杨星 赵晓冬 编著

AutoCAD2004二次开发（VB版）及在海工模型试验数据处理中的应用

海洋出版社

AutoCAD2004 二次开发(VB 版)及在海工模型试验数据处理中的应用

杨　星　赵晓冬　编著

海洋出版社
2005 年 · 北京

图书在版编目(CIP)数据

AutoCAD 2004 二次开发(VB 版)及在海工模型试验数据处理中的应用/杨星,赵晓冬编著. —北京:海洋出版社,2005.12
ISBN 7-5027-6521-2

Ⅰ.A… Ⅱ.①杨… ②赵… Ⅲ.海洋工程—数学模型—数据处理—应用软件,AutoCAD 2004 Ⅳ.P75-39

中国版本图书馆 CIP 数据核字(2005)第 157726 号

责任编辑:阎 安
责任印制:刘志恒

海洋出版社 出版发行
http://www.oceanpress.com.cn
(100081 北京市海淀区大慧寺路 8 号)
北京华正印刷厂印刷 各地新华书店经销
2005 年 12 月第 1 版 2005 年 12 月北京第 1 次印刷
开本:787mm×1092mm 1/16 印张:16.875
字数:420 千字 印数:1~2000 册
定价:28.00
海洋版图书印、装错误可随时退换

前　言

在工程设计、科研工作中经常需要对 AutoCAD 工程图像文件进行数据处理和图形绘制，如从 AutoCAD 地形图上提取高程点数据，相同观测点的数据更换，在 AutoCAD 中绘制矢量场图等，采用人工操作往往花费大量的时间。本书主要讲述了 AutoCAD2004 的二次开发（VB 版）及在海工模型试验数据处理中的应用。书中提供的原代码是在 VB 6.0 和 AutoCAD2004 环境下编译的。

本书的读者对象是具备了 AutoCAD 和 VB 基础知识的工科院校的学生和工程设计、科研单位的研究人员。您如果是入门者，或正在考虑入门，应该准备一本 VB 和 AutoCAD 的基础教程，对于书中出现的一些基本概念，可以参考这些教程。当然，我们也会在书中尽量解释您遇到的这些基本概念。那么是否需要对 VB 和 AutoCAD 有了深入了解才能开始我们的探索？答案当然是否定的。我们安排的内容由易到难，适合各种水平的开发者，是一循序渐进的引导过程。即使您是一个初学者，或仅仅掌握较少关于 VB 和 AutoCAD 的知识，也可以开始这次神奇的探索，探索精彩的二次开发的世界。阅读完本书，也许您会惊讶的发现，原来二次开发是如此简便，原来我的工作可以更有趣，更简单。更重要的是，通过阅读本书，您将掌握一门通用的开发技术——ActiveX Automation，ActiveX 是微软制定的一种实现程序间通信、调用的规范，它提供了一种控制 AutoCAD 的技术手段，通过这种技术手段，对于 AutoCAD 中每一个对象，如 point、line、circle、都可以用 VB、VB、C + +、Jave、Excel VBA、AutoCAD VBA、Word VBA 等支持 ActiveX 的语言来控制。每一个对象都有其相应的属性、方法，可以读取或改变其属性，可以用方法来控制对象的动作，包括改变对象的颜色，读取对象的坐标等。在您阅读完本书后，发挥您的想象，您也将具备对其他应用软件的开发能力。我们期待着您在不久的将来带我们进入到另一个二次开发的精彩世界。

本书由杨星和赵晓冬主编。其中第 2、3、4 章和附录由河海大学交通学院杨星执笔，第 6、7、8 章及第 9 章的第 9.3 ~ 9.5 节由南京水利科学研究院赵晓冬执笔，第 1 章由河海大学交通学院王娅娜编写，第 5 章由南京市水建水利建筑工程监理有限公司蔡开玺编写，第 9 章 9.1 节由深圳市水利规划设计院金家莉编写，第 9 章 9.2 节由佛山市水利水电建筑设计有限公司陈永昌编写。

