Auto LISP
实例教程

李子铮 李超 张跃 编著

```
(if (= (cos #jd)
(setq #tt "JGTB (5T1
(setq #tt "JGTB (5T2
)
(start_image "image
(slide_image    0
(dimx_tile "imagea")
(end_image)
(set_tile "aa01" #
(set_tile "aa02" #
```

机械工业出版社
CHINA MACHINE PRESS

AutoLISP 实例教程

李子铮　李　超　张　跃　编著

机　械　工　业　出　版　社

前　　言

本书作者是普通的土建专业设计人员，亲身经历了从图板制图到AutoCAD计算机辅助绘图，再到二次开发的一系列过程。

最早的图板制图，最为辛苦。

到后来，通过AutoCAD计算机辅助绘图的应用，大大提高了制图质量和复制图纸的速度。AutoCAD是一个非常好用、非常流行的计算机辅助绘图软件。但该软件的强大功能并没有在广大设计人员手中得到发挥。作者从AutoCAD 10.0开始接触该软件，目前的版本是2006，不论它的功能如何强大，对于各专业普通的设计人员来说，只是从一块“图板”换到另外一块更为华丽的“图板”，并没有质的飞跃，可代价就是速度的下降。

AutoCAD绘图软件，在原始安装的状态下，是一款通用的、适合各种专业的、各种复杂情况的绘图软件，一款可以取代图板等手工绘图工具的绘图界面，一款“一药治百病”的软件。而我们真正需要的，也恰恰缺少的是“专病专药”的软件。

设计过程的专业性、地域差别、行业习惯等的不同，无法得到AutoDesk公司的支持，AutoDesk公司只是不断加强通用功能方面的开发，使得“专病专药”的实现变得没了着落，但AutoCAD留下的二次开发接口，让我们看到了希望。

一个非常现实的问题摆在面前：

一方面，是成百上千的软件开发高手。越是高手，目光越是着眼于高级开发语言、通用性软件、利润丰厚的领域，就算有某位编程高手肯于屈就，想去研究点此方面的东西，面对AutoCAD的软件接口语言，面对空白的功能规格书，也只能望而却步，因为他不知道如何施展他的各种技巧，更不肯放下“高级语言”来学习“低级语言”。

另一方面，是成千上万的设计人才、制图专家，眼巴巴地在重复着各种体力劳动，却不知如何从中解脱出来，就算有几个人懂一些软件开发，又有几人能成气候。各种不同领域的AutoCAD二次开发功能需求太多。在中国目前软件不值钱的情况下，使用软件的收获高于编制软件的付出之时，你又能指望谁?

综上所述，解决问题的唯一出路是把自己变成编程高手。值得庆幸的是，AutoCAD二次开发语言——AutoLISP是一种非常简单的、一看就懂、一学就会的“低级语言”。AutoCAD二次开发语言接口也支持VB、VC、C+、C++等编程语言，但作为普通的专业设计人员，如果目前不是VB、VC、C+、C++等语言的高手，那么还是研究AutoLISP语言更现实一些。用VB、VC、C+、C++等语言进行AutoCAD二次开发，对于不同的版本程序修改的工作量都很大，而且程序调试复杂；相对而言AutoLISP程序几乎不用进行任何修改，就可以在不同版本中进行移植，而且程序调试简单、直观。

市面上介绍 AutoCAD 的书籍不少，但都不针对某个具体专业。作者在此方面摸索了多年，认为针对具体专业的二次开发最重要的是，最大深度挖掘专业中的隐含信息，将其参数化，由原本需要的一笔一笔绘制，变成输入参数，由程序自动计算、运行，从而实现几笔乃至于上百笔、上千笔的自动输出。只有如此，才能最大程度提高制图效率。而对于这方面的参数化软件编制，不需要过多的技巧，完全没有必要先去系统学习编程；恰恰这一点，是我们各专业设计人员的强项。开发程序，不应为把自己打造成编程高手，去啃那填鸭式的、厚厚的编程手册，往往翻不上五六页，就已经兴趣索然，而编程的最大动力是兴趣。作为专业制图人员，编程只是自我提高工作效率的手段，而绝对不是目的，因为自己编出的软件很可能在市场上一分不值，而由于提高效率，较以前有更多的产出，却会给人们带来利益。所以，软件编制只要能够实现计算、批量输出绘图的目的，不必过于追求编程的技巧或代码的精练。

