断提供对 Visual LISP 更多的功能支持。

AutoCAD 2004 的新特性

2003 年,Autodesk 公司发布了 AutoCAD 系列软件的最新版本——AutoCAD 2004,此软件是理想的二维制图和三维设计工具,它不仅有强大的图形设计功能,而且用户界面更加友好,更容易学习和使用。和以前的版本相比,该软件具有以下特性:

- 高效的 DWG 存储格式

AutoCAD 2004 对 DWG 格式的文件进行了优化,比运用旧版软件创建的文件小 52%,因此,无论图形存储在本地还是网络驱动器中,都可以节省访问图形的时间。

- 功能强大的工具面板

AutoCAD 2004 中提供了新的工具面板。这些工具面板的透明度可以调整,能够增大屏幕工作区域,并且可以充分进行定制。例如插入图块的操作,只需从工具面板将图块拖入图纸即可,而不必使用插入命令。利用包括图层管理、尺寸标注和对象修改的 Express Tools 工具可以减少完成工作所需的步骤。

更新了“重做”功能,可以跟踪修改历史,恢复多次“撤消”操作。而且,第一个 AUGI 希望列表项目(多行文本)已经包括定位点和缩排功能,另外,删除了“文本编辑”对话框。

- 更强的色彩支持

AutoCAD 2004 中提供用于制作演示图纸的高质量图形,并提供新的色彩支持。在两种颜色或同一颜色的明暗色彩之间用指定的梯度填充,还可以运用描影 Viewport 出图功能打印演示质量的描影、三维等视图。而且,通过 1600 多万种可供选择的 24 位真色彩和包括 PANTONE(r)、RAL CLASSIC、RAL DESIGN 的颜色系统库,用户可以向 AutoCAD 对象应用自己想要的颜色。

- 安全共享数据功能

AutoCAD 2004 通过新的密码保护、数字特征文件和增强的 DWF 文件格式(Web 图形格式),可以使用户安全地共享数字设计数据。用户可以使用密码来保护自己的文件,并且可以利用与墨水签字功效相同的数字特征文件来验证图纸的来源、真实性和未修改状态。

用户使用 DWF 文件,可以通过因特网与需要查看和出图而不用编辑 AutoCAD 图纸的团

队成员交换图档文件。DWF 提供仅支持查看和出图的锁定轻型格式,可生成与 DWG 文件相同的按比例视觉保真度。而且,可以将多幅图纸发布为单一 DWF 文件,以简化传输。

- 更方便地共享数据

在 AutoCAD 2004 中,用户可以从更新的 AutoCAD DesignCenter(tm)中直接将现有的设计内容(如图块、标准、布局甚至整个 DWG 文件)拖入图纸;也可以使用新的选项卡访问 DesignCenter Online(访问大量预绘制 i-drop 内容的接入点),这样,用户可以简单地从 autodesk.com 或其他厂商的网站中将内容拖入自己的制图会话,而不用下载、保存和插入等操作。

- 自动标准检查

运用 AutoCAD 2004,用户可以检查标准。在进行制图时,Standards Manager 将会自动在后台运行,当用户违反或遗漏某个标准时,它会自动显示一条消息,通知用户违反了标准并提出纠正建议。而且,用户可以选择要检查的标准类型(如尺寸和文本样式、图层或线型)并通过电子邮件发送或打印标准核查报告。

Visual LISP 的特性

Visual LISP 作为一个集成开发环境,使编写、修改代码以及测试和调试程序更加方便,它主要提供以下功能:

- 语法检查器。该检查器可识别 AutoLISP 的语法错误和调用内置函数时的参数错误。
- 文件编译器。改善了程序的执行速度,并提供安全高效的程序发布平台。
- 专为 AutoLISP 设计的源代码调试器。利用它可以在窗口中单步调试 AutoLISP 源代码,并同时在 AutoCAD 图形窗口中显示代码运行结果。
- 文本编辑器。可采用 AutoLISP 及 DCL(可编程控制对话框)语法着色,并提供其他 AutoLISP 语法支持功能。
- AutoLISP 格式编排程序。用于设置程序格式,以改善程序的可读性。
- 全面的检验和监视功能。可以方便地访问变量和表达式的值,以便浏览和修改数据结构。这些功能还可以用来浏览 AutoLISP 数据和 AutoCAD 图形的图元。
- 上下文相关帮助。提供 AutoLISP 函数的信息,强大的自动匹配功能方便了符号名查找等操作。
- 工程管理系统。使维护多文件应用程序更加容易。
- 可将编译后的 AutoLISP 文件打包成单个模块。
- 桌面保存和恢复能力。可保存和重用任意 Visual LISP 任务的窗口环境。
- 智能化控制窗口。它给 AutoLISP 用户提供了极大的方便,大大提高了用户的工作效率。控制台的基本功能与 AutoCAD 的文本窗口类似,并提供了许多交互功能。例如,历史记录功能、完整的行编辑功能等。

