

AutoCAD应用程序开发系列

AutoCAD

VBA 程序开发技术

主编 李长勋

国防工业出版社

<http://www.ndip.cn>

华北水利水电学院图书馆



208645894

TP391.72

L130

AutoCAD 应用程序开发系列

AutoCAD VBA 程序开发技术

主编 李长勋



国防工业出版社

·北京·

864589

内 容 简 介

VBA (Visual Basic for Application) 是微软公司集成在微软系列软件中的强有力的开发工具, AutoCAD 软件也集成了 VBA 开发工具, 同时提供了适用于 VBA 开发的 ActiveX Automation 对象模型。在最新版本的 AutoCAD 2000/2002 中, 这部分功能又得到了增强。

本书从使用的角度出发, 系统地介绍了如何使用 VBA 开发 AutoCAD 2000/2002 应用程序的知识, 主要包括 VBA 的有关概念、基本编程知识、AutoCAD ActiveX 对象模型中各对象的使用方法等, 并提供了大量简单易学的典型应用实例。在内容安排上由浅入深, 而且图文并茂, 是一本非常难得的 VBA 教程。

本书适合于从事开发 AutoCAD 的初学者和中级水平的读者。

图书在版编目 (CIP) 数据

AutoCAD VBA 程序开发技术/李长勋主编. —北京: 国防工业出版社, 2004. 1

(AutoCAD 应用程序开发系列)

ISBN 7-118-03310-3

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

中国版本图书馆 CIP 数据核字 (2003) 第 097983 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 24 547 千字

2004 年 1 月第 1 版 2004 年 1 月北京第 1 次印刷

印数: 1—3000 册 定价: 32.00 元

(本书如有印装错误, 我社负责调换)

前 言

AutoCAD 是美国 Autodesk 公司开发的一个图形设计软件，在全球 CAD 用户中拥有很大的影响。该软件目前最新的版本是 AutoCAD 2002，它提供了强大的图形设计功能，是一个效率非常高的图形设计软件。与以前的版本相比，AutoCAD 2002 在以下几个方面更是远远超出了同类的其他软件：

- 关联标注功能：提供了一个全真的关联标注功能，使得已有尺寸的标注可以随着几何对象的改变而改变，从而大幅提高了绘图的效率。
- 块属性管理功能：提供了修改块属性的简便方法，只要选中一个已经插入的块，编辑它的属性值，所有的改变就可以显示在绘图区内。
- 图层管理器：提供了将图形从一个图层转换到另一个图层的功能，此外还可以使用记忆功能保存上一个图层的状态。
- XML 设计：提供了一个通过 Web 页面传送图形几何信息的功能。

AutoCAD 2002 在提供这些强大功能的同时，也提供了一个功能十分强大的集成开发工具——Visual Basic for Application (VBA)。

今天，高级应用程序的新时代已经到来。以 VBA 为代表的一些更快、更敏捷、更灵活的应用程序正在取代那些原始而古老的近乎于僵硬的软件包，这些高级应用程序的最重要的特点便是它们具有更强的灵活性。

尽管这些新的应用程序和以往的程序一样，十分庞大并且需要编程人员细细琢磨。但是这两者还是有本质的区别，其中最大的不同点是：新的高级应用程序有着更强的变化能力以迎合用户的需求，同时，具有和其他高级程序协同工作的能力。

由于 VBA 易学易用、功能强大，所以很多软件开发商都从微软公司购得了 VBA 的使用许可，将其作为一种开发工具提供给用户使用。同时，Autodesk 公司从 AutoCAD R14.01 版开始，内置了 VBA 开发工具，并提供了适用于 VBA 开发的 ActiveX Automation 对象模型。

AutoCAD 2000/2002 软件中的 ActiveX Automation 对象模型十分有用，因为 Autodesk 公司开放了每一个 AutoCAD 对象模型，这些对象模型显示了每一个对象的方法和属性，并且能够依照用户的愿望和想法去编程，从而使得用户可以随心所欲地编制出个性化的 AutoCAD 2002 软件。

