



AutoCAD

定制与应用程序设计

■ 杨广旋 杨 浩 孙 静 编著



科学出版社
www.sciencep.com

AutoCAD 定制与应用程序设计

杨广旋 杨 浩 孙 静 编著

科学出版社

北 京

内 容 简 介

本书全面介绍了绘图软件 AutoCAD 的定制和开发语言 AutoLISP、Visual LISP 及 VBA。内容包括 AutoCAD 定制、AutoLISP 及 Visual LISP 语言、DCL 对话框、AutoCAD VBA 语言、ActiveX 对象与自动化接口技术、VLISP 和 VBA 应用程序设计实例及参数化绘图。另外,为方便用户学习和工作,书后还附有三个附录及光盘。

本书从入门开始,结合实例系统地介绍了 AutoCAD 各开发语言的概念和功能,并用两章列举了多种类型的设计实例,详细说明了应用程序的开发过程和技巧,并试图给出该类程序设计的一种模式。光盘中的实例可直接在设计中应用,还可以根据实际需要稍加修改后变成自己的应用程序。本书力求使读者能成为 AutoCAD 应用程序的开发能手。

本书可作为工程技术人员、大专院校师生和广大计算机用户的技术参考书,也可作为 AutoCAD 应用软件开发人员的自学教材。

图书在版编目(CIP)数据

AutoCAD 定制与应用程序设计/杨广旋等编著. —北京:科学出版社, 2005

ISBN 7-03-014887-8

I. A… II. 杨… III. 计算机辅助设计—应用软件, AutoCAD—程序设计 IV. TP391.72

中国版本图书馆 CIP 数据核字(2005)第 003322 号

责任编辑:赵卫江 / 责任校对:柏连海

责任印制:吕春珉 / 封面设计:王 浩

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2005年2月第 一 版	开本: 787×1092 1/16
2005年2月第一次印刷	印张: 34 3/4
印数: 1—4 000	字数: 806 000

定价: 56.00 元(含光盘)

(如有印装质量问题,我社负责调换<双青>)

前 言

AutoCAD 以其功能强大、适用面广、操作简单易学而被各个行业广大的设计与绘图工作者所喜爱,使其成为目前国内外使用最为广泛的 CAD 软件。大多数用户在实际的设计和绘图实践中已深切地感受到 AutoCAD 提供的绘图和编辑命令的魅力。

AutoCAD 毕竟是一个通用的绘图软件,不可能面面俱到地照顾到各个行业的所有要求。AutoCAD 的基本功能对于已经定型的标准件、通用件、成系列的产品、具有复杂型面的图形以及要通过复杂计算后才能绘制的零件图表现出了明显的不足。对于这类问题,用户可使用随软件所提供的多种开发语言通过编程的方法来解决。

AutoCAD 应用程序的设计也属于一类软件开发技术,在国外已很普遍,形成了一大产业,极大地促进了制造技术的发展,我国这方面的工作还很落后。为提高我国的 CAD 技术,提高产品的竞争能力,需要更多的人掌握这方面的技术。一方面因工作的需要,广大的 AutoCAD 用户要解决一类零件的设计问题,提高设计与绘图的效率和质量,就必须掌握这些程序设计语言;另一方面,众多各专业的新手需要在自己的工作中一试身手,有所作为,更需要学习 AutoCAD 的程序设计语言来解决工作中的实际问题。

多年来,介绍 AutoCAD 开发语言的书籍时有出现,但多因起点过高,使专业设计工作者和初学者都有所困惑。本书的特点是起点低,从入门开始,系统地介绍 AutoCAD 各开发语言的概念和功能。为使用户能准确地理解和掌握所学的内容,书中列有大量的例句、例题。为使用户在学完本书后能解决工作中的实际问题,书中用两章的篇幅列举了多种类型的设计实例,详细地说明了应用程序的开发过程和技巧,并试图给出该类程序设计的一种框架模式。光盘中的实例可直接在设计中应用,根据实际需要稍加修改后即可变成自己的实用程序。

