

普通高等教育“十一五”规划教材

# AutoCAD 应用开发技术

翟 震 主 编

毋彩虹 翟 萍 张晓黎 副主编



中国电力出版社

[www.infopower.com.cn](http://www.infopower.com.cn)

普通高等教育“十一五”规划教材

# AutoCAD 应用开发技术

---

程 震 主 编

毋彩虹 程 萍 张晓黎 副主编



中国电力出版社

[www.infopower.com.cn](http://www.infopower.com.cn)

## 内容提要

本书全面地介绍了 AutoCAD 的各种开发技术,包括 Visual LISP、VBA、ARX、图形通用格式 DXF 以及 AutoCAD 固有的自定义方法等,是目前国内比较全面介绍 AutoCAD 开发技术的资料。书中通过大量实例,方便了用户针对各种不同的工程领域,进行工程 CAD 系统的开发。

本书在内容叙述上采用循序渐进、图文并茂的方法,同时强调全书的系统性与实用性,使读者能够迅速、深入地掌握 AutoCAD 的各种开发方法。

本书可作为高等院校本科生、研究生教材,也可用于有关专业的科研、技术人员学习和参考。

## 图书在版编目(CIP)数据

AutoCAD 应用开发技术 / 翟震主编. —北京: 中国电力出版社, 2007.

普通高等教育“十一五”规划教材

ISBN 978-7-5083-5650-1

I. A… II. 翟… III. 计算机辅助设计—应用软件, AutoCAD—高等学校—教材 IV. TP391.72

中国版本图书馆 CIP 数据核字(2007)第 063403 号

丛 书 名: 普通高等教育“十一五”规划教材

书 名: AutoCAD 应用开发技术

出版发行: 中国电力出版社

地 址: 北京市三里河路 6 号

邮政编码: 100044

电 话: (010) 68362602

传 真: (010) 68316497, 88383619

服务电话: (010) 58383411

传 真: (010) 58383267

E-mail: infopower@cepp.com.cn

印 刷: 汇鑫印务有限公司

开本尺寸: 185mm×260mm 印 张: 14.5 字 数: 349 千字

书 号: ISBN 978-7-5083-5650-1

版 次: 2007 年 9 月北京第 1 版

印 次: 2007 年 9 月第 1 次印刷

印 数: 0001—3000 册

定 价: 23.00 元

## 敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

# 前 言

近年来,随着 CAD 应用技术的不断深入,各种工程 CAD 软件层出不穷。其中,基于 AutoCAD 平台进行的二次开发,由于其方法多样灵活,使用方便,已经得到越来越多的重视。

AutoCAD 是目前国内外使用较为广泛的一种计算机辅助设计软件,仅在国内的市场占有率即达到 70%以上,该软件从早期的 1.0 版发展到目前的 2008 版,其良好的开放性结构不但保证了系统本身的不断扩充和完善,而且也方便了用户针对不同的工程领域,进行工程 CAD 系统的开发。主要的开发工具不仅包括编程类的 Visual LISP、VBA、ARX,还有 AutoCAD 固有的自定义文件,以及绘图软件通用格式 DXF 文件。

本书强调系统性、基础性和实用性,在结构和内容上以作者多年来从事 CAD 教学及科研实践中的体会和经验为基础,既注重基础知识的介绍,又结合实际中遇到的问题,将详细介绍与具体实例相结合,在内容上力争做到由浅入深和循序渐进,读者可以根据系统开发的规模,结合自身水平,灵活选择不同的开发工具。

本书可作为高等院校教材,也可作为培训与继续教育用书。读者对象包括在校本科生、研究生,以及有关专业的科研、技术人员。读者在学习的同时,必须进行上机操作和实践,才能收到较好的效果。

本书由翟震担任主编,全书共分六章。其中,第 1 章、第 2 章、第 5 章的第 7 节由翟震编写;第 3 章的第 3、4 节,第 5 章的第 1~6 节由翟萍编写;第 4 章由张晓黎编写;第 3 章的第 1、2 节,第 6 章由毋彩虹编写。

