

Mastering AutoCAD 2000 Objects

# AutoCAD 2000 对象开发 从入门到

〔美〕 Dietmar Rudolph 著

曾琦 等译

# 精通

对于经验丰富的AutoCAD用户来说，本书是相当重要的资源，因为他们需要更进一步地访问图形文件。本书中包含关于图形文件中物体的重要信息、访问对象的五种方法以及其他一些建议和指令。



电子工业出版社

Publishing House of Electronics Industry  
URL: <http://www.phei.com.cn>

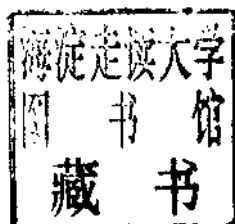
$\tau_f = 91.72$   
LDF/1

**Mastering AutoCAD 2000 Objects**

# AutoCAD 2000对象开发 从入门到精通

[美] <sup>D</sup>Dietmar <sup>鲁道夫</sup>Rudolph 著

曾 琦 等译



電子工業出版社

**Publishing House of Electronics Industry**

北京 • BEIJING

1056773

## 内 容 提 要

本书是一本介绍读、写和操作AutoCAD图形文件的手册。它不仅全面、系统地介绍了AutoCAD的各种对象,还详细介绍了访问这些图形对象的五种主要方法:AutoLISP和Visual LISP、ObjectARX、ActiveX Automation、ObjectDBX和DXF文件。本书的介绍基于AutoCAD 2000版本,其安排循序渐进、内容详尽,结构合理,是AutoCAD软件使用者的有力辅助工具。

本书适用于使用AutoCAD的中高级用户,尤其适用于具有一定经验的AutoCAD软件开发人员。



Copyright©2000 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

### 图书在版编目(CIP)数据

AutoCAD 2000对象开发从入门到精通/(美)鲁道夫(Rudolph, D.)著.—北京:电子工业出版社,2000.9

书名原文:Mastering AutoCAD 2000 Objects

ISBN 7-5053-6172-4

I. A… II. ①鲁… ②曾… III. 计算机辅助设计-应用软件-AutoCAD 2000 IV. TP391.72

中国版本图书馆CIP数据核字(2000)第69315号

书 名: AutoCAD 2000对象开发从入门到精通

著 作 者: [美] Dietmar Rudolph

译 者: 曾 琦 等

责任编辑: 王 军

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河金马印装有限公司

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036 电话: 68279077

北京市海淀区翠微东里甲2号 邮编: 100036 电话: 68207419

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 18.75 字数: 470 千字

版 次: 2000年9月第1版 2000年9月第1次印刷

书 号: ISBN 7-5053-6172-4

TP·3312

定 价: 30.00元

版权贸易合同登记号 图字: 01-1999-2986

凡购买电子工业出版社的图书,如有缺页、倒页、脱页请向购买书店调换,若书店售缺,请与本社发行部联系。

JS414/26  
04

## 译者序

本书是Dietmar Rudolph多年努力的结果。经过长达12年的潜心研究和创作，作者深入了解AutoCAD的内核，并因此将AutoCAD中每个对象的含义、属性以及访问AutoCAD对象的多种途径公布于众。虽然本书并不能称为AutoCAD的操作和使用方法的全书，但它深入剖析了AutoCAD的各种对象以及访问这些对象的不同方法，因此，它当之无愧是学习AutoCAD核心内容的首选书籍。

本书层次清楚，结构分明。全书共由19章组成，首先介绍图形数据库的基本概念和访问数据库对象的主要方法，然后详细介绍了AutoCAD的各种对象及其属性和含义。附录部分提供了一些辅助信息，以帮助读者更好地学习和理解AutoCAD图形数据库。

总之，无论读者是初次学习AutoCAD二次开发，还是已经具有一定的开发经验，本书都能为读者提供最大的帮助。

在翻译过程中，我们本着严谨认真的态度，反复推敲了一些专业术语，部分程序代码在AutoCAD 2000中进行了调试，以便深入理解作者的编程思路。除封面上署名的译者外，许书明、王维、孔祥丰、邱丽、袁建华、王玉娟、曹木军、胡辰浩、王惠卿、李万红、王军、翟志强、寸少娟、齐磊、刘亚平、周志林等也参与了本书翻译和校对工作。由于水平有限，翻译不妥之处敬请读者批评指正。

