



普通高等教育“十二五”规划教材

AutoCAD

二次开发

实用教程

郭秀娟 徐勇 郑馨 等编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”规划教材

AutoCAD 二次开发 实用教程

郭秀娟 徐勇 郑馨 编
李力东 于全通 张朝

机械工业出版社

北京机械工业出版社

1. 4105 . 11

机械工业出版社

ISBN 978-7-

—A—1

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01

—CIP—001234-01



机械工业出版社

本书是讨论基于 Visual LISP 的 AutoCAD 二次开发程序设计技术的教程,旨在帮助用户进行专业辅助设计程序的制作和使用,达到精通 Visual LISP 程序设计,使 AutoCAD 真正成为用户的专业设计软件。

本书详细讨论了 Visual LISP 程序设计的基本方法和应用技巧,AutoLISP 语言的基本函数及利用 AutoLISP 语言进行 AutoCAD 二次开发的方法,同时结合编者多年的教学经验提供了大量的例题和范例,侧重于专业应用的方法、实际应用中的难点和解决方案的讨论。本书既可以作为高等院校的有关教材、高级应用培训教材,也适合作为专业程序设计用户的参考用书。

全书语言叙述精练、实例讲解过程翔实,力争做到初学者能够看懂,程序设计的专业人员能够得到启发,为广大从事 AutoCAD 二次开发的读者提供有力的指导。

本书适合作为工科院校建筑类及相关专业学生的教材,也可供建筑、机械设计、电子电路设计、平面图设计、三维造型等行业及相关专业人员,AutoLISP 初学者,3D 图形爱好者学习和使用。

图书在版编目 (CIP) 数据

AutoCAD 二次开发实用教程/郭秀娟等编. —北京:机械工业出版社, 2014. 1

普通高等教育“十二五”规划教材

ISBN 978-7-111-44795-5

I. ①A… II. ①郭… III. ①AutoCAD 软件—高等学校—教材
IV. ①TP391.72

中国版本图书馆 CIP 数据核字 (2013) 第 272112 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 贡克勤 责任编辑: 贡克勤 任正一

版式设计: 常天培 责任校对: 卢惠英

封面设计: 陈沛 责任印制: 张楠

北京玥实印刷有限公司印刷

2014 年 3 月第 1 版第 1 次印刷

184mm × 260mm · 18.5 印张 · 457 千字

标准书号: ISBN 978-7-111-44795-5

定价: 38.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066 教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294 机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649 机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203 封面防伪标均为盗版

前 言

AutoCAD 二次开发实用教程立足于解决实际问题，以实例讲解为主，通过循序渐进的实例开拓思路，使读者在实例中快速掌握利用 AutoCAD 进行二次开发的基本方法。

本书的开发工具为 Visual LISP 语言，它是为加速 AutoLISP 程序开发而设计的软件开发工具，是一个完整的集成开发环境。在 Visual LISP 环境下可以便捷、高效地开发 AutoLISP 程序，经过编译得到运行效率更高、代码更加紧凑、源代码受到保护的应用程序。

Visual LISP 既兼容 AutoLISP 程序，又扩充了许多新的功能，是新一代的 AutoLISP 语言。利用 AutoLISP 可以进行各种工程的分析计算、自动绘制复杂的图形，驱动对话框、控制菜单、定义新的命令，为 AutoCAD 扩充智能化和参数化的功能。

AutoCAD 在工程设计领域得到了普遍应用，为其专业设计提供了方便。由于 AutoCAD 是一个通用绘图软件，不具专业特色，使作图效率不高。而 AutoCAD 开放的结构为使用者提供了广阔的开发空间及许多二次开发的工具，AutoLISP 就是其中比较常用的一个，它能够为用户开发出具有专业特点的高效率应用软件。因此，了解 AutoLISP 的程序结构，掌握 AutoLISP 程序设计方法，开发出适合专业特点的 CAD 软件，已成为专业技术人员和学生渴望掌握的一个工具。目前，国内一些高等院校的工程设计相关专业也开设了 AutoLISP 语言课程。然而，能够作为教材及参考书使用的则不多见，无法满足人们实现深入学习及开发的设想和愿望。

编者通过查阅有关资料和参考手册，经过多年的教学实践，对 AutoLISP 语言有了较为深入的理解，积累了一些难得的实践资料，并在教学及工程实践中得以验证。目前，还有一些用户停留在将 AutoCAD 作为绘图工具的阶段，这样只是提高了绘图的效率，但距离真正意义上的计算机辅助设计还有较大差距。因此，实现图形参数化、智能化、分析计算与绘图一体化是本书要介绍的主要内容。

本书有以下两个主要特点：

(1) 实用性 书中所有实例均以实际应用为背景，具有较高的实用价值和一定的技术含量。初学者可以从调试、运行这些实例程序开始，然后修改、扩充这些实例，逐步掌握 AutoLISP 程序设计技术。

(2) 便于教学 本书是在学校教学和工程实践背景下编写的。编者参照多

年的教案确定了本书的内容和章节的次序，因此本书具有便于教学和实训的特点。

本书的程序是在 AutoCAD 2007 的 Visual LISP 集成环境下建立、调试和运行通过的。所有程序的源代码都适用于当前的 AutoCAD 版本。

全书共 13 章。郭秀娟负责第 1~5 章的编写，徐勇负责第 8~10 章的编写，郑馨负责第 6、12 章的编写，李力东负责第 13 章及附录部分的编写，张朝负责第 11 章的编写，于全通负责第 7 章的编写及全书的程序调试运行与图形绘制工作。本书在编写过程中，得到了吉林建筑大学计算机学院老师及相关专业人士的帮助和指导，编者在此深表谢意。由于编写水平有限，书中的不当和疏漏之处在所难免，恳请各方面的专家予以指教并请广大读者提出宝贵的意见。

