计算机工程设计与应用开发丛书

AutoCAD 2000i 应用开发与实例

清源计算机工作室 编著

机械工业出版社
China Machine Press

计算机工程设计与应用开发丛书

AutoCAD 2000i 应用开发与实例

清源计算机工作室　编著

机 械 工 业 出 版 社

本书主要讲解如何使用 AutoLISP 和 Visual LISP 工具进行应用程序开发的方法，全书共分 15 章。第 1 章至第 6 章主要讲述基本的 AutoLISP 开发知识和对话框设计与管理知识；第 7 章至第 15 章，讲述如何使用 Visual LISP 集成开发环境进行 LISP 应用程序开发，其中包括 Visual LISP 开发界面的使用、Visual LISP 编写代码的使用，以及应用程序的调试，另外还详细讲述了 ActiveX 的使用方法以及反应器特征，并结合实例讲述如何开发应用程序。本书以实例的方式，由浅入深，系统而全面地讲述了使用 AutoLISP 和 Visual LISP 进行二次开发的基本知识，用户可以从中学习到许多 AutoCAD 2000 为二次开发用户在 Visual LISP 中增加的新的、强大的功能。

本书适用于使用 AutoCAD 进行产品设计，并使用 AutoLISP 和 Visual LISP 进行应用程序二次开发的工程技术人员和软件开发人员。本书还可供高等院校和培训学校相应专业的师生参考。

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
责任编辑：边　萌　　封面设计：康　健
责任印制：路　琳
中国建筑工业出版社密云印刷厂印刷・新华书店北京发行所发行
2001 年 5 月第 1 版第 1 次印刷
787mm×1092mm 1/16 ・21.25 印张・523 千字
0001—5000 册
定价：37.00 元（1CD，含配套书）
ISBN7－900066－32－2/TP ・30

前　言

随着计算机技术的发展，计算机软件在工程设计领域的应用越来越广。在机械、电子、建筑等行业，应用计算机软件进行产品设计的 CAD 软件也非常丰富，使产品设计人员能够高效率地进行各自领域的产品分析、设计等工作。这些应用于工程设计领域的 CAD 软件有 AutoCAD、Protel、MATLAB 等。这些软件极大地提高了机械、电子等行业的产品设计的质量与效率，是目前 CAD 领域应用最为广泛的软件，是工程设计领域中最有用的辅助设计软件。为了帮助工程设计人员学习这些软件，快速掌握这些软件的使用与开发技术，我们特编写这套“计算机工程设计与应用开发丛书”。AutoCAD 2000i 主要应用于机械产品的设计和开发，以及 AutoCAD 2000i 的二次开发；Protel 99 SE 主要应用于电子原理图的设计、电路板的设计和绘制，以及电子逻辑分析和仿真等；MATLAB 6.0 主要应用于工程方面的数学计算、自动控制系统的分析，以及图形与图像处理等。

本套书主要面向工程设计人员，涉及的知识不但包括软件应用知识，还包括专业基础知识，以及软件在相关领域的二次开发技术知识，是机械、电子领域技术人员的最佳参考书。本套书在介绍软件的使用过程中，结合丰富的实例进行讲解，使读者能快速掌握相关的知识。本套书不但讲述软件的基础应用知识，还讲述了软件的中、高级应用知识，是一套面向中、高级读者的全面而系统的参考书。本套书包括《AutoCAD 2000i 图形绘制与应用》和《AutoCAD 2000i 应用开发与实例》；《Protel 99 SE 原理图和 PCB 设计》和《Protel 99 SE 电路设计与仿真》；《MATLAB 6.0 基础及应用》和《MATLAB 6.0 高级应用——图形图像处理》，基本覆盖了三种软件在机械领域和电子领域的应用。全套书均附有光盘，以实例为主，将软件的实际应用很生动地展现在读者面前。

本套书由清华大学从事该领域工作多年的博士生和硕士生编写，具有贴近读者的特点。书中列举了大量的实例，是使用这三种软件的工程设计人员不可多得的参考书。

清源计算机工作室

编 者 的 话

AutoCAD 2000i 是 Autodesk 于 2000 年推出的最新版本软件，AutoCAD 软件集图形处理、产品设计、图形数据管理以及网络技术于一体，成了连接世界的产品设计平台，在 CAD 发展领域具有深远的意义。AutoCAD 2000i 不但可以用来进行产品设计绘图，而且还提供了二次开发工具，可实现产品参数化设计等二次开发，并可以针对用户的需求实现产品的二次软件开发，大大提高了产品设计的效率。AutoCAD 2000i 提供了多种开发工具，如 AutoLISP 及 Visual LISP、ObjectARX 2000 和 VBA 等，这些都是基于 AutoCAD 2000i 平台的二次开发软件。