本书编写过程中，得到了南京水利科学研究院王向明高工的大力支持。

本书主要作者杨星在此特别感谢父母和姐姐多年来对自己学业和生活上的支持和关心。

由于编者水平有限，采用的实例仅局限于海工模型实验的数据处理，书中难免有不当之处，谨请各位读者批评指正。

编著者

2005 年 9 月

目　次

第 1 章　VB 开发 AutoCAD 的基础知识

1.1　欢迎进入 VB 开发 AutoCAD 的精彩世界

欢迎进入 VB 开发 AutoCAD 的精彩世界。应该强调一下，控制 AutoCAD 的方法有多种，本书将要介绍的只是其中一种。首先开始简单介绍一下如何控制 AutoCAD。本书主要讲述了通过 ActiveX Automation 的技术规范实现 AutoCAD 控制的方法。ActiveX 是微软制定的程序间通信、调用的规范，它提供了一种控制 AutoCAD 的技术手段，通过这种技术手段，对于 AutoCAD 中每一个对象，如点（Point）、直线（Line）、圆（Circle）、图层（Layer）等，都可以用 VB、Delphi、C++、Java、Excel VBA、AutoCAD VBA、Word VBA 等支持 ActiveX 的语言来控制。每一个对象都有其相应的属性、方法，例如，直线（line）是 AutoCAD 中的一个对象，它的属性包括线的颜色（Color）、起点坐标（StartPoint）、结束点坐标（EndPoint）、直线所在图层（Layer）等，方法则有移动（Move）、复制（Copy）、镜像（Mirror）、直线与另外某个对象的交点（IntersectWith）等。我们即可以读取或改变对象属性，又可以用方法来控制对象的动作。这样，就可以借助 ActiveX Automation 技术通过编程在 AutoCAD 里做很多有趣的事情，包括在 AutoCAD 里画两条直线对象，并求出它们的交点，在所有的图形对象中找到半径等于 2 cm 或者大于 2 cm 的圆，并确定它们的数量和位置，查询图形中文本对象的内容或者直接修改 AutoCAD 图层上图形对象的颜色，更改图形对象到另一个图层等。注意这里提到的“对象”这个词，我们以后会一直用到它。可以把它理解为所能看到的一切 AutoCAD 上的东西，包括图形对象（点，线，面，实体等），也包括图层（Layer），线型（Linetype），模型空间（Model Space）等等一切非图形对象，当然也包括 AutoCAD 应用程序（Application）本身，所有“对象”都有属性、方法。非图形对象和图形对象是紧密联系的。在下面的章节里，我们会详细讨论 AutoCAD 的对象，对象之间的关系以及如何利用对象的属性和方法。下面试一下我们的第一个小程序，看看我们是怎样把这句话“欢迎进入 VB 开发 AutoCAD 的精彩世界”（或许说它是个文字对象更专业）写到 AutoCAD 里的。

```
{声明公共变量，obj_Acad:Application 对象，即 AutoCAD 软件本身；obj_Doc:Document 对象，当前的图形文件；obj_ModelSpace:Model Space 对象，当前图形文件的模型空间}
Dim obj_Acad As Object, obj_Doc As Object, obj_ModelSpace As Object
Dim boo As Boolean

Public Sub AutoCADConnect()          '连接 AutoCAD 的子程序
On Error Resume Next
'若 AutoCAD 已启动，则直接得到 Application 对象
Set obj_Acad = GetObject(, "autocad.application")
If Err Then
  Err.Clear
```

```
    On Error Resume Next
    '若 AutoCAD 未启动,则运行 AutoCAD 程序
    Set obj_Acad = CreateObject("autocad.application")
    If Err Then
    Err.Clear
    MsgBox " 不能运行 AutoCAD, 请检查是否安装! ",vbOKOnly, " 警告! "
    Exit Sub
    End If
End If
Set obj_Acad.Visible = True    '设置 AutoCAD 为可见(或者在后台运行,不可见)
Set obj_Doc = obj_Acad.ActiveDocument    '设置 obj_Doc 为当前图形文件( Document 对象)
'设置 obj_ModelSpace 为当前图形文件的模型空间( Model Space 对象)
Set obj_ModelSpace = obj_Doc.ModelSpace
MsgBox " 运行结束 !", vbOKOnly, " 工程 1!"
boo = True    ' 判断已经生成 AutoCAD 程序对象
End Sub

Private Sub Command1_Click()
Call AutoCADConnect            '调用连接 AutoCAD 的子程序
End Sub

Private Sub Command2_Click()
Dim TextString As String
Dim InsertionPoint(0 To 2) As Double    '定义我们要把文本插入到的点位置坐标和文本对象
Dim obj_text As Object                  ' 定义文本的高度
Dim Heigth As Double
  If Not boo Then
    MsgBox " 请先生成 autocad 程序对象 ", vbOKOnly, "autocad 程序对象 ?"
    Exit Sub
  End If
TextString = Text1.Text    'Text1.Text = 欢迎进入 VB 开发 AutoCAD 的精彩世界
InsertionPoint(0) = 5#: InsertionPoint(1) = 5#: InsertionPoint(2) = 0#
Heigth = 0.5
'AddText 是模型对象( Model Space 对象)提供的一种在模型空间( Model Space )上绘制文字的方法
Set obj_text = obj_ModelSpace.AddText(TextString, InsertionPoint, Heigth )
obj_Acad.zoomextents
MsgBox " 运行结束 !", vbOKOnly, " 工程 1!"
End Sub
```