译者

## 引言

AutoCAD是世界上使用范围最广的计算机辅助设计和辅助绘图软件。根据Autodesk的资料，在日常生活中，有两百多万的人运用AutoCAD去定义并记录技术设计方案，或存储图像、站点计划以及其他图形数据。这些数据被存入AutoCAD的图形文件中。同样，根据Autodesk所提供的信息可知，现在的计算机文件中，AutoCAD图形文件多达20多亿。

AutoCAD提供了大量用于创建、操作及查询图形文件内容的命令和功能，但是这些内置命令仅仅是使用这些文件的无数方法中的几种。嵌入图形数据的一些明显的运用包括统计和报表生成、NC程序和绘图文件的生成，以及与其他CAD或图形软件之间的数据交换。

因此，访问图形文件的内核就变得极为重要。1998年OpenDWG联盟的建立提供了访问图形文件内核的最完善文档资料，OpenDWG联盟是Autodesk的一个竞争性组织，它的成立促使了更多的访问AutoCAD图形文件方法的生产。

本书介绍了多种读、写、评估及操作AutoCAD图形文件的方法。其中的一些方法仅仅需要用户能够使用文本编辑器或简单的一张纸，另一些方法要求用户懂得一些计算机编程语言。当然，本书不会介绍这些编程语言，读者可参考其他有关编程语言方面的书籍。

然而，不管用户采用的是哪一种方式来访问图形文件中嵌入的数据，总会面临同样的问题：数据的含义是什么？计算机图形中有些什么样的对象呢？定义这些对象需要什么属性呢？属性取值的种类和格式又是如何呢？

本书的目的就是回答上述问题：AutoCAD图形文件的内容。读者可以从中了解到评估图形（通常称为图形数据库）时会遇到的所有知识。本书介绍了查找或操作对象的各种方法，例如构成图形数据库的线和圆。但本书的主要目的是介绍每个对象的含义和属性。

### 难道还不明显吗？

但是读者是否真的需要一本完全介绍图形文件内容的书吗？组成图形的线和圆有多么复杂呢？有两点必须注意：图形文件中远不止线和圆这两种对象，尺寸、样条线、立方体以及地形模型都能成为图形文件的一部分。为了处理和操作这些对象，必须先理解它们。其次，即使是那些看起来类似线和圆的简单对象，AutoCAD也存储并使用了大量的属性，其中许多属性是很难理解的。

一个典型的例子是点对象。如果要列出定义一个点所需的全部属性，究竟需要多少属性呢？在2D视图中，即在一张纸上，必须有两个距离描述点：其中一个从纸张的左边缘开始（即 $x$ ），另一个从底边开始（即 $y$ ）。用数学术语来说，这些距离被称为坐标。在三维空间中，必须用三个坐标去定义点：左侧距离、右侧距离和高度。

如果更多地联系到计算机图形，而不是数学定义，就会在列表中增加另一个属性：点在屏幕或纸上所显示的颜色。另外，大多数绘图程序还采用一个层属性去分组相关的对象。

但是你能设想给点一个方向吗？大概不能。AutoCAD却可以做到，它不仅可以为每个点保存方向，而且保存的是两个方向。这使情况变得更加复杂了，因为AutoCAD甚至可以给点赋予高度值和厚度值。觉得有些奇怪，不是吗？有厚度的点是什么样的呢？正如读者将看到的，它是某类平面在3D空间中的排列。读者不会将它们称为点，但AutoCAD可以。

AutoCAD的一些对象很易于理解，至于其他的一些对象，就需要读者对AutoCAD创建、操作和保存图形数据具有更深入的了解，这就是本书的写作目的。本书为读者提供了AutoCAD图形文件的全部资料，甚至可能更多。

## 访问AutoCAD对象

自从1982年AutoCAD开始发展以来，Autodesk的程序员就知道他们无法支持每一个应用程序在图形文件中保存的数据。因此，他们为软件增加了开放性。第一版的AutoCAD就允许用户采用非AutoCAD命令去访问和操作图形。

随着时间的推移，Autodesk增加了大量访问AutoCAD图形数据的方法，并且Autodesk的开发者还不断地添加新的方法。但是，不管采用哪种方法，都有些共性：所有访问方法都可以读、写内部数据。除了少数例外，用户可以采用任意方法访问整个图形。所以，选择何种方法取决于用户的工作环境。