编者

本书主要介绍了 AutoCAD 二次开发的主要方法和应用。全书共分 13 章，第 1 章介绍了 AutoCAD 二次开发的概念、特点、应用及开发环境；第 2 章介绍了 AutoCAD 二次开发的开发环境；第 3 章介绍了 AutoCAD 二次开发的开发语言；第 4 章介绍了 AutoCAD 二次开发的开发工具；第 5 章介绍了 AutoCAD 二次开发的开发方法；第 6 章介绍了 AutoCAD 二次开发的开发应用；第 7 章介绍了 AutoCAD 二次开发的开发案例；第 8 章介绍了 AutoCAD 二次开发的开发应用；第 9 章介绍了 AutoCAD 二次开发的开发应用；第 10 章介绍了 AutoCAD 二次开发的开发应用；第 11 章介绍了 AutoCAD 二次开发的开发应用；第 12 章介绍了 AutoCAD 二次开发的开发应用；第 13 章介绍了 AutoCAD 二次开发的开发应用。本书可作为 AutoCAD 二次开发的教学用书，也可作为 AutoCAD 二次开发的工程技术人员参考。

试读结束，需要全本请在线购买：www.ertongbook.com

目 录

前言	1	3.2.6 表处理函数举例	36
第1章 Visual LISP 语言概述	1	习题	38
1.1 LISP 语言	1	第4章 程序流程控制	40
1.2 AutoLISP 语言	1	4.1 顺序结构	40
1.3 Visual LISP 语言	2	4.1.1 GET 族输入函数	40
1.4 Visual LISP 的编程环境	4	4.1.2 图形处理函数	45
1.4.1 Visual LISP 集成开发环境的界面	4	4.1.3 显示控制函数	49
1.4.2 输入和修改程序代码	6	4.1.4 举例	51
习题	8	4.2 分支结构	52
第2章 数据类型、表	9	4.2.1 判断函数	52
2.1 数据类型	9	4.2.2 条件函数	53
2.1.1 原子	9	4.3 循环函数	56
2.1.2 表和点对	11	4.4 函数递归定义	59
2.1.3 其他类型	12	4.4.1 递归的概念	59
2.1.4 AutoLISP 的程序结构	13	4.4.2 递归模型	59
2.2 变量	18	4.4.3 递归算法的程序设计	60
2.2.1 符号	18	4.5 综合举例	61
2.2.2 变量的数据类型	18	习题	75
2.2.3 变量赋值	19	第5章 AutoLISP 文件	77
2.2.4 显示变量的值	20	5.1 AutoLISP 文件的特点	77
2.2.5 在交互方式下将变量的值 传递给 AutoCAD	20	5.2 程序中的注释	78
2.2.6 AutoCAD 的系统变量	20	5.3 在 AutoCAD 环境下加载 AutoLISP 文件	78
习题	21	5.4 搜索、获得文件的函数	80
第3章 AutoLISP 基本函数	23	5.5 打开、关闭文件的函数	81
3.1 数值函数	23	5.6 用于文件的输入输出函数	84
3.1.1 计算函数	23	5.7 综合举例	86
3.1.2 布尔运算函数	27	习题	87
3.1.3 三角函数	28	第6章 实体和设备访问函数	89
3.1.4 数值函数举例	29	6.1 基本概念	89
3.2 表处理函数	30	6.2 选择集操作函数	92
3.2.1 提取表中数据的函数	30	6.3 实体名操作函数	97
3.2.2 构造和修改表的函数	32	6.4 实体数据函数	99
3.2.3 提取并修改表中数据的函数	33	6.5 符号表的访问	108
3.2.4 表循环处理函数	35	6.6 图形屏幕和输入设备的访问	110
3.2.5 其他表处理函数	36		

6.7 综合举例	113	10.7 访问 AutoCAD 对象	183
6.7.1 实体名和选择集在开发 AutoCAD 程序中的应用	113	习题	186
6.7.2 生成局部放大视图的简便方法	114	第 11 章 编辑及维护 AutoLISP 程序	188
6.7.3 求圆或圆弧中心线	117	11.1 编译链接程序	188
习题	122	11.1.1 Visual LISP 编译器	188
第 7 章 AutoLISP 实训	123	11.1.2 加载运行已编译程序	190
7.1 设置作图环境	123	11.1.3 链接函数调用	191
7.2 设置图层、颜色、 线型和线宽	126	11.2 生成应用程序	191
7.3 AutoLISP 程序设计 的 6 个步骤	127	11.2.1 创建新应用程序	191
7.4 AutoLISP 程序实例	127	11.2.2 加载和运行 Visual LISP 应用程序	196
习题	136	11.2.3 修改应用程序选项	196
第 8 章 Visual LISP 基本操作	137	11.2.4 重新编译应用程序	197
8.1 进入和退出 Visual LISP	137	11.2.5 更新应用程序	197
8.2 Visual LISP 的用户界面	137	11.3 多文档环境下的程序设计	197
8.3 Visual LISP 的控制台操作	141	11.3.1 理解命名空间	198
8.4 Visual LISP 的文件操作	142	11.3.2 查看多名称空间对函数的 影响步骤	199
8.5 退出 Visual LISP	144	11.3.3 运行应用程序于自身的 名称空间中	199
习题	144	11.3.4 使文档可以访问函数	201
第 9 章 编辑源程序代码	145	11.3.5 查看 vl-doc-export 在独立名称 空间 VLX 中的作用	201
9.1 文本编辑工具	145	11.3.6 使用其他 VLX 应用程序访问独立 名称空间的函数	202
9.2 文本操作	147	11.3.7 引用文档名称空间中的变量	202
9.3 设置代码格式	150	11.3.8 在名称空间中共享数据	202
9.4 检查语法错误	152	11.3.9 MDI 环境下的错误处理	203
习题	156	11.3.10 在自身名称空间中运行的 VLX 的错误处理	204
第 10 章 调试程序	157	11.3.11 在 MDI 环境下对于使用 AutoLISP 的限制	204
10.1 Visual LISP 调试功能简介	157	习题	205
10.2 通过实例学习调试程序	158	第 12 章 使用 ActiveX	206
10.3 Visual LISP 调试功能	162	12.1 在 AutoLISP 中使用 ActiveX 对象	207
10.3.1 开始调试任务	163	12.2 AutoCAD 对象模型	207
10.3.2 断点循环	163	12.2.1 对象属性	207
10.3.3 使用断点	164	12.2.2 对象方法	210
10.4 使用 Visual LISP 数据 查看工具	166	12.2.3 对象集合	211
10.4.1 监视程序	167	12.3 访问 AutoCAD 对象	211
10.4.2 跟踪程序	170		
10.5 修改变量和函数的特性	176		
10.6 “检验”窗口	178		