程序界面如图 1-1 所示。

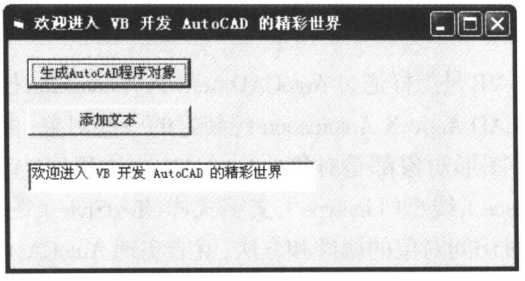

图 1-1　程序界面

运行程序前，建议您先打开 AutoCAD 程序，然后执行 AutoCAD 菜单命令【格式】/【文字样式】，显示【文字样式】对话框，从【字体名】下拉表中选择“新宋体”。然后运行我们的程序，先单击按钮【生成 AutoCAD 程序对象】，再按钮单击【添加文本】，最后程序运行结果如图 1-2。

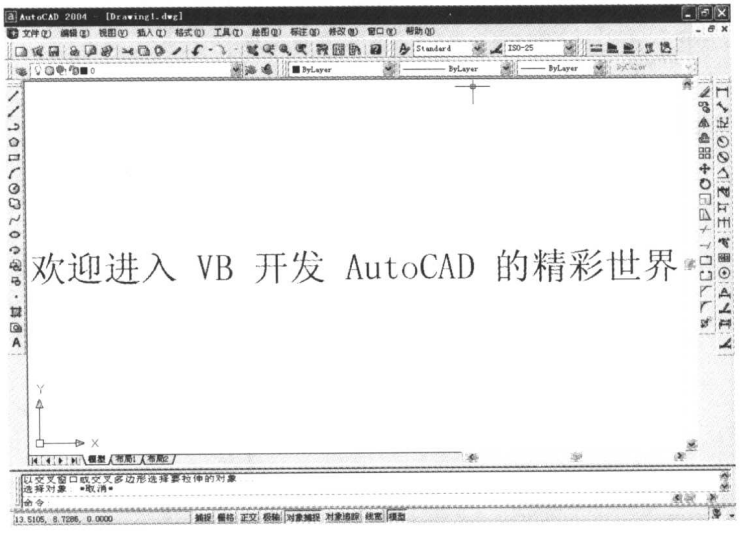

图 1-2　运行结果

当第一次做出这个程序，就感觉到了 VB 开发 AutoCAD 的方便。VB 友好的界面，AutoCAD 所支持的 ActiveX Automation 技术，使我们开发程序就是这么简单，只用了很少的程序代码就实现了所要的功能。以上的代码作为框架，还可以继续添加绘制直线，绘制圆等其他功能，仅仅只是添加几行代码的问题。在这里，没有在代码里设置任何的有关绘图的参数（每项参数都是一种对象），包括图纸的大小，图纸用的单位，字体的颜色，字体的样式，字体所在的图层等等。对于初学者，建议您在 AutoCAD 里先设置好这些参数。这样，您不论加入的是直线、文字，还是其他的图形对象，在没有在程序代码里设置相关绘图的参数的情况下，AutoCAD 会默认采用当前设置的绘图的参数，例如，您设置了当前图层是“0 层”，那么所有的对象就绘制在“0 层”，所有对象的颜色就会采用当前图层的颜色，线型也是当前图层采用的线型，同样的，如果您在 AutoCAD 里设置好字体的样式为“新宋体”，那么用程序添加的文字其字体默认就是“新宋体”，除非您已经在程序里采用了其他字体。

1.2 VB 开发 AutoCAD 的技术手段

在这一节里，主要讲解 VB 是怎样通过 AutoCAD ActiveX Automation 技术来控制 AutoCAD 的。VB 所能操作的实际上是 AutoCAD ActiveX Automation 包装好的一些对象，直线（Line）、圆（Circle）、文字（Text）和标注（DimStyle）等图形对象都是对象。AutoCAD 应用程序（Application）、图纸空间（Paper Space）、模型空间（Model Space）、线型（Linetype）、文字式样（TextStyle）、图层（Layer）、块（Block）等非图形对象也都是对象。通过访问对象的属性和方法，就能实现 AutoCAD 的全部功能。在本节要对“对象”这个词的意义做详细论述，还要引入“集合”的概念。“集合”是“对象”的集合，其本身也是“对象”，“集合”包含 Add（添加对象）、Delete（删除对象）、Item（取对象）、Count（对象数量）等属性和方法，包含的对象可以是不同种类的对象。图纸空间（Paper Space）、模型空间（Model Space）、块（Block）都是集合。图 1-3 的模型空间包含了三个对象，它们是 Circle（圆对象）、Line（直线对象）、Text（文字对象），绘制图形实际上就是通过“Add 方法”往集合中添加对象。