本书在编写过程中得到四川大学陈道洁教授、付守默教授的大力支持,赵海滨同学为本书整理了部分资料,特此表示感谢。

限于作者水平,加之时间仓促,难免有错误与不足之处,恳请读者批评指正。

作 者  
2007 年 5 月

# 目 录

前 言	
第 1 章 绪论	1
1.1 AutoCAD 简介	1
1.2 AutoCAD 二次开发工具简介	3
习题	5
第 2 章 Visual LISP 开发技术	6
2.1 引言	6
2.2 Visual LISP 基本函数	10
2.3 Visual LISP 的绘图功能	20
2.4 逻辑运算与分支函数	26
2.5 循环结构	30
2.6 图形数据库	34
2.7 输入输出函数	40
2.8 DCL 对话框设计	48
习题	58
第 3 章 VBA 开发技术	59
3.1 概述	59
3.2 控制 AutoCAD 环境	62
3.3 创建和编辑 AutoCAD 图元	73
3.4 使用 VBA 开发应用程序	94
习题	100
第 4 章 ARX 开发技术	101
4.1 引言	101
4.2 ObjectARX 程序设计	109
4.3 ARX 图形处理	121
4.4 ARX 应用程序对外部数据库的引用	125
习题	129
第 5 章 外部接口文件 DXF	131
5.1 概述	131
5.2 HEADER 段	133
5.3 CLASSES 段	144
5.4 TABLES 段	145
5.5 BLOCK 段	157

5.6	ENTITIES 段	159
5.7	DXF 接口程序	170
	习题	175
<b>第 6 章</b>	<b>自定义 AutoCAD</b>	<b>176</b>
6.1	概述	176
6.2	组织程序文件和支持文件	177
6.3	自定义线型	183
6.4	自定义填充图案	188
6.5	自定义用户界面	190
6.6	自定义工具栏	196
6.7	创建下拉菜单和快捷菜单	198
6.8	创建宏	200
6.9	自定义状态行	207
6.10	形和形字体	216
	习题	222
	参考文献	223

# 第 1 章 绪 论

## 1.1 AutoCAD 简介

### 1.1.1 AutoCAD 发展概况

20 世纪 80 年代初,美国 Autodesk(欧特克)公司为微机上应用 CAD 技术而开发出 AutoCAD 绘图程序软件包。AutoCAD 作为 CAD 的旗舰产品,在全世界拥有数以百万计的用户,已经成为当今国际上广为流行的绘图工具,并以其独特的优势被全球的设计工程师所采用。

AutoCAD 最初主要用于绘制二维图形,逐步发展到处理三维图形。与传统的手工绘图相比,用 AutoCAD 绘图速度更快、精度更高,而且更富个性化。我国自 20 世纪 80 年代中期引进该软件以来,已经在航空航天、造船、建筑、机械、电子、化工、美术、轻纺等很多领域得到了广泛应用,并取得了丰硕的成果和巨大的经济效益。

AutoCAD 具有良好的用户界面,通过交互菜单、工具栏或命令行方式便可以进行各种操作。它的多文档设计环境,让非计算机专业人员也能很快地掌握。在不断实践的过程中可以全面地掌握它的各种应用和开发技巧,从而不断提高工作效率。

AutoCAD 具有广泛的适应性,它可以在各种操作系统支持的微型计算机和工作站上运行,并支持分辨率为  $320 \times 200 \sim 2048 \times 1024$  的各种图形显示设备 40 多种,以及数字仪和鼠标器 30 多种,绘图仪和打印机数十种,这就为 AutoCAD 的普及创造了条件。

AutoCAD 相继推出的版本如下:

1982 年 11 月,推出 AutoCAD 1.0 版本。

1983 年 4 月,推出 AutoCAD 1.2 版本。

1983 年 8 月,推出 AutoCAD 1.3 版本。

1983 年 10 月,推出 AutoCAD 1.4 版本。

1984 年 10 月,推出 AutoCAD 2.0 版本。

1985 年 5 月,推出 AutoCAD 2.17 版本和 2.18 版本。提供了 AutoLISP 程序设计的支持,从此增加了可以在一个通用的 CAD 平台上添加自己所需的特殊功能的能力。

1986 年 6 月,推出 AutoCAD 2.5 版本。

1987 年 5 月,推出 AutoCAD 2.6 版版本,AutoLISP 得到一些改善,这也正是国内流行 AutoCAD 及 AutoCAD 环境下开发编程的开始。