Visual LISP 具有一个交互式的智能控制台,包含有一个项目窗口、代码分色的文本编辑器、调试器、源程序窗口及许多其他的特性。

本书内容结构

本书介绍了 AutoLISP 语言的基础知识(数据类型、程序结构、数据存储方式等)、Visual LISP 集成开发环境的使用以及编写 AutoLISP 程序的一些技巧,另外,还介绍了 ActiveX 对象

及 AutoCAD 反应器方面的知识。

本书适用于具有一定 AutoCAD 操作经验,并且熟悉 Windows 系统操作的用户。本书并不要求用户具有使用 LISP 或者 AutoLISP 语言编程的经验,当然,具有这方面编程经验的读者将会更容易掌握本书的内容。

本书按照介绍内容的不同,可大致分为以下几个部分:

- AutoLISP 语言基础,介绍 AutoLISP 的数据类型、存储方式以及程序流程控制方面的内容。
- AutoLISP 函数,按功能的不同分类介绍 AutoLISP 程序中常用的基本函数。
- Visual LISP 集成开发环境,介绍如何使用 Visual LISP 来开发、编译、调试 AutoLISP 程序以及如何使用工程来管理多个程序文件。
- AutoCAD 对话框,介绍如何进行对话框设计以及在 AutoLISP 程序中应用对话框。
- ActiveX 对象,介绍了应用 ActiveX 对象的基本知识以及如何使用 AutoLISP 程序来实现 ActiveX 方法。
- 反应器,介绍了反应器的基本知识以及如何在 AutoCAD 中使用反应器。

参与本书编写工作的还有:霍军军、宁可、刘昕、周旋、王青、高新波、王大智、龚瑞男、富晓静、周剑、徐大丰、李清。

目 录

第 1 章 AutoLISP 语言概述	1
1.1 AutoLISP 语言的出现与发展	1
1.2 AutoLISP 语言的特点	2
1.3 简单的 AutoLISP 程序介绍	3
1.4 AutoLISP 程序的执行	6
1.5 AutoLISP 程序的内存分配	7
1.6 本章小结	8
1.7 习题	8
第 2 章 数据类型、运算符与表达式	9
2.1 AutoLISP 数据类型	9
2.1.1 整型	9
2.1.2 实型	9
2.1.3 字符串	10
2.1.4 表	11
2.1.5 文件描述符	11
2.1.6 实体名	11
2.1.7 选择集	12
2.2 AutoLISP 数据存储结构	12
2.2.1 节点	12
2.2.2 表	13
2.2.3 点对	13
2.3 AutoLISP 变量	14
2.3.1 变量的命名	14
2.3.2 变量的数据类型	15
2.3.3 变量的赋值	15
2.3.4 预定义变量	16
2.4 数据类型的转换	17
2.5 AutoLISP 的程序控制	20
2.5.1 逻辑函数	20
2.5.2 关系函数	21
2.5.3 条件与循环函数	24
2.6 本章小结	28
2.7 习题	28