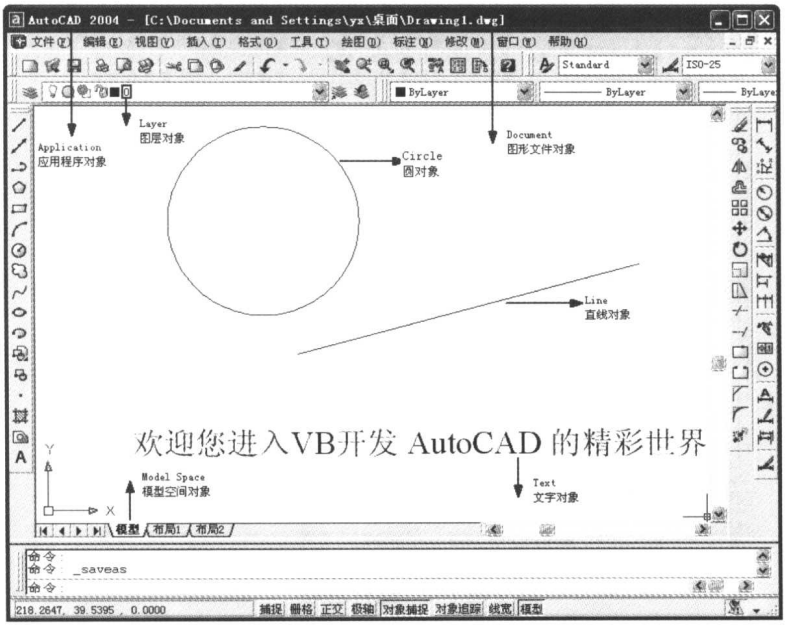

图 1-3 对象示意图

1.2.1 AutoCAD ActiveX Automation 技术概述

AutoCAD ActiveX Automation 使用户能够从 AutoCAD 的内部或外部以编程方式来操作 AutoCAD。它是把 AutoCAD 全部的功能和函数打包成一个名为“Application”的对象，可以把 Application 对象看作是 AutoCAD 本身，这是一个处在所有 AutoCAD 对象最顶层的对象，可以称它为应用程序对象。其他对象均为 Application 对象的子对象，子对象也可能再包含子对象。通过这些对象，我们开发的应用程序可以实现 AutoCAD 的全部功能。在上一节讲到的程序里首先写了“Set obj_Acad = GetObject（，"AutoCAD.Application"）”这样一句代码，获得了一个 Application 对象；然后，通过代码“Set obj_Doc = obj_Acad.ActiveDocument”获得 Application 对象的子对象图形文

件对象——Document 对象；接下去，通过代码 "Set obj_ModelSpace = obj_Doc.ModelSpace" 获得的是 Document 对象的子对象模型空间对象——Model Space 对象；最后，我们通过代码 "Set obj_text = obj_ModelSpace.AddText（TextString, InsertionPoint, Heigth）" 获得了 Model Space 对象的子对象文字对象——Text 对象，实现了文字对象——"欢迎进入 VB 开发 AutoCAD 的精彩世界"的输出。可见，理解 AutoCAD 对象的层次结构是我们编程的基础。抛开 ActiveX Automation 的内部技术，作为程序开发人员，能看到的就是这样一些对象，它们"暴露"在 AutoCAD 的外面，被我们的程序访问。注意，这种层次关系只是一种包含关系，与 VB 等语言里的父子继承关系截然不同，AutoCAD 中对象的属性和方法不会向它所包含的子对象传递。