1987 年 9 月,推出 AutoCAD 9.0 版本和 9.03 版本。

1988 年 8 月,推出 AutoCAD 10.0 版本。使得 AutoLISP 程序可以在扩展内存中允许,这为编写稍大一点规模的应用程序提供了可能。

1990 年 10 月,推出 AutoCAD 11.0 版本。第一次在 PC 版的 AutoCAD 上提供了 C 语言开

发环境 (ADS) 的支持, 这使得在 AutoCAD 的上开发大规模的综合性的应用程序成了可能, AutoCAD 下的应用程序可以利用标准 C 语言提供的能力, 编制出第一代工具无法实现的应用效果。

1992 年 11 月, 推出 AutoCAD 12.0 版本。第一次提供了面向对象的 C++ 开发环境的支持 (ARX), 这就是 AutoCAD 第三代开发工具。AutoCAD 本身在内核上也发生了本质的变化, 尽管最终用户并不会感觉到, 但 AutoCAD 确实走向了一个开放式的面向对象的 CAD 平台, 为今后的进一步发展奠定了技术基础。

1996 年 6 月, 推出 AutoCAD R13 版本。ARX 是个新生事物, 本身不完善, 在不同的小版本上的 ARX 程序不兼容, 直至 R13C4 版才较为稳定。

1998 年 1 月, 推出 AutoCAD R14 版本。R14 对 ARX 的支持更加可靠和稳定, 经过一段时间的摸索, 二次开发商逐步具备开发 ARX 应用程序的能力。另外开始将 VBA 嵌套在 AutoCAD 之中, VBA 是根据微软公司的规则, 是每一个 Windows 下的应用软件都必须具有的功能。

1999 年 1 月, 推出 AutoCAD 2000 版本。将 AutoLisp 发展为 Visual Lisp, 增加了编辑功能, 同时 VBA、ARX 更加完善。

2001 年 9 月, 推出 AutoCAD 2002 版本。

2003 年 5 月, 推出 AutoCAD 2004 版本。

2005 年 10 月, 推出 AutoCAD 2006 版本。

2006 年 3 月, 推出 AutoCAD 2007 版本。

2007 年 3 月, 推出 AutoCAD 2008 版本。

AutoCAD 2008 是 Autodesk 公司推出的、目前较为成熟的版本, 它扩展了 AutoCAD 以前版本的优势和特点, 并且在用户界面、性能、操作、用户定制、协同设计、图形管理、产品数据管理等方面进一步得到加强。

### 1.1.2 AutoCAD 基本绘图功能

(1) 提供了绘制各种二维图形的工具, 并可以根据所绘制的图形进行测量和标注尺寸, 具备对图形进行修改、删除、移动、旋转、复制、偏移、修剪、圆角等多种强大的编辑功能。

(2) 缩放、平移等动态观察功能, 并具有透视、投影、轴测图、着色等多种图形显示方式。

(3) 提供栅格、正交、极轴、对象捕捉及追踪等多种精确绘图辅助工具。

(4) 提供块及属性等功能提高绘图效率, 对于经常使用到的一些图形对象组可以定义成块并且附加上从属于它的文字信息, 需要的时候可反复插入到图形中, 甚至可以仅仅修改块的定义便可以批量修改插入进来的多个相同块。

(5) 使用图层管理器管理不同专业和类型的图线, 可以根据颜色、线型、线宽分类管理图线, 并可以控制图形的显示或打印与否。

(6) 可对指定的图形区域进行图案填充。

(7) 提供在图形中书写、编辑文字的功能。

(8) 创建三维几何模型, 并可以对其进行修改和提取几何和物理特性。

(9) 动态输入功能, 具备了当前主流 CAD 软件的一般特征。



## 1.2 AutoCAD 二次开发工具简介

AutoCAD 具备强大的用户定制功能, 提供多种方式以使用户按照自己的思路去解决问题。AutoCAD 开放的平台使用户不仅可以用 Visual LISP、VBA、ARX 等语言开发适合特定行业使用的大型 CAD 产品, 也可以利用其自定义功能设计小型的参数化图库, 还能使用各种编程工具生成 DXF 格式文件与 AutoCAD 交互。