12.3.1	访问 AutoCAD 应用程序	211	12.11.2	将一系列函数应用到集合中的每一个对象	243	
12.3.2	应用程序对象以下的其他 ActiveX 对象	212	12.11.3	获取集合中的成员对象	244	
12.3.3	过程总结	212	12.11.4	释放 VLA 对象和释放内存	245	
12.3.4	编程技巧	213	12.11.5	处理 ActiveX 方法返回的错误	245	
12.3.5	在 Visual LISP 函数中使用 ActiveX	214	12.12	举例	246	
12.3.6	确定所需的 Visual LISP 函数	214		习题	247	
12.4	ActiveX 对象访问	215	第 13 章 使用反应器		249	
12.4.1	查看对象特性	216	13.1	反应器基础	249	
12.4.2	访问图形对象	217	13.1.1	反应器的类型	249	
12.4.3	访问其他 AutoCAD 对象	218	13.1.2	反应器的回调事件	250	
12.4.4	使用检验工具了解 AutoCAD 对象的属性	222	13.1.3	反应器的回调函数	252	
12.4.5	通过 Help 功能了解 AutoCAD 对象	225	13.2	生成反应器	255	
12.5	在 Visual LISP 函数中使用 ActiveX 方法	225	13.2.1	创建对象反应器	255	
12.5.1	查找所需要的函数	226	13.2.2	创建其他反应器	258	
12.5.2	确定函数参数	227	13.2.3	将数据附着到反应器对象	259	
12.5.3	将 Visual BASIC 环境下的语句改写为 AutoLISP 表达式	227	13.2.4	在多重名称空间中 使用反应器	259	
12.5.4	转换数据类型为 ActiveX 型	228	13.3	查询、修改和控制 反应器的状态	260	
12.6	AutoCAD 实体名和 VLA 对象之间的转换	236	13.3.1	查询反应器	260	
12.7	修改图形对象的属性	237	13.3.2	修改反应器	262	
12.8	确定方法或属性是否适用于特定对象	238	13.3.3	控制反应器的状态	264	
12.9	确定是否可以修改对象	239	13.4	临时反应器和永久反应器	265	
12.10	使用参数带回返回值的 ActiveX 方法	240	13.5	反应器的使用规则	266	
12.11	使用集合对象	241	13.6	定义反应器实例	267	
12.11.1	将某一个函数应用到集合中的每一个对象	241		习题	271	
				附录	272	
				附录 A	AutoLISP 函数概要	272
				附录 B	标准 ASCII 码表	283
				附录 C	联机程序错误代码	285
				参考文献	288	

第 1 章 Visual LISP 语言概述

1.1 LISP 语言

LISP (List Processing Language) 是表处理语言, 主要用于人工智能 (AI)、专家系统、机器人、博弈和定理证明等领域。LISP 最初是作为书写字符与表的递归函数的形式出现的, 也称为符号式语言。该语言于 1958 年由美国麻省理工学院 (MIT) 的 AI 小组提出, 1960 年由 MIT 的 John. McCarthy 教授整理成为 LISP1.0 版发表。以后陆续出现的 LISP1.5、LISP1.6、MacLISP、InterLISP, CommonLISP, GCLISP, CCLISP 等变种。在众多流行的 LISP 语言版本中, 使用最广泛的是 InterLISP、MacLISP 和 CommonLISP。LISP 是继 FORTRAN 之后出现的第二个古老的计算机高级语言, 至今使用五十多年仍受重视, 并为人工智能语言的发展做出了不可磨灭的贡献。因此, LISP 是一门历史悠久, 用途广泛, 功能极强的人工智能程序设计语言。

LISP 语言的发展经历了如下几个时期:

- 1) 酝酿时期 (1956—1958), 这个时期形成了 LISP 的基本思想。
- 2) 实现与应用时期 (1958—1962), 这个时期的发展基本上是单线的。
- 3) “百家争鸣”时期 (1962—1984), 在这个时期 LISP 的发展呈现多样化, 形成了多种 LISP 语言, 支持 LISP 的机器也越来越多。不同的机构、团体在开发 LISP 语言的同时, 在一定程度上对 LISP 的发展也做出了贡献。
- 4) 标准化时期 (1984 至今), LISP 语言的发展进入了标准化时代。1981 年夏 B. K. Steele 开始编写 Common LISP 手册, 并试图进行标准化, 直到 1983 年初, Common LISP 语言基本达到了预期的目标: ① 公用性: 成为一种公用的 LISP 语言; ② 可移植性; ③ 一致性: 解释器和编译器应保证语义相同; ④ 丰富的表达能力: 吸取各种方言的优点; ⑤ 兼容性: 与 ZetaLISP、MacLISP 和 InterLISP 兼容; ⑥ 效率: 提供一个优化编译器; ⑦ Common LISP 是一个系统构造语言; ⑧ 稳定性。

1987 年, ISO 成立了一个工作小组 WG16, 讨论 LISP 的标准化、国际化。其目标是未来的 LISP 语言应该具有商业价值分层, 内部是一个 Kernel LISP。在 1992 年 ISO 草拟了一个 Kernet LISP 草案, 并不断完善, 形成标准, 这个标准被大多数解释器和编译器所接受。

LISP 语言具有其他高级语言不可比拟的特征。它具有深厚的理论基础, 丰富的表达能力, 较强的可塑性, 也提供了操作系统的许多设施, 如命令解释器、文件管理、多任务等。所有这些特征为符号计算和人工智能研究提供了一个方便的工具。

1.2 AutoLISP 语言

AutoLISP 是由 Autodesk 公司为二次开发 AutoCAD 而专门设计的编程语言, 它起源于 LISP 语言, 并嵌入在 AutoCAD 的内部, 是 LISP 语言和 AutoCAD 有机结合的产物。

AutoLISP 采用了和 CommonLISP 最相近的语法和习惯约定, 具有 CommonLISP 的特性, 但针对 AutoCAD 又增加了许多功能。它既有 LISP 语言人工智能的特性, 又具有 AutoCAD 强大的图形编辑功能。可以把 AutoLISP 程序和 AutoCAD 的绘图命令透明地结合起来, 使之成为一体, 还可以实现对 AutoCAD 图形数据库的直接访问和利用; AutoLISP 语言可以自动绘制各种复杂的图形, 定义新的 AutoCAD 命令, 驱动对话框, 为 AutoCAD 扩充智能化、参数化。

AutoLISP 解释程序位于 AutoCAD 软件包中。需要指出的是 AutoCAD R2.17 及更低版本中并不包含 AutoLISP 解释程序, 只有 AutoCAD R2.18 及更高版本才可以使用 AutoLISP 语言。

1. AutoLISP 的优点

- 1) 源于 LISP 的 AutoLISP 语言语法规则简单, 灵活且易学易用。
- 2) 功能函数强大, 编写环境简单。
- 3) 可根据需求对 AutoCAD 进行二次开发, 实现对 AutoCAD 的图形实体和各种参数表的数据进行存取和编辑, 易于交互。
- 4) 解释执行。

2. AutoLISP 的缺点

- 1) 功能单一, 综合处理能力差。
- 2) 解释执行, 程序运行速度慢。
- 3) 缺乏很好的保护机制, 源程序保密性差。
- 4) LISP 用表来描述一切, 并不能很好地反映现实世界和过程, 与人的思维方式不一致。

AutoCAD 软件包中包含的命令大多数是用于绘制和修改图形的, 但仍有某些命令未被提供。如, AutoCAD 中没有图形文本对象内绘制矩形及作全局改变的命令, 但是通过使用 AutoLISP 语言编程就可实现在图形文本对象内绘制矩形或作全局选择性改变的操作。事实上, 可以用 AutoLISP 编写任何程序, 或把它嵌入到菜单中, 这样定制的系统效率会有很大的提高。

1.3 Visual LISP 语言

Visual LISP 是 Autodesk 公司在 1997 年的 AutoCAD14 版本中推出的。是为加速 AutoLISP 程序开发而设计的软件开发工具, 是一个完整的集成开发环境 (IDE)。它增强并扩展了 AutoLISP 语言, 可以通过 Microsoft ActiveX Automation 接口与对象交互, 并扩展了 AutoLISP 响应事件的能力。Visual LISP 集成开发环境包括文本编辑器、格式编排器、语法检查器、源代码调试器、检验和监程管理系统、上下文相关帮助等。Visual LISP 用户界面良好, 用过 Microsoft 软件的用户只需很短的时间即可掌握它。

Visual LISP 兼容 AutoLISP 程序。在 Visual LISP 集成环境下可以便捷、高效地开发 AutoLISP 程序, 可以经过编译得到运行效率高、代码紧凑、源代码受到保护的应用程序。

Visual LISP 是新一代的 AutoLISP 语言, 它对 AutoLISP 语言的功能进行了扩展, 可以通过 Microsoft ActiveX Automation 接口与 AutoCAD 对象进行交互, 可以通过反应器函数扩展

AutoLISP 响应事件的能力。使用 Visual LISP 中对 AutoLISP 进行扩展的功能时, 必须调用 `vl-load-com` 函数, 或将调用该函数的表达式写 `acad * doc. lsp` 文件内 (*为通配符, 代表 AutoCAD 的不同版本)。