总而言之，我们操作 AutoCAD，实际上是在操作 AutoCAD 的对象，沿着从父对象到子对象的链接，用户可以访问接口中的所有对象。对象本身包含自己的方法和属性，通过方法可以实现对象的一些操作，通过属性可以获取对象状态信息或者改变对象当前的状态。正如我们前面提到过的，直线（Line）是 AutoCAD 中的一个对象，它的属性可以是颜色（Color）、起始点坐标（StartPoint）、结束点坐标（EndPoint）、所在图层（Layer）等性质，方法则有移动（Move）、复制（Copy）、镜像（Mirror）、交点（IntersectWith）等操作。如果我们想要移动一条直线，一种是改变直线的起始和结束点坐标，即通过改变直线的 "属性" 使对象动作，另一种是直接利用直线对象的方法 "移动（Move）"，即通过方法实现对象的动作。

ActiveX Automation 分为两层：自动服务器（Automation Server）和自动控制端（Automation Controller）。如果一个应用程序支持 ActiveX Automation 技术，那么其他应用程序就可以通过其对象（Object）对其自动操作。就 VB 开发 AutoCAD 而言，我们开发的程序为自动控制端，AutoCAD 是服务器，应用程序正是通过对 AutoCAD 的各级对象进行操作而控制 AutoCAD 工作的。AutoCAD 从 R14 版开始引入了 Automation 技术，使得更多的编程环境可以访问 AutoCAD 图形。而在 ActiveX Automation 出现以前，开发人员只能使用 AutoLISP 或 C++ 接口。AutoLisp 不如其他可视化编程语言如 VB、Delphi 等功能强大，在定制复杂用户界面，功能较复杂的项目时力不从心，而 C 语言编程极为烦琐，不适应当前可视化编程的需要。所以，Automation 技术在 AutoCAD 上的实现，大大简化了程序设计的工作，使 AutoCAD 的二次开发技术变的更易于推广。同时，在与其他 Windows 应用程序（例如 Microsoft Excel 和 Word）共享数据变得更加容易。图 1-4 给出通过 ActiveX Automation 技术开发 AutoCAD 的流程图。