### 1.2.1 Visual LISP 程序设计

LISP 语言是早期的人工智能语言之一, 为更好地开发 AutoCAD, 自 AutoCAD2.18 版以后均内嵌了 AutoLISP 程序, 是第一种能对 AutoCAD 进行二次开发的语言, 在我国有相当多的软件都是采用 AutoLISP 程序编写的。到 2000 版发展为 Visual LISP 程序, 向下兼容各版本的 AutoLISP 程序, 适合编写小型的 CAD 软件。

Visual LISP 语言最典型的应用之一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。另一个典型应用就是驱动 AutoCAD 提供 PDB 模块构成 DCL (Dialog Control Language) 文件创建自己的对话框。

#### 1. Visual LISP 的优点

- (1) 语言规则十分简单, 易学易用。
- (2) 直接针对 AutoCAD, 易于交互。
- (3) 解释执行, 立竿见影。

#### 2. Visual LISP 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, 跟人的思维方式也不一致。
- (5) 不能直接访问硬件设备、进行二进制文件的读写。

Visual LISP 的这些特点, 使其仅适合于有能力的终端用户完成一些自己的开发任务。

### 1.2.2 VBA 程序设计

VBA (Visual Basic for Application) 是 2000 版以后具备的功能。VBA 通过 AutoCAD ActiveX Automation 接口将消息发送到 AutoCAD。允许 VBA 环境与 AutoCAD 同时运行, 并通过 ActiveX Automation 接口对 AutoCAD 进行编程控制。它是一个引入了对象的设计环境, 为用户提供类似 Visual Basic (VB) 语言所拥有的功能, VBA 可以执行并存取 AutoCAD 环境的所有资源, 适合编写中型的数据库类的 CAD 软件。

其优点是可以很容易地使用 Windows 系统资源, 使 AutoCAD 能较好地与其他 Windows 下的应用软件交互。VBA 对话框设计能力比 DCL 好得多, 多图档环境下的应用程序也比 VLISP 提供了更多的支持。但由于它是使用 ActiveX 与 AutoCAD 交互, 程序设计比较复杂。会成为与 VLISP 并列的另一种专业用户程序设计语言, 不太容易掌握。因为 VBA 并不等于

VB, 所以熟悉 VB 的人, 在使用 VBA 作 AutoCAD 程序设计时, 也需要学习大量的新知识, 只不过 VBA 的规则与 VB 比较相近。

在设计模拟类的程序设计中, VBA 明显不如 AutoLISP 那样顺利, 毕竟 AutoLISP 脱胎于人工智能语言。另外, 如果 VBA 作者因为“疏忽”, 没有提供相应的方法和属性, AutoLISP 能够访问的对象数据、能够激活某些 AutoCAD 对话框, VBA 却不能。VBA 虽然不是编译运行, 数据结构和程序写法却像 ARX 一样严格, 这对于工程师来说, 需要有一个适应过程, 只能按照 VBA 的特点去组织自己的数据, 而不能像 VLISP 那样按照专业设计要求去创建数据结构。

VBA 中不能像 VLISP 那样随心所欲地使用 AutoCAD 命令, 只能像脚本文件的模式一样, 以字串方式向 AutoCAD 命令行发送一串响应, 这个字串的内容也只能在自己的程序中生成和进行数据转换。这样, 许多交互操作的技巧, 在 VBA 中不能直接借用。

VBA 是以对象作为数据处理的基本单元, 在 VLISP 中堪称特色的 CAGD 程序设计技术, 在 VBA 中则无法实现。如 Trim、Extend、Dim-Diameter 之类的命令都不能真正实施, 要想完成这样的操作, 相关数据只能在程序中自己计算。这样, 程序设计就不能充分利用 AutoCAD 的操作技巧了。

### 1.2.3 ARX 程序设计

ObjectARX® (AutoCAD 运行时扩展) 是开发 AutoCAD 应用程序的编译语言编程环境。ObjectARX 编程环境包括许多动态链接库 (DLL), 它们和 AutoCAD 运行在同一地址空间, 并直接操作 AutoCAD 内核数据结构和代码。这些库利用 AutoCAD 的开放式体系结构, 提供对 AutoCAD 数据库结构、图形系统和 AutoCAD 几何图形引擎的直接访问权限, 以扩展 AutoCAD 在运行时的分类和功能。另外, 也可以使用 DLL 来创建新命令, 这些新命令的操作方式与 AutoCAD 的基本命令操作方式相同。