目前 AutoCAD 所附的开发语言有四种:AutoLISP、Visual LISP、VBA 及 ObjectARX。因篇幅所限,本书只介绍前三种。ObjectARX 及有关与数据库集成工作的内容作者将在另外的两本书中介绍。

本书共分 9 章,内容包括 AutoCAD 定制、AutoLISP 及 Visual LISP 语言、DCL 对话框、AutoCAD VBA 语言、ActiveX 对象与自动化接口技术、VLISP 和 VBA 实例及参数化绘图。另外还附有三个附录,附录 A 给出了 AutoCAD ActiveX 对象模型的方法、属性和事件的简要说明,附录 B 给出了 Visual Basic 语言中的对象、方法、属性、事件及语句的简要说明,附录 C 给出了光盘的用法。光盘中给出了书中的全部例题。

由于作者水平有限,书中错误和不当之处,恳请读者批评指正。

编 者

目 录

1	AutoCAD定制与开发概述	1
1.1	AutoCAD的定制	1
1.2	AutoCAD二次开发技术	1
1.3	AutoCAD二次开发语言	3
1.3.1	AutoLISP语言与Visual LISP语言	3
1.3.2	AutoCAD VBA语言	3
1.3.3	AutoCAD Object ARX语言	4
2	AutoCAD的定制	5
2.1	有关定制的基本知识	5
2.1.1	AutoCAD的运行环境	5
2.1.2	命令搜索过程	7
2.1.3	定制支持文件	8
2.1.4	基本的定制方法	10
2.2	自定义命令	10
2.2.1	程序参数文件acad.pgp的结构	10
2.2.2	定义外部命令	10
2.2.3	定义命令缩写别名	12
2.3	线型的定制	13
2.3.1	线型的定义格式	13
2.3.2	标准线型	14
2.3.3	创建新的简单线型	16
2.3.4	创建新的复合线型	18
2.4	填充图案的定制	20
2.4.1	填充图案的定义格式	20
2.4.2	定义填充图案	21
2.5	形和字体的定制	23
2.5.1	生成字符的方法	23
2.5.2	形及形文件的定义	24
2.5.3	编译、加载和调用形文件	30
2.6	菜单的定制	32
2.6.1	菜单文件	32
2.6.2	菜单文件的格式	33
2.6.3	定制用户菜单	35
2.6.4	定制按钮菜单和定点设备菜单	36

2.6.5	定制下拉菜单和浮动菜单.....	37
2.6.6	图像控件菜单.....	42
2.6.7	菜单帮助.....	44
2.7	工具栏的定制.....	45
2.7.1	工具栏的定义格式.....	45
2.7.2	定制工具栏.....	47
2.7.3	创建自定义按钮.....	52
2.8	DIESEL语言及状态栏的定制.....	56
2.8.1	DIESEL语言函数.....	56
2.8.2	DIESEL中的错误信息显示.....	61
2.8.3	状态栏的定制.....	61
2.8.4	菜单宏中的DIESEL表达式.....	64
2.9	幻灯片和脚本文件.....	67
2.9.1	幻灯片及脚本文件概述.....	67
2.9.2	幻灯片的制作和播放.....	68
2.10	工具选项板的定制.....	71
2.10.1	工具选项板的界面.....	71
2.10.2	工具选项板的操作.....	72
2.10.3	工具选项板的定制.....	75
2.11	各种符号库的建立和定制实例.....	78
2.11.1	加工符号的定制.....	78
2.11.2	工艺符号的定制.....	84
2.11.3	自定义复合线型.....	88
3	AutoLISP与Visual LISP语言.....	91
3.1	概述.....	91
3.1.1	AutoCAD的二次开发.....	91
3.1.2	关于AutoLISP.....	91
3.1.3	关于Visual LISP.....	92
3.2	Visual LISP的程序开发界面.....	93
3.2.1	下拉菜单.....	94
3.2.2	工具栏.....	102
3.2.3	控制台窗口.....	102
3.2.4	状态栏.....	103
3.2.5	程序编辑窗口.....	103
3.3	Visual LISP的数据类型和变量声明.....	103
3.3.1	原子.....	104
3.3.2	表 and 点对.....	105
3.3.3	选择集.....	106