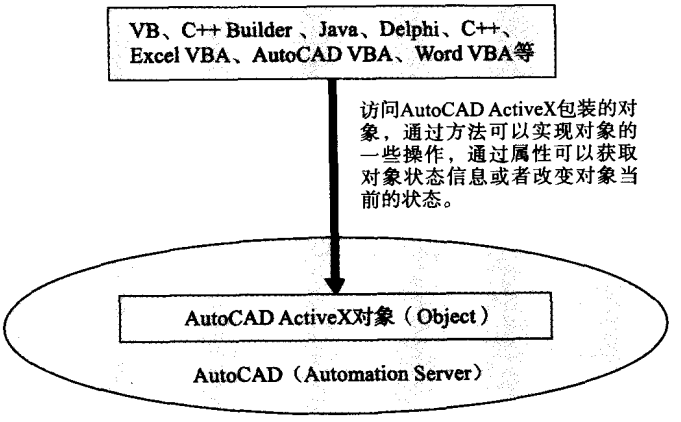

图 1-4　开发 AutoCAD 的流程图

1.2.2 AutoCAD 对象模型

AutoCAD 对象模型是一种层状结构模型，Application 对象处在最顶层，其他各层 AutoCAD 的对象都是它的子对象。任何上一层对象都是下面各层所有对象的父对象。父对象与子对象不是通常意义的继承关系，也就是说，父对象不会把它的属性和方法传递给它的子对象。这种层次关系实际上是包含关系。AutoCAD ActiveX 接口中有许多不同类型的对象。直线、圆弧、文字和标注等图形对象都是对象。AutoCAD 应用程序、图纸空间、模型空间、线型、标注样式、图层、块等非图形对象也都是对象。沿着从父对象到子对象的链接，用户就可以访问接口中的所有对象。仔细研究图 1-5 列出的 AutoCAD 对象模型结构图，了解它们的层次关系，对以后的编程会很有好处。

