

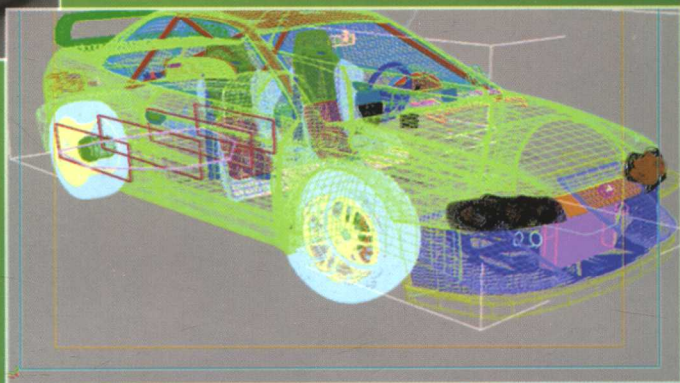
实例通系列

AutoCAD VBA

二次开发教程



张帆 主编
郑立楷 卢择临 王成煌 编著



清华大学出版社

AutoCAD VBA 二次开发教程

张 帆 主 编
郑立楷 卢择临 王成煌 编著

清华大学出版社

北 京

内 容 简 介

本书以 AutoCAD 2005 中文版为开发平台,通过 73 个精彩的编程实例,向读者展示了 AutoCAD VBA 开发的高级技术。

全书共 16 章,将所有的实例归入基础相关、控制 AutoCAD 环境、创建和编辑二维对象、对象特性、文字、标注和公差、块和属性、处理 AutoCAD 中的事件、三维空间的操作、打印、高级绘图组织、使用组件扩展程序功能、和 Excel 交换数据、数据库应用程序、发布 VBA 应用程序等大类,从不同的角度分析 VBA 编程的技术和技巧。本书提供了多个精彩实例的源码,涉及 VBA 的基础操作、VBA 的应用领域、VBA 的特殊功能,以及 CAD 软件开发的一些基础知识。

本书层次清晰、实例丰富,适用于建筑、机械、电子等多个领域的工程设计人员和软件开发工程师,同时也可供对计算机辅助设计有兴趣的大学生、研究生以及各工程单位相关技术人员使用。对于 VBA 的读者来说,这是一本具有较强指导性和实用性的图书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

AutoCAD VBA 二次开发教程/张帆主编;郑立楷,卢择临,王成煌编著. —北京:清华大学出版社,2006.1
ISBN 7-302-11982-1

I. A… II. ①张… ②郑… ③卢… ④王… 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(2005)第 118850 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客 户 服 务: 010-62776969

责任编辑: 魏江江

封面设计: 杨 兮

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 31 字数: 773 千字

版 次: 2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

书 号: ISBN 7-302-11982-1/TP·7760

印 数: 1~3500

定 价: 49.00 元(含光盘)

前 言

读者对象

本书适用于使用 VBA 和 ActiveX 技术对 AutoCAD 进行二次开发的工程技术人员。

如果对 AutoCAD VBA 开发毫无概念,建议首先阅读清华大学出版社出版的《AutoCAD VBA 开发精彩实例教程》。

本书内容

本书的内容分为 16 章,另有一个附录,从各个方面向读者展示 AutoCAD VBA 开发的知识。

第 1 章深入分析了 AutoCAD VBA 开发中两个重要的知识点——宏和集合对象。宏是 AutoCAD VBA 开发中一个最基础但又非常重要的概念,而集合对象在开发过程中使用非常频繁,有必要在学习的开始阶段打下坚实的基础。

第 2 章包括文档操作、缩放视图、命名视图,以及和用户交互的操作。“文档操作”部分可以看作是集合对象使用的继续深入探讨,缩放视图使用 ActiveX 模型中提供的缩放方法模拟了 AutoCAD 自身的几种缩放功能,“和用户交互”一节提出了在 GetPoint 方法中使用关键字的解决方案。

第 3 章对选择集的使用进行了全面的剖析,为 SendCommand 提供了更广泛的应用,分析了直线、圆、圆弧和样条曲线向多段线的转化,提供了一些常用的多段线操作,提出制图法解决问题的新思路,编写了根据起点、终点和弧长创建圆弧的函数,最后又给出绘制放样展开图的方法。