Visual LISP 既是 LISP 编辑器又是编译器, 提供一套简单的可视化环境来开发和维护原有的 AutoLISP 源程序。比其他的 AutoCAD 编程语言 (AutoLISP, ADS, VB, ARX) 更灵活, 更先进, 更易用, 并且 Visual LISP 对硬件没有任何特殊的需求, 只要能运行 AutoCAD 的系统即可运行 Visual LISP。Visual LISP 作为一个完整的集成开发环境 (IDE), 具有自己的窗口和菜单, 但它并不能独立于 AutoCAD 运行。当用户从 Visual LISP IDE 中运行 AutoLISP 程序时, 经常需要与 AutoCAD 图形交互或在命令窗口响应程序提示。当 Visual LISP 把控制传给 AutoCAD 时, AutoCAD 已被最小化, 用户必须手动恢复并激活 AutoCAD 才可继续, Visual LISP 不会自动恢复 AutoCAD 窗口。相反, Visual LISP 窗口中会出现并保持一个 Visual LISP 的符号, 直到激活 AutoCAD 并响应了在 AutoCAD 命令提示处的提示。

1. Visual LISP IDE 的主要组成部分和功能

- (1) 语法检查器 可识别 AutoLISP 语法错误和调用内置函数时的参数错误。
- (2) 文件编译器 改善了程序的执行速度, 并提供了安全高效的程序发布平台。
- (3) 源代码调试器 专为 AutoLISP 设计, 利用它可以在窗口中单步调试 AutoLISP 源代码, 同时还在 AutoCAD 图形窗口显示代码运行结果。
- (4) 文字编辑器 可采用 AutoLISP 和 DCL 语法着色, 并提供其他 AutoLISP 语法支持功能。
- (5) AutoLISP 格式编排程序 用于调整程序格式, 改善其可读性。
- (6) 全面的检验和监视功能 用户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。还可浏览 AutoLISP 数据和 AutoCAD 图形的图元。

2. Visual LISP 的特点

Visual LISP 是 Autodesk 公司为 AutoLISP 提供的一个完整的开发环境, 从 R14 版开始, AutoCAD 支持 Visual LISP 开发工具。Visual LISP 是一个可视化的 LISP 语言开发环境, 它是 AutoLISP 语言的扩展和延伸。Visual LISP 具有以下特性:

- (1) 在完全可视化的开发环境下编写、调试程序。
- (2) Visual LISP 程序经过编译后, 提高了运行性能和保密性。
- (3) LISP 代码通过 AutoCAD R14 的 Object ARX 接口, 提高了程序的运行速度。
- (4) Visual LISP 是一个被建立并装载的 Object ARX 应用程序, 可以在 AutoCAD 外部装载和更新。
- (5) Visual LISP 与 AutoLISP 完全兼容, 因为 Visual LISP 将它的程序语法设计成与 AutoLISP 相同, 并且还新增了许多函数和系统变量, 这使得 AutoCAD 的应用程序开发工作变得更加容易。
- (6) 功能强大的整合开发环境。Visual LISP 整合了 AutoLISP 程序开发时所需的几大主要工具和功能, 包括:

① Visual LISP 采用 `Compile-during-Load` 技术, 达到与 AutoLISP 完全兼容的境界。

② Visual LISP 采用可支持 AutoLISP 与 DCL 色彩编码以及其他 AutoLISP 语法的全屏幕文本编辑器。这样将方便用户输入 AutoLISP 源程序, 并通过色彩编码对源程序的不同部分加

以区分，达到改善 AutoLISP 源程序可读性的目的。

③ Visual LISP 支持多种检查器，其中，语法检查器可以用来检查 AutoLISP 程序结构错误和内部函数中的变量错误。综合检查器可以提供对数据结构中变量和表达式值的浏览和编辑功能。

④ 将 Visual LISP 的动态调整功能用于专门调整 AutoLISP 源程序及其灵活性。它可以在一个窗口单一执行 AutoLISP 的源代码，在 AutoCAD 窗口中同时显示程序执行的效果。

⑤ Visual LISP 先进的源程序编译器可以将 AutoLISP 的源程序编译成二进制文件。这将大力改善程序的执行速度与安全性。

综上所述：Visual LISP 是一种将 AutoLISP 语言的优点完全保留，克服其缺点，并与最新的程序设计技术相结合的全新的整合开发系统，它已经成为 AutoCAD 的下一代语言标准。Visual LISP 也将充分地利用现有的 AutoLISP 资源，极力保护用户投资，所以它的推出已引起广大的 AutoCAD 用户及专业开发人员的强烈兴趣。

3. Visual LISP 为 AutoLISP 应用程序提供 3 种文件格式选项

- (1) 读取 LSP 文件 (.lsp) 包含 AutoLISP 程序代码的 ASCII 文本文件。
- (2) 读取 FAS 文件 (.fas) 单个 LSP 程序文件的二进制编译版本。
- (3) 读取 VLX 文件 (.vlx) 一个或多个 LSP 文件和/或对话框控制语言 (DCL) 文件的编译集。

注意：名称相似的 AutoLISP 应用程序文件的加载由它们的编辑时间决定。除非指定完整的文件名（包括文件扩展名），否则将加载最近编辑过的 LSP、FAS 或 VLX 文件。

1.4 Visual LISP 的编程环境

1.4.1 Visual LISP 集成开发环境的界面

1. 启动 Visual LISP

由于 Visual LISP 集成于 AutoCAD 系统内部，因此用户必须先启动 AutoCAD，然后才能进入 Visual LISP IDE 环境。启动 Visual LISP 的方式为：

菜单：“Tools (工具)” → “AutoLISP” → “Visual LISP Editor (Visual LISP 编辑器)”

命令行：vlide (或 vlisp)

启动 Visual LISP，进入 Visual LISP IDE 环境，其集成开发环境 (IDE) 的主界面如图 1-1 所示。

各组成部分的说明如下：

- (1) 菜单栏 通过选取各菜单项的选项执行 Visual LISP 命令。
- (2) 工具栏 提供了对常用 Visual LISP 命令的快速调用。Visual LISP 共提供了 5 个工具栏：“Standard (标准)”、“Search (搜索)”、“View (视图)”、“Debug (调试)”和“Tools (工具)”如图 1-2 所示，每个工具栏各自代表不同功能的命令组。