本书所介绍的内容主要是采用 AutoLISP 语言编写，与版本关系不大。本书的顺序也不是按照命令或是函数的姓氏笔画顺序介绍，而是按照本套绘图软件的功能顺序。

开始学习本书之前的说明和建议：本书只探讨了 AutoCAD 二次开发编程的部分内容及技巧，不求全面介绍。研习本书最好看两遍，第一遍的目标，囫囵吞枣、依葫芦画瓢，树立成就感，掌握基本知识，能够参照程序说明及附录说明看懂现成的源程序；第二遍的目标，知其然亦知其所以然，成为编程高手。这时就可以借助于 AutoCAD 附带的“开发人员帮助”或其他书籍，来探求 AutoCAD 编程的其他内容和更多的技巧。

需要再次强调的是，作为制图人员，学习编程的目的始终是：将想法变成现实，而决不是成为面面俱到的编程专家。想要达到上述目标，就本书所介绍的内容，加以扩展已经足够了。本书作者也不是编程专家，而是普通的工程设计、制图者，但了解读者需要什么。因此，学习本书将会引导读者逐渐步入某个专业的编程高手行列。

AutoCAD 是开放式结构的通用绘图软件，它的功能可以自定义并加以扩展，因此用户可以根据需要扩展和调整 AutoCAD 软件的功能。

“AutoCAD 开发人员帮助”（acad_ dev. chm），提供了有关如何自定义和扩展 AutoCAD 的完整信息。是最具权威的随软件配套安装的指导性文件。本书主要讲述 AutoLISP/DCL 进行应用程序的开发，仅简单介绍“菜单文件”的自定义，“字型文件”的自定义，作为相关基本知识的准备，也为方便读者对照解读本书内容，对“AutoCAD 开发人员帮助”（acad_ dev. chm）包含的“菜单文件”，“AutoLISP Reference”两个部分，就本书涉及到的部分做了简单的介绍，附在书后附录中，以方便读者学习相关部分时查阅。

附录中的内容，并非本书主打方向，介绍较为简单，只是为方便读者对照查询而设置。附录中的部分内容参考了网上部分文章，在此一并谢过。如果你的英文水平还可以的话，最好针对自己使用的版本，查看英文原版的“AutoCAD 开发人员帮助”。

可以登录 http：//lizizheng. bloghome. cn/获得帮助，提出你在学习中遇到的问题，以及你的建议。

也可以通过 lizizheng@ tom. com 与作者进行联系。

编　者

目　　录

第1章 线 层

1.1 问题的提出

计算机辅助绘图，无论有多么的高明、迅速，无外乎是对手工绘图的模拟和扩展。绘图的三个基本的要素就是“线型”、“线宽”和“形状”。

（1）线型　最常用的有实线、虚线、点划线、双点划线、双线等。

（2）线宽　多数情况，细线、中粗线、粗线三种即可满足需要。

（3）形状　直线、圆、椭圆、弧等 AutoCAD 各种命令非常完备。

通过表 1-1 的比较可以略见一斑。

表 1-1　绘图基本要素的比较

	手工图板绘图	AutoCAD 计算机辅助绘图
线型	通过笔尖的放下、抬起表现线型的变化	采用不同的图层关联相应的“线型”定义
线宽	利用不同粗细的铅笔或针管笔来完成不同的线宽绘制	① 对线宽进行定义，绘制有宽线 ② 赋予图层特定的线宽，与输出设备关联（绘制时可采用 0 线宽）
形状	利用圆规、模板	不同的命令

1.2 功能规格书

本章介绍的各功能函数，一般都比较短小，而且各功能函数之间都有或多或少的联系，所以本章各函数的功能规格书统一给出，并统一介绍软件编制。

绘图时尽量采用宽度为 0 的线宽设置，而通过颜色识别线宽，图层颜色与输出设备关联，即可实现对线的有宽度输出。同时便于减小绘图文件大小，提高机器的响应速度，同时也降低了编程的难度，因为绘制有宽线需要更多的信息。可利用状态栏“线宽”按钮切换显示，来查看实际宽度效果。

