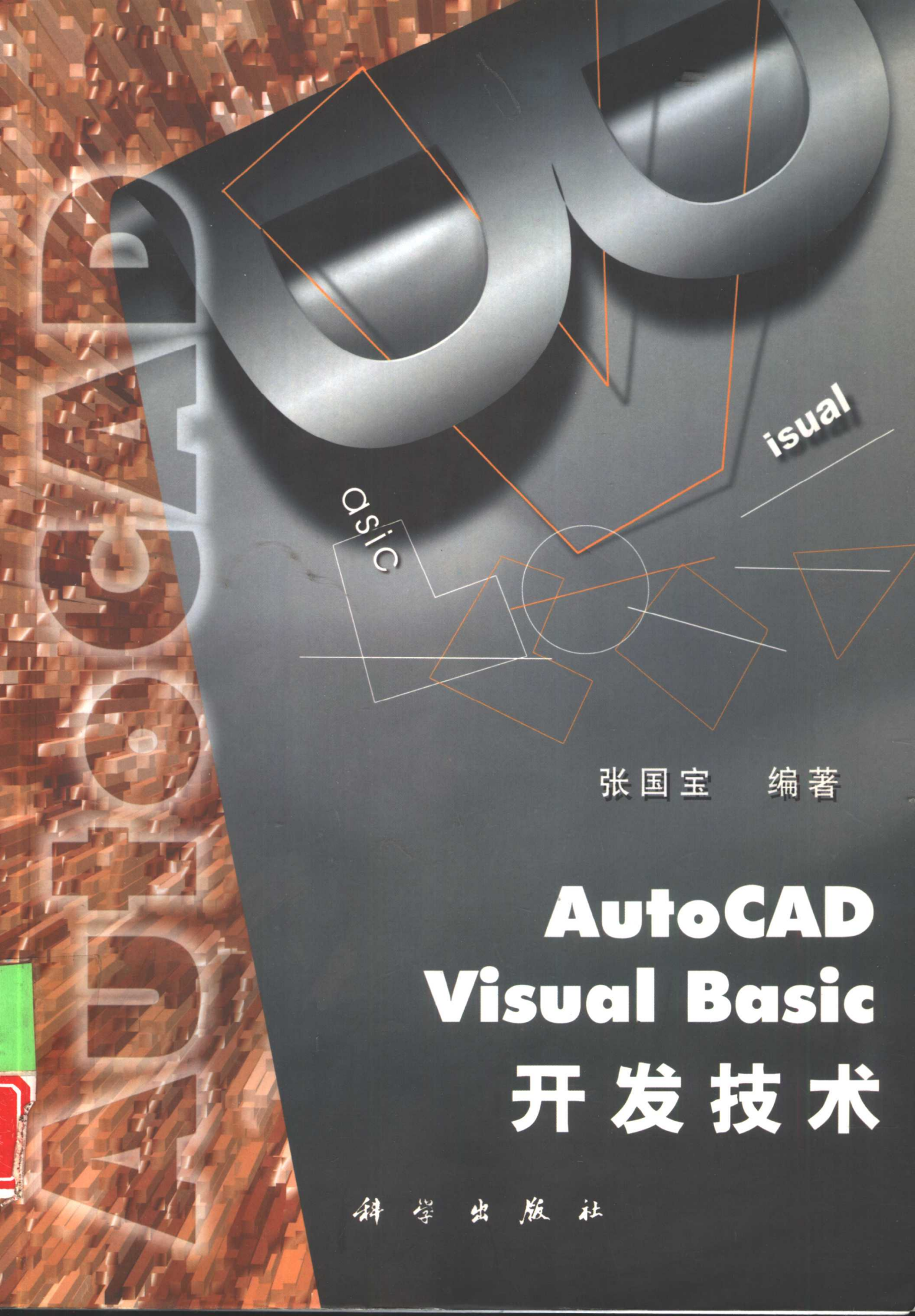

张国宝 编著

AutoCAD Visual Basic 开发技术

科学出版社

AutoCAD Visual Basic 开发技术

张国宝　编著

科学出版社

2000

内 容 简 介

提到 AutoCAD 的开发技术，大家最熟悉的就是 LISP，基于 C/C++ 的 ADS，ARX 等工具，其实用 Visual Basic 同样可以进行 AutoCAD 的二次开发，且功能相当强大。

本书详细介绍了用 Visual Basic 开发 AutoCAD 这一全新、便捷的开发方式。在结构安排上，既考虑到 AutoCAD 作为工程制图工具的一面，又考虑到 VB 作为编程语言和环境的一面，以对象模型为贯穿全书的主线，以完成一个具体的 AutoCAD 操作为目的。