3.3.4	图元名	106
3.3.5	VLA对象	106
3.3.6	文件描述符	107
3.4	Visual LISP的程序结构	107
3.5	Visual LISP的内存分配	108
3.6	Visual LISP的处理函数	108
3.6.1	基本函数	109
3.6.2	实用函数	115
3.6.3	选择集、对象和符号表函数	119
3.6.4	Visual LISP扩展到AutoLISP函数	122
3.6.5	其他函数	125
3.6.6	外部定义命令	130
3.7	AutoLISP/VLISP程序的编辑、加载和运行	131
3.7.1	AutoLISP程序的编辑、加载和运行	131
3.7.2	Visual LISP程序的编辑、编译、加载和运行	132
4	DCL对话框	135
4.1	概述	135
4.2	定义对话框	136
4.2.1	控件及其属性	136
4.2.2	对话框控制语言(DCL)简介	146
4.2.3	设计对话框	149
4.3	对话框的PDB函数	152
4.3.1	对话框的打开和关闭函数	152
4.3.2	控件和属性处理函数	152
4.3.3	列表框和下拉列表框处理函数	153
4.3.4	图像控件处理函数	153
4.3.5	特殊应用数据处理函数	153
4.4	对话框的AutoLISP驱动程序设计	153
4.4.1	对话框驱动程序设计概要	153
4.4.2	动作表达式和回调	154
4.4.3	控件的处理	155
4.4.4	隐藏对话框	158
4.4.5	列表框和下拉列表框的处理	159
4.4.6	图像控件和图像按钮	160
4.4.7	通过AutoLISP程序快速预览对话框	161
4.4.8	当对话框打开时函数的限制	162
4.5	对话框设计实例	162



5 AutoCAD VBA语言	169
5.1 概述	169
5.1.1 VBA语言概述	169
5.1.2 VBA中的有关术语	172
5.2 VBA界面	173
5.2.1 VBA界面	173
5.2.2 VBA下拉菜单	174
5.2.3 VBA工具栏	180
5.2.4 VBA编辑器中的窗口	181
5.3 VBA数据类型	182
5.4 VBA常量和变量	183
5.4.1 VBA常量	183
5.4.2 VBA变量及声明	184
5.5 VBA运算符与表达式	186
5.5.1 VBA算术运算符	186
5.5.2 VBA关系运算符	187
5.5.3 VBA连接运算符	187
5.5.4 VBA逻辑运算符	187
5.5.5 VBA表达式	188
5.6 VBA字符串处理	189
5.6.1 字符串运算	189
5.6.2 字符串处理	189
5.7 VBA程序流程控制	193
5.7.1 VBA顺序结构	193
5.7.2 VBA判断结构	196
5.7.3 VBA转移结构	198
5.7.4 VBA循环结构	198
5.8 VBA数组	200
5.8.1 VBA数组的概念	201
5.8.2 VBA静态数组	201
5.8.3 VBA动态数组	202
5.9 VBA常用内部函数	202
5.9.1 VBA数学函数	202
5.9.2 VBA转换函数	203
5.9.3 VBA日期时间函数	203
5.10 VBA文件操作	204
5.10.1 VBA文件操作函数	204
5.10.2 VBA文件打开和关闭函数	204

5.10.3	VBA文件访问函数的适用性	206
5.10.4	VBA使用文件系统对象（FSO）操作文件	207
5.11	VBA工程、模块与过程	208
5.11.1	VBA工程概述	208
5.11.2	模块与过程概述	213
5.11.3	标准模块	214
5.11.4	VBA代码中的过程	215
5.11.5	Sub和Function过程的调用	223
5.11.6	Property过程的调用	224
5.11.7	调用其他模块中的过程	224
5.11.8	过程的嵌套调用和递归调用	225
5.11.9	过程的参数传递	226
5.12	VBA控件	226
5.12.1	VBA控件概述	226
5.12.2	命令按钮	229
5.12.3	列表框与组合列表框	231
5.12.4	单选按钮与复选框	235
5.12.5	文本框	237
5.12.6	图像控件	242
5.12.7	其他控件	246
5.13	VBA用户窗体（UserForm）	247
5.13.1	窗体概述	247
5.13.2	窗体设计	248
5.13.3	窗体的加载和卸载	249
5.14	VBA宏	249
5.15	VBA程序的开发过程	250
5.16	VBA工程的加载和卸载	251
5.16.1	加载工程	251
5.16.2	卸载工程	252
5.16.3	分发应用程序	252
6	ActiveX对象与自动化接口技术	253
6.1	概述	253
6.1.1	ActiveX对象与自动化接口技术	253
6.1.2	ActiveX对象与自动化接口技术的形成和发展	254
6.2	AutoCAD的对象模型	255
6.2.1	AutoCAD的ActiveX对象模型	255
6.2.2	AutoCAD对象的层次结构	257
6.3	AutoCAD对象的功能	258