在不同的图层，具有不同的颜色，代表不同的“线型”和“线宽”；另外，尺寸、文本、特定图块固定在特定的文本图层。

本章要介绍给读者的主要内容，如图 1-1 所示。同时可参见如下功能要求说明。

功能要求说明

（1）A～F 项（画线）　首先进入指定层（具有特定线型、线宽、颜色定义的图层），然后开始画线（即执行 LINE 命令）。

(2) G～L 项（变线） 将所选实体移入指定层，并赋予该层的线型、线宽、颜色定义，以达到修改线型、线宽、颜色等线特征的目的。

(3) M 项（切断线，在交点） 将一实体在与另外一实体的交点处一分为二。先选中该实体，然后选择该实体与其他实体的交点。

(4) N 项（切断线，在拾取位置） 将一实体在选择该实体的选择点上切断，一分为二。

(5) O 项（伸缩线端点） 将选中的线一端动态伸长或缩短。

(6) 0～6 项（置当前层） 将指定层设置为当前层，以便配合 AutoCAD 各种命令绘制不同形状实体，同时显示当前层标记“√”。

(7) P 项（颜色变白） 配合打印输出，各层颜色设置为白色，实体颜色随层。为了更清晰输出，避免浪费彩色墨盒，同时避免以“灰度”输出的模糊。

(8) Q 项（恢复颜色） 配合 P 项，恢复各层颜色的系统设置颜色，以便继续绘图。

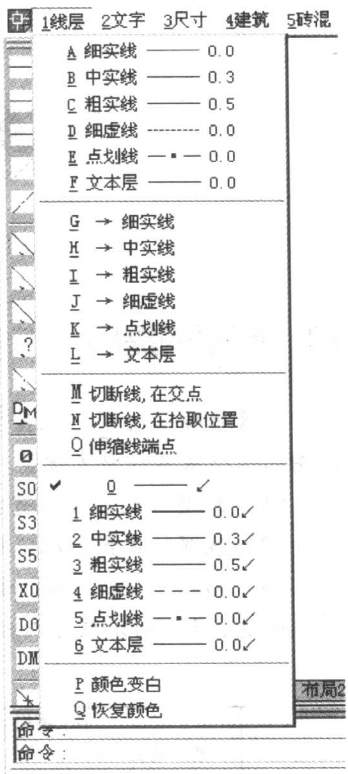


图 1-1

本章介绍的都是一些最基本的功能，虽然本系统软件可以通过其他章节介绍的功能达到批量输出的效果，但本章介绍的功能在整个体系中是不可或缺的。

当然，在应用各功能的时候，也要学会“组合拳”。例如，在工程实际绘图中，往往需要将一条线段在特定位置切断，变成两条线型不同的线段，执行命令的过程就是：切断线段→选取需要改变线型部分→由原定义层移至新层。也就是经 M、N 项处理线实体的切断，再利用 G～L 各项命令实现对某一实体的分离，并修改其中一部分的图层设置。当然，也可以将切断与变层配合使用，编制在同一个功能程序之中，这也正是留给读者依葫芦画瓢、举一反三的空间。

1.3 准备工作

建立一个空白的图形文件：JGTB. DWG（或 JGTB. DWT），利用已有的知识，一一定义如表 1-2 所示的图层，将此图定义为模板，省去了每次绘图的定义过程。

表 1-2 图层及特性设置

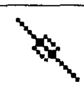
序 号	图层名称	颜 色	线型或特点	宽度/mm	备 注
1	JS0	白色 7	细实线	0.0	
2	JS3	品红 6	中实线	0.3	
3	JS5	蓝色 5	粗实线	0.5	
4	JX0	青色 4	细虚线	0.0	

（续）

序 号	图层名称	颜 色	线型或特点	宽度/mm	备 注
5	JD0	红色 1	点划线	0.0	
6	JDM	白色 7	文本、尺寸	0.0	块、文本、尺寸等
7	0	白色 7	细实线	0.0	

同时，按照表 1-3 依次准备 bmp 图形，用于工具菜单的定制。每个图块大小为 16×15（像素）。编辑方法，①采用 mspaint. exe 制作；②在 AutoCAD 绘图界面下，以鼠标右键单击工具条→以鼠标左键单击自定义→再次以鼠标右键单击工具条→编辑按钮图像→工作目录下保存名称。