本书主要讲述如何使用 AutoCAD 2000i 优秀的二次开发工具 AutoLISP 及 Visual LISP 进行程序设计，以及开发基于 AutoCAD 平台的 CAD 软件。AutoLISP 作为一种 LISP 编程语言，是 AutoCAD 2000i 内嵌的编程工具，是 AutoCAD 2000i 整体的一部分。AutoLISP 易于使用并很灵活。使用 AutoLISP，你可以编写适合于图形处理的很有效的宏程序及函数。另外，AutoCAD 2000i 为使用 AutoLISP 语言提供了一个非常有用的可视化开发工具——Visual LISP 2000。使用该开发工具可以更高效地开发 LISP 应用程序。Visual LISP 应用现代化的 LISP 引擎以支持多文档、对象和事件，并且提供强大的集成开发环境（IDE）。利用 Visual LISP 支持 ActiveX 的优点，可以开发运行效率更高的应用程序，而且可以生成更加安全的可执行程序（VLX 和 FAS）。

本书围绕如何使用 AutoLISP 和 Visual LISP 工具开发应用程序进行讲解。全书共分 15 章。第 1 章至第 6 章主要讲述基本的 AutoLISP 开发知识和对话框设计与管理知识；从第 7 章至第 14 章主要讲述如何使用 Visual LISP 集成开发环境进行 LISP 应用程序开发，其中包括 Visual LISP 开发界面的讲解，Visual LISP 代码的编写，以及应用程序的调试，另外还详细讲述了 ActiveX 的使用方法以及反应器特征；第 15 章为使用 AutoLISP 和 Visual LISP 开发应用程序实例，读者从中可以领略到如何使用 AutoLISP 和 Visual LISP 进行二次开发的真谛。本书系统而全面地讲述了使用 AutoLISP 和 Visual LISP 进行二次开发的基本知识，用户可以从中学习到 AutoCAD 2000i 为二次开发用户在 Visual LISP 中增加的许多新的、强大的功能。

在阅读本书时，读者可以先学习前面的 AutoLISP 基础和对话框设计部分，也可以先学习第 7 章至第 11 章的 Visual LISP 集成开发环境的基本部分，然后再从前到后顺序学习，对于某些 Visual LISP 调试和窗口选项的设置，可以直接学习第 14 章。

本书适用于使用 AutoCAD 进行产品设计，并使用 AutoLISP 和 Visual LISP 进行应用程序二次开发的工程技术人员和软件开发人员。本书还可供高等院校和培训学校的相关专业师生参考。

本书由江思敏主编，郑巍、胡荣等同志参与编写了部分章节。由于水平有限，时间仓促，书中缺点和不足在所难免，敬请广大读者批评指正。

编 者

目　录

第 1 章　AutoLISP 基础

AutoLISP 作为一种 LISP 编程语言，是 AutoCAD 内嵌的编程工具，是 AutoCAD 整体的一部分。它易于使用并很灵活，使用 AutoLISP 可以编写适合于图形处理的很有效的宏程序及函数。本章主要介绍 AutoLISP 编程语言的基本概念。因为 AutoLISP 代码无需编译，所以用户可以直接在命令行中输入代码并立即看到结果。通过练习本章的实例，用户可以快速地学习并掌握 AutoLISP 的基本知识。

1.1　概述

AutoLISP 是基于 Common LISP 发展而来的，它不但具有与 Common LISP 相一致的句法和语法规则，而且具有许多适合于 AutoCAD 的专用函数，可以实现 AutoLISP 程序与 AutoCAD 绘图命令的结合，使设计与绘图成为一体；还可以通过 AutoLISP 程序实现对 AutoCAD 图形数据的直接访问、修改等处理，以便对图形进行绘制、编辑等实时处理，实现图形的交互设计。使用 AutoLISP 进行 AutoCAD 二次开发，重要的是先掌握 AutoLISP 的语法规则及如何使用 AutoLISP 函数进行程序设计，然后再结合 AutoCAD 的绘图命令就可以实现图形的参数化绘制等开发目的。