第 4 章包含了两个有用的程序。按图层缩放提供了显示整个图层上所有实体的功能;使用 ObjectDBX 技术导入其他图形中的信息,相当于提供了一种后期“绑定”样板文件的手段。

第 5 章介绍了与文字使用有关的程序设计。创建文字样式能够获得当前计算机中可用的字体列表;文字按插入点自动对齐提供了对自定义数据类型的数组进行排序的一个思路;提取多行文字中的文字内容对 AutoCAD 中多行文字的格式代码进行了分析。使用 ObjectDBX 实现的批量文字替换,与上一个版本比较起来,修正了一些 Bug,并且是在不打开文件的情况下完成操作的,速度有了明显的提高。

第 6 章介绍了标注和公差的使用。新的圆弧长度标注程序涉及到多行文字格式代码的使用;锁定标注文字的程序对于创建局部放大图形时非常有用,技术方面涉及到匿名块、尺寸标注对象的组成、GetSubEntity 方法的使用等;列举所有的公差符号程序仍然涉及到多行文字的格式代码,从程序中还可以发现访问公差内容的方法。

第 7 章介绍了块和属性相关的程序设计,简单总结了简单块和属性块的使用,提供了将已经存在的实体添加到块定义中的方法,探讨了匿名块的使用。块和属性的结合使用非常灵活,自动标题栏和块属性生成表格两个程序演示了这种应用。实现“斜切”功能应该属于

块应用的一个副产品,利用插入块参照时可以指定不同 X、Y、Z 坐标的特点,实现特殊的图形变换。

第 8 章介绍了 AutoCAD 事件处理的方法。利用几个典型的实例介绍了应用程序层事件(禁用命令)、文档层事件(双击事件和快捷菜单事件)和对象层事件(临时“反应器”)的处理方法,然后用一个序号球的实例模拟了永久“反应器”。

第 9 章处理在三维空间中的程序设计。在三维操作中,提供了在 UCS 中创建基本图元的方法,让编程人员也能享受 UCS 带来的便利。三点法创建 UCS、在任意平面上创建图案填充、创建正多面体和创建偏心圆台都以数学知识为基础,解决几个在三维建模中应用的典型问题。

第 10 章讨论打印和布局的问题。本章仅提供了批量打印的程序。

第 11 章介绍了扩展数据、扩展字典、对象命名字典、SummaryInfo 对象和绘图顺序等高级绘图组织技术。其中,扩展数据和扩展字典是标识实体的一种手段,SummaryInfo 对象提供了访问图形特性的接口,绘图顺序则用于调整图形对象的显示层次。

第 12 章探讨了在 VBA 开发应用程序过程中可能会用到的几种技术和技巧。调试应用程序和应用程序的错误处理对于程序开发来说必不可少,这里用作者自身的经历来介绍其中的技巧;DwgThumbnail、Accont、Slide、MSScript 是几个很有特色的组件,能够为 VBA 程序增加一些特殊的功能;创建和使用定时器一节直接用 Windows API 函数在类模块中自己创建定时器。

第 13 章将注意力集中在 Excel、Word 和 AutoCAD 的数据交换,以及在 VBA 使用 Windows API 实现特殊效果。VBA 和 Excel 的数据交换似乎一直是一个热点问题,本章提供了 Excel 表格向 AutoCAD 的转换、使用 Excel 数据绘图、转换 AutoCAD 数据到 Word 和 Excel 中几个实例。在使用 Windows API 方面,提供了去掉对话框的关闭按钮和添加快捷菜单的实例。

第 14 章介绍在 VBA 中创建数据库应用程序的方法。由于 ADO 技术是比较理想的一种技术,因此本章着重介绍 ADO 和 SQL 语句结合开发数据库应用程序。此外,数据记录和 AutoCAD 实体的对应一节探讨了实现属性数据和地理数据的对应,这对于在 AutoCAD 上实现 GIS 的读者非常有用。