鉴于作者水平有限，加之时间仓促，错误和不当之处在所难免，敬请读者批评指正。

作 者

目 录

第 1 章 AutoCAD 开发工具简介	1
1.1 Auto LISP 语言及 Visual LISP 语言	1
1.1.1 Auto LISP 语言的特点	1
1.1.2 Visual LISP 语言的特点	1
1.2 Object ARX 开发工具	2
1.2.1 Object ARX 的特点	2
1.2.2 Object ARX 的调用机制	2
1.2.3 ARX 类库及函数库	3
1.3 VBA 开发工具	3
1.3.1 AutoCAD ActiveX 自动化技术简介	3
1.3.2 VBA 的功能	4
1.3.3 VBA 作为 AutoCAD 开发工具的优势	4
第 2 章 AutoCAD VBA 入门	6
2.1 什么是 VBA	6
2.1.1 VBA 的功能	6
2.1.2 VBA 与 Visual Basic 的区别	7
2.2 VBA 工程	7
2.3 VBA 管理器	7
2.3.1 打开 VBA 管理器	8
2.3.2 创建新工程	8
2.3.3 加载已有的工程	9
2.3.4 嵌入工程	10
2.3.5 保存嵌入的工程	10
2.3.6 提取工程	10
2.3.7 卸载工程	11
2.4 宏	11
2.4.1 建立新宏	12
2.4.2 运行宏	13
2.4.3 编辑及调试宏	13
2.4.4 宏删除	14
2.4.5 设置工程选项	14
2.5 VBA IDE 开发环境	15

2.5.1	打开 VBA IDE	15
2.5.2	查看工程信息	15
2.5.3	定义工程信息	16
2.5.4	添加新元素	17
2.5.5	导入元素	17
2.5.6	编辑元素	18
2.5.7	程序代码窗口与窗体窗口	19
2.5.8	在 VBA IDE 中运行宏	20
2.5.9	工程的命名和保存	20
2.5.10	引用其他 VBA 工程	22
2.5.11	如何设置 VBA IDE 选项	23
2.6	一个简单的 VBA 程序示例	26
第 3 章	VBA 的基本语法知识	28
3.1	变量的数据类型	28
3.1.1	变量的基本数据类型	28
3.1.2	用户自定义数据类型	29
3.2	变量的作用域及声明	30
3.2.1	变量的作用域	30
3.2.2	变量的声明	31
3.2.3	变量命名的惯例	32
3.3	常量的声明及作用域	32
3.3.1	常量的声明	32
3.3.2	常量的作用域	33
3.4	语句的注释及长语句的分行	33
3.4.1	语句的注释	33
3.4.2	长语句的分行	33
3.5	运算符和表达式	34
3.5.1	算术运算符	34
3.5.2	关系运算符	34
3.5.3	逻辑运算符	35
3.6	基本语句	35
3.6.1	赋值语句	35
3.6.2	条件语句	37
3.6.3	循环语句	41
3.7	数组	46
3.8	子程序和函数	48
3.8.1	子程序	48
3.8.2	函数	52

第 4 章 ActiveX 的基础知识	54
4.1 AutoCAD VBA 工程术语与命令.....	54
4.1.1 AutoCAD VBA 工程术语.....	54
4.1.2 AutoCAD VBA 命令.....	54
4.2 AutoCAD 对象模型.....	55
4.2.1 Application 对象.....	58
4.2.2 Document 对象.....	58
4.2.3 Collection 对象.....	59
4.2.4 图形对象和非图形对象.....	59
4.2.5 Preference、Plot 和 Utility 对象.....	60
4.3 存取对象.....	62
4.3.1 在对象层次中引用对象.....	63
4.3.2 访问应用程序对象.....	64
4.4 使用集合对象.....	64
4.4.1 存取集合.....	65
4.4.2 添加新成员到集合对象.....	65
4.4.3 在集合对象中循环.....	65
4.4.4 删除集合对象中的成员.....	66
4.5 属性与方法.....	67
4.6 父对象.....	67
4.7 定位类型库.....	67
4.8 在模型空间中返回第一个图元.....	68
4.9 深刻理解模块和变体类型.....	68
4.9.1 模块的概念.....	68
4.9.2 对模块的概览.....	69
4.9.3 保存对模块所做的改变.....	71
4.10 使用变体数据类型.....	71
4.10.1 变体类型.....	72
4.10.2 在数组数据中使用变体.....	72
4.10.3 转换数组为变体.....	72
4.10.4 解释变体数组.....	73
4.11 使用其他程序语言.....	74
4.11.1 转换 VBA 代码到 VB 中.....	74
4.11.2 VBA 相对于 VB 对照代码示例.....	75
第 5 章 设置 AutoCAD 环境	77
5.1 图形文件操作.....	77
5.1.1 打开图形.....	77
5.1.2 创建新图形.....	78