第 3 章 Visual Lisp 集成开发环境	29
3.1 Visual Lisp 简介	29
3.2 启动 Visual Lisp	30
3.3 Visual Lisp 界面介绍	30
3.3.1 菜单区	31
3.3.2 工具栏区	32
3.3.3 控制台窗口	32
3.3.4 跟踪窗口	33
3.3.5 状态栏	33
3.3.6 文本编辑器	33
3.3.7 Visual Lisp 的其他窗口	33
3.4 加载和运行 AutoLISP 程序	34
3.4.1 打开一个 AutoLISP 程序	34
3.4.2 加载并运行程序片断	34
3.4.3 加载并运行整个程序	35
3.5 Visual Lisp 文本编辑器	36
3.5.1 文本编辑器概要	36
3.5.2 程序代码格式化	36
3.5.3 Visual Lisp 的注释	37
3.5.4 语法结构分色和匹配	38
3.5.5 语法检查	42
3.5.6 文本编辑器的快捷键	43
3.6 退出 Visual Lisp	44
3.7 本章小结	44
3.8 习题	44
第 4 章 AutoLISP 函数	45
4.1 基本函数	45
4.1.1 数学函数	45
4.1.2 字符串处理函数	48
4.1.3 表处理函数	50
4.1.4 符号操作函数	54
4.1.5 函数处理函数	56
4.1.6 出错处理函数	60
4.1.7 应用程序管理函数	63
4.2 实用函数	65
4.2.1 几何函数	65
4.2.2 用户输入函数	68
4.2.3 内存管理函数	76

4.2.4	文件处理函数	78
4.2.5	设备访问函数	81
4.3	显示控制及命令函数	84
4.3.1	提示信息函数	84
4.3.2	菜单控制函数	87
4.3.3	图形窗口和文本窗口控制函数	89
4.3.4	Command 函数	93
4.3.5	存取系统信息函数	95
4.4	本章小结	98
4.5	习题	98
第 5 章	调试 Visual LISP 程序	100
5.1	Visual LISP 调试功能简介	100
5.2	调试器的使用	100
5.2.1	进入调试模式	102
5.2.2	单步调试	104
5.2.3	自动执行模式	104
5.3	断点的设置	105
5.3.1	程序中断的基本概念	105
5.3.2	可继续中断循环的调试	106
5.3.3	不可继续中断循环	106
5.3.4	断点操作	107
5.4	数据检验	109
5.4.1	跟踪日志文件	109
5.4.2	使用监视窗口	110
5.4.3	使用跟踪堆栈窗口	111
5.4.4	使用“符号服务”对话框	116
5.4.5	使用检验器	117
5.4.6	访问 AutoCAD 对象	122
5.5	本章小结	126
5.6	习题	126
第 6 章	编译及维护 AutoLISP 程序	127
6.1	编译链接程序	127
6.1.1	Visual LISP 编译器	128
6.1.2	加载运行已编译程序	129
6.1.3	链接函数调用	130
6.2	生成应用程序	130
6.2.1	创建新应用程序	131
6.2.2	更新及修改应用程序选项	135

6.3	多文档环境下的程序设计	136
6.3.1	命名空间	136
6.3.2	应用程序的独立命名空间	138
6.3.3	使用图形文档可以访问非 C:定义的函数	138
6.3.4	在其他 VLX 程序中调用独立命名空间函数	139
6.3.5	使用图形文档命名空间中的变量	140
6.3.6	在命名空间中共享数据	140
6.3.7	在多图形文档环境中的自定义错误处理	141
6.4	管理多个 Visual LISP 文件	142
6.5	使用 Visual LISP 工程	143
6.5.1	新建一个工程	143
6.5.2	向工程中添加源程序文件	144
6.5.3	选择编译器选项	144
6.5.4	工程窗口	145
6.6	操作工程文件	146
6.6.1	在工程中查找源文件的内容	146
6.6.2	由工程生成应用程序	147
6.7	优化应用程序	148
6.7.1	优化由工程生成的应用程序	148
6.7.2	选择编译模式	149
6.7.3	选择链接模式	150
6.7.4	安全优化	150
6.8	本章小结	151
6.9	习题	151
第 7 章	AutoLISP 程序设计	152
7.1	AutoLISP 应用程序开发概述	152
7.1.1	AutoLISP 与 AutoCAD 应用开发	152
7.1.2	应用程序开发的一般步骤及质量要求	153
7.2	扩展 AutoLISP 函数功能	155
7.2.1	利用表模拟数组功能	155
7.2.2	对表及表中元素进行排序	157
7.2.3	利用表进行文件读写处理	161
7.3	用 AutoLISP 进行参数化绘图	163
7.3.1	参数化绘图的原理与一般步骤	163
7.3.2	实例:标注序号程序	165
7.3.3	参数化绘制三维图形	167
7.3.4	参数化绘图时的注意事项	169
7.4	AutoCAD 实体数据库操作	170