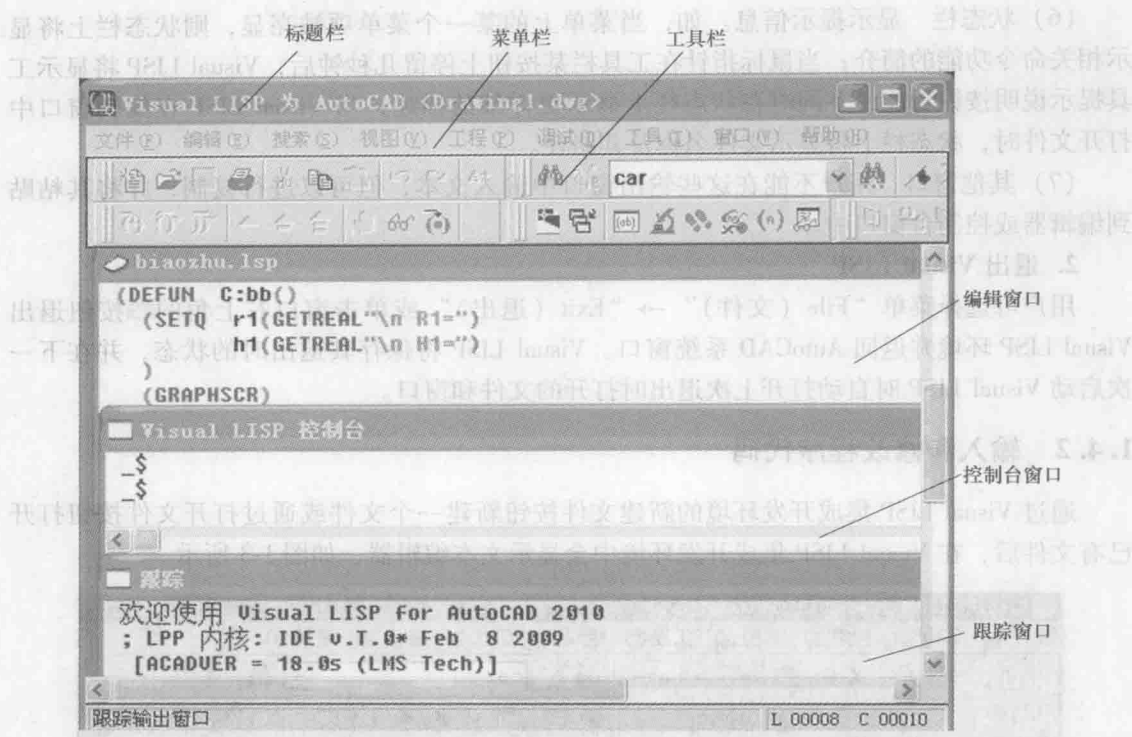


图 1-1 Visual LISP IDE 的主界面



图 1-2 Visual LISP 的工具栏

(3) 编辑窗口 用于编辑 LISP 文件代码。如果用户同时编辑多个文件，则 Visual LISP 使用多个编辑窗口来分别显示文件。


(4) 控制台窗口 类似于 AutoCAD 的命令窗口，可在其中输入 AutoLISP 命令，也可以不使用菜单或工具栏而直接在控制台窗口中调用 Visual LISP 命令。

(5) 跟踪窗口 在启动 Visual LISP 后，窗口将显示 Visual LISP 当前版本的信息。如果 Visual LISP 在启动时遇到错误，它还会包含相应的错误信息。

(6) 状态栏 显示提示信息。如,当菜单上的某一个菜单项被亮显,则状态栏上将显示相关命令功能的简介;当鼠标指针在工具栏某按钮上停留几秒钟后,Visual LISP 将显示工具提示说明按钮功能,并同时状态栏上显示更详细的描述;当 Visual LISP 在编辑窗口中打开文件时,状态栏上将显示文件名称及其路径。

(7) 其他窗口 用户不能在这些输出窗口中输入文本,但可以进行复制,并将其粘贴到编辑器或控制台窗口中。

2. 退出 Visual LISP

用户可选择菜单“File (文件)”→“Exit (退出)”或单击窗口右上角的  按钮退出 Visual LISP 环境并返回 AutoCAD 系统窗口。Visual LISP 将保存其退出时的状态,并在下一次启动 Visual LISP 时自动打开上次退出时打开的文件和窗口。

1.4.2 输入和修改程序代码

通过 Visual LISP 集成开发环境的新建文件按钮新建一个文件或通过打开文件按钮打开已有文件后,在 Visual LISP 集成开发环境中会显示文本编辑器,如图 1-3 所示。