第 15 章提供发布 VBA 应用程序的各种技术,包括菜单宏的使用、在工具栏中使用自定义图标、基本图库的实现(动态修改菜单文件)、在命令行或菜单中执行 VBA 程序、实现 VBA 程序的先选择再操作、更改 AutoCAD 应用程序的图标和标题栏、使用 InstallShield 完全发布 VBA 应用程序。

第 16 章介绍了使用 VB 开发 AutoCAD 的一些知识(将 VBA 代码转换到 VB 中,VB 窗体和 AutoCAD 的焦点切换,使用 ActiveX DLL 封装带有窗体的 VB 程序)、搜索某个文件夹中所有的文件和 VBA IDE 的编程,利用这些知识,完全有可能开发出功能更为强大的应用程序。

附录

本书的附录包含了 AutoCAD 2005 帮助文档中《ActiveX 和 VBA 参考》(英文版)部分的内容,并且已经翻译为中文,供读者在学习过程中参考。

配套光盘

本书的配套光盘给出了除附录内容之外的所有源程序代码,以及在编程中可能用到的资源文件。

使用配套光盘,就可以直接将代码复制到读者本人的计算机中学习,而不必键入源程序代码。此外,读者可以自己写代码,将程序运行结果与光盘中程序执行结果相对照。

本书适用版本

本书所有程序都能够运行在 AutoCAD 2005 中文版和英文版平台上。

AutoCAD 2005 VBA 中新增了一些对象和方法,但是在不使用这些新增功能的情况下,仍然可以在低版本的 AutoCAD 中运行这些程序。本书作者不保证所有的程序都能在 AutoCAD 2004 及其以下版本中使用(很明显的一个例子,创建表格对象的程序不能在 AutoCAD 2004 及其以下的版本中使用,因为那些版本中根本没有表格对象)。

关于明经通道

“明经通道”是国内著名的 CAD 技术网站(<http://www.mjtd.com>),它提供了大量的学习资料和 CAD 程序。网站论坛现有注册用户 30 000 以上,每天都有很多高手在论坛中交流各种技术。

明经通道是免费的技术站点,欢迎各位读者来做客。请记住明经通道的服务宗旨:你的需要,我们创造。

本书的技术支持

可以通过以下三种方式获得关于书中问题的技术支持:

- 在明经通道论坛(<http://www.mjtd.com/bbs>)的“VBA 和 ActiveX 技术”版块发帖。
- 发送邮件到作者的邮箱(统一发送到 zf9568@263.net)。
- 在作者的个人网站“CAD 大观园”论坛(<http://www.cadhelp.net/bbs>)的相关版块发帖。

关于本书作者

本书是集体劳动的结晶!下面是每一位作者所撰写的内容:

- 郑立楷(mccad)(明经通道站长,VBA 和 Visual LISP 开发方面的专家):3.2、3.3、3.5、3.9、5.2、5.3、第 6 章、7.7、11.3、15.4、15.5。
- 王成煌(efan2000)(明经通道“VBA 和 ActiveX 技术”版块版主):12.3~12.6、13.1、13.3、13.4、15.6。
- 卢择临(ahlzl)(明经通道贵宾,安徽省淮北工业学校教师):3.1、3.10、3.11、7.6、第 9 章。
- 李倚伏(leeyefu)(明经通道“数据库/EXCEL/ObjectDBX 编程技术”版块版主):14.1~14.4。

- 李昭鸿(lzh741206)(明经通道“VBA 和 ActiveX 技术”板块版主):8.4、16.1、配套光盘中的“块内按点生成填充”程序。
- 张帆(zfbj)(明经通道“编程申请”板块版主,“CAD 大观园”站长):第 1 章、第 2 章、3.4、3.6、3.7、3.8、第 4 章、5.1、5.4、7.1~7.5、8.1~8.3、第 10 章、11.1、11.2、11.4、12.1、12.2、12.7、13.2、14.5、15.1~15.3、15.7、16.2、16.3、附录,并负责全书统稿。

本书和《AutoCAD VBA 开发精彩实例教程》

同《AutoCAD VBA 开发精彩实例教程》一书相比较,这本书并不是内容的重复,而是一个提高,前者专注于基础和中级应用,本书则更多地介绍高级技巧,很多内容在帮助系统和市场上同类的图书中是找不到答案的。