ObjectARX 库可以与其他 AutoCAD 编程接口 (如 AutoLISP 或 VBA) 结合使用, 从而启用跨 API 集成, 适合编写大型的 CAD 集成软件。

由于 ARX 是使用 Visual C++ 作为基本程序设计语言, 无论其效率、数据处理和硬件的控制能力都是最好的, 是唯一能针对多图档环境的程序设计方法, 是从根本上扩展 VLISP / VBA 的程序设计方法。

但是, 与 AutoCAD 命令交互和对象数据库访问功能, ARX 与 VBA 并没有实质性的区别, 就是说, 同样很难借用 AutoCAD 强大的 CAGD 功能和交互操作的技巧。另外, 要想真正发挥这种开发模式的作用, 对于程序设计者的要求 (计算机专业知识、软件专业知识等) 也较高, 而且程序运行风险较大。要求程序自己做好各种数据的测试, 自己设计好各种条件下的出错处理和恢复, 否则, ARX 程序的崩溃常常会连带 AutoCAD, 甚至操作系统一起崩溃。由于 ARX 技术是大部分开发者的工具, 因此, 真正有用的编程技巧也不大容易通过一般的交流得到。可见, 这种开发模式不适合在专业设计工程师中普及。

### 1.2.4 AutoCAD 自定义功能

利用 AutoCAD 的自定义功能可以实现自定义工具栏、创建命令别名、自定义线型、自定义填充图案、自定义用户界面、自定义状态行等, 可以解决用户设计中的界面、图库等问题。

### 1.2.5 DXF 格式文件

DXF 文件是文本或二进制文件，其中包含可由其他 CAD 程序读取的图形信息，可以由其他高级语言生成。如果其他用户正使用能够识别 DXF 文件的 CAD 程序，那么以 DXF 文件保存的图形就可以共享该图形，因此 DXF 文件可以作为不同软件之间的接口文件使用。

## 习 题

1. AutoCAD 主要经过哪几个发展阶段？各有什么特点？
2. AutoCAD 开发工具各自的优缺点是什么？

## 第 2 章 Visual LISP 开发技术

### 2.1 引 言

#### 2.1.1 Visual LISP 概述

LISP 是一种计算机的表处理语言，它主要是应用于人工智能领域，最早是 1960 年由美国麻省理工学院的 John McCarthy 设计实现的，至今仍在广泛使用。AutoLISP 是 AutoCAD 内嵌的一种解释语言，由美国 Autodesk 公司开发，自 AutoCAD2.18 版开始应用，其目的是使用户充分利用 AutoCAD 进行二次开发，它采用了与 LISP 语言中的 Common LISP 最接近的语法和习惯约定，同时又针对 AutoCAD 增加了许多新的功能，可以直接调用大部分 AutoCAD 命令，因此它既具有一般高级语言的基本结构和功能，简单易学，又具有 AutoCAD 强大的图形处理能力，国内很多科研院校以 AutoLISP 为语言开发了很多专业的 CAD 系统，取得很好的效果。

然而计算机技术在不断地向前发展，程序的规模越来越大，逻辑结构也更加复杂，AutoCAD 中的二次开发也不再仅仅局限于交互性了。由于 AutoLISP 是一种解释语言，决定了它的运行速度比较慢；解释运行导致的另一个缺点是 AutoLISP 程序不能编译，软件开发者的劳动成果得不到保护。随着 CAD 应用层次的提高，不但要求 AutoCAD 能画图，而且要进行相关的数据处理，这并不是 AutoLISP 的长项。AutoLISP 还有一个比较大的缺点就是使用不方便，由于 AutoCAD 本身并未提供一个专门的 AutoLISP 程序设计环境，致使程序的编写、加载和调试都不方便。

为了克服以上问题，并适应 AutoCAD 环境下的较大型用户 CAD 应用系统的开发，AutoDesk 公司于 1998 年 3 月推出 Visual LISP 语言，加在 AutoCAD2000 版上发布。新设计的 Visual LISP 采用与 AutoLISP 完全兼容的模式，原来的 AutoLISP 程序可以在 Visual LISP 环境下运行，同时还新增了许多函数和系统变量，使得 AutoCAD 的开发工作更加容易。