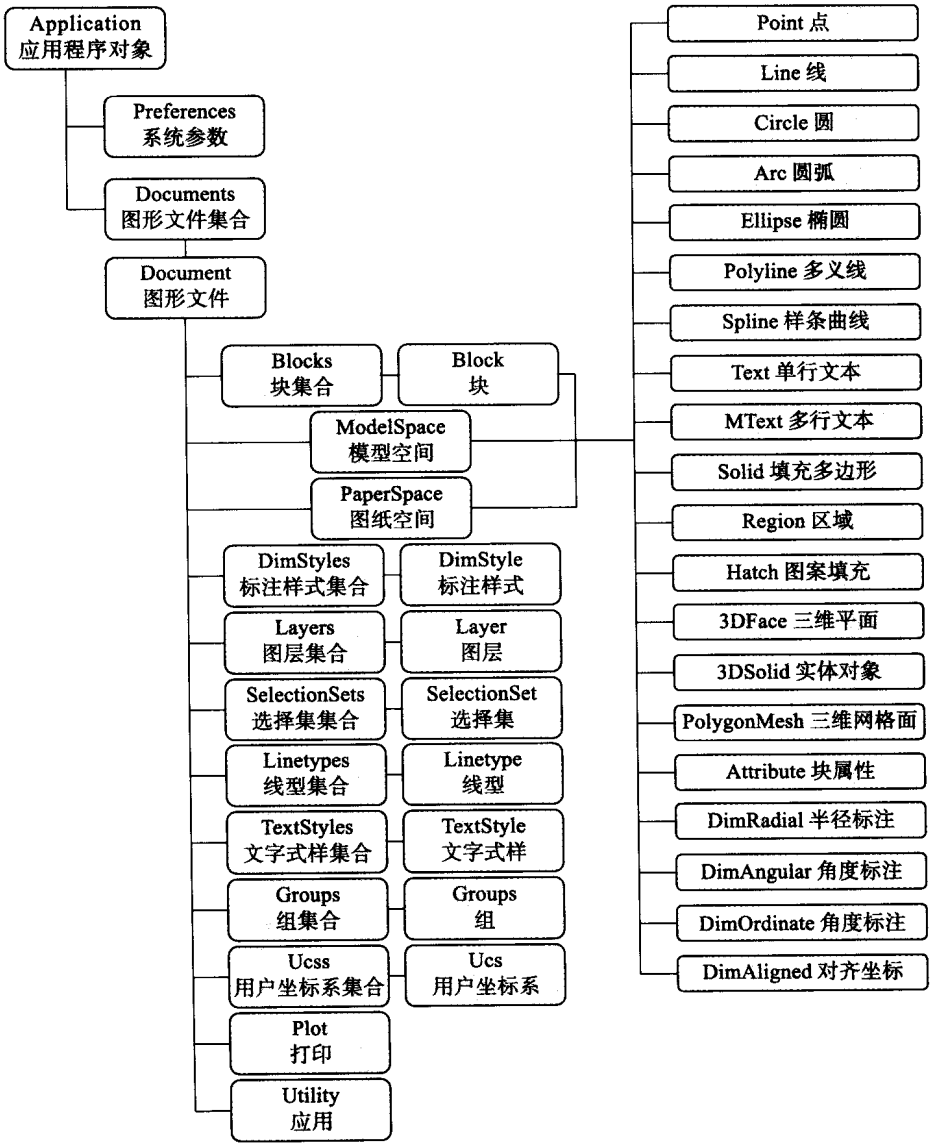

图 1-5 AutoCAD 对象模型

配合图 1-5,说明下面四个对象“Application 对象、Document 对象、Model Space 对象、Text 对象”的关系。Application 对象处在最高层,是所有对象的根对象或者父对象,Application 层之下是 Preferences 和 Document 对象。Document 对象下包含了 Model Space 对象,Model Space 对象包含了 Text 对象。现在再回顾上一节编的第一个程序,仔细想想是如何获得那四个对象的,在程序中获得对象的顺序是符合 AutoCAD 对象模型层次关系的,换句话说,沿着从父对象到子对象的链接,用户可以访问接口中的所有对象。

● Application,应用程序对象:可以看作是 AutoCAD 本身,处于对象模型的最高层,它包含了 AutoCAD ActiveX Automation 对象所提供的全部功能。

● Preferences,系统参数设置对象:处于 Application 层之下,包含了当前 AutoCAD 的参数设置。与 AutoCAD 中的【工具】/【选项】对话框作用相同。

● Documents ,图形文件对象集合:处于 Application 层之下,是当前 AutoCAD 中已经打开的图形文件的“集合”,用 Item 方法可以获取集合中的任何图形文件,这一点类似 VB 的集合的概念,每一个集合都提供了向其中添加对象的方法。大多数集合使用的是 Add 方法,但添加图形对象通常使用名为 Add< 图形名 > 的方法。例如,要添加直线,使用 AddLine 方法,集合还有一些其他的常用方法和属性,Count 属性用于获取集合中的对象个数(从零开始),其他的集合对象类似。

● Document,图形文件对象:提供了大多数 AutoCad 的文件功能,可以通过它实现对文件的更新(New)、打开(Open)、输出(Export)、输入(Import)等操作,当前 AutoCAD 中打开的图形文件可以通过 Documents 图形文件对象集合的 Item 方法获得,或者 Application 对象的 ActiveDocument 属性获得。与 AutoCAD 中的【文件】下拉菜单提供的功能相同。

