

· 高等学校计算机基础教材精选 ·

Visual LISP程序设计 (第2版)

李学志 主编

方戈亮 孙力红 编著



清华大学出版社

· 高等学校计算机基础教材精选 ·

Visual LISP程序设计 (第2版)

李学志 主编

方戈亮 孙力红 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍了 AutoCAD 最新版本(2010)的 Visual LISP 程序设计技术。全书共分 12 章,前 3 章介绍 AutoLISP 语言的基础知识,第 4 章介绍 Visual LISP 的开发环境,第 5 章~第 7 章介绍程序的编辑、调试和设计的方法与技巧。第 8 章介绍了如何定义新的 AutoCAD 命令及创建图层、线型、文字样式、剖面线、尺寸标注等各种 AutoCAD 对象,以及如何实现参数化图形设计的方法和技术,第 9 章介绍如何访问和修改图形数据库,第 10 章介绍如何使用 ActiveX 对象,第 11 章介绍如何将反应器附着到 AutoCAD 图形对象上,第 12 章介绍对话框设计技术。

本书具有实用性强和便于教学的特点,既可作为大学本科、专科或继续教育的教材,也可作为从事 CAD 领域工作的工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Visual LISP 程序设计 / 李学志主编;方戈亮,孙力红编著. —2 版. —北京:清华大学出版社, 2010.10

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-23368-8

I. ①V… II. ①李… ②方… ③孙… III. ①LISP 表处理语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 152977 号

责任编辑:焦虹 徐跃进

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260

印 张:22.5

字 数:554 千字

版 次:2010 年 10 月第 2 版

印 次:2010 年 10 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

产品编号:038982-01

出版说明

高等学校计算机基础教育教材精选

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验优秀的教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,包括面向各高校开设的计算机必修课、选修课以及各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本,并保持不断更新),坚持宁缺毋滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是 jiaoh@tup.tsinghua.edu.cn。联系人:焦虹。

清华大学出版社

前言

Visual LISP 程序设计(第2版)

Visual LISP 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,可以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 既兼容以前的 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、参数化的功能。

目前,还有一些用户停留在将 AutoCAD 作为绘图工具的阶段,逐条线、逐个圆地将图纸录入计算机,这样做只是提高了绘图效率,减轻了设计人员的劳动强度,距离真正意义上的计算机辅助设计尚有较大差距,因为设计人员的主要精力本应用于产品的构思与创新设计中。

如何将繁琐的绘图、标注与复杂的分析计算由程序来完成,实现图形参数化、智能化、分析计算与绘图一体化,如何提高产品的自动设计成分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。对于初学者可以从调试、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企业培训的背景下编写的。作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培训的特点。

本书的程序是在 AutoCAD 2010 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较早的 AutoCAD 版本。

由于本书具有上述特点,因此其适用面较宽,既可作为大学本科、专科或继续教育的教材,也可作为从事 CAD 领域的工程技术人员的参考书。

本书共 12 章。其中第 1~第 5 和第 12 章由李学志编写,第 6~第 8 章由孙力红编写,第 9~第 11 章和附录由方戈亮编写。李学志负责全书的统稿工作。在编写本书的过程中,作者得到了清华大学计算机辅助教学中心的大力支持,在此深表感谢。

感谢读者选择使用本书,欢迎对本书的内容提出宝贵的意见和建议。如果在阅读本书时发现问题,或者在实际应用时遇到困难,请登录清华大学出版社网站(www.tup.com.cn)与我们联系。

作者

2010 年 8 月于清华园

目录

Visual LISP 程序设计(第2版)

第1章 概述	1
1.1 关于 LISP	1
1.2 关于 AutoLISP	1
1.3 关于 Visual LISP	2
第2章 数据类型、表达式和函数	3
2.1 数据类型	3
2.2 变量	5
2.2.1 符号	5
2.2.2 变量的数据类型	5
2.2.3 为变量赋值	6
2.2.4 预定义的符号	6
2.2.5 显示变量的值	7
2.2.6 在交互方式下将变量的值传递给 AutoCAD	7
2.3 表达式	8
2.3.1 表达式的构成	8
2.3.2 表达式的前缀表示法	8
2.3.3 表达式的求值过程	8
2.3.4 表达式的求值规则	9
2.4 函数	10
2.4.1 定义 AutoLISP 函数	10
2.4.2 调用 AutoLISP 函数	11
2.4.3 递归定义 AutoLISP 函数	11
2.4.4 调用 AutoCAD 命令	12
2.4.5 定义 AutoCAD 命令	12
习题	13
第3章 程序的流程控制和 AutoLISP 文件	14
3.1 程序的流程控制	14