表 1-3 工具菜单图形列表

图名	JGTB _ JS0. bmp	JGTB _ JS3. bmp	JGTB _ JS5. bmp	JGTB _ JX0. bmp	JGTB _ JD0. bmp
图形					
图名	JGTB _ GS0. bmp	JGTB _ GS3. bmp	JGTB _ GS5. bmp	JGTB _ GX0. bmp	JGTB _ GD0. bmp
图形					
图名	JGTB _ GDM. bmp	JGTB _ 0. bmp	JGTB _ S0. bmp	JGTB _ S3. bmp	JGTB _ S5. bmp
图形		0	S0	S3	S5
图名	JGTB _ X0. bmp	JGTB _ D0. bmp	JGTB _ DM. bmp	JGTB _ JD. bmp	JGTB _ DD. bmp
图形	X0	D0	DM		
图名	JGTB _ LS. bmp				
图形					

1.4 编程

1. 菜单宏的实现

test1. mun

行号	test1. mun 内容
1	***MENUGROUP=TEST
2	***POP1
3	[&A 线层]
4	[&1 细实线—— 0.0]^C^C-LAYER;S;JS0;;_ LINE
5	ID _ JGTBjs0 [&2 细实线 —— 0.0]^C^C^P(command “ -layer”” s”” js0”””” line”)

（续）

行号	test1. mun 内容
6	[&3→ 细实线]^C^C^Pselect \change p ;p LA JS0 c BYLAYER LT BYLAYER ;
7	[$ (if, $ (eq, $ (getvar, clayer), JS0), ! .) &4 细实线 —— 0.0] ' - LAYER; S; JS0; ;
8	[--]
9	[&5 切断线,在交点]^C^C - OSNAP INT _ break _ f \@
10	[&6 切断线,在拾取位置]^C^C $ S = X $ S = break0 _ break; \@ ;
11	[&7 伸缩线端点]^C^C _ lengthen; dy
12	*** TOOLBARS
13	** JGTB _ xc
14	[_ Toolbar("线层", _ Left, _ Show, 0, 0, 1)]
15	ID _ JGTBjs0 [_ Button("细实线", JGTB _ JS0. bmp, JGTB _ JS0. bmp)]^C^C^P - LAYER; S; JS0; ; _ LINE
16	*** HELPSTRINGS
17	ID _ JGTBjs0 [在 JX0 层绘制细实线 —— 0.0]

对照上述菜单文件和图 1-2，要先了解一下菜单文件的结构。在原始安装情况下，通常使用的菜单是“acad. mnu”、“acad. mns”或“acad. mnc”。也可以打开“acad. mnu”文本文件，对照着学习。“acad. mns”或“acad. mnc”是 AutoCAD 加载“acad. mnu”后编译生成的菜单文件，可以先不必理会它们。

菜单文件包括若干部分。第一部分始终是 Menugroup 部分，它为菜单文件指定唯一的菜单组名。菜单组名是一个最多可包含 32 个字母数字字符的字符串，不能包含空格和标点符号。

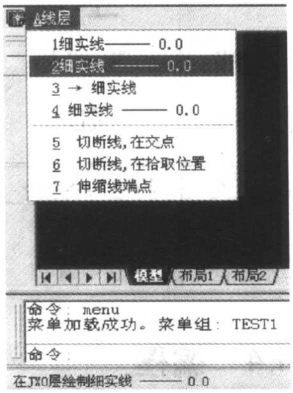


图 1-2

后续部分定义了 AutoCAD 界面的特定区域，并包含通常由名称标记、标签和菜单宏组成的菜单项，详见表 1-4。

菜单文件的各部分由使用格式 *** section _ name 的部分标签进行标识。多个按钮部分、辅助部分、弹出部分和数字化仪部分均被编号，例如 *** POP1。

//后面的部分为解释性说明，不会对菜单的加载和运行产生任何影响。不论是在菜单文件编制过程中，还是在程序编制过程中，适当填写一些注释性的内容，只会有好处，不会有丝毫的坏处。这也是作为一个程序编制人员必备的良好习惯。当过了一段时间再次修改程序的时候，注释会给你节约不少时间，记住，不要把自己锁在门外。

值得特别提示的是，在软件编制中，除作为提示的字符串外，一律不要采用中文，作为程序一部分的各种符号，包括逗号、点、括号、引号等尤其要注意。另外，应尽量避免使用形状相近的字符。