Visual LISP 的主要工具和环境包括以下五个方面：

(1) Visual LISP 采用加载即编译技术，实现了与 AutoLISP 完全兼容的功能。

(2) Visual LISP 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 原始程序，并通过色彩编码对原始程序的不同部分加以颜色区分，以增强 AutoLISP 原始程序的可读性。

(3) Visual LISP 支持多种检查器。其中语法检查器可以用来检查 Auto LISP 程序结构错误和内部函数的变量错误。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，具有灵活的特性，它可以在一个窗口单一执行 AutoLISP 的源代码，而在 AutoCAD 窗口中同时显示程序代码所执

行的效果。

(5) Visual LISP 先进的原始程序编辑器可以将 AutoLISP 的源程序编译成二进制文件, 将改善程序的执行速度和安全性。Visual LISP 的原始文件 (.LSP) 和已编译文件 (.FAS) 都可以利用系统提供的 Application Wizard 软件, 将其包装成一个单一的 ARX 模块。

综上所述, Visual LISP 是一种将 AutoLISP 语言的优点完全保留、缺点完全克服, 并与最新的程序设计相结合的全新的整合开发系统, 为用户开发的 AutoLISP 程序不至于废弃, AutoDesk 公司宣布将 Visual LISP 作为 AutoCAD 的语言标准, 在今后的版本中始终保留。

### 2.1.2 Visual LISP 的数据类型

Visual LISP 的数据类型主要有以下九种。

#### 1. 整型数 (INT)

整型数是由 0、1、2、3、4、5、6、7、8、9、+、- 等字符组成, 不包含小数点。Visual LISP 的整数是 32 位带符号的数, 取值范围从 +2147483647 到 -2147483648 (注意: getint 函数只接受 16 位的数, 即 +32767 到 -32678)。当用户在 Visual LISP 表达式中直接使用整数时, 该值被称为常量。数字 2、-56 和 1200196 都是有效的。

23 -105 0 5000

以上是合法的整型数。

8A 20-5 34.6

以上是非法的整型数。

#### 2. 实型数 (REAL)

实型数是用整型数后跟小数组成, 若实数的数值小于 1, 则小数点前必须加 0, 例如:

35.6 -26.8 0.05

以上是合法的实型数。

45. 5+0.4 .A

以上是非法的实型数。

#### 3. 符号 (SYM)

相当于其他高级语言的变量, 符号包括除 "("、")"、"'"、";" 之外的任何打印字符, 其长度没有限制, 为了节省内存空间, 一般不超过 6 个, 字母的大小写等效。在其他高级语言中, 所有的符号或变量都需要事先定义, 且在使用中不能相互改变, 而 Visual LISP 语言中的符号不需要事先定义, 仅通过简单命名就可以使用, 并且可以在程序的任何位置相互转化, 例如:

abc A18 18A a-b\_1 a+b 25? +-\*/

以上是合法的符号, 但最好采用匈牙利命名法定义的符号 (变量)。

(a1 a;b

以上是非法的符号。

#### 4. 字符串 (STR)

字符串是由双引号引起来的字符组成的, 例如:

"abc" "a12BCD345" "A=B"

以上是合法的字符串。

说明：前四种数据类型在 Visual LISP 语言中被称为原子。原子中包含数字原子（整数数和实型数）、符号原子和串原子。原子和下面的表是 Visual LISP 语言最基本的数据，它们又总称为符号表达式（S-表达式）。

### 5. 表（及用户定义的函数）(LIST)

表是指放在一对相匹配的左右括号中的一个或多个元素的有序集合。表中的每一个元素可以是任何整数、实数、字符串或表。元素和元素之间要用空格隔开，元素和括号之间可以不用空格。例如：

```
(A ("B" C)、(1 2 3) 25)
```

上例表中有四个元素，即 A、("B" C)、(1 2 3)、25。其中，第二、三个元素又是一个表，表中的元素几乎无限。表可以嵌套很多个表，从外往里依次称为 0 层（也称顶层）、1 层、2 层、…，一般表中元素是指表中的顶层元素。

表中元素是有顺序的。为便于对表中的元素进行存取，每一个元素都有一个序号，从左向右，第一个元素序号为 0，第二个元素序号为 1，以此类推，第 n 个元素序号为 n-1。