3.1.1	分支结构	14
3.1.2	循环结构	16
3.2	AutoLISP 程序文件	17
3.2.1	AutoLISP 文件的特点	17
3.2.2	程序中的注释	18
3.2.3	在 AutoCAD 环境下加载 AutoLISP 文件	19
习题		21
第 4 章	有关 Visual LISP 的基本操作	23
4.1	进入和退出 Visual LISP	23
4.2	Visual LISP 的用户界面	24
4.3	控制台操作	25
4.4	文件操作	26
习题		27
第 5 章	编辑源程序代码	28
5.1	文本编辑工具	28
5.2	文本操作	30
5.3	设置代码格式	32
5.4	语法检查	34
习题		35
第 6 章	调试程序	36
6.1	概述	36
6.2	监视窗口	37
6.3	在不设置断点的情况下分步调试程序	39
6.4	断点循环	41
6.5	利用断点调试程序	42
6.6	跟踪程序运行	44
6.7	修改变量和函数的特性	50
6.8	检验窗口	52
6.9	浏览 AutoCAD 对象	54
习题		56
第 7 章	建立应用程序	57
7.1	通过 vlisp_compile 函数编译 LISP 文件	57
7.2	用应用程序生成器创建应用程序	60

7.3 利用工程管理器管理和维护 LISP 文件	64
7.3.1 工程管理器功能	64
7.3.2 创建一个工程	66
7.3.3 工程窗口	67
7.3.4 通过工程创建应用程序	68
7.3.5 优化应用程序	68
习题	70
第 8 章 编程实例	71
8.1 设置作图环境	71
8.2 设置图层、颜色、线型和线宽	73
8.3 定义字样	77
8.4 尺寸标注	80
8.5 调用 AutoCAD 的图形编辑功能	87
8.6 调用图案填充功能	90
8.7 定义绘制曲线的命令	91
8.8 定义参数化绘图的命令	93
8.9 读写文件功能	99
习题	103
第 9 章 访问和修改图形数据库	104
9.1 获取图元的名字	104
9.2 图元表	105
9.2.1 获取图元表	106
9.2.2 对图元表的操作	107
9.2.3 举例	108
9.3 选择集	109
9.3.1 常用的有关选择集的函数	109
9.3.2 举例	110
9.4 符号表	111
9.4.1 符号表操作函数	111
9.4.2 举例	112
习题	113
第 10 章 在 AutoLISP 中使用 ActiveX 对象	114
10.1 AutoCAD 对象模型	114
10.2 通过函数访问 AutoCAD 对象	115
10.2.1 访问 AutoCAD 应用程序	116

10.2.2	访问其他 AutoCAD 对象	117
10.3	使用工具了解 AutoCAD 对象	120
10.3.1	使用检验工具了解 AutoCAD 对象	120
10.3.2	通过 Help 功能了解 AutoCAD 对象	122
10.4	在 Visual LISP 函数中使用 ActiveX 方法	123
10.4.1	通过 Help 功能详细了解 AutoCAD 图形对象的方法	123
10.4.2	将 Visual Basic 环境下的语句改写为 AutoLISP 表达式	124
10.4.3	ActiveX 能够自动进行转换的 AutoLISP 的一些数据类型	124
10.4.4	Visual LISP 扩充的数据类型	125
10.4.5	AutoCAD 图元名和 VLA 对象之间的转换	130
10.5	获取和修改图形对象的特性	131
10.5.1	获取图形对象特性	131
10.5.2	修改图形对象的特性	132
10.6	确定是否可以修改对象	133
10.7	用参数传递返回值的 ActiveX 方法	134
10.8	判断某方法或特性是否可应用于某对象	135
10.9	使用集合对象	137
10.9.1	将某一个函数应用到集合中的每一个对象	137
10.9.2	将一系列函数应用到集合中的每一个对象	138
10.9.3	获取集合中的成员对象	139
10.10	释放 VLA 对象和释放内存	140
10.11	处理 ActiveX 方法返回的错误	140
10.12	举例	142
	习题	143

第 11 章	使用反应器	144
11.1	反应器	144
11.2	定义反应器的回调函数	146
11.2.1	定义对象反应器的回调函数	146
11.2.2	定义其他反应器的回调函数	147
11.2.3	AutoCAD 预定义的回调函数	147
11.3	创建反应器	148
11.3.1	创建对象反应器	148
11.3.2	创建其他反应器	151
11.3.3	将数据附着到反应器对象	152
11.4	查询、修改和控制反应器的状态	152
11.4.1	查询反应器	152
11.4.2	修改反应器	153