5.1.3	保存图形	78
5.2	设定 AutoCAD 参数	79
5.3	控制应用程序窗口	80
5.3.1	调整应用程序窗口的位置	81
5.3.2	最小化和最大化 AutoCAD 窗口	82
5.3.3	查找 AutoCAD 窗口的当前状态	82
5.4	管理图形窗口	83
5.4.1	更改文档窗口的位置和大小	83
5.4.2	最小化和最大化文档窗口	83
5.4.3	查找文档窗口的当前状态	83
5.4.4	使用缩放	84
5.4.5	使用命名视图	90
5.4.6	使用平铺视口	90
5.4.7	更新文档窗口的几何图形	94
5.5	重置活动对象	95
5.6	设定和返回系统变量	96
5.7	精确制图	96
5.7.1	调整捕捉和基点	96
5.7.2	使用正交模式	97
5.7.3	绘制构造线	98
5.7.4	计算点和值	103
5.7.5	计算面积	104
5.8	提示用户输入	106
5.8.1	GetString 方法	106
5.8.2	GetPoint 方法	107
5.8.3	GetKeyword 方法	108
5.8.4	控制用户输入	109
5.9	访问 AutoCAD 命令行	111
5.10	工作于无打开文件状态	112
5.11	输入其他文件格式	112
5.12	输出到其他文件格式	113
5.13	定义用户坐标系统	114
第 6 章	创建 AutoCAD 图形	117
6.1	创建基本图形对象	117
6.1.1	确定容器对象	117
6.1.2	创建点对象	118
6.1.3	创建直线对象	120
6.1.4	创建曲线对象	124
6.2	创建实体填充区域	128

6.2.1	创建实体填充对象.....	129
6.2.2	面域对象.....	131
6.3	创建阴影.....	138
6.3.1	创建 Hatch 对象.....	138
6.3.2	定义阴影边界.....	139
第 7 章	编辑 AutoCAD 图形.....	143
7.1	命名对象.....	143
7.1.1	清理命名的对象.....	143
7.1.2	重命名对象.....	143
7.2	选择集对象.....	145
7.2.1	创建选择集.....	146
7.2.2	添加对象到选择集.....	147
7.2.3	过滤选择集.....	149
7.2.4	从选择集上移去对象.....	151
7.2.5	有关选择集的技巧举例.....	151
7.3	编辑对象.....	153
7.3.1	复制对象.....	153
7.3.2	偏移对象.....	156
7.3.3	镜像对象.....	158
7.3.4	阵列对象.....	161
7.3.5	移动对象.....	164
7.3.6	旋转对象.....	165
7.3.7	删除对象.....	167
7.3.8	比例缩放对象.....	168
7.3.9	转换对象.....	170
7.3.10	延伸和修剪对象.....	173
7.3.11	分解对象.....	174
7.4	高级对象编辑.....	176
7.4.1	编辑多段线.....	176
7.4.2	编辑样条曲线.....	179
7.4.3	编辑阴影.....	181
第 8 章	组织 AutoCAD 图形元素.....	187
8.1	使用图层.....	187
8.1.1	图层和线型分类.....	187
8.1.2	创建和命名图层.....	188
8.1.3	使图层成为当前图层.....	190
8.1.4	控制图层的可见性.....	190
8.1.5	打开和关闭图层.....	191