7.4.1	创建选择集	171
7.4.2	过滤选择集	171
7.4.3	选择集操作	174
7.5	AutoCAD 对象操作	174
7.5.1	图元名称操作	175
7.5.2	图元数据处理	176
7.5.3	非图形对象的处理	181
7.5.4	操作 AutoCAD 对象实例	184
7.6	AutoCAD 扩展实体数据应用	186
7.6.1	扩展实体数据的结构	186
7.6.2	应用名的注册	187
7.6.3	获取及修改扩展数据	188
7.6.4	扩展数据中的句柄	190
7.6.5	扩展数据的内存管理	190
7.7	Xrecord 对象	191
7.8	自动加载 LISP 程序	191
7.9	本章小结	193
7.10	习题	193
第 8 章	AutoCAD 对话框设计	194
8.1	AutoCAD 对话框概述	194
8.2	对话框的组成	195
8.2.1	预定义活动控件	196
8.2.2	控件组	198
8.2.3	用于修饰和说明的控件	198
8.3	控件的属性	199
8.3.1	预定义标准属性	199
8.3.2	关键字和值属性	201
8.3.3	功能属性	201
8.3.4	布局属性和尺寸属性	202
8.3.5	预定义控件属性	203
8.4	对话框设计原则与规范	206
8.4.1	通用准则	206
8.4.2	通用规范	207
8.4.3	预定义控件的使用规范	208
8.5	对话框控制语言	210
8.5.1	DCL 文件结构	210
8.5.2	DCL 语法	211
8.5.3	对话框预览及出错处理	213

8.5.4	对话框布局	215
8.6	管理对话框	219
8.6.1	概述	219
8.6.2	对话框打开时限制使用的函数	220
8.6.3	动作表达式与回调	221
8.6.4	控件处理	223
8.6.5	嵌套对话框	226
8.6.6	隐藏对话框	227
8.6.7	列表框和弹出式列表框	230
8.6.8	图像控件和图像按钮	232
8.6.9	特定应用数据	235
8.7	对话框管理应用实例	235
8.7.1	绘图实例	235
8.7.2	行差标注实例	237
8.8	本章小结	242
8.9	习题	242
第 9 章	Visual LISP 中应用 ActiveX 对象	243
9.1	ActiveX 对象简述	243
9.2	AutoCAD 对象模型	244
9.2.1	对象的属性及方法	245
9.2.2	AutoCAD 2004 中变更的对象及方法	246
9.3	ActiveX 对象访问	248
9.3.1	查看对象特性	248
9.3.2	访问图形对象	249
9.4	用 AutoLISP 实现 ActiveX 方法	252
9.4.1	查找所需要的函数	252
9.4.2	确定函数的参数	253
9.4.3	转换数据类型为 ActiveX 型	254
9.4.4	转换其他数据类型	259
9.4.5	访问与更新图形对象	259
9.4.6	确定方法或属性是否适用于特定对象	260
9.4.7	使用参数带返回值的 ActiveX 方法	261
9.4.8	使用集合对象	262
9.4.9	释放对象及释放内存	264
9.4.10	转换对象引用	264
9.4.11	ActiveX 方法出错的处理	266
9.5	使用 ActiveX 与其他应用程序交互	268
9.5.1	引入一个类型库	268

9.5.2	与应用程序建立连接	270
9.5.3	应用程序实例	270
9.5.4	在不引入类型库时使用 ActiveX 方法	272
9.6	本章小结	273
9.7	习题	274
第 10 章	使用反应器	275
10.1	反应器基础	275
10.1.1	反应器类型	275
10.1.2	反应器的回调事件	277
10.1.3	反应器的回调函数	278
10.2	生成反应器	279
10.2.1	生成对象反应器	280
10.2.2	向反应器添加数据	281
10.3	在多命名空间中使用反应器	282
10.4	查询、修改与删除反应器	283
10.4.1	检验反应器	283
10.4.2	使用函数查询反应器	284
10.4.3	修改反应器	284
10.4.4	禁用反应器	286
10.5	临时反应器与永久反应器	286
10.6	反应器使用规则	287
10.7	反应器实例	288
10.8	本章小结	290
10.9	习题	290