11.4.3	控制反应器的状态	155
11.5	临时反应器和永久反应器	156
11.6	反应器的使用规则	157
11.7	定义反应器实例	159
	习题	162
第 12 章	对话框设计	164
12.1	概述	164
12.2	对话框控件	165
12.2.1	可以定义动作行为的控件	165
12.2.2	组合类控件	168
12.2.3	装饰性和信息类控件	172
12.2.4	AutoCAD 预定义的控件	175
12.3	控件的属性	177
12.4	对话框设计的原则与规范	184
12.4.1	一般原则	184
12.4.2	通用规范	185
12.4.3	预定义控件的使用规范	186
12.5	对话框控制语言	188
12.5.1	对话框文件	188
12.5.2	DCL 文件的结构	189
12.5.3	DCL 语法	190
12.6	对话框驱动程序	192
12.6.1	驱动对话框的程序流程图	192
12.6.2	对话框驱动函数	193
12.6.3	调用含有对话框的 AutoLISP 函数	197
12.6.4	编写对话框回调函数的注意事项	199
12.6.5	对话框的特殊处理	202
12.7	几种典型控件的应用实例	209
	习题	226
附录 A	AutoLISP 函数	227
A1	赋值、求值与禁止求值函数	227
A2	数值计算函数	228
A3	关系运算函数	231
A4	逻辑运算函数	232
A5	几何运算函数	234
A6	判断函数	237

A7	流程控制函数	239
A8	表处理函数	240
A9	字符串处理函数	242
A10	数据类型转换函数	245
A11	交互输入函数	247
A12	有关文件操作的函数	252
A13	打印输出函数	255
A14	有关函数的函数	256
A15	处理错误函数	260
A16	选择集操作函数	261
A17	与 AutoCAD 直接相关的函数	265
A18	图元操作函数	270
A19	符号表操作函数	273
A20	扩展数据操作函数	273
A21	内存管理函数	274
A22	对话框操作函数	276
A23	菜单控制函数	279
A24	数字化仪控制函数	280
A25	控制环境变量的函数	280
A26	词典操作函数	281
A27	应用程序操作函数	283
A28	Visual LISP 扩展函数	285
A29	有关 ActiveX 的函数	304
A30	与反应器有关的函数	319
 附录 B AutoLISP 程序出错代码		 339
 参考文献		 342

2. T

T 为常量,它作为逻辑变量的值,表示成立,相当于其他程序设计语言的 true。

3. Pause

Pause 与 command 函数配合使用,用于暂停,等候用户的输入。

4. Pi

Pi 被定义为常量 π 。

每个变量都占用一些内存空间。为了节省存储空间,可重复使用变量名,或者将不再使用的变量设置为 nil,以便释放该变量所占的内存空间。

必须注意,一般的编程语言不允许将内部函数名或流程控制的关键字作为变量名,而 AutoLISP 没有这样的限制。因此,为变量命名时不要与 AutoLISP 定义的函数名和预定义的符号名相同,否则,后面的定义将取代已有的定义,从而引起混乱。例如,在执行表达式 (setq sin 0 Pi 0) 之后, sin 不再是正弦函数, Pi 也不再是圆周率,而是值为 0 的整型变量。

2.2.5 显示变量的值

要想在 AutoCAD 命令提示区显示变量的值,需要在变量名前添加叹号!。例如了解前面已赋值的变量 x、v 的值,操作如下:

命令:

```
!x ↵  
1.5
```

命令:

```
!v ↵  
"Center"
```

命令:

```
!z ↵  
nil (假定变量 z 尚未被定义)
```

2.2.6 在交互方式下将变量的值传递给 AutoCAD

在变量前加一个感叹号“!”,即可将变量的值传递给 AutoCAD。例如:

命令:

```
(setq p '(20 10) r 5) ↵ ;为变量 p 和 r 赋值
```

1.1 关于 LISP

LISP(List Processing Language)是广泛应用的一种程序设计语言,主要用于人工智能、机器人、专家系统、博弈、定理证明等领域。LISP 也被称为符号式语言,因为它处理的对象是符号表达式。LISP 语言语法简单,编写程序便捷、灵活,数据类型丰富,利用 LISP 语言可以很容易地定义或调用新的函数。