AutoLISP 是一种人工智能语言，它是一种基于符号处理的、面向对象的语言，而不是过程性的语言，在处理语句的过程中是通过对对象进行求值实现的。在 AutoCAD 的开发工具中，AutoLISP 是唯一的一种非编译的、解释性的语言，所以在运行 AutoLISP 应用程序时，只需直接调用其编写的函数，就可以实现目标操作，不需要一个专门的编译工具。如果再结合 AutoCAD 的绘图命令及相关的内部函数，不仅可以用来开发设计、计算、绘图、DCL 对话框等应用程序，而且能生成 Windows 类对话框。

利用 AutoLISP 进行 AutoCAD 的二次开发，可以帮助用户充分利用 AutoCAD，从而大大节省绘图时间并提高工作效率。AutoLISP 的一个典型应用是实现图形参数化设计。在工程设计中，图形的绘制常存在大量的重复劳动，在设计进程中，还存在着大量的修改设计，而通过使用参数化设计，则可以使工程设计人员减少不必要的重复劳动，提高设计效率。

本章重点介绍 AutoLISP 函数的使用和 AutoLISP 应用程序的编写方法，为学习后面的程序设计开发打好基础。

1.2　AutoLISP 表达式

在命令行中输入文本时，AutoCAD 将该文本与内部的可用命令名列表作比较，以解释该文本。如果输入的文本与列表中的某项相匹配，则 AutoCAD 执行该命令。当 AutoCAD 接收到 AutoLISP 代码时，它将该代码传递给 AutoLISP 解释器。

AutoLISP 解释器的核心是计算器。该计算器先读取一行代码，对它求值，然后返回一个结果。该代码必须符合 AutoLISP 表达式的格式要求，它可以从文件中读取，也可以由用户从 AutoCAD 命令行中输入。

所有 AutoLISP 表达式的格式都如下所示：

(function arguments)

每个表达式都以一个左括号开始，由一个函数名和一个该函数的可选参数组成。并且每个参数都可以是一个表达式。表达式以右括号结束。每个表达式都返回一个可由外层表达式使用的值。最后解释的表达式的值返回给调用表达式。

(fun1 (fun2 arguments)(fun3 arguments))

如果在 Visual LISP 控制台提示或 AutoCAD 命令提示下输入此代码，则由 AutoCAD AutoLISP 解释程序处理该代码。第一个函数 fun1 有两个参数，另两个函数 fun2 和 fun3 各有一个参数。函数 fun2 和 fun3 被函数 fun1 所包含，因此它们的返回值作为参数传递给 fun1。函数 fun1 由这两个参数计算函数值，并将该值返回给输入代码的窗口。

如果在 AutoCAD 命令提示中输入 AutoLISP 表达式，则 AutoLISP 将计算该表达式并显示结果，然后重新显示命令提示。下例展示了 *（乘）函数的用法，该函数接受一个或多个实数作参数。

_$ (* 2 27)

54

因为此函数没有外层表达式，所以它将结果返回给命令行。嵌套在其他表达式中的表达式将它们的结果返回给外层表达式。下例用 +（加）函数的结果作 *（乘）函数的一个参数。

_$ (* 2 (+ 5 10))

30

如果输入的闭（右）括号数量不对，AutoLISP 将显示如下提示：

(_>

其中，n 是一个整数，表明还有几层左括号尚未匹配。如果出现此提示，用户必须输入 n 个右括号后才能对表达式求值。如

_$ (* 2 (+ 5 10

((_>))

30

常见错误是忽略了文本字符串中的引号 (")。在这种情况下，右括号被解释为字符串的一部分，而不会被用来和开括号配对。要改正此错误，请按【Shift+Esc】 键取消该函数，然后重新输入正确的表达式。

1.3 AutoLISP 数据类型

AutoLISP 计算器按括号中代码的次序和数据类型来处理表达式。在用户充分利用 AutoLISP 之前，必须了解数据类型之间的区别以及如何使用它们。下面详细讲述 AutoLISP 的数据类型。