致谢

本书能够顺利出版,绝对是集体劳动的成果,明经通道几位在 VBA 方面有造诣的版主和贵宾(有几位因为工作的忙碌没能参与这次的写作)都拿出自己擅长的程序,并亲自撰写文档,确保文档能够最好地反映出程序设计的思想。感谢几位合作者能够在工作忙碌的情况下抽时间高质量地完成写作任务,并且为我的统稿工作提供技术支持,没有你们的努力,这本书不可能有这么高的含金量。

在图书撰写过程中,明经通道的多位网友为本书的编写提出了宝贵的意见并提供了素材资料,例如 wyj7485 网友提供了创建定时器控件的原始素材,更多的网友不再一一列举,在此一并表示感谢,你们的支持是我们前进最重要的动力!

最后要感谢的是本书的读者,感谢您选购这本图书,这是对我们辛苦工作的最好的认可,希望这本书能对你有所帮助,祝读书愉快!

编 者

目 录

第 1 章 AutoCAD VBA 开发基础	1
1.1 宏	1
1.2 集合对象的使用	4
第 2 章 控制 AutoCAD 环境	8
2.1 文档操作	8
2.2 缩放视图	12
2.3 命名视图	14
2.4 和用户交互	17
第 3 章 创建和编辑二维图形对象	25
3.1 根据起点、终点和弧长创建圆弧	25
3.2 利用制图法通过三点创建圆	28
3.3 获得点到多段线的最短距离	31
3.4 选择集	36
3.5 使用 SendCommand 方法	48
3.6 多段线	53
3.7 直线、圆和圆弧转化为多段线	60
3.8 样条曲线转化为多段线	67
3.9 在 AutoCAD 2004 以前版本中计算优化多段线长度	70
3.10 圆管的放样展开	74
3.11 天圆地方的放样展开	78
第 4 章 对象特性	84
4.1 按图层进行缩放	84
4.2 使用 ObjectDBX 技术导入其他图形中的信息	87
第 5 章 文字	98
5.1 创建文字样式	98
5.2 文字按插入点自动对齐	105
5.3 提取多行文字中的文字内容	111
5.4 使用 ObjectDBX 实现批量文字替换	116
第 6 章 标注和公差	124
6.1 标注圆弧长度	124

6.2	锁定标注文字不随对象缩放而动态变化	127
6.3	列举所有的形位公差符号	132
第 7 章	块和属性	135
7.1	简单块的定义和插入	135
7.2	将已经存在的实体添加到块定义中	139
7.3	使用匿名块	143
7.4	属性块的使用	146
7.5	自动标题栏	148
7.6	AutoCAD 中“斜切”功能的实现	159
7.7	块属性生成表格	164
第 8 章	使用 AutoCAD 中的事件	169
8.1	处理应用程序层事件	170
8.2	处理文档层事件	173
8.3	处理对象层事件	179
8.4	序号球(永久反应器)	182
第 9 章	在三维空间工作	191
9.1	在 UCS 中创建基本图元	191
9.2	创建 UCS	199
9.3	在任意的平面上创建图案填充	203
9.4	创建正多面体	208
9.5	创建偏心圆台、圆柱和圆锥	217
第 10 章	布局和打印	230
10.1	批量打印	230
10.2	布局	235
第 11 章	高级绘图与组织技术	237
11.1	扩展数据	237
11.2	扩展字典和命名对象字典	243
11.3	使用 SummaryInfo 对象保存图形信息	249
11.4	绘图顺序	256
第 12 章	VBA 与应用程序	262
12.1	调试 VBA 应用程序	262
12.2	VBA 应用程序的错误处理	270
12.3	显示缩略图——DwgThumbnail 组件	275