作为菜单文件，并非所有的标签都是不可或缺的，本书仅简单介绍 *** MENUGROUP；*** POPn；*** TOOLBARS；*** IMAGE；*** HELPSTRINGS 等几个部分，更详细的内容参见 AutoCAD“开发人员帮助”。

表 1-4 菜单文件标签

菜单部分标签	用户界面区域	备　注
*** MENUGROUP	菜单组名	必须有，见 test1. mnu 文件第 1 行
*** BUTTONSn	定点设备按钮菜单	可有可无
*** AUXn	系统定点设备菜单	可有可无
*** POPn	下拉菜单和快捷菜单	必须有，见 test1. mnu 文件第 2 ~ 11 行
*** TOOLBARS	工具栏定义	可有可无，见 test1. mnu 文件第 12 ~ 15 行
*** IMAGE	图像控件菜单	可有可无，以后章节中介绍
*** SCREEN	屏幕菜单	可有可无
*** TABLETn	数字化仪菜单	可有可无
*** HELPSTRINGS	当亮显下拉菜单或快捷菜单项时，或者当光标位于工具栏按钮上时，显示状态栏中的文字	可有可无，见 test1. mnu 文件第 16 ~ 17 行
*** ACCELERATORS	快捷键（或加速键）定义	可有可无

菜单文件中无需包含每个可能的菜单部分。建议创建小菜单文件，以便在需要时加载和卸载（使用 MENULOAD 和 MENUUNLOAD 命令）。使用较小的文件能够更好地控制系统资源，更方便程序调试，并且更容易进行消化、学习。

在工作目录下，按照上述内容编制 test1. mnu 文件。(可以使用“记事本 notepad. exe”)。

首先，在命令提示状态下输入：menuload ↙（或直接采用 MENU 命令加载）。

然后，在 <菜单组> 中加载 test1. mnu，在“菜单栏”中选中 test1. mnu 插入“线层”菜单，看一看出现了什么变化?

会出现一个“线层”下拉菜单。对照“test1. mnu”文件，不难发现下拉菜单中显示的各项内容对应于“test1. mnu”文件中的位置，均在一对方括号中间，只是“&”变成了相近符号的下划线。

还出现了一个工具条，顶部有一个按钮。对照“test1. mnu”文件，可以看出“test1. mnu”文件中列出的“JGTB _ JS0. bmp”文件名变成了按钮图像。

像往常一样，单击每一个命令，看看都能做些什么?

“命令 1”，“命令 2”，工具按钮，均执行了同样任务——将“JS0”层设置为当前层，然后执行命令“line”。即一对方括号后面的字符串内容相同，都是“^C^C - LAYER;S;JS0;;_ LINE”。不同点是，当鼠标停留在“命令 2”或按钮上面的时候，状态栏显示命令功能提示信息“在 JX0 层绘制细实线——0. 0”，上述内容可以在“test1. mnu”第 17 行找到，存在于一对方括号中间，仔细观察，可以发现，它们是通过“ID _ JGTBjs0”字符串进行关联的。

在 AutoCAD 中命令状态下，将方括号后面的字符串逐一输入，分号用空格或〈Enter〉键代替，就会看到命令执行的过程。4，5，6，7，9，11，15 能够正常执行。注意“ - ”与“_”的区别：“ - ”将阻止一些命令的对话框弹出，而允许在命令行输入参数；“_”对国际语言的支持，自动将其后的命令翻译为其他语言。

第 6 行，select，可以提示并允许选择，组成选择集供后面命令使用。

第 7 行，$（if，$（eq，$（getvar，clayer），JS0），!.）是一种条件判断，当前层是“JS0”层时，该菜单显示“√”。

一对方括号后的（‘）使得本命令不但可以单独执行，还可以在其他命令执行过程中执行。

第 10 行，是一种菜单引用方式。

看到^C^C 后面的内容是不是很眼熟？对了，很多执行部分的语句都是大家所熟悉的命令执行语句的变形内容。

注：菜单宏中的分号（;）相当于〈Enter〉键。如果行是以控制字符、反斜杠（\）、正号（+）或分号（;）结尾，则 AutoCAD 不在行末添加空格和^C^C。这相当于按〈ESC〉键两次。用来终止正在运行的命令。