LISP 在它几十年的发展过程中产生了多种版本,如 MacLISP、InterLISP、ZetaLISP、和 CommonLISP 等。其中 CommonLISP 是近几年美国的几所大学和工业界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,因而功能较强且拥有各版本的一些优点,是目前 LISP 语言较完整的版本。

1.2 关于 AutoLISP

AutoLISP 是为二次开发 AutoCAD 而专门设计的编程语言,它起源于 LISP 语言,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 的特点,但又针对 AutoCAD 增加了许多功能。它既有 LISP 语言人工智能的特性,又有 AutoCAD 强大的图形编辑功能的特点。它可以把 AutoLISP 程序和 AutoCAD 的绘图命令透明地结合起来,使设计和绘图完全融为一体,还可以实现对 AutoCAD 图形数据库的直接访问和修改。

利用 AutoLISP 可以进行工程分析计算,可以自动绘制复杂的图形、定义新的 AutoCAD 命令,可以驱动对话框、控制菜单。通过 AutoLISP 可以为 AutoCAD 扩充具有一定智能化、参数化的功能,使设计人员的主要精力用于产品的构思和创新设计上,实现真正意义上的计算机辅助设计。

1.3 关于 Visual LISP

Visual LISP 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。Visual LISP 包括文本编辑器、格式编排器、语法检查器、源代码调试器、检验和监视工具、文件编译器、工程管理系统、上下文相关帮助、自动匹配功能和智能化控制台等。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 兼容以前的 AutoLISP 程序。在 Visual LISP 集成环境下开发 AutoLISP 程序,就不再像以前那样编辑程序时,借用其他系统的文本编辑程序编写程序代码;调试程序时需要用户自己决定在程序的什么位置插入打印语句,以便查看变量的内容;在程序运行正常后,还必须将插入的调试代码删除或注释掉。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更高、代码更加紧凑、源代码受到保护的应用程序。

Visual LISP 是新一代 AutoLISP,它对 AutoLISP 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,或者将调用该函数的表达式写在 `acad2010doc.lsp` 文件内。

AutoLISP 的数据类型丰富,除了具有一般程序设计语言的整型、实型、字符串等数据类型之外,还有表、函数、文件描述符、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。

2.1 数据类型

1. 整型(INT)

整型即整数,整数由数字和正负号组成,正号+可缺省。整数为 32 位带符号的数字,其范围从 $-2\,147\,483\,648 \sim +2\,147\,483\,647$ 。

2. 实型(REAL)

实型数是带小数点的数。在 -1 和 1 之间的小数,小数点之前的 0 不能省略,例如 $.5$ 或 $- .5$ 是错误的,应该写成 0.5 或 -0.5 。实数用双精度的浮点数表示,并且至少有 14 位有效位数的精度。注意,Visual LISP 不显示所有的有效位。

实型数也可以用科学计数法表示,即数字后可有一个 e 或 E ,其后是数的指数。例如: 0.12×10^9 表示为 $0.12E9$ 。

3. 字符串(STR)

字符串又被称为字符常数,它是由双引号括起来的字符序列。字符串中字母的大、小写和空格符都是有意义的。

字符串中字符的个数(不包括双引号)称为字符串的长度。字符串可以是空的,即`""`,称为空串,其长度为 0 。

任何字符都可以用 `\nnn` 的格式表示,其中反斜杠 `\` 是 ASCII 码的前导标识字符, `nnn` 是该字符八进制的 ASCII 码。例如,字符串 `ABCD` 也可表示为 `\101\102\123\104`。一些常用的控制字符,像反斜杠、双引号,除了可以用 `\nnn` 的格式表示之外,还可以用 `\转义字符` 的格式表示为 `\\`、`\"` 等特殊字符,见表 2-1。

4. 表(LIST)

表以左括号(开始,以配对的右括号)结束,表可以空,也可以有若干元素;元素可以是

表 2-1 常用控制字符的表示方法

控制字符	用\为前导的转义字符表示	用\为前导的 ASCII 码数值表示
反斜杠\	\\	\134
双引号""	\"	\042
Esc 键	\e	\033
换行	\n	\012
回车键	\r	\015
Tab 键	\t	\011

注意,其中的字符 e、n、r、t 必须小写。

简单的,也可以是复杂的,还可以是其他表。例如 $(+ 1 2 3)$ 、 $(\sin (* 0.5 \pi))$ 、 $(A B)$ 、 $((A B)C(C D))$ 、 (0 "LINE") 、 $(10 1.5 2.6 0.0)$ 、 $()$ 都是合法的表。