6.3.1	Application对象	258
6.3.2	Document对象	258
6.3.3	集合对象	258
6.3.4	图形对象	259
6.3.5	非图形对象	260
6.3.6	Preferences对象	260
6.3.7	Plot对象	260
6.3.8	Utility对象	260
6.4	AutoCAD对象的访问	261
6.4.1	访问对象的层次结构	261
6.4.2	访问Application对象	263
6.4.3	集合对象的访问	263
6.5	ActiveX对象的属性、方法和事件	264
6.5.1	ActiveX对象的属性、方法和事件的查询	264
6.5.2	ActiveX事件的使用	274
6.6	在Visual LISP中使用ActiveX对象	280
6.6.1	在Visual LISP中处理ActiveX对象的函数	280
6.6.2	在Visual LISP中访问ActiveX对象的一般方法	281
6.6.3	在Visual LISP中使用检验工具查询对象的属性	285
6.7	在Visual LISP中使用ActiveX变量的方法	286
6.7.1	安全数组	286
6.7.2	变体 (Variant)	288
6.7.3	使用带变量的安全数组	290
6.7.4	变量转换	291
6.8	在Visual LISP中对ActiveX对象、方法及属性的访问	292
6.8.1	查看和更新对象的属性	292
6.8.2	使用ActiveX方法返回参数值	295
6.8.3	列出对象的方法和属性	296
6.8.4	确定方法和属性是否可应用到指定的对象	298
6.8.5	使用集合对象	298
6.8.6	提取集合中的成员对象	301
6.8.7	释放对象和解除内存	302
6.8.8	转换对象标识类型	302
6.8.9	处理通过ActiveX方法返回的错误	304
6.9	在Visual LISP中引用其他应用程序	305
6.9.1	引入一个类型库	305
6.9.2	使用ActiveX连接其他应用程序	306
6.9.3	编制一个样例应用程序	307

6.9.4 使用与引入类型库无关的ActiveX函数.....	309
6.10 VBA与其他应用程序交互.....	311
6.10.1 VBA与Visual LISP应用程序交互.....	311
6.10.2 VBA与其他应用程序交互.....	311
6.11 VBA与Windows API交互.....	314
6.11.1 Windows动态链接库DLL.....	314
6.11.2 Windows API编程.....	316
6.11.3 VBA与Windows API交互.....	317
7 Visual LISP应用程序设计实例.....	318
7.1 键槽特征设计.....	318
7.2 齿轮特征设计.....	348
7.3 螺栓的绘制.....	363
7.4 弹簧的自动绘制.....	387
7.5 齿轮参数表的自动绘制.....	400
8 VBA应用程序设计实例.....	423
8.1 键槽特征设计.....	423
8.2 齿轮特征设计.....	435
8.3 螺栓的绘制.....	445
8.4 弹簧的自动绘制.....	461
8.5 齿轮参数表的自动绘制.....	469
9 参数化绘图.....	486
9.1 概述.....	486
9.2 参数化绘图原理.....	486
9.2.1 参数化绘图原理.....	486
9.2.2 参数化绘图程序设计步骤.....	487
9.3 参数化绘图与图形数据库.....	487
9.4 尺寸驱动技术.....	488
9.4.1 尺寸驱动技术简介.....	488
9.4.2 尺寸驱动技术的实现方法.....	488
9.5 三维立体图的参数化.....	493
9.6 具有复杂空间曲面零件的绘制.....	494
9.6.1 具有复杂空间曲面零件图绘制概述.....	494
9.6.2 仿照零件加工方法绘制复杂曲面.....	494
9.6.3 绘图程序设计的一般步骤.....	494
9.6.4 绘图实例——蜗杆零件的绘制.....	495
附录A AutoCAD对象模型参考.....	509
附录B Visual Basic语言参考.....	534
附录C 光盘使用方法.....	541