特殊字符反斜杠（\）用于创建停顿，以便用户输入参数。

一对方括号中间的内容，是显示在下拉菜单中的提示索引信息，完全可以改成自己喜欢的内容。当然，文字有利于编程或展示，形象的符号更有利于应用绘图。

看到上面的实例和解释，相信大家已经对菜单文件有了一定的了解，应该可以独立完成前面的功能规格书了。如果想懂得每个语句的含义，可以马上打开“AutoCAD 开发人员帮助”文件（acad_dev.chm）中“自定义手册”中的“菜单文件”部分。如果已经懂得了菜单文件的结构，但还是不明白诸如：line 、layer、change 等命令的含义，那就有必要补习一下 AutoCAD 一些操作方面的知识了。

下面就给出完整的菜单文件实例，对照一下吧。

JGTB.MNU 文件：

```
// JGTB.MNU for the AutoCAD JGTB Tools

***MENUGROUP=JGTB

***POP1
[&1 线层]
ID_JGTBjs0 [&A 细实线 ——   0.0]^C^C^P-LAYER;S;JS0;;_LINE
ID_JGTBjs3 [&B 中实线 ——   0.3]^C^C^P-LAYER;S;JS3;;_LINE
ID_JGTBjs5 [&C 粗实线 ——   0.5]^C^C^P-LAYER;S;JS5;;_LINE
ID_JGTBjx0 [&D 细虚线 --------- 0.0]^C^C^P-LAYER;S;JX0;;_LINE
ID_JGTBjd0 [&E 点划线 —·— 0.0]^C^C^P-LAYER;S;JD0;;_LINE
ID_JGTBjdm [&F 文本层 ——   0.0]^C^C^P-LAYER;S;JDM;;_LINE
[--]
ID_JGTBcs0 [&G → 细实线]^C^C^Pselect \change;P;;p;LA;JS0;c;BYLAYER;LT;BYLAYER;;
ID_JGTBcs3 [&H → 中实线]^C^C^Pselect \change;P;;p;LA;JS3;c;BYLAYER;LT;BYLAYER;;
ID_JGTBcs5 [&I → 粗实线]^C^C^Pselect \change;P;;p;LA;JS5;c;BYLAYER;LT;BYLAYER;;
```

```
ID_JGTBcx0 [&J → 细虚线]^C^C^Pselect \change;P;;p;LA;JX0;c;BYLAYER;LT;BYLAYER;;
ID_JGTBcd0 [&K → 点划线]^C^C^Pselect \change;P;;p;LA;JD0;c;BYLAYER;LT;BYLAYER;;
ID_JGTBcsd [&L → 文本层]^C^C^Pselect \change;P;;p;LA;JDM;c;BYLAYER;LT;BYLAYER;;
[--]
ID_JGTBqj [&M 切断线,在交点]^C^C_OSNAP INT _break \_f \@
ID_JGTBqs [&N 切断线,在拾取位置]^C^C $S=X $S=break0 _break;\@ ;
ID_JGTBls [&O 伸缩线端点]^C^C_lengthen dy
[--]
ID_JGTB0 [$(if,$(eq,$(getvar,clayer),0),!.) &0 —— ↙]'-LAYER;S;0;;
ID_JGTBS0 [$(if,$(eq,$(getvar,clayer),JS0),!.)&1 细实线 —— 0.0 ↙]'-LAYER;S;JS0;;
ID_JGTBS3 [$(if,$(eq,$(getvar,clayer),JS3),!.)&2 中实线 —— 0.3 ↙]'-LAYER;S;JS3;;
ID_JGTBS5 [$(if,$(eq,$(getvar,clayer),JS5),!.)&3 粗实线 —— 0.5 ↙]'-LAYER;S;JS5;;
ID_JGTBX0 [$(if,$(eq,$(getvar,clayer),JX0),!.)&4 细虚线 --- 0.0 ↙]'-LAYER;S;JX0;;
ID_JGTBD0 [$(if,$(eq,$(getvar,clayer),JD0),!.)&5 点划线 —·— 0.0 ↙]'-LAYER;S;JD0;;
ID_JGTBDM [$(if,$(eq,$(getvar,clayer),JDM),!.)&6 文本层 —— 0.0 ↙]'-LAYER;S;JDM;;
          [--]
ID_JGTBbianb [&P 颜色变白]^C^C^P-LAYER;c;7;JS3,JS5,JX0,JX3,JD0,JSD;;^p
ID_JGTBhuif [&Q 恢复颜]^C^C^P-LAYER;c;6;JS3;C;5;JS5;c;1;JD0;c;4;JX0;c;3;JX3;c;2;
JSD;;^P

***HELPSTRINGS
ID_JGTBjs0 [在 JS0 层绘制细实线 ——     0.0]
ID_JGTBjs3 [在 JS3 层绘制中实线 ——     0.3]
ID_JGTBjs5 [在 JS5 层绘制粗实线 ——     0.5]
ID_JGTBjx0 [在 JX0 层绘制细虚线 ---    0.0]
ID_JGTBjd0 [在 JD0 层绘制点划线 —·—    0.0]
ID_JGTBjdm [在 JDM 层绘制文本层 ——     0.0]
ID_JGTBcs0 [修改目标线至 JS0 细实线层]
ID_JGTBcs3 [修改目标线至 JS3 中实线层]
ID_JGTBcs5 [修改目标线至 JS5 粗实线层]
ID_JGTBcx0 [修改目标线至 JX0 细虚线层]
ID_JGTBcd0 [修改目标线至 JD0 点划线层]
ID_JGTBcdm [修改目标线或其他图元至 JDM 文本层]
ID_JGTB0 [置当前层为 0 层 —— 0.0]
ID_JGTBS0 [置当前层为细实线层 JS0 ——   0.0]
ID_JGTBS3 [置当前层为中实线层 JS3 ——   0.3]
ID_JGTBS5 [置当前层为粗实线层 JS5 ——   0.5]
ID_JGTBX0 [置当前层为细虚线层 JX0 ---0.0]
ID_JGTBD0 [置当前层为点划线层 JD0 —·—0.0]
ID_JGTBDM [置当前层为文本层 JDM ——     0.0]
ID_JGTBqj [切断目标线,在交点]
ID_JGTBqs [切断目标线,在拾取位置]
```

```
ID_JGTBls [伸缩线的端点]
ID_JGTBbianb [各线层颜色变白,利于打印]
ID_JGTBhuif [各线层恢复系统原始设置颜色]
//jgtb.mnu 结束
```