12.4	显示停靠窗口——Account 组件	279
12.5	显示幻灯片 Slide 组件	285
12.6	解析表达式——MSScript 组件	289
12.7	创建和使用定时器	294
第 13 章	与其他应用程序和 Windows API 交互	300
13.1	转换 Excel 表格到 AutoCAD 中	300
13.2	使用 Excel 数据绘图	313
13.3	转换 AutoCAD 数据到 Word 和 Excel 中	317
13.4	去掉关闭按钮(X)和添加快捷菜单	326
第 14 章	创建数据库应用程序	332
14.1	创建数据库的 ADO 连接	332
14.2	创建数据表	336
14.3	使用记录集从数据表中读取数据	341
14.4	使用 ADODC 控件	346
14.5	数据记录和 AutoCAD 实体的对应	356
第 15 章	VBA 程序的发布	365
15.1	菜单宏	365
15.2	在工具栏中使用自定义图标	368
15.3	基本图库的实现	375
15.4	在命令行或菜单中执行 VBA 程序	383
15.5	实现 VBA 程序的先选择后操作和透明命令	392
15.6	更改 AutoCAD 应用程序的图标和标题栏	398
15.7	发布 VBA 应用程序	403
第 16 章	VBA 开发相关技术	430
16.1	在 VB 中创建应用程序	430
16.2	搜索某个文件夹中所有的文件	441
16.3	VBA IDE 的编程	448
附录 A	AutoCAD ActiveX 和 VBA 参考	457

第 1 章 AutoCAD VBA 开发基础

本章将对 AutoCAD VBA 开发中两个重要的基础知识进行分析,确保用户在后面的程序中能自由地使用 VBA 管理器来创建和管理宏,并且深入理解 AutoCAD 中集合对象的含义,掌握访问集合对象及其元素的方法。

1.1 宏

1.1.1 实例说明

开始学习 AutoCAD VBA 之前,必须熟悉宏及其应用。本节介绍宏的创建、编辑、运行,以及宏的管理等知识。

1.1.2 编程思路

宏是在选择工具栏某个选项时执行特定操作的一系列命令,可以只是完成任务的按键记录,也可以是复杂的命令,例如,AutoLISP、DIESEL 或 ActiveX 编程代码等的组合。宏的定义方式与 VB 中过程(Sub)的定义相一致,一般用来实现某些特定功能,也可作为 VBA 应用程序的启动过程。只有用 Public 关键字声明的过程才能被作为宏。

在 AutoCAD 的“宏”对话框的宏列表中,仅会显示 ThisDrawing 或者标准模块中的宏。ThisDrawing 和标准模块中的代码有何区别? ThisDrawing 模块中无法创建全局变量,但是可以放置 AutoCAD 事件处理的代码;标准模块与之相反。

严格来说,本节的内容并不在于编程,而是在于介绍与 AutoCAD VBA 开发关系密切的一些概念和开发工具的使用。

1.1.3 步骤

(1) 在 AutoCAD 2005 中,选择“工具”|“宏”|“VBA 管理器”菜单项,系统会弹出“VBA 管理器”对话框,如图 1.1 所示。目前工程列表中未包含任何项目,表示 AutoCAD 2005 没有加载任何的 VBA 工程。

(2) 新建 VBA 工程。单击“新建”按钮,系统会自动添加一个名称为 ACADProject 的全局工程。单击“Visual Basic 编辑器”按钮,进入 VBA 集成开发环境,在“工程资源管理器”窗口中,显示了当前加载的 VBA 工程,如图 1.2 所示。

(3) 修改工程名称。在“工程资源管理器”窗口中选择未命名的工程,然后在“属性”窗口的“名称”一栏中输入“宏的基本概念”,并按 Enter 键,完成工程名称的修改。此时,在 AutoCAD 2005 中打开“VBA 管理器”对话框,就能在工程列表中看到具体的工程名称,如图 1.3 所示。