1

AutoCAD 定制与开发概述

AutoCAD 是一个通用的绘图软件, 其功能强大, 适用面广, 受到了多个行业用户的欢迎。其所提供的命令都是绘图的基本功能, 方便实用, 可以满足用户绘制各种图形的需求。但对具体用户而言, 由于行业不同, 产品不同, 工作的对象及使用的标准差异很大, 再加上用户的工作习惯及历史图形的使用问题等, 都影响绘图效率的进一步提高。另外, 还有一些具有复杂型面的图形无法使用通用的绘图命令实现, 这些都是掌握了 AutoCAD 绘图功能的用户急需解决的问题。为此, AutoCAD 的开发从 AutoCAD 的早期版本开始就提供了定制和二次开发语言。多年来, 随着版本的更新, 定制与开发功能也不断完善和发展。至今, 定制与开发功能已是广大用户进一步提高设计效率、提高产品性能、立于不败之地的必备技术。

1.1 AutoCAD 的定制

定制实质上就是 AutoCAD 的个性化、用户化和专业化。也就是用户根据自己专业和产品的需要以及使用的习惯, 改造 AutoCAD 的通用界面、操作方法, 补充和完善常用的线型、填充图案、形和字体等等, 满足用户专业、产品及个人习惯上的需求。

定制的内容包括: 建立自定义的线型、填充图案、形和字体, 满足用户专业的需要; 创建自定义菜单和工具栏, 将用户所常用的命令、符号、图块等定制成下拉菜单、图像菜单、快捷菜单和工具条等, 以方便调用; 创建自定义状态栏; 创建命令宏; 定义命令别名; 创建执行外部程序命令以及制作外部应用程序帮助文件等。定制是已学会初步绘图功能的用户进一步提高绘图效率和质量的必然需求。因此, 掌握定制方法十分重要。

1.2 AutoCAD 二次开发技术

Autodesk 公司自从 1982 年底推出 AutoCAD 1.0 版本以来, 不断地对其进行改进、发展和完善: 从最初的只完成二维绘图的 1.0、2.0 版本, 发展成为后来能完成三维复杂造型的 R12.0 到 R14.0 版本。20 世纪末以来 Autodesk 公司又推出了跨世纪的、大型的、一体化的网络版本 AutoCAD 2000/2002, 使其成为千禧之年功能最为强大的计算机辅助设计软件。2003 年初又发行了 AutoCAD 2004 版, 功能得到进一步完善和增强。

AutoCAD 是一种交互式的通用绘图软件, 尽管它的功能非常强大, 可以高速度、高质量地完成从前人们用手工方式不可想象的复杂工作。但是, 在这个技术飞速发展的

年代, 人们的要求也愈来愈高, 对一个产品的要求也愈来愈多, 使用 AutoCAD 通用功能还不能满足人们各方面的要求, 这主要表现在:

(1) AutoCAD 是一种绘图软件, 目前主要还是用于辅助绘图, 还不是完全意义上的辅助设计软件。其虽然集成了 Windows 的图形设备接口 (GDI), 可以完成二维、三维图形的复杂造型, 也具备一些与其他应用程序交换文件的功能, 但它不具备各种工程分析和计算、优化设计、动态仿真和文档制作 (如完成各种报告、表格、文件和说明书等) 的能力。

(2) AutoCAD 是一种交互式的绘图软件, 不具备参数驱动的绘图功能, 当然更没有特征设计的能力。

(3) AutoCAD 是一种通用的绘图软件, 适用于多个行业, 因而对用户来说也就不可能有个性和专业特性。虽然也提供了不少的定制方法, 但是对于某些专业要求, 如标准件库、专业的零件图库、专业的符号库等是需要用户自己建立的。