无论是初学者，还是资深的软件开发工程师，都会在本书中有所收益。读者并不需要掌握高深的 VB 知识，即使从没接触过 VB，也能按照本书提供的程序样例轻松操作。

虽然本书以 Visual Basic 5.0 和 AutoCAD R14 为操作对象，但对于 VB 6.0 和 AutoCAD 2000 同样适用。

图书在版编目（CIP）数据

AutoCAD Visual Basic 开发技术/张国宝编著. -北京：科学出版社，2000

ISBN 7-03-007981-7

Ⅰ. A… Ⅱ. 张… Ⅲ. ①自动绘图-软件包，AutoCAD ②Basic 语言-软件开发 Ⅳ. TP391.72

中国版本图书馆 CIP 数据核字（1999）第 67156 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

北京双青印刷厂 印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 1 月第 一 版 开本：787×1092 1/16

2000 年 1 月第一次印刷 印张：24 1/4

印数：1—5 100 字数：560 000

定价：32.00 元

(如有印装质量问题，我社负责调换(环伟))

前　言

AutoCAD 软件包是目前最为流行的工程图形处理软件，它强大的生命力不仅在于其完善的绘图功能、良好的用户界面、易学易用的操作特点，而且还在于它开放的结构体系，为用户提供了多种二次开发的工具。

目前，开发者普遍使用的工具有两种：AutoLISP 语言和 C/C++语言。

AutoLISP 由于推出较早，具有广泛的使用基础，是目前应用较多的开发工具。AutoLISP 的优势在于语言结构简单，易学易用，开发周期短，不依赖于操作平台，有大量的不加密的样板程序可供参考。但作为解释语言，其运行速度和保密性方面有一定缺陷，一般专门的、商品化的应用程序采用较少。R14 版本后，AutoDesk 公司又推出过可编译的 VisualLISP 语言，比以前的 LISP 语言有较大的改进，并可以将 LISP 源代码编译成 .ARX 文件。AutoCAD 2000 中，VLISP 功能更为强大，编译后的 .ARX 文件短小精悍，运行效率高。

从 R11 版本起，AutoDesk 公司又提供了基于 C/C++语言的开发系统 ADS（AutoCAD Development System）。在 C/C++上开发、编译后，程序在运行速度和保密性方面得到加强。从 R13 版本开始又新增了另一种基于 C/C++语言的 AutoCAD 运行时扩展系统 ARX（AutoCAD Runtime Extension System）。与 ADS 不同的是，ARX 不需要通过 AutoLISP 解释程序而直接与 AutoCAD 核心进行通讯，因此在运行速度和性能上均大大优于前者。R14 版本推出的 ObjectARX 编程工具采用全新的面向对象技术，同样是基于 C/C++的开发工具，它能实现以前难以实现的功能，运行效率更高，是 AutoDesk 公司广为推崇的开发工具。

基于 C/C++开发的应用程序在运行速度和性能上极具优势，但由于 C/C++语言本身难于掌握，加之开发周期长、可移植性差等原因，一般用户较难把握，普及面较窄。

随着微软平台一统天下局面的出现，微软提出了各应用程序间通讯的统一标准，较原先 OLE（对象链接与嵌入）更为全面的新概念——ActiveX 自动化技术，它倡导在 Windows 平台下，各应用程序间相互调用、相互协作。AutoDesk 对此积极响应，于是在 AutoCAD R14 的开发后期，诞生了又一开发方法——利用 ActiveX Automation 技术。

1．ActiveX Automation 技术

ActiveX Automation 是微软公司的一个技术标准，以前被称为 OLE（对象链接与嵌入），其宗旨是在 Windows 系统的统一管理下协调不同的应用程序，准许这些应用程序之间相互沟通、相互控制。每个应用程序决定自己的哪些信息暴露出来，这些暴露出来的信息可以被其他程序操纵。这个标准被微软公司极力推广，许多软件开发商也积极响应。

这样做的好处是显而易见的：把多个应用程序组织起来，各用其长，协调在一起为

我们工作。假如要进行一项建筑设计，首先我们要利用 AutoCAD 绘制建筑图纸，然后用 Word 起草报告，其中要列出各种建筑材料、办公桌椅的数量和尺寸。如果能从做好的图形中直接读出这些信息插入到报告中，最大程度地达到数据共享，效率岂不更高？

ActiveX Automation 通过在两个程序间安排对话，达到一个程序控制另一个程序的目的；但这个对话不是双向的，对话双方也不是平等的。客户程序（Client）是开始要求对话的应用程序，服务程序（Server）是响应 Client 的应用程序。ActiveX Automation 的代码在 Client 上运行，而这个代码所控制的动作由 Server 执行。