方括号内的内容，只是识别信息或是提示信息，应该尽量简洁、明了，也可以尝试采用一些符号，更有利于识别。M、N、O 项代码有可能与版本有关。为了回避版本问题，可以考虑采用 AutoLISP 编程。

2. AutoLISP 程序的实现

上一节介绍了本章开篇功能规格书要求的各项功能的菜单实现方法，对于同样的功能，本节将介绍使用 AutoLISP 编程实现的方法。

实例 1

test1. lsp

行号	test1. lsp 内容	
1	；细实线绘图程序	；之后的内容为解释内容，不参与运行。
2	(command "layer" "s" "JS0" "")	；设置当前层为 JS0 层
3	(command " line")	；画线

在工作目录下，按照上述内容编制 Test1. lsp 文件。（可以使用“记事本 notepad. exe”;推荐使用 AutoCAD 附带的“Visual LISP 编辑器”）。不必编入行号和表格，行号和表格只是为了解释软件编制而设置的。

在命令提示状态下输入：(load"test1. lsp")↙

执行结果：将 JS0 设为当前层→在 js0 层执行画线命令 line。

实例 2

test2. lsp

行号	test2. lsp 内容	
1	(defun JS0 ( )	；定义一个可执行函数（JS0）
2	(setvar "cmdecho" 0)	；设置程序不透明执行
3	(command "layer" "s" "JS0" "")	；设置当前层为 JS0 层
4	(command "line")	；画线
5	(princ)	；释放
6	)；defun 结束	

在工作目录下，按照上述内容编制 test2. lsp 文件。

首先，在命令提示状态下输入：(load"test2. lsp")↙

然后，在 command：状态下输入：(js0) ↙

执行结果：将 JS0 设为当前层→在 js0 层执行画线命令 line。

实例3

test3. lsp

test3. lsp 内容

```
(defun c:JS0( )                          ;定义一个可执行命令 JS0
(setvar "cmdecho" 0)                     ;设置程序不透明执行
(command "layer" "s" "JS0" "" )          ;设置当前层为 JS0 层
(command "line")                         ;画线
(princ)                                  ;释放
);defun 结束
```