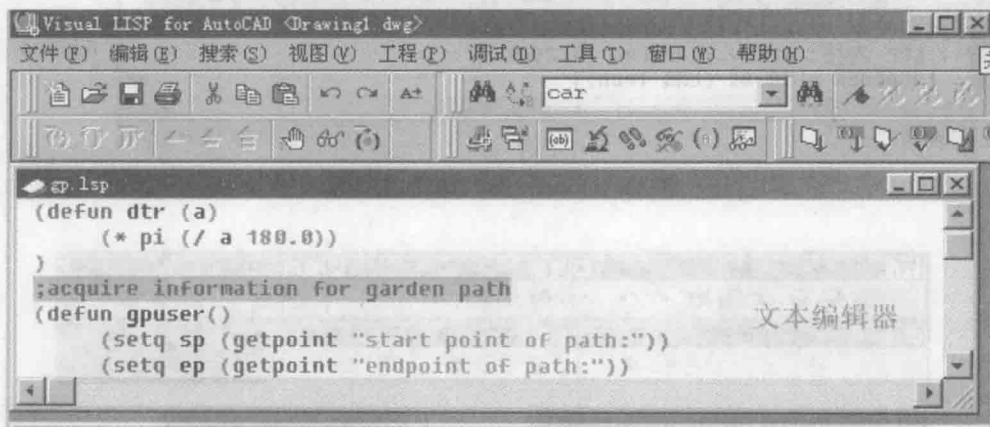


图 1-3 文本编辑器窗口

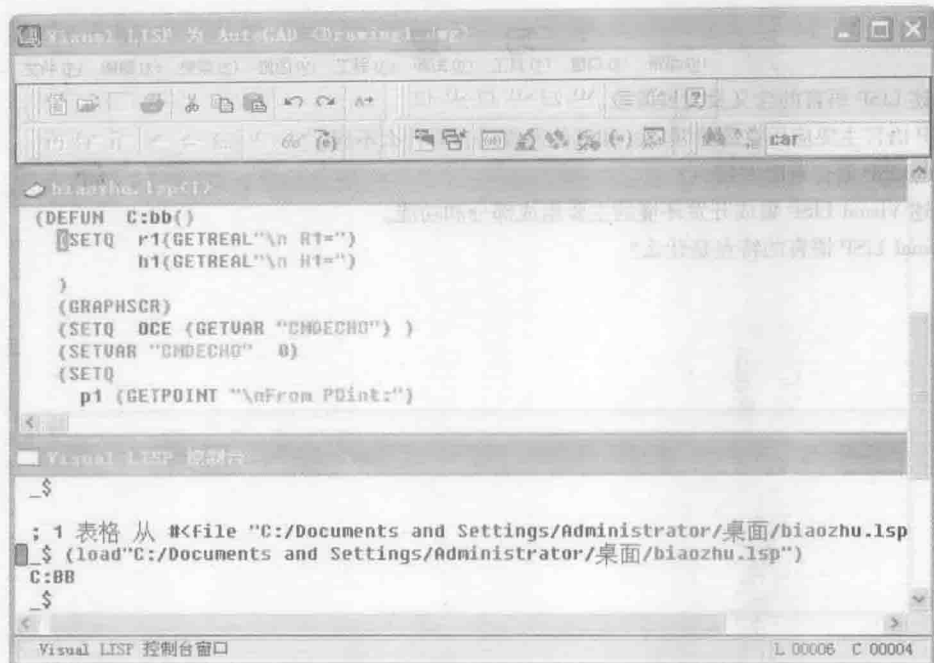
在 Visual LISP 环境中,加载和运行 AutoLISP 程序。

1. 加载 AutoLISP 程序

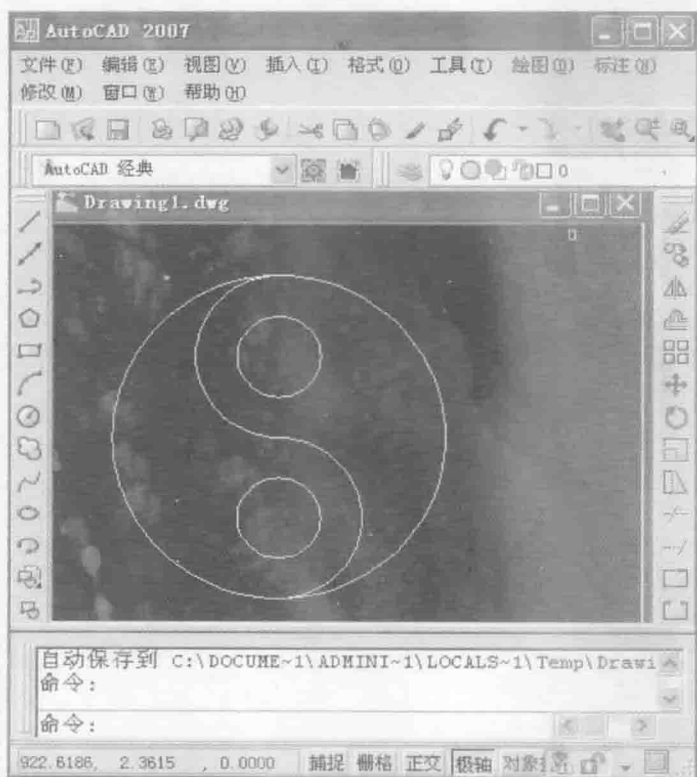
- 1) 直接在 AutoCAD 命令行输入 Load “文件名 [.lsp]”。
- 2) 启动 Visual LISP 集成开发环境。
- 3) 编写新程序或打开已有的 AutoLISP 程序。
- 4) 选择“工具”→“加载编辑器中的文字”。

2. 运行 AutoLISP 程序

- 1) 按照“defun c: <函数名>”格式定义函数。
在 \$_提示符后键入:(c:函数名)后按 <Enter> 键。
- 2) 需要参数时在函数名与参数间应加空格,如图 1-4 所示。



a)