一个应用程序控制另一个应用程序的过程一般是这样的：首先，一个应用程序决定发动 ActiveX Automation 操作，这个应用程序自动成为 Client，被它调用的应用程序成为 Server。Server 收到对话请求后，决定暴露那些对象给 Client。在给定时刻，由 Client 决定实际使用哪些对象。然后 ActiveX Automation 命令被传给 Server，由 Server 对这些命令作出反应。Client 可以持续地发出命令，Server 则忠实地执行每一条命令。最后由 Server 明确提出终止对话。

2. AutoCAD 对 ActiveX Automation 的支持

AutoCAD R14 最大的改进就是对 AutoCAD 本身增加了 ActiveX 自动化服务功能（ActiveX Automation Server Capabilities），也就是说完全可以作为服务程序，用户可以从其他 ActiveX 客户程序中操作 AutoCAD。

应用程序通过 ActiveX Automation 显露其功能的程序差别很大。例如微软的 Excel 等允许客户程序处理其内部数据及用户界面，可以改变一个 Excel 工作表上单元格的内容，也可以在 Excel 菜单上加上一项，甚至做更大的改动，使其在外表上看起来完全不像 Excel，倒像被人开发好的应用程序。而另外有些应用程序，比如微软的 Visul Baisc，只对客户程序显露其很小的一部分。

AutoDesk 采取偏于前者的策略，通过 ActiveX Automation 可以操作图形中的所有实体，读取、设置系统变量，控制视图区的大小和位置。但对界面元素，如菜单的内容、工具条的显示位置等，AutoCAD 不暴露给客户程序。也就是说，客户程序可以以任何方式处理图形，但不能扩大 AutoCAD 本身的环境。你不能编写一个程序，让它记录当前界面上有哪些工具条、各工具条的摆放位置，以便以后恢复这些设置。因为这些信息并没有暴露给客户程序。

除了不能控制 AutoCAD 界面元素以外，Automation 尽可以随心所欲地做一切工作。用户完全可以把 AutoCAD 当成自己程序中的一个盛放图形的窗口（甚至看不到窗口的存在），用到时打开、编辑、打印出图，不用时关闭，继续做自己的事情，完全不必理会 AutoCAD 本身。

3. 可作为客户程序的 VB 和 AutoCAD VBA

ActiveX 服务程序比客户程序要多得多，因为给一个程序编写 ActiveX 服务程序比把它做成 ActiveX 客户程序要简单得多。尽管如此，仍然有各类 ActiveX 客户程序供我们选择，可以用它来控制包括 AutoCAD 在内的各种 ActiveX 服务程序。我们常见的可作为客户程序的软件包括：Visual Basic 4.0 以上版本、Excel 95 或 97、Word 95 或 97、Access95

或 97、PowerPoint97 等。

VB 是最为常用的支持 ActiveX Automation 技术的开发工具，其语法简单、功能强大，提供建立可执行代码的源代码编译器。

在 AutoCAD Release 14 发行时，除了提供一个正规版本外，还提供了一个 AutoCAD R14 Preview VBA 版本（我们称之为“AutoCAD VBA”）。由于 AutoCAD VBA 是嵌入在 AutoCAD 内部的，共享 AutoCAD 的内存空间，可以更好地与 AutoCAD Release14 沟通。其运行速度与 ObjectARX 应用程序非常接近，较其他开发方式如 ADS，LISP 在速度上要提高许多。同时，AutoCAD VBA 也提供了应用程序与其他 VBA 程序（如 Microsoft Excel）的沟通路径，当使用其他应用程序的对象类型库时，AutoCAD 可以作为该程序的客户程序，而不仅仅是 AutoCAD 本身。

VB 和 AutoCAD VBA 都是基于 ActiveX Automation 技术的开发环境，语法相同，易学易用。如果掌握了怎样用 VB 开发 AutoCAD，那么在 AutoCAD VBA 中编程就不费任何力气——两种开发方式的原理是一样的，只需要简单的改动就可以将两个环境下的程序代码互换。

鉴于多数读者对 VB 5.0 可能更为熟悉，本书以 VB 5.0 为客户程序，详细论述用 VB 手段开发 AutoCAD 的过程，并在附录 D 中介绍了 AutoCAD VBA 的有关内容。在使用本书的同时，最好在你的系统里安装了 Visul Basic 5.0 版本，以便边学边练。

4．本书的结构安排

本书主要介绍了用 VB 5.0 开发 AutoCAD R14 这一全新、便捷的开发方式。在结构安排上，既考虑了 AutoCAD 作为工程制图工具的一面，又考虑到 VB 5.0 作为编程语言和环境的一面，以对象模型为贯穿全书的主线，以完成一个具体的 AutoCAD 操作为目的。