1.3.1 整数

AutoLISP 的整数由 0、1、2、3、…、9、+、-等字符组成，不包含小数点，不允许

出现其他类型的字符，对于正整数，＋可以省略。

AutoLISP 的整数是 32 位带符号的数，其取值范围为 ＋2147483648 到－2147483647 虽然 AutoLISP 在其内部使用 32 位整数，但在 AutoLISP 和 AutoCAD 之间传递的整数却被限制为 16 位，即用户不能把大于＋32767 或小于－32768 的整数值传递给 AutoCAD 当用户在 AutoLISP 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 AutoLISP 整数。

1.3.2　实数

实数是带有小数点的数。在－1 和 1 之间的实数必须以零开始。实数是以双精度浮点格式保存的，可以提供至少 14 位有效位数的精度，尽管在 AutoCAD 命令行中只能显示 8 位有效数字。实数可以用科学计数法表示，科学计数法的格式中可包括 e 或 E 及指数（例如，0.0000055 与 5.5e－6 相同）。当用户在 AutoLISP 表达式中直接使用实数时，该值被称为常量。数字 3.5、0.25、－58.456 和 35000000.086 都是有效的 AutoLISP 实数。

1.3.3　字符串

字符串是在双引号中的一组字符。在引号包括的字符串中，用反斜杠 (\) 字符可以添加控制字符（或换码代码）。当用户在 AutoLISP 表达式中直接使用引号包括的字符串时，该值被称为文字字符串或字符串常量。

字符串中字符的个数（不包括分隔符）称为字符串长度，字符串可以为任意长度，它们的存储空间是动态分配的。字符串常量的最大长度为 132 个字符，如果超过 132 个字符，则后面的字符无效。

字符串中可以包括 ASCII 中的任何字符，通常格式为“\nnn”，其中 nnn 是其字符的八进制 ASCII 码。如字符串“5AB7”也可表示为“\065\101\102\067”。因为“\”已经作为标志字符，在字符串中有特殊作用，所以如字符串需要包含它，则必须使用“\\”或“\134”来代替“\”，对于双引号，因为它也用于字符串定界，所以在字符串需要包含它时，可以使用“\042”来表示。另外 AutoLISP 还提供了特定的控制字符的简单表示形式：

\e　　表示 Esc（等价于\033）
\n　　表示换行 LF（等价于\012）
\r　　表示回车 CR（等价于\015）
\t　　表示制表 HT（等价于\011）

注意：这 4 个特定控制符中“\”后的字符均应该小写。

一般地，字符串在 AutoLISP 程序中常用于文件名、标志符及控件名（与 DCL 相关）。

1.3.4　表

表是 AutoLISP 存储和处理数据最有效的方式，因为 AutoCAD 的数据是以链表的方式进行存储的，所以 AutoLISP 使用表这一数据类型，对开发 AutoCAD 非常有效。因为图形绘制涉及的图形主要由点、线、面组成，而点是最基本的元素。可以使用 AutoLISP 的表

进行点坐标处理。

AutoLISP 的表是包括在括号中的以空格分隔的一组相关值。表提供了一种保存大量相关值的有效方法。AutoCAD 将三维点表示为三个实数组成的表。

(1.0 1.0 0.0)、("this" "that" "the other") 和 (1 "ONE") 都是有效的表。

在处理图形坐标时，AutoLISP 有以下约定：

- 二维点坐标 AutoLISP 用（X Y）表示。
- 三维点坐标 AutoLISP 用（X Y Z）表示。
- 表可以由 AutoLISP 函数 List 等生成及处理。

1.3.5 选择集

选择集是包含一个或多个对象（实体）的组。用户可以通过 AutoLISP 例程交互地将对象添加到选择集中，或从选择集中删除对象。

下面的例子用 ssget 函数返回包含图形中所有对象的选择集。

```
_$ (ssget "X")
< Selection set: 1>          ；（返回值）
```

下面例子为将最近获得的图形实体对象选择集赋给符号“Lent”：

```
(setq Lent (ssadd (entlast)))
<Selection set: 1>          ；（返回值）
```

1.3.6 实体名

实体名是分配给图形中对象的标签。它实际上是一个指向 AutoCAD 所维护的文件中的指针。通过该指针，AutoLISP 可从图形数据库中找到该对象的数据纪录和矢量（如果实体在屏幕上）。此标签可以由 AutoLISP 函数调用，使得 AutoLISP 函数可以用不同的方式选择要处理的对象。AutoCAD 内部将对象当作实体。

下例用 entlast 函数来获取最新输入图形对象的实体名。

```
_$ (entlast)
<Entity name: 60000016>          ；（返回值）
```