在工作目录下，按照上述内容编制 test3. lsp 文件。

首先，在命令提示状态下输入：(load"test3. lsp")↙

然后，在 command：状态下输入：(js0) ↙

执行结果：将 JS0 设为当前层→在 js0 层执行画线命令 line。

实例4

test4. lsp

test4. lsp 内容

```
(defun c:lay( / #fh #lay )
(setvar "cmdecho" 0)
(if (NULL (setq #fh(open "JGTB. INI" "r")))        ;以读取方式打开文件"JGTB. INI"
(progn (alert "JGTB. INI 文件不存在. ")(exit)))     ;如未找到文件,警告并退出
(if (NULL(setq #lay(read - line #fh)))              ;读取文件首行信息
;首行内容为("JS0" "JS3" "JS5" "JX0" "JD0" "JDM")读者可以根据个体情况自行设置
(progn
(close #fh)                                         ;关闭文件
(alert "JGTB. ini 文件 1 行信息不存在<层名>. ")     ;显示警告提示内容
(exit)))                                            ;如首行信息无,退出
(close #fh)
(prin1 #lay)                                        ;命令行回显变量#lay 内容
(setvar "users5" #lay)                              ;层设置信息字符串存入系统变量
(princ))

(defun c:JS00( / #lay #js0)                         ;定义一个可执行命令 JS00
```

（续）

行号	test4. lsp 内容	
15	(setvar "cmdecho" 0)	;设置程序不透明执行
16	(getvar "users5" #lay)	;从系统变量提取层设置信息字符串
17	(setq #lay (read #lay))	;将字符串变成表
18	(setq #js0 (nth 0 #lay))	;提取表的第一个元素
19	(prompt "层名:") (prin1 #lay)	
20	(command "layer" "s" #JS0 "")	;设置当前层为 JS0 层
21	(command "line")	;画线
22	(princ)	;释放
23	);defun 结束	

在工作目录下，按照上述内容编制 test4. lsp 文件。

首先，在命令提示状态下输入：(load" test4. lsp") ↙

然后，在 command：状态下输入：lay ↙

第三，在 command：状态下输入：js00 ↙

执行结果：将 JS0 设为当前层→在 js0 层执行画线命令 line。

test1、test2、test3、test4，以及前面介绍的菜单文件中代码都执行相同的层设置及画线命令，但 js00 提供了操作人员更加灵活的图层设置权利，此种设置，可以承接设计人员原有的系统设置，不必强行修改来适应新的软件系统，是值得提倡的选择。

实例 5

test5. lsp

行号	test5. lsp 内容	
1	(DEFUN C_C(#lay / #S #lay #a #l #ss #d)	;定义带有参数#lay 的子函数 c_c
2	(SETVAR "CMDECHO" 0)	;设置系统变量"CMDECHO"的值为 0
3	(SETQ #s(SSGET '((-4 . "<OR")(0 . "LINE")(0 . "POLYLINE") (0 . "LWPOLYLINE")	
4	(0 . "ARC")(0 . "CIRCLE") (-4 . "OR>"))))	;有条件选取实体集合并存入变量#s
5	(if (/= #s nil)	;如果变量#s 不为空执行 5——15 行内容
6	(progn	
7	(setq #l(sslength #s))	;计算#s 集合所含实体个数，存入变量#l
8	(setq #a 0)	;变量#a 赋值 0
9	(while(< #a #l)	;如果#a < #l 循环执行 9——16 行内容；如#a > #l 跳出循环至 19 行
10	(setQ #ss(ssname #s #a))	;在#s 集合中依次提取实体，存入变量#ss
11	(SETQ #d(cdr(assoc 0 (ENTGET #sS))))	;得到变量#ss 代表的实体的句柄，存入变量#d
12	(command "_change" #ss "" "p" "_layer" #lay "color" "BYLAYER" "ltype" "BYLAYER" "")	
13		;用命令形式将#d 代表的实体移到参数#lay 指示的层
14	(if(or(= #d "LWPOLYLINE")(= #d "POLYLINE"))	