第 1 章用一个简单的实例开始我们的学习过程。这部分内容介绍了 AutoCAD Automation 编程的整个过程，从引用 AutoCAD 类型库、书写编程代码到程序编译、运行。

第 2 章介绍与 AutoCAD Activex 程序开发有关的 VB 知识，从对象这一基本概念开始，到 AutoCAD 部件对象的引用和访问、AutoCAD 对象模型及各对象的主要功能，以及用对象的方法和属性来完成某个操作的过程。

第 3 章介绍了 VB 程序与 AutoCAD 的连接，讲解怎样在 VB 程序中启动 AutoCAD 应用程序，以及 AutoCAD 应用程序对象可以完成的操作。

第 4 章以文档对象为核心，讲述了与文件操作有关的内容，如图形文件的打开、关闭，系统变量的存取，图形数据交换等。

第 5 章和第 6 章介绍了在模型空间和图纸空间创建图形实体对象，以及利用实体对象的方法和属性进行实体编辑。

第 7 章以非实体集合及其成员为核心，介绍了图层、线型、文本样式等 12 个非实体集合及其成员的创建、编辑。

第 8、9、10 三章分别以优先设置对象、实用工具对象和出图对象为基础，介绍了设置 AutoCAD 运行环境、用户输入技术及数据类型转换、出图控制等内容。

第 11 章介绍了程序设计中较为实用的高级编程技术，如选择集操作、扩展数据访

问、坐标转换、外部文件访问等方面的内容。

第 12 章和 13 章列举了 AutoCAD Activex 所有对象的方法、属性的功能及其用法，用户在程序设计中可以随时查阅这些资料。

另外，本书安排了 5 个附录，附录 A 以列表的形式简要说明了 AutoCAD 所有对象的方法和属性的功能，读者可以从这个表中获取完成某一操作所使用的方法或属性，然后按照第 12、13 章的具体语法去操作；附录 B 介绍了 AutoLISP 的函数与 ActiveX 方法、属性的对应关系，以便熟悉 AutoLISP 的读者可以很快接受 Automation；附录 C 介绍了 AutoCAD 系统变量；附录 D 介绍了 AutoCAD R14.01 版本的 AutoCAD VBA 方面的有关内容；在附录 E 中列举了程序设计专用的 DXF 组码，相信每个高级编程人员对这方面是很感兴趣的。

本书由张国宝编著，参加资料准备的有徐洁、张素艳、章卫东、张素娟、高杰、张素梅等。

本书在编写过程中得到了张志修、张晓今、赵振庆等领导的大力支持，在此谨表感谢。

书中疏漏和错误之处，敬请读者批评指正。

编者

1999 年 10 月

目　录

第 1 章　编程实例

万事开头难。作为本书的开始，我们首先书写一个完整的 AutoCAD ActiveX 应用程序，让你对编程有一个直观、初步的认识。程序的功能很简单：在 AutoCAD 绘图区内注写字符“AutoCAD Automation”。

麻雀虽小，五脏俱全。这个实例演示了 AutoCAD Automation 应用程序开发的一个完整的过程：从进入 Visual Basic 5.0 集成开发环境、引用 AutoCAD R14 对象类型库开始，到设计可视化界面、设置控件属性、书写程序代码，直到运行和编译程序。程序编译成 EXE 文件后，可以在 Windows 操作系统环境和 AutoCAD 应用程序环境下运行，对此本章也做了详细说明。