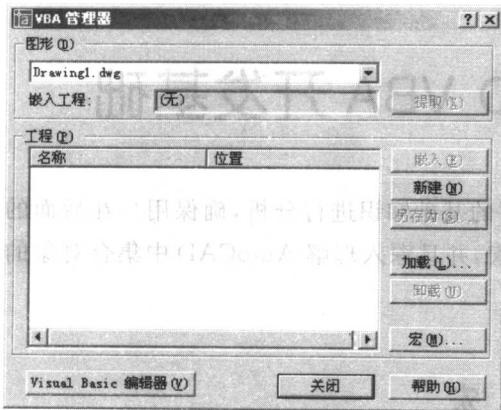


图 1.1 “VBA 管理器”对话框

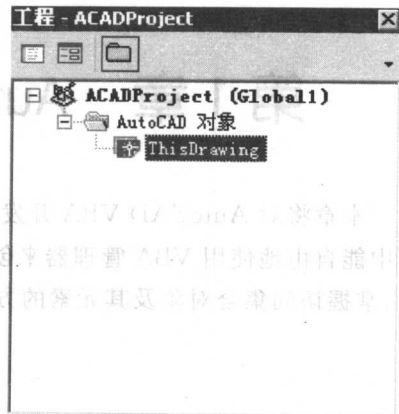


图 1.2 显示当前加载的 VBA 工程

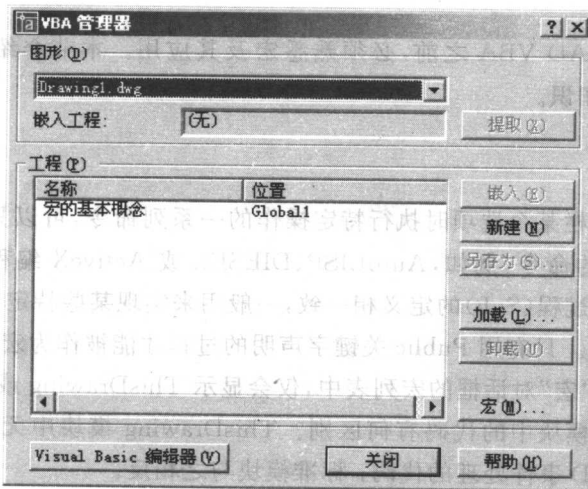


图 1.3 显示修改后的工程名称

提示：在创建新工程之后，应该为其指定新的名称，这样在加载多个工程之后，仍可在“VBA 管理器”对话框和“工程资源管理器”窗口中区分不同的工程。

(4) 保存和另存工程。如果当前的工程尚未保存过，可以在 VBA 集成开发环境中选择“文件”|“保存 Global”菜单项，或者单击“标准”工具栏中的“保存”按钮，在系统弹出的“另存为”对话框中指定工程的位置，完成工程的保存。

此外，在“VBA 管理器”对话框中，从工程列表中选择所要保存的工程，单击“另存为”按钮，在弹出的“另存为”对话框中，同样可以完成工程的保存。要实现工程的换名保存(另存)，就必须使用这种方法。

(5) 打开已有的 VBA 工程。在“VBA 管理器”对话框中，单击“加载”按钮，在系统弹出的“打开 VBA 工程”对话框中，选择所要加载的 VBA 工程，单击“打开”按钮，即可将其加载到 AutoCAD 中。

(6) 关闭 VBA 工程。在“VBA 管理器”对话框中，选择所要关闭的工程，单击“卸载”按

钮,就能关闭指定的 VBA 工程。

(7) 输入示例代码。进入 VBA 集成开发环境,双击工程资源管理器的 ThisDrawing 图标,在弹出的代码窗口中输入如下所示的代码。

```
Public Sub MacroDemo( )
    MsgBox "Hello,Welcome to AutoCAD VBA!"
End Sub
```

(8) 运行宏。在 VBA 集成开发环境中,将光标定位到要运行的宏内部(也就是位于 Sub 和 End Sub 语句之间的部分),按 F5 键就可以执行宏。另外,也可选择“工具”|“宏”菜单项,在系统弹出的“宏”对话框中运行宏。

1.1.4 实例效果

在 VBA 集成开发环境中,选择“工具”|“宏”菜单项,系统弹出如图 1.4 所示的“宏”对话框。从宏列表中选择本节创建的 MacroDemo 宏,单击“运行”按钮,执行该宏。