(4) AutoCAD 不具有大型数据库的功能, 只能表示一个实体的图形属性 (如线型、颜色、线宽、图层、视图等), 而不能完全反映其他非图形信息 (如材料、价格、质量、厂商、市场、生产和销售计划等)。这对于网上联合设计、产品的更新换代和市场开拓无疑都是一个缺陷。

AutoCAD 作为专业的绘图软件, 没有也不可能在其内部解决上面提到的所有问题, 这也不是应用软件发展的方向。但是在实际的工程设计中, 的确是要解决这些问题的。怎么办呢? 方法就是在完善本身功能的基础上, 用专用的接口软件将各专业软件连接起来协同工作。为此, AutoCAD 提供了非常强大的二次开发功能。

AutoCAD 提供的二次开发语言从最初引进 LISP 表处理语言而生成的 AutoLISP 语言, 到随后引进 C 语言而生成的 ADS 语言。这两种语言都属于结构化的编程语言, 是基于顺序、条件、循环和转移语句等结构化编程方式的语言。自 20 世纪 80 年代推出面向对象编程技术以来, 面向对象的程序设计语言也不断地出现, 如 C++, 微软公司的 Visual Basic (以下简称 VB)、Visual C++、Visual J++ 等。面向对象编程语言内定义了大量的类和成员函数, 其内封装了绝大部分 Windows API 函数和 Windows 标准控件。人们利用类的继承性、多态性使编程像在某个组件平台上, 用各种标准的控件和少量的编程组装应用程序一样, 从而使应用程序的开发变得更加简单方便, 极大地缩短了开发周期, 提高了代码的可靠性和可重用性。该种类型的程序也突破了结构化程序比较呆板的工作方式, 根据需要在程序中设置某些对象, 并用该对象发生的事件所产生的消息, 驱动相应的处理过程来工作, 具有灵活性, 更贴近于人们的工作思维方式。VBA、ARX 就是面向对象的编程语言, 提供了一系列用于定义新类的宏以及对 AutoCAD 现有类的扩充功能, 并提供了内容丰富的类库。

20 世纪 90 年代, 随着面向对象编程技术的蓬勃发展, AutoCAD 也紧随其后, 淘汰了传统的格式化编程方式的 ADS 语言, 引进了可视化的 VB 语言, 从而生成了功能强大的 VBA 语言。同时又推出了以 C++ 语言为基础的功能更为强大的 ObjectARX 语言。由于这些面向对象语言的推出, AutoCAD 添加了 ActiveX 对象模型及自动化接口技术 Active Automation。因为 AutoLISP 语言历史长久, 已有大量的 AutoLISP 应用程序正在

应用, 为不用户的这部分投资报废, 将 AutoLISP 改造成可使用 ActiveX 对象模型的 Visual LISP 语言。

由于以上技术和语言的推出, 使 AutoCAD 能够访问支持 ActiveX 对象的其他 Windows 应用程序, 如 Word、Excel 等。与其他应用程序和数据库集成工作, 从而进一步扩大了 AutoCAD 的应用领域。

利用这些二次开发工具不但可以完成许多定制功能, 还可以制作用户专用的标准件库、通用零部件库、专用符号库等; 可以编制许多专用的绘图程序、参数驱动绘图程序、特征设计程序、自动装配程序, 还可以编制同外部专业应用软件和数据库连接的接口软件等。

1.3 AutoCAD 二次开发语言

1.3.1 AutoLISP 语言与 Visual LISP 语言

AutoCAD 2004 的开发语言有 AutoLISP、Visual LISP、VBA 及 ObjectARX 四种。AutoLISP 语言是最早推出的开发语言, 之所以保留至今是因为, 其简单实用, 编程容易, 用户只要在命令行输入程序即可随时看到运行的结果。另外用户手中有大量的专用程序可用, 为充分利用这些资源, 需要继续保留 AutoLISP 语言。为了适应发展的要求, 多年来也在不断地增加和完善功能函数, 这为 AutoCAD 成为真正意义上的 CAD 软件增色不少。