8.1.6	冻结和解冻图层	192
8.1.7	锁定和解锁图层	193
8.1.8	指定图层颜色	193
8.1.9	指定图层线型	194
8.1.10	删除图层	194
8.2	使用颜色	194
8.2.1	指定颜色	194
8.2.2	设定当前颜色	195
8.3	使用线型	196
8.3.1	使线型成为当前线型	199
8.3.2	重命名线型	199
8.3.3	删除线型	199
8.3.4	更改线型	199
8.3.5	指定线型比例	199
8.4	分配图层、颜色和线型给对象	203
8.4.1	更改对象的图层	203
8.4.2	更改对象的颜色	204
8.4.3	更改对象的线型	205
8.4.4	图层管理标准化	206
8.5	使用块	211
8.5.1	块功能	211
8.5.2	块中的图层、颜色和线型	212
8.5.3	块嵌套	212
8.5.4	块定义	212
8.5.5	插入块	214
8.5.6	炸开块参照	216
8.5.7	重新定义块	219
8.6	块属性	221
8.6.1	属性定义和属性参照	221
8.6.2	编辑属性定义	224
8.6.3	提取属性信息	227
第9章	在 AutoCAD 图形中添加文本	232
9.1	处理文字样式	232
9.1.1	创建和修改文字样式	232
9.1.2	指定字体	233
9.1.3	使用 TrueType 字体	236
9.1.4	使用 Unicode 和大字体	237
9.1.5	设定文字高度	237
9.1.6	设定倾斜角度	239

9.1.7	设定文字生成标记	240
9.2	使用单行文字	242
9.2.1	创建单行文字	242
9.2.2	格式化单行文字	243
9.2.3	对齐单行文字	244
9.2.4	编辑单行文字	246
9.3	使用多行文字	247
9.3.1	创建多行文字	247
9.3.2	格式化多行文字	248
9.3.3	编辑多行文字对象	251
9.4	使用 Unicode 字符、控制代码和特殊字符	251
9.5	替换字体	252
9.6	拼写检查	258
第 10 章	标注与公差	259
10.1	标注的基本知识	259
10.1.1	标注的元素	260
10.1.2	定义标注系统变量	261
10.1.3	设置标注文字样式	261
10.1.4	引线标注	261
10.1.5	关联标注	262
10.2	创建标注对象	262
10.2.1	创建线性标注	263
10.2.2	创建径向标注	264
10.2.3	创建角度标注	265
10.2.4	创建坐标标注	267
10.3	利用标注样式	269
10.3.1	复制标注样式	269
10.3.2	替代标注样式	272
10.4	编辑标注	275
10.5	在模型空间和图纸空间中标注	277
10.6	创建引线及注解	277
10.6.1	创建引线	278
10.6.2	添加注解到引线	279
10.6.3	关联引线	279
10.6.4	编辑关联引线	281
10.6.5	编辑引线	281
10.7	创建和编辑形位公差	282
10.7.1	创建形位公差	282
10.7.2	编辑公差	283

第 11 章 定义菜单和工具栏	284
11.1 理解 MenuBar 和 MenuGroups 集合.....	284
11.2 加载和创建菜单组.....	286
11.2.1 加载菜单组.....	286
11.2.2 创建新的菜单组.....	286
11.3 改变菜单条.....	287
11.3.1 在菜单条上插入菜单.....	287
11.3.2 从菜单条上删除菜单.....	288
11.3.3 在菜单条上重排菜单项.....	289
11.4 创建和编辑下拉菜单和快捷菜单.....	290
11.4.1 建立新菜单.....	290
11.4.2 在菜单中增加新的菜单项.....	291
11.4.3 在菜单中增加分隔符.....	294
11.4.4 对菜单项赋值一个快捷键.....	294
11.4.5 建立层叠式子菜单.....	295
11.4.6 返回新菜单.....	296
11.4.7 在菜单上删除菜单项.....	297
11.4.8 菜单项的属性.....	298
11.5 建立并编辑工具栏.....	300
11.5.1 建立新工具栏.....	301
11.5.2 在工具栏中增加新的工具栏按钮.....	301
11.5.3 对一个工具栏增加分隔符.....	303
11.5.4 定义工具栏按钮的图像.....	304
11.5.5 建立弹出工具栏.....	306
11.5.6 浮动和固定工具栏.....	307
11.5.7 从工具栏上删除一个工具按钮.....	310
11.5.8 工具栏项的属性.....	310
11.6 建立宏.....	311
11.6.1 宏字符所映射的 ASCII 等效值.....	312
11.6.2 宏的终止.....	313
11.6.3 暂停等待用户输入.....	313
11.6.4 取消一个命令.....	314
11.6.5 宏循环.....	314
11.7 对菜单项和工具栏项增加状态栏帮助.....	315
11.8 在右键菜单中增加条目.....	316
第 12 章 三维操作	318
12.1 三维坐标系统.....	318
12.2 用户坐标系统.....	320