### AutoLISP和Visual LISP

利用AutoCAD内置的编程语言（AutoLISP）可以访问整个AutoCAD图形内容。AutoLISP和它的新变体Visual LISP都是LISP语言。LISP是一种编程语言，尤其适用于处理CAD图形中的各种数据元素，例如坐标列表、数值、关键字等等。AutoLISP包括一些从当前图形中读、写数据的一些特殊语言结构，包括entget、entmod和entmake。

AutoLISP和Visual LISP具有访问图形数据库的所有方式，但无法用这两种LISP创建任意的定制对象。下面列出它们的两个缺点：

- AutoLISP和Visual LISP只在AutoCAD会话中有效
- 只能访问AutoCAD当前加载的图形

采用AutoLISP和Visual LISP的一个最大的好处是：只需五行代码就可以产生、抽取和操作图形对象。

### ObjectARX方法

ObjectARX是创建AutoCAD本身的基础。作为一套C++库，任何想为AutoCAD增加功能的第三方都可以利用ObjectARX。ObjectARX的数据访问部分被称为AcDb库，这是AutoCAD数据库的缩写。AcDb库包含了创建、评估以及操作AutoCAD图形对象的所有功能。

程序员可以运用ObjectARX访问图形的每一个细节部分。AcDb库甚至允许同时访问多个图形，而不管它们是否是AutoCAD当前所打开的。ObjectARX几乎无所不能，包括创建完整的新的对象类。但是ObjectARX也存在着一些缺陷。首先，它只能用在AutoCAD环境下，用户只能在AutoCAD内采用ObjectARX，或在基于AutoCAD OEM内核的产品中使用它。第

三个缺点是编程上的不足，用户只能用C++语言，并且需要大量的培训和时间去掌握这个编程环境。

## COM和ActiveX Automation

ActiveX Automation以前称为OLE Automation，现在也称为COM Automation，Microsoft用它来定义标准的进程间通信。Microsoft Windows平台上运行的不同程序之间可以互相输出数据并完成特定的功能。与其他许多程序一样，AutoCAD通过ActiveX Automation显示了文档的大部分内容，并因此形成了自动化服务器（automation server）。在AutoCAD内，文档常被称为图形。

任何其他采用该接口的程序称为自动化客户（automation client），它可以从AutoCAD查询对象数据。另外，ActiveX Automation允许客户创建并操作对象。正如ObjectARX和LISP，ActiveX Automation可以随时访问图形数据，可以利用它远程控制AutoCAD的绝大多数函数。

ActiveX Automation是DCOM（Distributed Component Object Model）的一部分，可以把ActiveX Automation看作是封装了DCOM小部分功能的软件包。程序员可以直接通过DCOM访问AutoCAD或其他组件，而不必采用ActiveX Automation。因为所有主要编程环境也提供了访问AutoCAD图形数据的友好的ActiveX Automation接口，所以通常不需要直接通过DCOM进行访问。但是，图形中有些对象不能通过AutoCAD的ActiveX Automation接口访问，它们仅对于DCOM开放，本书只在这些特殊的例子中使用DCOM。

ActiveX Automation的一个优点是：程序员可通过它的标准化接口自由选择编程语言或开发环境。当然，唯一的要求是编程语言或开发环境必须支持ActiveX Automation的客户程序。ActiveX 客户程序的最常用环境是Visual Basic和Delphi，以及可嵌入在许多应用程序内的Visual Basic for Applications（VBA），例如Microsoft Excel、Microsoft Word，甚至AutoCAD本身。然而，ActiveX Automation客户也能用Java、C或其他语言编写。采用ActiveX Automation评估或操作图形的缺点有：在对象的访问和创建过程中受到许多局限，并且它还依赖于AutoCAD。因为客户程序不知道图形的内部信息，所以它需要AutoCAD处于运行状态，以便打开图形数据库并与它对话。

## ObjectDBX方法

到目前为止讨论的所有方法都必须加载AutoCAD。但是，许多人希望在不加载甚至不购买AutoCAD的情况下访问图形文件。根据这种需要，就产生了图形访问库的市场。一些公司开发并出售C库，用于在没有AutoCAD的条件下读、写AutoCAD的图形文件。