1.1　引用 AutoCAD R14 类型库

运行 Visual Basic 应用程序，在新建窗口选择“标准 EXE”，单击“打开”按钮（如图 1.1），进入 VB 综合开发环境（IDE）。

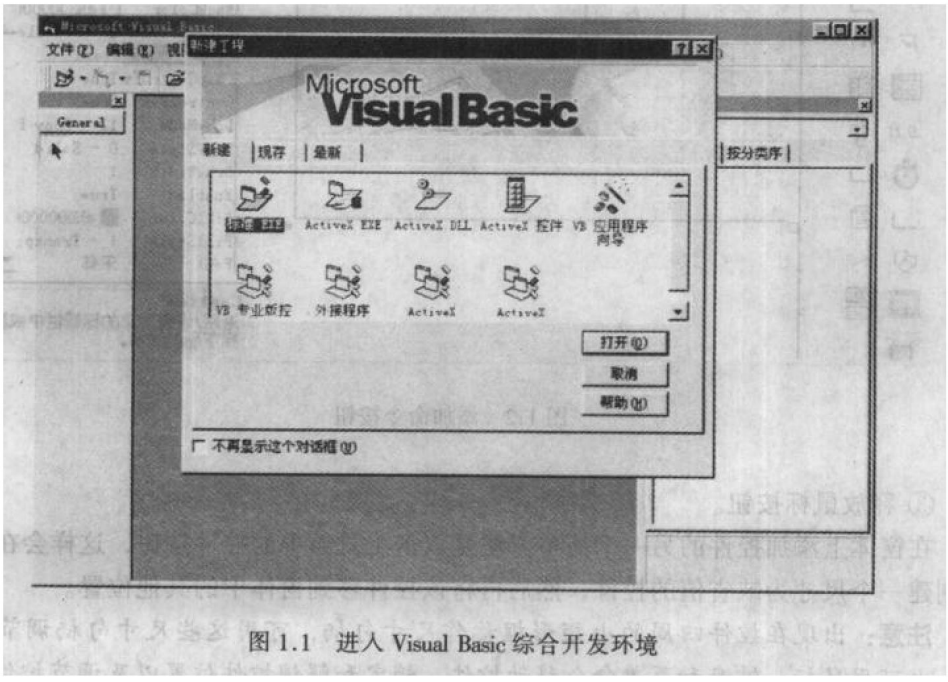

图 1.1　进入 Visual Basic 综合开发环境

进入 VB 编程界面后，第一步工作是引用 AutoCAD R14 对象类型库。操过程很简单：

① 从“工程菜单”中选择“引用”；

② 在引用对话框中选择 AutoCAD R14 Object Library，如果该项没有列出，选择“浏览”。在打开的文件对话框中选择 acad.tlb 文件（位于 AutoCAD R14 可执行目录下）。

1.2 设计可视化界面

编写 Visual Basic 应用程序的第一步是创建窗体，这些窗体是应用程序的门面。进入 VB5 IDE 后，系统自动给我们创建一个名为“Form1”的窗体，我们只需在窗体上绘制构成界面的控件就可以了。对于本应用程序，只用到工具箱中的一个控件——命令按钮。