分配给图形中对象的实体名仅仅在当前编辑阶段有效，下一次打开图形时，AutoCAD 分配给对象新的实体名。用户可以使用指向对象的句柄来处理对象，句柄是随图形保存的。

1.3.7 VLA 对象

图形中的对象可以表示为 Visual LISP ActiveX（VLA）对象，该对象是 Visual LISP 的一种数据类型。当用户使用 ActiveX 函数进行工作时，用户必须参考 VLA 对象，而不是例如 entlast 等函数返回的 ename（实体名）指针。

关于如何使用 ActiveX 对象进行操作，可以参考第 12 章。

1.3.8 文件描述符

文件描述符是分配给 AutoLISP 所打开文件的标签。当 AutoLISP 函数需要读写文件时，必须引用该文件的标签。下例以只读方式打开文件 myinfo.dat：

```
_$ (setq file1 (open "c:\\myinfo.dat" "r") )
#<file "c:\\myinfo.dat">          ；（返回值）
```

在该实例中，文件描述符保存在 file1 变量中。

直到用户在 AutoLISP 程序中使用 Close 函数关闭文件，文件一直是打开的，Close 函数用来关闭一个文件。下面的代码为关闭文件描述符保存 file1 变量中的文件，返回值为 Nil。

```
_$ (close file1)
Nil          ；（返回值）
```

1.3.9　符号和变量

AutoLISP 通过符号来引用数据。符号名不区分大小写，可以由字母、数字和标注符号的任何序列组成（除以下字符以外的任何标准字符）。

“(” 及 “)”	开括号和闭括号在 AutoLISP 中用于定义表
“ . ”	句号用做点对
“ ’ ”	单引号用于求值，是 Quote 的简写
“ ; ”	分号用做程序的注释标志

如：B56、!C68、8bh8 等均为合法的符号原子，B·C、（ABC）、C;67 则为非法字符。符号名不能仅由数字组成。

从技术上来说，AutoLISP 应用程序是由函数、符号或常量值（例如字符串、实数和整数）组成的。为了表述得更加清楚，本文用术语符号来表示保存静态数据的符号名，例如内置和用户定义的函数。术语变量用于表示保存程序数据的符号名。下例用 setq 函数将字符串值 "this is a string" 分配给变量 str1：

```
_$   (setq str1 "this is a string")
"this is a string"          ；（返回值）
```

AutoLISP 提供了一些字符来终止一个原子或分隔多个原子，如：“（”、“）”、“ ’ ”、“ ″ ”、“ ; ”和空格。

注意：多个空格与一个空格分隔原子的作用是相同的。

AutoLISP 程序中不区别符号的大小写。如 ABCD、AbCD、abcd 三者是等价原子。

符号原子的长度在 AutoLISP 中没有限制，所有的字符均有意义，这一点与其他语言不同。如 DEVELOPMENT 与 DEVELOPMENTTOOL 是两个不同的原子。

AutoLISP 提供了两个特殊的字符，即 T 及 Nil，它们的值预先由系统设定为 T 和 Nil。

在执行程序过程中，用户常需要观察或获取某个符号的值，AutoLISP 允许使用 “!” 加符号原子进行查看。如：

```
Command: !aaa
″50″          ；（返回值）
```

则″50″为 aaa 的当前值。

对于 AutoLISP 符号原子的存储，系统一般采用节点方式进行存储，通常一个节点只能保存不超过 6 位的符号原子，如果符号原子超过了 6 位，则系统就另外开辟一个存储区

来保存符号原子，该节点则包含有指向该存储区的指针。所以，一般定义符号最好不超过6位，以避免占用过多的存储空间。

注意：用户在程序中定义的符号不能与系统定义的函数或符号名相同，否则后面定义的符号将取代系统的定义而引起变量混乱。为了方便阅读程序代码，用户在程序中应该选择有意义的符号和变量。

● **保护符号** 如果用户试图改变一些被 AutoLISP 语言使用的符号值，则系统会提出警告。这些符号为系统保护符号，包括数学操作符（例如＋、－等）和 T、Nil 值。用户可以使用 Visual LISP 符号的服务特征来识别一个符号是否是保护符号。