12.3	坐标转换.....	323
12.4	创建三维对象.....	325
12.4.1	三维模型.....	325
12.4.2	创建线框模型.....	326
12.4.3	创建曲面模型.....	326
12.4.4	创建实体模型.....	328
12.5	三维对象编辑.....	340
12.5.1	三维旋转.....	340
12.5.2	三维阵列.....	342
12.5.3	三维镜像.....	344
12.5.4	三维干涉检验.....	345
第 13 章	使用事件.....	348
13.1	AutoCAD 中的事件.....	348
13.2	编写事件处理器的方法.....	348
13.3	处理应用程序级事件.....	349
13.3.1	应用程序级事件.....	349
13.3.2	激活应用程序级事件.....	350
13.4	处理文档级事件.....	353
13.4.1	文档级事件.....	354
13.4.2	在 VBA 以外的环境启用文档级事件.....	354
13.4.3	在 VBA 以外的环境下编制文档级事件程序.....	356
13.4.4	在 VBA 环境下编制文档级事件程序.....	356
13.5	激活对象级事件.....	358
附录	Auto LISP 函数与 ActiveX/VBA 比较.....	363

第 1 章 AutoCAD 开发工具简介

工程 CAD 软件在工程设计中已经成为不可缺少的强有力工具。许多设计单位及科研院所都推出了各自的 CAD 软件, 涉及各专业。一个好的 CAD 软件必须有一个好的图形支撑系统, Autodesk 公司开发的 AutoCAD 系列软件一直是 CAD 市场中的主流产品, 并以其功能强、用户界面友好及二次开发工具齐全等优点, 成为使用最广泛的 CAD 软件和二次开发平台。随着 AutoCAD 的日益普及, 在其上进行二次开发的工具也不断更新。从早期的 Auto LISP、ADS、DCL 到现在流行的 Visual Lisp、Object ARX(AutoCAD Run-time Extend)、VBA 等, 均可十分方便地对 AutoCAD 进行二次开发。

本章将首先对一些最具有影响力的开发工具做一些简要的介绍, 在此基础上分析 VBA 开发 AutoCAD 的优势所在。

1.1 Auto LISP 语言及 Visual LISP 语言

1.1.1 Auto LISP 语言的特点

AutoCAD 系统自问世以来, 在世界范围内都得到了广泛的应用。究其原因, 最主要的是 AutoCAD 结构的开放性, 它为用户提供了许多进行二次开发的工具, 其中包括内嵌编程语言 Auto LISP。

AutoCAD 本身图形功能是很强的, 由于 Auto LISP 语言的问世, 使 AutoCAD 功能有了一个质的飞跃, 用户可以根据自己的需要对 AutoCAD 进行多种功能的开发, 国外这种工作一度开展得很活跃, 称之为第三次开发。

从本质上讲, Auto LISP 语言是由 COMMON LISP 语言的一个子集再增加了一些与图形处理有关的特殊函数组成, 并内嵌在 AutoCAD 之中, 即是嵌入 AutoCAD 内部的通用 LISP 程序设计语言的子集。Auto LISP 语言灵活简洁、表达能力强, 通过定义函数实现各种功能。除可以完成各种科学计算外, 还能调用 AutoCAD 各项命令, 直接访问图形数据, 从而大大地扩充了图形处理功能。

但由于 Auto LISP 是一种解释性语言, 其保密性差、运行速度慢、自顶而下的程序设计思想、调试不便等缺点使得它不适合开发大型的计算密集型的系统。另外, 其表达式的表示法也不易让一般开发人员习惯和接受。

1.1.2 Visual LISP 语言的特点

为了克服 Auto LISP 存在的缺点, Autodesk 推出 Auto LISP 的新版本——Visual LISP。

Visual LISP 提供了 Auto LISP 的所有功能, 并与现有的 Auto LISP 程序兼容, 其面向对象的可视化集成开发环境(IDE)将使编译、调试用 LISP 编写的 AutoCAD 程序更加方便, 功能更强。其主要特色如下:

- (1) 集成开发环境(IDE)可生成 Object ARX 应用程序;
- (2) 源代码彩色拼法检查;
- (3) 内置调试器(Debugger);
- (4) 代码自动格式化和缩紧编写;
- (5) DCL 对话框预视;
- (6) 程序结构浏览;
- (7) 动态的符号生成;
- (8) 变量直接求值;
- (9) 源文件打开的数目和大小无限制;
- (10) 通过 ActiveX 自动化接口与其他软件集成, 如 Microsoft Office 等。

1.2 Object ARX 开发工具

为了适应面向对象程序设计(OOP)技术的发展形势, AutoCAD R14 提供了一个面向对象的开发环境 Object ARX(R13 版中称为 ARX, AutoCAD Run-time Extension)取代 ADS, 所用的语言为 C++, 只支持 Microsoft Visual C++编译器。这也就是说, ARX 是基于 Microsoft Visual C++的一个功能强大的二次开发工具, 用 Object ARX 环境开发的 AutoCAD 应用程序将比 ADS 程序具有更好的性能和更强的功能, 因为:

- (1) Object ARX 开发环境提供了一个超过 220 个类的 C++类系统;
- (2) Object ARX 应用程序的命令不再由 Auto LISP 调用而与 AutoCAD 内部命令处于同一级别;
- (3) 更重要的是, 它可构造用户自定义的对象 / 实体。处理这种对象 / 实体的方法完全与处理 AutoCAD 内部的对象 / 实体一样。

1.2.1 Object ARX 的特点

Object ARX 是以 C++为基础的面向对象开发环境及应用程序接口, 具有面向对象编程方式的数据可封装性、可继承性及多态性等特点。

用 Object ARX 开发的软件具有模块性好、独立性强、连接简单、使用方便、内部功能高效及代码可重复性强等优点, 并支持 MFC(Microsoft Foundation Class)基本类库, 能简捷并高效地实现许多复杂功能。

1.2.2 Object ARX 的调用机制

Object ARX 编程环境与 ADS、Auto LISP 的编程环境有很大的不同, 最大区别是 ARX 应用程序是一种可共享 AutoCAD 地址空间和建立对 AutoCAD 直接函数调用的动态链接库(DLL), 因而 ARX 应用程序与 AutoCAD 之间的通信更加直接, 运行效

率更高。

1.2.3 ARX 类库及函数库

ARX 包括 5 个类库和 1 个函数库,它们是:

- (1) AcRx 类库: 系统级的类和 C++宏指令集, 用于 DLL 应用程序的初始化、连接及实时类的注册和标注。
- (2) AcEd 类库: 定义和注册新的内部命令。
- (3) AcDb 类库: 直接访问 AutoCAD 图形数据库结构。
- (4) AcGi 类库: 提供图形界面工具, 用以绘制 AutoCAD 实体。
- (5) AcGe 类库: 提供普通的二维及三维几何操作, 也提供简单的几何对象, 如点、线、曲面。
- (6) ADS—RX 函数库: 提供可共享 AutoCAD 地址空间的 ADS 函数。

1.3 VBA 开发工具

目前, 有很多工具可以对 AutoCAD 进行二次开发, 如前文所介绍过的 Auto LISP、ARX、Visual LISP 等。但是这些工具中各自都有相应的优缺点, 简单来说, Auto LISP 语言是一种嵌入 AutoCAD 里的解释程序语言, 具有编程简单的特点, 但在开发大项目时显得力不从心。ADS 是一种 C 语言程序设计环境, 它与 AutoCAD 之间并不直接通信, 具有很大的灵活性, 但不适应当前可视化编程的需要。ARX 程序是 Windows 动态链接库程序, 执行 ARX 应用程序所需系统的开销最小, 速度最快, 但是开发过程相当复杂。

AutoCAD2000 内嵌的 VBA 语言是在标准 Visual Basic 基础上, 结合 AutoCAD 的特点发展起来的一种 Windows 平台上的高效开发工具, 它可以充分利用 Windows 操作系统提供的强大功能, 且简单易用, 非计算机专业人员也能利用 VBA 语言, 高效快速地对 AutoCAD 进行二次开发。

VBA 是通过 AutoCAD ActiveX Automation 接口来建立和 AutoCAD 对象间的联系。ActiveX 是建立在 COM 对象模型之上的一个标准通信协议, 它允许对象之间通过一定的接口相互通信。而 AutoCAD ActiveX 提供在 AutoCAD 外控制编程的机制, 通过使用 AutoCAD 对象, 实现控制 AutoCAD。