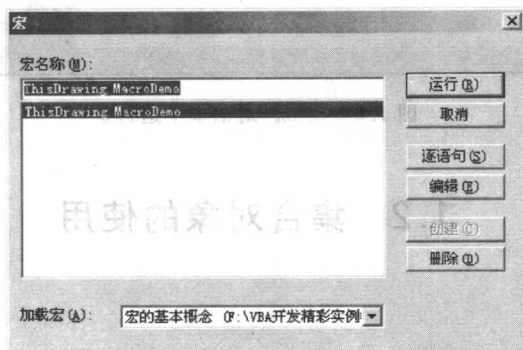


图 1.4 选择所要执行的宏

系统会弹出一个信息提示框,如图 1.5 所示。



图 1.5 运行宏的结果

1.1.5 小结

宏是 VBA 中重要的概念,在 AutoCAD VBA 开发中,一般被用作某个功能的入口点,类似于 VB 中启动对象(Sub Main)的概念。

除了在 VBA 集成开发环境中创建和执行宏之外,还可以在 AutoCAD 2005 中选择“工具”|“宏”|“宏”菜单项,打开如图 1.6 所示的“宏”对话框。可以看出,在“宏”对话框

中同样可以创建和运行宏,但是这多用于应用阶段,也就是说程序的使用者多用这种方法来运行宏。

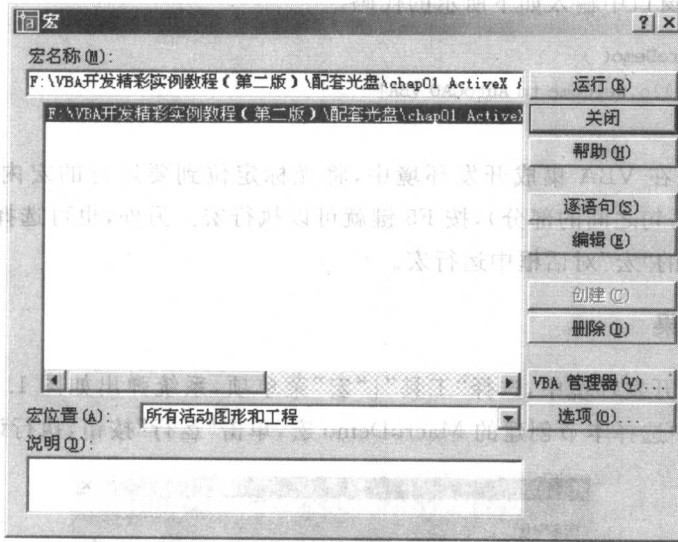


图 1.6 在“宏”对话框中运行宏

1.2 集合对象的使用

1.2.1 实例说明

集合对象是一种预定义的对象,包含相似对象的所有实例(或是相似对象所有实例的上级对象)。在 ActiveX 接口中,提供了大量的集合对象,重要的集合对象包括 Documents(当前 AutoCAD 任务中打开的所有文档)、ModelSpace(模型空间中的所有图形对象)、Blocks(图形中所有的块)、Groups(图形中的所有编组)、Layers(图形中的所有图层)、Linetypes(图形中的所有线型)、SelectionSets(图形中所有的选择集)、TextStyles(图形中所有的文字样式)、UCSs(图形中所有的用户坐标系)、Views(图形中所有的视图)和 Viewports(图形中所有的视口)。

本节介绍访问集合、集合对象,添加和删除集合对象,以及遍历集合对象的知识。

1.2.2 编程思路

大部分的集合对象都是通过 Document 对象访问的,对于每一个集合对象,Document 对象都有一个相应的特性。例如,可以定义一个变量,并将其设置为当前图形的 Groups 集合:

```
Dim groupCollection As AcadGroups
Set groupCollection = ThisDrawing.Groups
```

还有一些集合对象是通过 Application 对象来访问的,例如 Documents、MenuBar 和

MenuGroups。对于这些对象,在 Application 对象中包含了相应的特性。

要向集合中添加新的元素,可以使用 Add 方法,例如:

```
Dim newGroup As AcadGroup
Set newGroup = ThisDrawing.Groups.Add("MyGroup")
```

要访问集合中特定的某个元素,可以使用集合的 Item 方法,既可以通过集合元素的名称来访问,也可以通过元素的索引值来访问特定的元素,例如:

```
Dim objGroup As AcadGroup
Set objGroup = ThisDrawing.Groups.Item("MyGroup")
```

要从集合中删除某个元素,可以使用元素本身的 Delete 方法,例如:

```
Dim objGroup As AcadGroup
Set objGroup = ThisDrawing.Groups.Item("MyGroup")
objGroup.Delete
```