我们首先用工具箱向“Form1”窗体中添加命令按钮控件：

① 单击工具条上的命令按钮控件 ；

② 将指针移到窗体上，这时指针变成十字线，将十字线放在控件的左上角（如图 1.2），拖动十字线画出适合你要的控件大小的方框；

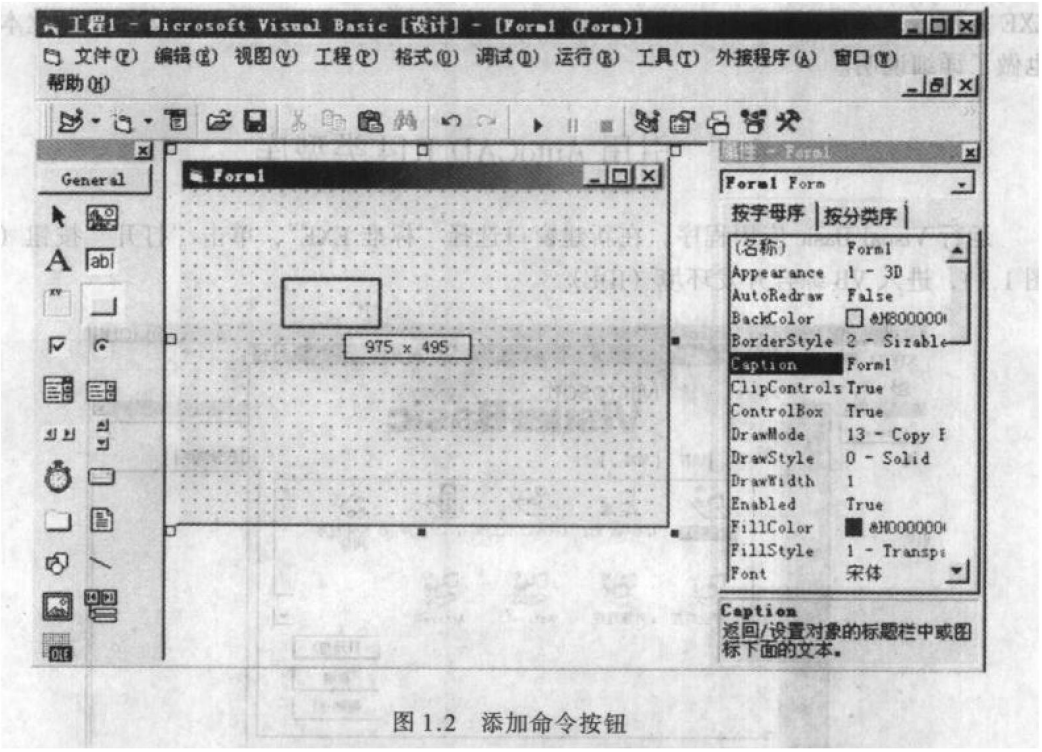

图 1.2 添加命令按钮

③ 释放鼠标按钮。

在窗体上添加控件的另一个简单方法是双击工具箱中的控件按钮，这样会在窗体中央创建一个尺寸为缺省值的控件，然后再将该控件移到窗体中的其他位置。

注意：出现在控件四周的小矩形框称作尺寸句柄，可用这些尺寸句柄调节控件尺寸；也可用鼠标、键盘和菜单命令移动控件、锁定和解锁控件位置以及调节控件位置。

重复两次上面的工作，我们在窗体中添加 3 个命令按钮。这些按钮的大小和位置可能不令人满意，我们用句柄来进行调整。现在已生成了我们应用程序的界面，如图 1.3。

我们设计这样的可视化界面，意图是这样的：当单击“Command1”命令按钮时，程序与 AutoCAD 应用程序开始沟通；当单击“Command2”命令按钮时，执行某个具体

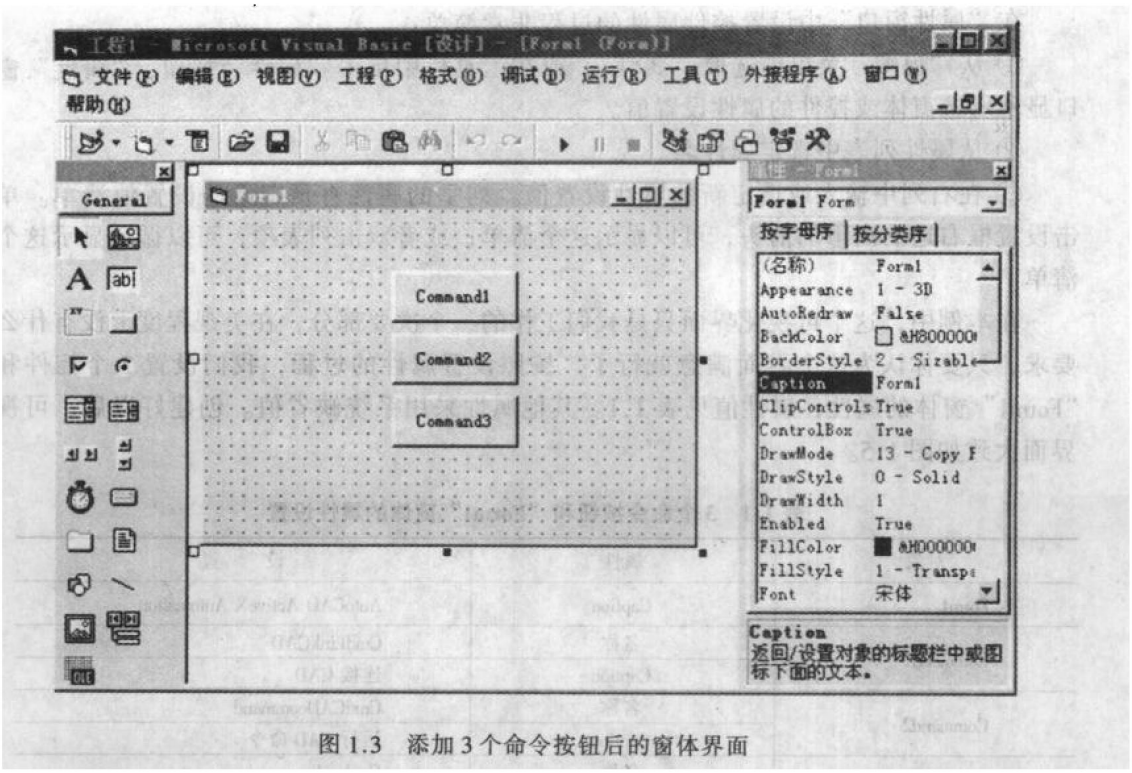

图 1.3　添加 3 个命令按钮后的窗体界面

的 AutoCAD ActiveX 操作；当单击“Command3”命令按钮时，结束程序运行。按照这个设计意图，我们首先来设置控件的属性。

1.3　设置控件属性

下一步是给添加的控件对象设置属性。属性窗口（如图 1.4）给出了所有窗体和窗体上控件的当前属性。在“视图”菜单中选择“属性窗口”命令（或单击工具栏上的“属性窗口”按钮）可以打开属性窗口。