第 1 章 AutoLISP 语言概述

AutoLISP 语言是嵌入在 AutoCAD 内部的 LISP 编程语言,它是 LISP 语言与 AutoCAD 有机结合的产物。自从 AutoLISP 嵌入 AutoCAD 之后,AutoCAD 便不再仅仅作为交互式的图形绘制、编辑软件,而真正成为能进行计算机辅助设计、绘图的 CAD 软件。

本章着重介绍 AutoLISP 的出现及发展历史,并对 AutoLISP 语言及 AutoLISP 程序特点做简要的介绍,内容包括:

- AutoLISP 语言的出现与发展
- AutoLISP 语言的特点
- 简单的 AutoLISP 程序介绍
- AutoLISP 程序的执行
- AutoLISP 程序的内存分配

1.1 AutoLISP 语言的出现与发展

LISP 语言(list processing language)是人工智能学科领域中广泛采用的一种程序设计语言,是一种计算机表处理语言,又称为符号式语言(symbolic language)。50 年代末,John McCarthy 为支持 Alonzo Church 的 λ 微积分学而发明了 LISP 语言。在以后十多年的发展过程中,LISP 语言产生了多种版本,比如 MacLISP、InterLISP、ZetaLISP、CommonLISP 等,主要用于人工智能、机器人、专家系统、博弈、定理证明等领域。其中,CommonLISP 是美国的几所大学(MIT、STANFORD、CMU、RUTGERS 等)和工业界(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,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。

LISP 语言的内部数据结构非常适合用表表示,而且表中还可以嵌套其他的表,这种构成表的符号表达式包括在一对括号中。这种符号表达式(symbolic expression)是 LISP 语言中最基本的数据类型,处理符号是 LISP 语言的特性之一,LISP 语言的程序和数据都以符号表达式的形式表示,也就是说,一个 LISP 程序可以把另一个 LISP 程序作为它的数据处理。所以,LISP 很容易去定义和调用用户编写的函数,这使得 LISP 语言的编程十分灵活。

AutoLISP 语言是嵌入在 AutoCAD 内部的 LISP 编程语言,是 LISP 语言与 AutoCAD 有机结合的产物。Autodesk 公司主要的奠基人 John Walker 早在 80 年代中期即认识到 LISP 语言与 AutoCAD 之间具有潜在的协同工作的能力,既可以提供非常简单的宏操作,又可以为类似的操作提供高级编程语言中广泛的资源,因而 Autodesk 公司从 AutoCAD R2.1 开始引入 AutoLISP 作为编程的接口。AutoLISP 既沿用了 LISP 的语法和习惯约定,具有 LISP 语言的特性,又针对 AutoCAD 增加了许多功能。例如 AutoLISP 可以方便调用 AutoCAD 的绘图命令,使设计和绘图完全融为一体;还可以实现对 AutoCAD 当前数据库的直接访问、修改,为实现对

屏幕图形的实时修改、交互设计、参数化设计以及在绘图领域中应用人工智能提供了方便。概括地说,AutoLISP 综合了人工智能语言 LISP 的特性和 AutoCAD 强大的图形编辑功能的特点,可谓是一种人工智能绘图语言。

AutoLISP 语言最典型的应用之一就是实现参数化绘图的程序设计,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,调整所需的参数值可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,而实际上,利用 AutoLISP 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)文件来创建自己的对话框。

自从 AutoLISP 嵌入 AutoCAD 之后,AutoCAD 便不再仅仅作为交互式的图形绘制、编辑软件,而真正成为能进行计算机辅助设计、绘图的 CAD 软件。

近几年,随着 Autodesk 产品人机交互性的进一步提高,在 AutoCAD R14 以后版本中均附带了 Visual LISP 作为 AutoLISP 的开发环境。Visual LISP 不仅继承了 AutoLISP 程序设计的优点,还允许用 AutoLISP 程序维护 AutoCAD 的资源。Visual LISP 提供了友好方便的操作编辑界面及强大的调试功能,这些特性使得编写、修改代码以及测试和调试程序变得更加容易,简化了自定义 AutoCAD 操作的开发过程,使得 AutoLISP 的开发效率更高。