● Blocks 、Block、Model Space、PaperSpace,分别是块集合对象、块对象、模型空间对象、图纸空间对象,其中 Block、Model Space、PaperSpace 是图形对象的集合,Blocks 是 Block 的集合。通常,我们主要通 Model Space 绘制二维图形和建立三维图形。

● DimStyles、DimStyle,标注样式对象集合及标注样式对象:用于尺寸标注与编辑。与 AutoCAD 中的【格式】/【标注样式】对话框作用相同。

● Layers、Layer, 图层集合对象及图层对象:用于与图形文件图层相关的操作,如添加新图层,更改图层名称,设置图层的颜色、线形,查询某个图层上的图形对象等,是编程中经常要用到的对象之一。与 AutoCAD 中的【格式】/【图层】对话框作用相同。

● SelectionSets、SelectionSet,选择集集合及选择集:用于图形编辑对象的选择,是对用户图纸上选择的图形对象的管理。有了选择集,就可以把需要修改或者需要查看属性的图形对象集合在一起,然后用 Item 的方法取得每一个对象,逐一实行操作。它也是我们编程中经常要用到的对象之一。

● Linetypes、Linetype,线型集合及线型:用于线型管理和设置。与 AutoCAD 中的【格式】/【线型】对话框作用相同。

● TextStyles、TextStyle,文字样式集合及文字样式:用于文字样式管理和设置。与 AutoCAD 中的【格式】/【文字样式】对话框作用相同。

● Groups、Group,组集合及组:负责当前活动的图纸文件中所有的组的建立和管理。

● Ucss、Ucs,用户坐标系集合及用户坐标系:用于建立和管理用户坐标系。

● Plot,打印对象,负责控制图形文件的输出和打印设置。

● Utility,应用对象,提供了用户与 AutoCAD 交互的途径,可以让用户在 AutoCAD 命令

行输入距离、字符、角度、坐标等参量。它包含了很多实用的函数，是计算和绘图的有力的辅——助工具。

对于以上罗列的对象，在以后的章节里还会详细讨论它们的属性和方法。只有熟练掌握好这些对象的方法、属性及它们之间的包含关系，才能在编程中灵活的运用这些对象。每个对象所提供的功能与 AutoCAD 相应的对话框或者绘图命令是一致的，所以，在学习的过程中，可以对照 AutoCAD 相应的对话框或者绘图命令来理解这些对象的功能，这是一个十分有效的学习方法。

1.2.3 VB 开发 AutoCAD 的基本框架的建立

下面，我们要着手建立一个 VB 开发 AutoCAD 的基本框架，以后所有的示例，都是建立在这个基本框架上，只是拓展了功能而已。

```
Dim obj_Acad As Object, obj_Doc As Object, obj_ModelSpace As Object
Dim boo As Boolean

Public Sub AutoCADConnec()              '连接 AutoCAD 的子程序
On Error Resume Next
Set obj_Acad = GetObject(, "autocad.application" )
If Err Then
  Err.Clear
  On Error Resume Next
  Set obj_Acad = CreateObject("autocad.application")
  If Err Then
    Err.Clear
    MsgBox " 不能运行 AutoCAD, 请检查是否安装！ ", vbOKCancel, " 警告！ "
    Exit Sub
    End If
End If
obj_Acad.Visible = True
Set obj_Doc = obj_Acad.ActiveDocument
Set obj_ModelSpace = obj_Doc.ModelSpace
MsgBox " 运行结束！ ", vbOKOnly, " 工程 1！ "
boo = True
End Sub

Private Sub Command1_Click()
Call AutoCADConnect        '调用连接 AutoCAD 的子程序
End Sub
```

● 声明公共变量，“obj_Acad, obj_Doc,obj_ModelSpace”，obj_Acad：Application 对象，即 AutoCAD 软件本身；obj_Doc：Document 对象，当前的图形文件；obj_ModelSpace ：Model Space 对象，当前图形文件的模型空间。