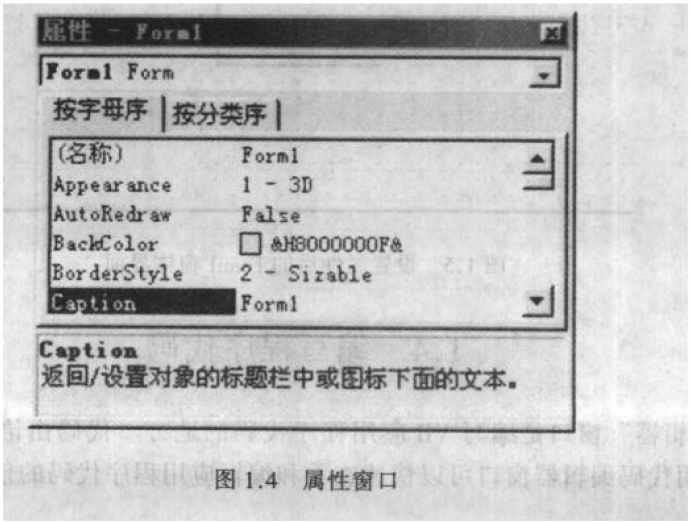

图 1.4　属性窗口

在“属性窗口”中设置控件属性的过程非常简单：

① 从“视图”菜单中选取“属性”，或在工具栏中单击“属性”按钮，“属性”窗口显示所选窗体或控件的属性设置值。

② 从属性列表中选定属性名。

③ 在右列中输入或选定新的属性设置值。列举的属性有预定义的设置值清单。单击设置框右边的向下的箭头，可以显示这个清单；或者双击列表项，可以循环显示这个清单。

在本例中，这个可视化界面只是我们工作的一个次要部分，在美观程度上没有什么要求，只要你认为这个界面满意就行了。按照设置属性的过程，我们设置 3 个控件和“Form1”窗体的属性，属性值见表 1.1。其他属性采用系统缺省值。创建好以后，可视界面大致如图 1.5。

表 1.1 3 个命令按钮和“Form1”窗体的属性设置

对象	属性	设 置
Form1	Caption	AutoCAD ActiveX Automaiton
Command1	名称	CmdLinkCAD
	Caption	连接 CAD
Command2	名称	CmdCADcommand
	Caption	运行 CAD 命令
Coommand3	名称	Cmdend
	Caption	退出

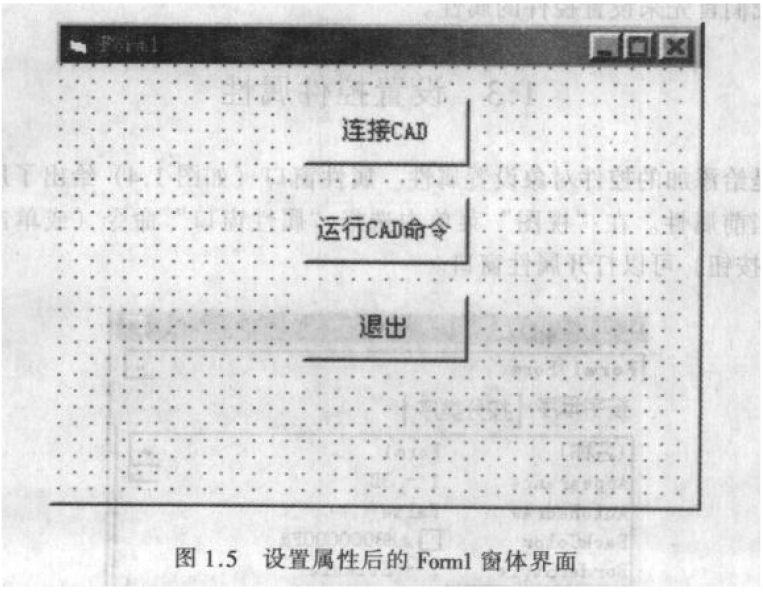

图 1.5 设置属性后的 Form1 窗体界面

1.4 编写程序代码

“代码编辑器”窗口是编写 VB 应用程序代码的地方。代码由语句、常数和声明部分组成。使用代码编辑器窗口可以快速查看和编辑应用程序代码的任何部分。

双击要编写代码的窗体或控件，则进入代码编辑器。

Visual Basic 应用程序的代码被分为称为过程的小的代码块。过程包含了事件发生（例如单击按钮）时要执行的代码。控件的事件过程由控件的实际名称（Name 属性指定）、下划线（_）和事件名组合而成。例如，在单击一个名为 Command1 的命令按钮时调用的事件过程，可称为 Command1-Click 事件过程。