在讲解 VBA 之前, 有必要简要地介绍一下什么是 AutoCAD ActiveX Automation。

1.3.1 AutoCAD ActiveX 自动化技术简介

ActiveX 自动化技术(ActiveX Automation)是微软建立的标准, 基于 COM 的自动服务程序, 是对象链接与嵌入(OLE: Object Link and Embed)的进一步发展, 它允许用户在 Windows 应用程序中通过暴露的对象控制另一个 Windows 应用程序。AutoCAD ActiveX 技术提供了一种机制, 可使编程者通过编程从 AutoCAD 的内部或外部操作控制 AutoCAD。

ActiveX Automation 提供了开发脚本、宏以及用 Automation 编程环境(如 Visual Basic

6.0)开发第三方应用程序的途径。通过 Automation, AutoCAD 提供了可由 Automation 控制器(例如 VB 和 Excel)进行操作的可编程对象。因此,使跨应用程序的宏编程成为可能,而这种功能在 Auto LISP 中是不存在的。简单地说,使用 Automation 可以将许多应用程序的功能合并到单个应用程序中。

ActiveX 是由一系列的对象按一定的层次组成的一种对象结构,每一个对象代表 AutoCAD 中的一个明确的功能,比如画线、图块定义等。其绝大部分功能均以方法和属性的方式封装在 ActiveX 对象中,只要使用某种方式,使对象“暴露”,就可以使用面向对象编程的语言对其中的方法、属性进行引用,从而达到对 AutoCAD 实现编程的目的。

1.3.2 VBA 的功能

在 Office 97、AutoCAD R14.1 和 AutoCAD 2000 中使用的 VBA 是基于 VB5.0 版本、完全面向对象体系的一种编程语言。换言之,作为主程序,Office 97 和 AutoCAD 根据自身的特点,提供对象体系结构,然后均可用相同的 VBA 格式调用,因而可以说 VBA 是一个万能的开发工具。当然,主程序能开发的功能到底有多强大,与它所提供的对象体系有直接的关系,但 VBA 的强大开发能力却是不容质疑。

下面是 VBA 的主要功能:

- (1) VBA 可提供强大的窗体创建功能,为应用程序建立对话框及其他屏幕界面。
- (2) 可创建自己的工具条。
- (3) 可创建功能强大的模块级宏指令,宏名实质上就是模块的过程名。
- (4) 提供建立类模块的功能,这对开发大型工程非常有用,因此类可提供重用组件。另外,对于 AutoCAD ActiveX 技术中的应用程序级和对象级事件调用,将使用类模块。
- (5) 具备完善的数据访问与管理能力,通过 DAO(数据访问对象),可以对 Access 数据库或其他外部数据库(像 dBase, FoxPro 等)实现访问与管理。此功能比直接使用 AutoCAD 的数据库管理系统要方便,且功能强大。
- (6) 可以使用 SQL 语句检索数据,与 RDO(远程数据对象)结合起来,能够建立客户机/服务器级的数据通信。
- (7) 能够使用 Win32 API 提供的功能,建立应用程序与操作系统之间的通信。

1.3.3 VBA 作为 AutoCAD 开发工具的优势

Visual Basic for Application(VBA)是由 Visual Basic 发展起来的,用来自动执行任务的一个编程环境。它提供了一些用来创建图形用户界面的可拖拉工具和用来与 AutoCAD 对象交互的编程语言。使用 VBA 开发 AutoCAD,用户可以随意地定制 AutoCAD 应用程序。本书中,你将会发现自动执行一些重复的任务是非常简单的,会大大地节省时间,这样就可以极大地发挥你的艺术才能和工程技巧,制作出更加复杂的图形。

一旦你开始编写 VBA 代码,很快就会发现 AutoCAD 和 VBA 的对象库中获取对象其实相当容易,而且在深入理解了 AutoCAD 的特征后立即就可以访问这些对象的功能。当你体会到 VBA 宏和应用程序所提供的好处时,就应专门花些时间去定制一些更重要的任务。在掌握这些以前,你可能已经对整个的可重用代码和应用程序有所了解。

提示:宏是一段代码语句,通常不太长,但在多次执行重复的任务时非常有用(在后