尽管如此，这些公司都遇到了一个大问题：Autodesk从不公布图形文件的内部结构。因此，DWG读/写库的创造者不得不逆向构造文件格式，从字面上去猜测图形文件中每一位的含义。虽然工作是复杂的，但结果却相当好，只是没有达到Autodesk公开图形文件的程度。但是，最终Autodesk提供了这样的工具箱，起初它叫DWG Unplugged，现在称为ObjectDBX。

程序员可以利用ObjectDBX创建在AutoCAD外读/写AutoCAD图形文件的程序。ObjectDBX是一组C++库，类似于ObjectARX的AcDb库。就像ObjectARX一样，采用ObjectDBX访问

图形数据库只在特定的环境中才能进行，这里特定的环境指Microsoft Visual C++和Microsoft Windows。如果程序在Sun工作站运行，那么就不能利用ObjectDBX访问图形。

## DXF文件

DXF文件最早产生。AutoCAD在1982年第一次发布时，并没有Microsoft Windows、AutoLISP、ObjectARX或ActiveX Automation，但仍然有读、写和操作AutoCAD图形文件的方法，用户不需要运行AutoCAD，也不需要特定的开发平台，那么他们是怎样做的呢？

Autodesk的程序员让AutoCAD将图形的全部内容复制到格式化的文本文件，该文件叫做Drawing Interchange File（图形交换文件），简称为DXF。从定义上来说，DXF文件包含了AutoCAD图形文件的所有内容，它可以被简单的计算机程序或其他人读取。操作和创建DXF文件是极其简单的。更改过的文件能再次加载到AutoCAD，并因此构造一个新的或修改过的图形数据库。

自从1982年以来的每一版AutoCAD都能够读、写DXF文件。DXF文件和相应的DWG文件相似，它们都包含了完全相同的数据库，仅在格式上有差别——除非有错误发作。与DWG文件不一样，DXF文件所显示的数据库由Autodesk进行了文档化，至少是部分地文档化。与AutoCAD一起提供的DXF文档一直以来都被认为是不完全的，有错误的，并且是极难理解的。

DXF几乎能被每一种图形程序读取，甚至可用在没有AutoCAD的群体间互换图形。采用DXF访问和操作图形的一大优点是：它可用于任意平台。因为DXF文件是普通的文本，所以它们可以在Unix、MacOS、DOS和Windows上被访问。由于可以获得完整的图形数据库，所以对它们的操作可以是无限制的。与前文介绍的其他访问方法不同，DXF也被AutoCAD LT支持。DXF的最大缺陷是：它没有错误处理。操作DXF文件时，可能会产生许多的错误。只有当AutoCAD或其他目标程序把DXF文件转换回图形数据库，用户才会发现错误定义的对象。

## 其他方法

除了上述五种方法外，也可以采用其他方法将对象从AutoCAD图形中读出，或将对象输入图形。不管怎样，只有上述五种方法才提供了图形数据库的完整的访问途径。任何其他方法，例如WMF、DWG、IGES、STEP、SAT、3DS等，都只是粗略的方法。采用这些方法时，图形的内容与原始图形有所不同，有时对象仅仅因为格式不支持就被忽略，更多情况是对象被更改，以适应目标格式。无论是那种情况，信息都在转换过程中被丢失，并且不能恢复。只有前文介绍的五种方法才会获得完整的图形文件信息，并可以在以后完全恢复这些信息。

## 有关版本的说明

AutoCAD在不断地发展，每一个新版本都有许多新的功能。从普遍的意义上来说，新的功能意味着图形数据库内容的增加。这是很容易理解的：新功能需要更多的空间去存储相关信息，而图形数据库应该是存储图形相关数据的唯一地方。