表的大小为表的长度，即表中顶层元素的个数。没有任何元素的表称为空表。空表用()或 nil 表示。在 Visual LISP 语言中 nil 是一个特殊的符号原子，它既是表又是原子。

表有两种类型：引用表和标准表。

(1) 标准表是 Visual LISP 程序的基本结构形式，Visual LISP 程序就是由标准表组成的，标准表用于函数的调用，其中第一个元素必须是系统内部函数或用户定义的函数，其他元素为该函数的参数。例如，下面算术函数的调用，即采用标准表的形式。

```
(+ 1 2 3)
```

表中第一个元素+为系统内部定义的加函数，1、2 和 3 为+加函数的参数。

(2) 引用表的第一个元素不是函数，常用作数据处理。在 Visual LISP 语言中，引用表相当于其他高级语言的数组。例如，表示图中点的坐标，二维点的坐标是用两个实型数构成的表(25.0 30.0)，表中第一个元素表示点的 X 坐标，第二个元素表示点的 Y 坐标。引用表一般用定义的方法产生，或用禁止求值符号定义。

(3) 点对也是一种表，表中有两个元素，中间为一圆点"."，圆点与元素之间用空格分开。例如：(A . B)，点对常用于构造联结表（修改实体等）。

### 6. 文件描述符 (FILE)

在打开文件时的返回值，相当于其他高级语言的文件号，用于数据的存取。

### 7. Visual LISP 的内部函数 (SUBR)

Visual LISP 的内部函数包括算术运算函数、表处理函数、循环函数、条件函数、字符串函数、图形函数、实体处理函数、输入输出函数等。

### 8. AutoCAD 的选择集 (PICKSET)

相当于 AutoCAD 中的目标选择，将所有选中的目标集中在一起，可以对其中任意实体进行操作。

### 9. AutoCAD 的实体名 (ENAME)

实体名是一个指针，指向由 AutoCAD 图形编辑器程序维持的文件指针。通过这个指针，可以找到该实体在图形数据库中的记录及在屏幕上的向量。在参数化设计中，一般通过修改图

形数据库来实现图形的变化，要比直接用编辑命令快速、准确。

### 2.1.3 简单的 Visual LISP 程序介绍

Visual LISP 语言没有“语句”术语，相当于其他高级语言的语句，一律采用标准表的形式，Visual LISP 程序往往是由一系列按顺序排列的标准表所组成的。例如：

```
计算 2 + 3 = ?
(defun fx1(a b / x y)      ;注意：a、b、/、x、y 之间一定要有空格
  (setq x a)              ;x=a
  (setq y b)              ;y=b
  (+ x y)                 ;计算 x+y
)                          ;defun
```

可以看出，Visual LISP 程序有以下特点。

- (1) 函数必须放在表中第一元素的位置。
- (2) 在 Visual LISP 程序中，一行可写多个标准表，也可将一个标准表分几行写。例如，上面程序可写成以下形式：

```
(defun fx0(a b / x y)(setq x a)(+ x y))
```

- (3) 注释语句前用英文“;”标明，在本行末尾结束。
- (4) defun 是系统内部函数，利用它可以定义用户自己的函数，如 fx1。
- (5) fx1 后面是参数表，其中 a、b 是形参，x、y 是局部变量，它们与分隔符“/”之间必须有空格，并且 a、b、x、y 在该函数外无效。参数表可以为空，如 fx1()。
- (6) Visual LISP 程序是以扩展名为“.LSP”的 ASCII 码文本文件的形式存储。

### 2.1.4 上机步骤

#### 1. 程序编辑

由于 Visual LISP 程序是纯文本文件，所以可用任何一种 ASCII 码文本编辑器来编辑，一般在 DOS（运行 cmd）环境下用 EDIT 编辑，在 Windows 2000 环境下用附件中的“记事本”来编辑，注意在存盘时加上文件名后缀.LSP。

打开 AutoCAD 2000 及以上版本，点下拉菜单中的“工具”|“AutoLISP(S)”|“Visual LISP 编辑器(V)”，再点“新建”|“文件”，即可进入 Visual LISP 编辑状态，程序写好后存盘。注意：不要在 Visual LISP 控制台下编辑程序。