遍历集合对象是指依次访问集合中的每一个元素,一般可以使用 VB 中的 For...Each 语句来完成。需要注意的是,不要在使用 For...Each 机制遍历集合的同时对任意对象使用图元编辑方法(例如复制、阵列、镜像等)。应在遍历完成后再尝试编辑集合中的对象,或者创建临时数组,并将其设置为等于该集合,然后遍历复制的数组并执行所需的编辑操作。

1.2.3 步骤

下面以图层的操作为例,介绍一下集合对象的使用。首先创建一个新工程,本节涉及到的所有代码均在 ThisDrawing 模块中添加。

(1) 新建一个图层,也就是向图层集合添加一个元素。使用 Add 方法,向当前图形的 Layers 集合添加一个新元素,其相关代码为:

```
Public Sub NewLayer( )
    '新建一个图层
    Dim objLayer As AcadLayer
    Dim layerCollection As AcadLayers

    Set layerCollection = ThisDrawing.Layers
    Set objLayer = layerCollection.Add("轴线")
End Sub
```

(2) 修改“轴线”层的名称。要修改某个特定图层的特性,必须首先声明一个变量,并且将该变量指向这个图层,其相关代码为:

```
Public Sub ChangeLayerName( )
    Dim objlayer As AcadLayer

    Set objlayer = ThisDrawing.Layers.Item("轴线")
    objlayer.Name = "中心线"
End Sub
```

(3) 统计图形中名称的前两个字符为“图层”的图层的数量。要实现这个功能,必须遍

历图形中所有的图层,将图层名称的前两个字符与“图层”字符串相比较,两者相等时将统计值加 1。其相关代码为:

```
Public Sub ChangeAllName()  
    Dim objlayer As AcadLayer  
    Dim i As Integer  
  
    For Each objlayer In ThisDrawing.Layers  
        If StrComp(Left$(objlayer.Name, 2), "图层") = 0 Then  
            i = i + 1  
        End If  
    Next objlayer  
  
    Debug.Print "有" & i & "个图层需要修改名称"  
End Sub
```

StrComp 函数能够比较两个字符串的内容,如果两字符串相同,返回值为 0。Left\$ 函数返回一个字符串,其中包含字符串中从左边算起指定数量的字符。Debug.Print 语句会在立即窗口中打印指定的文本。在 VBA 集成开发环境中,选择“视图”|“立即窗口”菜单项,能够显示立即窗口。

(4) 删除某个图层。如果图形中存在“轴线”层,就将其删除,其相关代码为:

```
Public Sub DeleteLayer()  
    On Error Resume Next  
  
    Dim objlayer As AcadLayer  
    Set objlayer = ThisDrawing.Layers.Item("轴线")  
  
    If Not IsEmpty(objlayer) Then  
        objlayer.Delete  
    End If  
End Sub
```

Not 运算符用来对表达式进行逻辑否定运算,IsEmpty 函数判断指定的变量是否已经初始化。之所以要使用 On Error Resume Next 语句进行错误处理(如果遇到错误,程序会转到紧接着发生错误的语句后面的语句,继续执行),是因为可能在当前图形中并不存在“轴线”图层,那么访问 Layers 集合中名为“轴线”的元素就会发生错误。

1.2.4 实例效果

在 VBA 集成开发环境中,将光标定位在要执行的宏内部,按 F5 键运行该宏,然后选择“格式”|“图层”菜单项,在图层特性管理器中观察宏的运行结果。

1.2.5 小结

在 VBA 编程中,经常用的操作是遍历图形数据库,获得图形数据库中满足特定条件的实体。例如,要将图形中所有的直线颜色变为红色,就可以使用下面的一段程序。


```
Public Sub ChangeLineColor( )  
    Dim ent As AcadEntity  
    Dim color As New AcadAcCmColor  
  
    color.ColorIndex = acRed  
  
    For Each ent In ThisDrawing.ModelSpace  
        If TypeOf ent Is AcadLine Then  
            ent.TrueColor = color  
        End If  
    Next ent  
End Sub
```