表中元素的个数称为表的长度。例如,表 $(+ 1 2 3)$ 的长度为 4,表 $(\sin (* 0.5 \pi))$ 的长度为 2,表 $((A B) C (C D))$ 的长度为 3,表 $()$ 的长度为 0。

用表可以很方便地构造出复杂的数据结构,例如, $(1.5 3.6)$ 可以表示为一个 X 等于 1.5、Y 等于 3.6 的二维点, $(2.5 2.0 1.0)$ 可以表示为一个 X 等于 2.5、Y 等于 2.0、Z 等于 1.0 的三维点。

5. 函数(SUBS)

函数相当于子程序或过程。函数分为内部函数和外部函数。AutoLISP 提供的或用 AutoLISP 定义的函数为内部函数。用 ADS、ADSRX 或 ARX 定义的函数为外部函数。例如 $\sin$ 、 $\cos$ 、 $\sqrt{\quad}$ 为内部函数。

运算符在 AutoLISP 里属于函数,例如 $+$ 、 $-$ 、 $*$ 、 $/$ 分别称之为加、减、乘、除函数, $<$ 、 $<=$ 、 $>$ 分别称之为小于、小于等于和大于函数。

其他计算机语言里的子程序、过程、程序流程控制的关键字,在 AutoLISP 里也属于函数,例如 if 、 while 分别称为条件和条件循环函数。

6. 文件描述符(FILE)

文件描述符是 AutoLISP 赋予被打开文件的标识号,它类似于文件指针。下面的例子是以“读”的方式打开文件 myfile.dat ,并将该文件的描述符赋予符号 f1 。

```
(setq f1 (open "myfile.dat" "r")) 返回<File:# 34614>
```

7. 图元名(ENAME)

图元名是 AutoCAD 为图形对象指定的十六进制的数字标识。AutoLISP 通过该标识,找到该图形对象在图形数据库中的位置,以便对其进行访问或编辑。

8. 选择集(PICKSET)

选择集是一个或多个图形对象命名的集合。可以通过 AutoLISP 程序建立选择集向

指定的选择集添加或移去图形对象,通过选择集可以对其内指定的成员进行访问或编辑。

9. VLA 对象

VLA 对象是 ActiveX 的基础。在 ActiveX 环境下,不仅直线、圆弧等称为 VLA 对象,图层、块、视图、模型空间、图纸空间、线型、字样和尺寸标注样式等也被称为 VLA 对象,甚至连 AutoCAD 应用程序本身也被认为是 VLA 对象。

2.2 变 量

2.2.1 符号

符号(SYMBOL)可以理解为标识,用来作为变量、函数的名字。它的命名规则是不能只含数字,可以由下列字符以外的任何可打印的字符所组成:

(、)、·、'、"、;

例如,a1、b2、c_3 是合法的符号,(a、)b、.c、'4、"5 是非法的符号。注意:按照这个规则,[1]、{2}、~3、!4、/5、1a、3c、-d、+e、b-2、c*3 都是合法的符号,这是与多数计算机语言的不同之处。

在 AutoLISP 中,符号的大小写等价,符号的长度没有限制,且所有的字符都是有意义的。

如果一个符号的长度不超过 6,就用结点本身来存储;如果符号的长度超过 6,就不能用结点来存储,而是在结点中存放一个指向实际存储符号名的指针。这将会多占用存储空间,且减慢了执行速度。因此,符号的长度最好不要超过 6。

2.2.2 变量的数据类型

数据类型是变量的重要特征,因为它关系到存放变量值的存储空间的大小。多数计算机语言都在为变量赋值前对变量进行显式或隐式的说明。AutoLISP 无须对变量进行事先的类型说明,变量被赋予值的类型即为变量的类型。

用 setq 函数对变量赋值,例如,(setq a 5)赋值之后的结果是:变量 a 是整型的,因为 5 是整型的;同样的原因,(setq b 2.5)赋值之后,b 是实型的;(setq c "ABC")赋值之后,c 就是字符串类型的。

在程序运行过程中,同一变量在不同的时刻可以被赋予不同类型的值,因此在程序运行过程中,变量的数据类型是可以被改变的。

例如,在一个程序里有以下连续两行:

(setq a 5)执行完该表达式之后,变量 a 是整型的。

(setq a "ABC")执行完该表达式之后,变量 a 是字符串类型的。

用 type 函数了解变量的类型,例如,(type a)返回 INT,说明变量 a 是整型的;(type b)