创建事件的过程如下：

① 在“对象”列表框中选定活动窗体中的一个对象名。② 在“过程”列表框中选择指定对象的事件名。③ 在 Sub 和 End Sub 语句之间输入代码。

1.4.1　编写声明部分程序代码

在代码窗口的“声明”部分书写图 1.6 所示的代码。

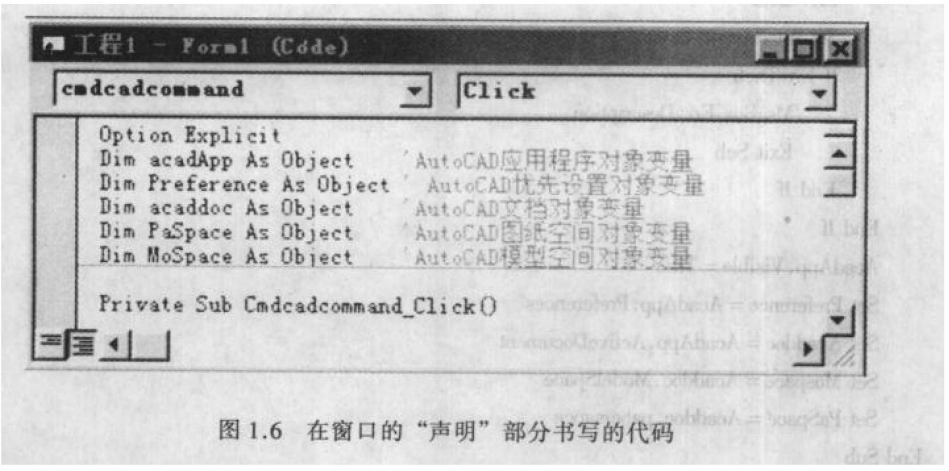

图 1.6　在窗口的“声明”部分书写的代码

第一句用以要求一切变量必须声明，否则不予承认。下面几句声明了对象型变量，用于引用相应的对象。如果工程文件中包括多个窗体，就要声明成 Public 型，以便其他窗体调用。有关声明方法请参考 VB 中的内容。

AutoCAD R14 曾发行过两个类型库，随 R14.01 版本发行的类型库支持事前绑定（ealy-bound），事前绑定可以使程序运行得更快，并能在编译期间就可以进行错误检查。

图 1.7　将对象声明成具体的类

所以，如果你使用的是 R14.01 版本，可以将各变量直接声明成某一具体的类，像图 1.7 那样。

1.4.2 CmdLinkCAD 按钮的 Click 事件

CmdLinkCAD 命令按钮的 Click 事件执行与 AutoCAD 应用程序的连接，这个过程是每个 AutoCAD ActiveX 程序所必须的，VB 对 AutoCAD 所有对象的操作都要通过这个连接点。代码如下：

```
Private Sub cmdlinkcad _ Click( )
    On Error Resume Next      '出错继续
    Set AcadApp = GetObject( ,"AutoCAD. Application")
    If Err Then      '如果出错
        Err. Clear
        Set AcadApp = CreateObject("AutoCAD. Application")
        If Err Then
            MsgBox Err. Description
            Exit Sub
        End If
    End If
    AcadApp. Visible = True
    Set Preference = AcadApp. Preferences
    Set Acaddoc = AcadApp. ActiveDocument
    Set Mospace = Acaddoc. ModelSpace
    Set PaSpace = Acaddoc. papersapce
End Sub
```

这一段代码定义了一个过程 CmdLinkcad _ Click：如果 AutoCAD 正在运行，GetObject 函数返回对 AutoCAD 应用程序（AutoCAD. Application）对象的引用；否则发生一个错误，发生错误后要做两件事：首先，用 ErrClear 方法清除错误信息，接着用 CreateObject 函数试图创建 AutoCAD 应用程序的引用。如果成功，AutoCAD 被启动，否则用消息框返回一个错误信息，告诉用户 AutoCAD 不能启动，退出连接过程。

AutoCAD 不能被启动的原因是多样的，包括：系统中不存在 AutoCAD，AutoCAD 在系统注册表中有错误，系统资源紧张等。AutoCAD 即使正在运行，对它的引用同样也可能失败。如 AutoCAD 正处于忙状态（如 LineType 对话框打开等），它就不会响应 VB 程序的请求。在程序设计时，应对有可能发生的错误进行补救，专门设计容错处理。如果你的编程环境不完全支持上面的代码，可以作适当的改动。

在缺省状态下，AutoCAD 启动后其 Visible 属性是 False，也就是说程序界面是不可见的。用下面语句使其可见：

```
AcadApp. Visible = True
```

如果不设置界面为可见状态，可能会遇到一个小麻烦——这里存在一个 AutoDesk 公司未曾预料到的问题：前面我们说过，AutoCAD 应用程序暴露的对象不包括界面对象（如工具条等），所以这些对象是否可见不受程序的控制，AutoCAD 启动后其程序界面不可见，但这些工具条却可以浮在屏幕上，造成一种很不体统的局面。当然，这些小插曲