Visual LISP 语言是为了适应编程语言发展的需要, 在 AutoLISP 的基础上添加了对 ActiveX 对象模型的支持而形成的具有面向对象功能的编程语言, 集 AutoLISP 语法与 ActiveX 对象模型于一体, 功能更为强大。Visual LISP 语言还提供了集成开发环境, 极大地减轻了程序开发的工作量, 提高了程序开发的效率。但 Visual LISP 毕竟是在 AutoLISP 基础上发展而成的, 既有 AutoLISP 编程简单的优点, 也有使用 ActiveX 方法因编程复杂而使程序代码增长的缺点。

1.3.2 AutoCAD VBA 语言

VB 语言是近年来非常流行的可视化通用编程语言, 其编程简单易懂, 容易学习, 是广大软件设计者非常喜爱的一种语言。AutoCAD 的开发者将 VB 语言引进 AutoCAD 形成了 VBA 语言。VBA 语言继承了 VB 语言的所有优点, 同时具备了方便调用 ActiveX 所有方法和属性的功能, 成为 AutoCAD 应用程序开发者们的宠。当然 VBA 与 AutoLISP 分属不同的语言系统, 也就失去了在 AutoCAD 环境下使用 AutoLISP 简单方便的优点, 使读写 DXF 文件和组码有一定困难。

以上三种语言都是解释性的编程语言, 使用它们编制的 AutoCAD 应用程序运行时需通过 ActiveX Automation 自动化接口与 ActiveX 对象打交道, 进而控制图形对象。它们与其他如 Word VBA、Excel VBA 应用程序一样, 完全属于外部应用程序, 运行的速度受到很大的影响。

1.3.3 AutoCAD Object ARX 语言

近年来随着面向对象编程技术的发展和完善, Autodesk 公司为了降低编程的难度, 提高程序的重用度以及进一步加快所编程序的运行速度, 又专门开发并推出了以 C++ 语言为基础的 ARX (AutoCAD Run-Time Extension), 即 AutoCAD 运行时扩展编程语言, 简称 ObjectARX 开发语言。

ARX 应用程序实质上是一个 Windows 下的动态链接库 (DLL) 文件, 与 AutoCAD 共享内存地址空间, 可直接调用 AutoCAD 的 API 函数, 使系统在某一时刻使用的资源最少, 并能使 AutoCAD 直接与 C++ 进行通信, 从而提高了程序的运行速度。

使用 ObjectARX 程序开发技术, 可以方便地利用微软的基础类库 MFC 创建快捷、高效、紧凑的 AutoCAD 应用程序及应用程序界面, 使 ARX 应用程序不仅具有交互性、综合性, 而且具有智能性。

使用 ObjectARX 集成开发环境开发的程序, 必须用 MS Visual C++ 6.0 版的编译器编译, 编译后的 ARX 程序可以加载到 AutoCAD 中供调用。

由于以上程序设计语言的内容太多, 本书只能介绍前三种语言。有关 ObjectARX 语言及程序设计中所涉及的数据库集成工作的内容将分别放在《AutoCAD 程序设计语言 ObjectARX》及《AutoCAD 与数据库集成及接口程序设计》中介绍。

2 AutoCAD 的定制

AutoCAD 是一个开放型的通用绘图软件。当安装完 AutoCAD 200X, 并且每次打开它时, 系统会自动进入默认的设置状态。默认设置可满足多方面的绘图要求, 但对于某个具体用户而言, 能不能使用某种方法, 根据用户的使用习惯和专业需要, 改变一下绘图界面, 增加一些新型的诸如线型、填充图案、字体等功能呢? 本章要介绍的定制功能就可以完成该项任务。

所谓定制, 实质上就是个性化、用户化和专业化。用户可以通过修改或创建某些 ASCII 文本文件来完成这些工作。

AutoCAD 200X 的基本定制特性包含以下一些内容:

- (1) 组织程序文件、支持文件和图形文件。
- (2) 修改用户界面, 创建自定义菜单、工具栏。
- (3) 在 AutoCAD 200X 内部执行外部程序命令。
- (4) 创建自定义线型、填充图案、形和字体。
- (5) 创建自定义的状态栏。
- (6) 创建组合执行 AutoCAD 200X 命令的脚本文件。
- (7) 重定义命令和定义命令的缩写和别名。