1.2 AutoLISP 语言的特点

在 Autodesk 公司逐渐推出其他如 ARX、ADS 等开放工具的情况下,许多设计者在针对 AutoCAD 做一些简单的辅助设计,或是进行复杂的开发时,都愿意选择 AutoLISP,主要的原因还是在于 AutoLISP 语言那独到的特点赢得了开发人员的喜爱。

首先,AutoLISP 语言是在普通 LISP 语言的基础上,扩充了许多适用于 CAD 应用的特殊功能而形成的一种以解释方式运行于 AutoCAD 内部的程序设计语言,继承了 LISP 语言擅长处理各种结构的数据表的特点。而且,AutoLISP 的数据类型是相当随意的,可以组织处理不同长度和结构的数据类型,用户可以按需求或最佳结构来设计和使用自定义的结构类型数据,甚至是几十万个巨大结构类型数组。因而,用户在开发程序时,不会感到在组织数据结构上的困难,AutoLISP 甚至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。

再者,AutoLISP 语言中的一切成分都是以函数的形式给出的,没有语句、过程这类程序设计中通常的概念。AutoLISP 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达式组合而成的。执行 AutoLISP 程序就是调用一些函数,函数再调用其他函数,即是在对函数求值的过程中实现函数的功能,进而实现程序的功能。

最后,AutoLISP 擅长描述人机交互操作的过程,对于各种用户输入接收、错误识别与恢复等方面的优秀功能(尤其是 DCL 语言加入以后),其他语言很难在此方面比及。而且,在 AutoLISP 程序中可以引用 AutoCAD 几乎所有的功能,进行多图形文档下的图形处理以及对象数据库、对象反应器等等的操作。

总的来说,AutoLISP 的特点主要还是体现其独特的语言结构以及执行方式上,下面就语言结构来体会一下 AutoLISP 的特点。

上面已经提到,AutoLISP 语言与一般的程序语言不同,没有语句、过程等概念,那 AutoLISP 程序是由什么组成的呢?它是由一个或者一系列按照特定顺序排列的表组成的。那什么是表呢?先来看看几个应用表的例子。例如:

```
(setq x 12.3 y 13.5)
(setq p (list x y))
(command "line" "0,0" p "")
```

通过以上 3 个表的实例,可以看出表的一些基本特点:表由左右两个括号组成,括号中是实现其功能的代码;表的第一个元素(如 `setq`、`command`)均为操作符(函数),其余的元素则是相应操作符(函数)的参数;表中的函数与参数、参数与参数之间至少要用一个空格(或回车)来分开。表的一般格式为:

```
(函数名 [<参数 1>] [<参数 2>]...)
```

而且和其他程序语言不同的是,AutoLISP 的表中还可以包含其他的表,就像上面的(`setq p (list x y)`),表中还包含另一个表(`list x y`)。

AutoLISP 程序就是由一个个这样的表组成的,如将上例中的 3 个表依次组合在一起,就可以看作是一个 AutoLISP 程序了。这个程序所实现的功能就是在(12.3 13.5 0)与原点之间画一条直线。同样地,复杂一些的程序也是由不同的表组成的,表是组成 AutoLISP 程序的最基本的元素。

在 AutoLISP 程序中,两个表之间和表内的多余空格与回车都不影响表的功能,因此,一个表可以占多行,也可以在一行内写下很多表。例如,上例中的表(`setq x 12.3 y 13.5`)可以写成下面的形式:

```
(setq x 12.3
      y 13.5)
```

同样地,也可以将上述程序中的 3 个表写在一行当中。出于程序代码可读性的考虑,应该采取前一种格式来书写程序,一行内一般最多写一个表,对于较长的表,可以将表分作几行来书写,以利于程序源代码的阅读。

AutoLISP 程序文件的扩展名为 `.lsp`,是以 ASCII 码文本形式来存储的,因此可以使用任何一个文本编辑软件来阅读和查看,如 Windows 操作系统的 `notepad` 程序、`UltrEdit-32 text editor` 与 `editplus` 等。

1.3 简单的 AutoLISP 程序介绍

在了解了 AutoLISP 程序最基本的元素表后,下面来看看几个简单的 AutoLISP 程序,从中体会 AutoLISP 程序的特性。