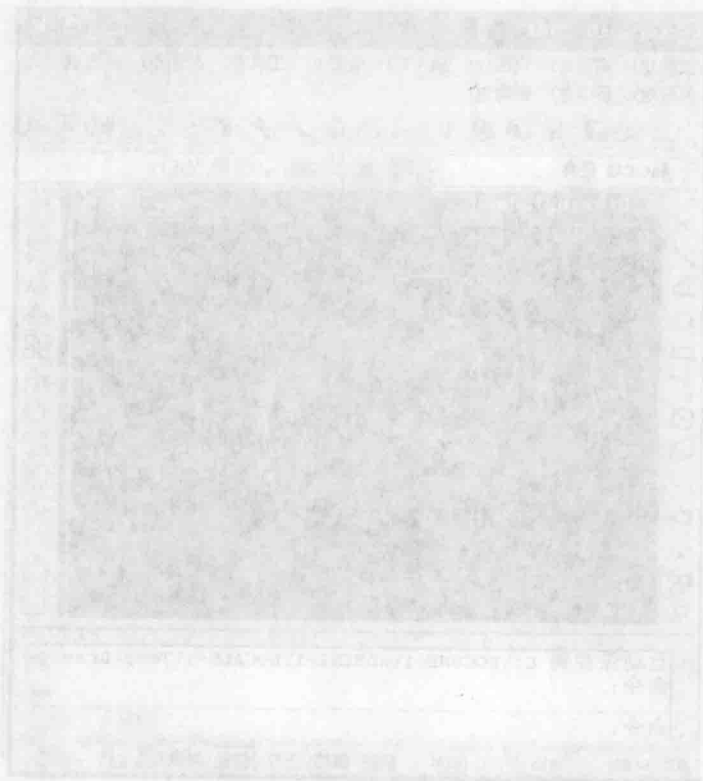
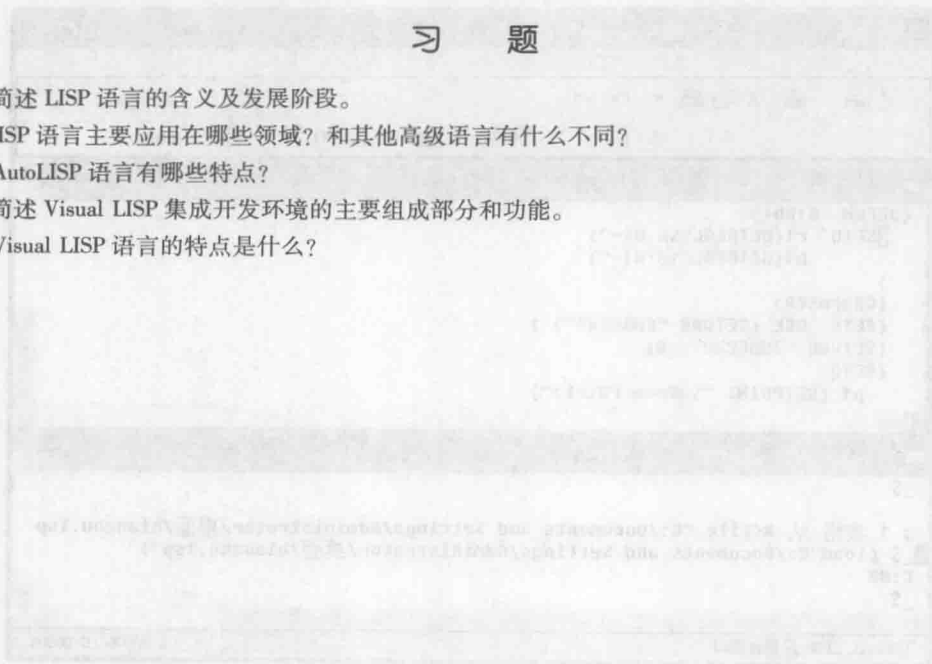
b)

图 1-4 程序编辑、运行窗口

a) 程序编辑窗口 b) 程序运行窗口

习 题

1. 简述 LISP 语言的含义及发展阶段。
2. LISP 语言主要应用在哪些领域？和其他高级语言有什么不同？
3. AutoLISP 语言有哪些特点？
4. 简述 Visual LISP 集成开发环境的主要组成部分和功能。
5. Visual LISP 语言的特点是什么？



第2章 数据类型、表

AutoLISP 语言的数据类型丰富,除了一般程序设计语言具有的整型、实型、字符串等类型外,还有表、函数、AutoCAD 选择集、AutoCAD 图元名、VLA 对象、函数分页表和外部函数等数据类型。

AutoLISP 语言常用数据类型有如下几种:

整数 (INT)

实数 (REAL)

符号 (SYM)

字符串 (STR)

表 (及用户定义的函数) (LIST)

文件描述符 (FILE)

AutoLISP 的内部函数 (SUBR)

AutoCAD 的选择集 (PICKSET)

AutoCAD 的实体名 (ENAME)

函数分页表 (PAGETB)

本节只介绍前 5 种数据类型,其他类型将在后面相应的章节中介绍。

在上述数据类型中,前 4 种称为原子 (ATOM),原子中包括数字原子 (整型数和实型数)、符号原子和串原子。

AutoLISP 语言最基本的数据类型是原子和表,称为符号表达式 (Symbolic-Expression),如图 2-1 所示。其中原子 (ATOM) 是表的最基本元素,它本身只是一个值或符号。



图 2-1 符号表达式

2.1 数据类型

2.1.1 原子

1. 整数

整数 (INT) 由数字和正负号组成,正号可以省略。AutoLISP 支持 32 位有符号整数,范围为 $-2147483648 \sim +2147483647$,如整数超出此范围,计算机将提示出错信息。

注意:在实际应用中,若设定和计算结果超出 AutoLISP 语言的整数范围时,可改用实型数。

2. 实型数

AutoLISP 支持双精度实数,并且至少有 14 位的精度。对于纯小数小数点前面的前导 0 不能省略。