2.1 有关定制的基本知识

在 AutoCAD 200X 中, 多数的定制功能需要用户通过修改或创建 ASCII 文本文件来实现, 因此, 用户必须熟悉使用文本编辑器 (如 Windows 2000 下的写字板) 或文字处理软件 (如 Word 2000), 通过这些软件可将所编辑的文本文件存储成 ASCII 格式。

注意: 在修改任何标准 AutoCAD 支持文件 (如菜单文件) 前, 应对其进行备份, 以便以后恢复。

2.1.1 AutoCAD 的运行环境

在默认情况下, AutoCAD 200X 程序和支持文件都已按照合理的逻辑要求被有效地组织起来。如果这种组织不满足用户的要求, 就需改变原来的设置。但是, 某些应用程序需要在特定的位置定位这些文件, 用户在修改后要验证所做的修改与应用程序的要求是否发生冲突。在没有指定驱动器和目录的完整路径的情况下, AutoCAD 仅在库搜索路径中查找所需要的文件。

1. 库搜索路径

当 AutoCAD 的某个命令被调用后, 系统都按照库路径所指定的顺序去搜寻该命令及所用的文件。通常的搜索顺序为:

- (1) 首先搜索当前目录 (启动 AutoCAD 200X 时, AutoCAD 启动程序所在的目录)。
- (2) 接着搜索包含当前图形文件的目录。
- (3) 然后搜索 AutoCAD “选项” 对话框的 “文件” 选项卡中所指定搜索路径上的目录。
- (4) 最后搜索包含 AutoCAD 程序文件的目录。

要说明的一点是: 若一个文件不在所搜索的路径中, 则必须在搜索该文件之前, 事先指定该文件名和路径, AutoCAD 才能找到该文件。例如, 你想插入图形文件 `parts.dwg` 到当前的图形中, 而该文件又不在库搜索路径中, 这时你必须指定该文件的完整路径和文件名。操作方法如下:

命令: `-insert` ✓

输入块名或 [?]: `/filesz/olddwgs/parts` ✓

2. 目录结构

在 AutoCAD 中, 文件的目录一般显示为树状目录结构。这里推荐用户将所补充的文件 (如用 AutoLISP 语言开发的应用程序和菜单文件) 从 AutoCAD 程序和支持文件中分离出来, 将其保存在不同的目录下。这样, 在跟踪某个发生冲突的问题或增强某个应用程序的功能时, 不会影响到其他文件。

AutoCAD 的默认位置为 Program Files 文件夹。用户可以在同一级别上创建新目录 (例如/AcadApps), 并将自定义的 AutoLISP 和 Visual Basic 应用程序、菜单文件和其他第三方应用程序存储到下一级的子目录中。如果需要为相互独立的工作文件维护多个图形文件目录, 可以先创建一个主目录 (例如/AcadJobs), 然后在其下为每项工作创建一个子目录。

3. 多配置文件

当用户配置输入和输出设备时, 所有的配置信息均保存在相应的配置文件中。在默认情况下, 配置文件 AutoCAD 200X.cfg 的默认位置列在 “选项” 对话框的 “文件” 选项卡上, 但用户可以通过另外的设置将其存储到不同的目录下。

通常 AutoCAD 只需用一个配置文件, 但有时也可用多个配置文件。如使用两个不同的输入设备时, 可用两个配置文件, 这样在更改输入设备时用不着重新更改配置文件。AutoCAD 200X 的许多系统变量和 “选项” 对话框中的配置选项也都保存在配置文件中。这些系统变量和配置选项的控制参数的不同值可保存在不同的配置文件中, 可使用 /c 开关指定启动时替换的配置文件, 不用重新设置。

配置和优化配置文件的方法有多种: 用户可以右击桌面上的 AutoCAD 200X 图标, 在打开的 “属性” 对话框的 “快捷方式” 选项卡中配置; 也可以通过控制面板在打开的 “系统特性” 对话框的 “设备管理器 (高级)” 选项卡中, 用设置环境变量的方式进行配