#### 2. 程序运行

在 AutoCAD 中的 Command 提示符下输入（下划线部分）：

```
Command:(load "文件名")
```

文件名可不加后缀，若文件在子目录中，则输入：

```
Command:(load "c:/子目录 1/子目录 2/文件名") 或
```

```
Command:(load "c:\\子目录 1\\子目录 2\\文件名")
```

这样就将 LISP 程序装入内存，再输入函数名即可运行。以上述程序为例，如计算 2+4=?，可在 Command 下输入：

```
Command:(fx1 2 4)
```

则输出

6

还可以利用菜单-工具- AutoLISP(S)-加载(V)，将编辑好的程序调入并运行。

## 2.2 Visual LISP 基本函数

Visual LISP 程序就是对各个函数的调用，所以 Visual LISP 语言又称为函数式语言。函数是 Visual LISP 语言处理数据的工具，学习和掌握 Visual LISP 语言，核心就是要掌握其函数。

Visual LISP 函数分为系统内部函数和用户自定义的外部函数。Visual LISP 提供了大量的系统内部函数，以满足编程的需要。

Visual LISP 语言调用函数是通过标准表实现的，其一般格式如下：

(函数名 [<参数 1>] [<参数 2>]... [<参数 n>])

标准表中的第一个元素必须是函数名，系统的求值器根据函数的特性进行求值并将结果作为返回值（计算结果）返回，函数中在函数名后面的各元素为该函数的参数，可以是具体数值，也可以是其他函数，参数的类型与数目取决于函数。

### 2.2.1 数值函数

数值函数用于处理整型数或实型数两种类型，它包括基本标准函数、三角函数及布尔函数。数值函数总是返回数的数据类型值，返回值是整型数或实型数取决于参数表中参数的数据类型。

数的运算遵循以下原则：

(1) 对于算术运算和幂次方函数，若参数表中的所有参数都为整型数，则求值器对参数表中的参数做整数运算，返回整数数值。

(2) 若参数表中有一个实型数，或者是进行指数运算、三角函数运算、幂运算等，则求值器对参数表中的参数做浮点运算，返回实型数。

(3) 若参数表中的参数多于两个，则从左到右，每两个参数进行数值运算，再把运算结果与下一个参数进行运算，直到最后一个参数为止。

(4) 进行求值运算应防止整型数超界。

### 2.2.2 算术运算函数（或符号）

#### 1. 加函数+

格式：(+ <数 1> <数 2> ...)

功能：对各数进行加法运算。例如：

Command: (+ 1 2 3)

6

Command: (+ 2.3 4.5)

6.8

Command: (+ 2 3.5 4)

9.5



## 2. 减函数 -

格式: (-<数 1><数 2>...)

功能: 用<数 1>减去<数 2>及以后的参数。例如:

```
Command: (- 10 2 3)
5
Command: (- 12.5 4.5)
8.0
Command: (- 2 3.5 4)
-5.5
Command: (+ 2 (- 4.5 1))
5.5
```

## 3. 乘函数 \*

格式: (\*<数 1><数 2>...)

功能: 对各数进行乘法运算。例如:

```
Command: (* 2 3 4)
24
Command: (* 12.5 (+ 2.5 1.5))
50.0
```

## 4. 除函数 /

格式: (/<数 1><数 2>...)

功能: 用<数 1>除以<数 2>及以后的参数。例如:

```
Command: (/ 8 2)
4
Command: (/ 10 4)
2
Command: (/ 10 4.0)
2.5
Command: (/ 16 5 2.0)
1.6
Command: (/ (/ 16 5) 2.0)
1.5
```

在第四个例子中, 求值器先进行 16 除以 5 的运算, 由于参数表中有一个实型数, 结果为 3.2, 再进行 3.2 除以 2.0 的运算, 结果为 1.6。

在第五个例子中, 求值器先进行 16 除以 5 的运算, 由于该参数表中全部是整型数, 结果为 3, 再进行 3 除以 2.0 的运算, 结果为 1.5。

**注意:** 要防止除数为零的情况。例如:

```
Command: (/ 25 (/ 4 9))
; 错误: 除数为零
Command: (/ 25 (/ 4 9.0))
56.25
```

## 5. 求余数函数 REM

格式: (rem <数 1><数 2>...)