当用户第一次启动 AutoCAD 时，保护符号接收不到特别的保护。如果用户在 AutoCAD 命令行设定保护符号，用户也接收不到符号具有特定状态的任何提示。但是，一旦用户启动了 Visual LISP 系统，这些就发生了变化。从启动 Visual LISP 直到退出 AutoCAD 操作，AutoLISP 会阻止任何修改保护符号的企图。保护符号的处理依赖于 Visual LISP 环境选项的设置状态，用户可以指定如下的选项：

（1）透明 保护符号像其他符号一样进行处理。

（2）打印信息 AutoLISP 当用户修改一个保护符号时，将给用户一个警告信息，但执行修改，例如，下面的语句为在这种状态下修改符号 T：

```
_$ (setq t "look up")
; 用户警告：给保护符号赋值：t <- "look up"
"look up"
```

（3）提示进入中断循环 该项为缺省项，当用户试图修改一个保护符号时，系统将显示如图 1-1 所示的对话框。

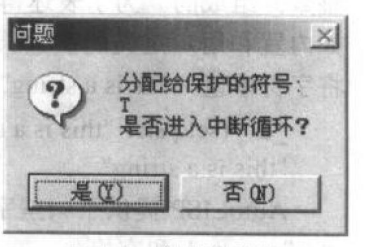

图 1-1 问题提示框

如果选择“否”，则符号被修改，处理过程继续。如果选择“是”，处理过程被终止，并且进入一个 Visual LISP 中断循环。如果为了设定符号并继续处理，则可以选 Visual LISP 工具栏上的“继续”按钮。如果想中断修改，则可以选“重置”按钮。

（4）错误 该选项禁止保护符号的修改，任何修改一个保护符号的企图均会导致一个错误信息。

关于 Visual LISP 的有关设置和操作可以参考后面有关章节，用户也可以将有关 Visual LISP 的章节提前学习。

1.4 AutoLISP 程序文件

虽然可以在 Visual LISP 控制台窗口中或在 AutoCAD 命令提示下输入 AutoLISP 代码，但如果将 AutoLISP 代码存储在文件中，在测试和调试一系列指令时，从文件中加载 AutoLISP 代码要比每次修改后都重新输入要简单得多。AutoLISP 代码通常保存在扩展名为“.lsp”的 ASCII 文本文件中。但可以从任何 ASCII 文本文件中加载 AutoLISP 代码。

1.4.1　格式化 AutoLISP 代码

AutoLISP 代码中括号的广泛使用使程序阅读很困难。处理该问题的传统技术是缩排，这样一行代码嵌套越深，则定位行位置越靠右。

如果用户使用 Visual LISP 文本编辑器输入代码，则 Visual LISP 会自动格式化用户输入的代码。Visual LISP 也可以格式化一段所选择的代码或整个程序文件。这样就改善了用户程序代码的外观，使它更便于阅读。

关于使用 Visual LISP 格式化代码的操作，用户可以参考 8.5 节的有关内容。

另外为了减少阅读程序的困难，通常采用插入空格的方法。空格允许程序代码成为一定的格式，以便使嵌套的程序行代码容易阅读。在 AutoLISP 程序中，在变量名、常量和函数名之间的多个空格与一个空格等效，行尾也被当作空格。

下面两个表达式的效果相同：

```
(setq test1 123 test2 456)
(setq
    test1 123
    test2 456
)
```

注意：尽管 Tab 键在 AutoLISP 中被解释为空格，但建议不要使用它，因为有些平台对它的解释不同，所以经常导致不可预料的结果。

1.4.2　注释

AutoLISP 程序文件中通常都包括注释。对于程序员和需要修改程序的用户，程序中的注释非常有用。注释可用于：

- 给出标题、作者和创建日期。
- 提供例程的使用信息。
- 对程序主体进行说明性注释。
- 添加调试信息。
- 输入各种字符来美化程序文件。

注释是以一个分号“;”开始的，并持续到行尾。如：

```
; This entire line is a comment      （这一行就是注释语句）
(setq area (* pi r r))   ; Compute area of circle   （；后面是注释）
```

任何在“;| ... |;”之间的文本均可忽略，因此这种注释可以包括在一行中或扩展到多行，这种注释在一行中注释很有用。如：

```
(setq tmode ;|一些注释|; (getvar "tilemode"))
```

下面的实例说明一个连续多行的注释：

```
(setvar "orthomode" 1) ;|注释从这里开始，
注释在本行持续，
此处终止注释|; (princ "\nORTHOMODE set On.")
```