**说明：**AutoCAD的图形数据库不包含从头重新构造图形的所有必要信息，例如，形状字符和/或形状符号就不是DWG文件的一部分。相反，这些信息存在于独立的计算机文件中，外部被引用的图形文件或位图图像也是如此。

当描述或使用AutoCAD图形数据库时，必须添加AutoCAD数据库的版本和/或AutoCAD的相关版本。下表是1984年以来发布的所有重要版本。

| <b>AutoCAD图形<br/>数据库版本</b> | <b>归属AutoCAD版本</b>               | <b>发布时间</b> |
|----------------------------|----------------------------------|-------------|
| AC1.50                     | AutoCAD Release 5, Version 2.0   | 1984        |
| AC2.10                     | AutoCAD Release 6, Version 2.1   | 1985        |
| AC1002                     | AutoCAD Release 7, Version 2.5   | 1986        |
| AC1003                     | AutoCAD Release 8, Version 2.6   | 1987        |
| AC1004                     | AutoCAD Release 9                | 1987        |
| AC1006                     | AutoCAD Release 10               | 1988        |
| AC1009                     | AutoCAD Release 11和12            | 1990        |
| AC1012                     | AutoCAD Release 13               | 1994        |
| AC1014                     | AutoCAD Release 14               | 1997        |
| AC1015                     | AutoCAD Release 15, AutoCAD 2000 | 1999        |

本书介绍的是AutoCAD 图形数据库的AC1015版本。虽然本书介绍的许多概念也适用于其他版本，但许多对象和属性在老版本中是没有的。

## 目 录

|                                  |    |
|----------------------------------|----|
| <b>第1章 了解AutoCAD数据库</b> .....    | 1  |
| 1.1 图形中有什么? .....                | 1  |
| 1.2 对象和容器 .....                  | 2  |
| 1.3 识别对象 .....                   | 5  |
| 1.4 类与继承 .....                   | 6  |
| 1.5 对象属性 .....                   | 7  |
| 1.6 本书是AutoCAD图形数据库的完备文档吗? ..... | 9  |
| 1.7 总结 .....                     | 9  |
| <b>第2章 访问图形数据库</b> .....         | 11 |
| 2.1 顺序化的方法——DXF .....            | 11 |
| 2.2 AutoLISP——DXF的另一个名称 .....    | 17 |
| 2.3 ActiveX Automation .....     | 24 |
| 2.4 面向对象的方法——ObjectARX .....     | 29 |
| 2.5 访问图形数据库的其他方法 .....           | 34 |
| 2.6 总结 .....                     | 36 |
| <b>第3章 图形数据库的数据类型</b> .....      | 37 |
| 3.1 图形数据库中的整数 .....              | 37 |
| 3.2 伪整数 .....                    | 40 |
| 3.3 图形数据库中的实数 .....              | 42 |
| 3.4 图形数据库中的字符串 .....             | 44 |
| 3.5 图形数据库中的句柄 .....              | 45 |
| 3.6 对象指针 .....                   | 45 |
| 3.7 图形数据库中的点 .....               | 45 |
| 3.8 图形数据库中的二进制数据 .....           | 49 |
| 3.9 总结 .....                     | 50 |
| <b>第4章 图形专用设置</b> .....          | 51 |
| 4.1 如何访问图形设置 .....               | 52 |
| 4.2 关键设置 .....                   | 53 |
| 4.3 附加设置 .....                   | 61 |
| 4.4 DXF文件中的设置序列 .....            | 78 |
| 4.5 总结 .....                     | 80 |

|                                   |     |
|-----------------------------------|-----|
| <b>第5章 对象的共有属性</b> .....          | 81  |
| 5.1 对象句柄 .....                    | 81  |
| 5.2 对象的类 .....                    | 82  |
| 5.3 对象的所有者 .....                  | 83  |
| 5.4 反应器链接 .....                   | 83  |
| 5.5 扩展词典链接 .....                  | 84  |
| 5.6 其他指针 .....                    | 85  |
| 5.7 扩展的对象数据 .....                 | 86  |
| 5.8 总结 .....                      | 88  |
| <b>第6章 没有可视化特性的对象——第一部分</b> ..... | 89  |
| 6.1 符号表 .....                     | 89  |
| 6.2 层表 .....                      | 91  |
| 6.3 线型表 .....                     | 94  |
| 6.4 文本样式 .....                    | 96  |
| 6.5 尺寸样式 .....                    | 99  |
| 6.6 用户坐标系 .....                   | 103 |
| 6.7 命名视图 .....                    | 104 |
| 6.8 视窗表 .....                     | 106 |
| 6.9 注册应用程序 .....                  | 109 |
| 6.10 总结 .....                     | 109 |
| <b>第7章 没有可视化特性的对象——第二部分</b> ..... | 110 |
| 7.1 DXF中的对象 .....                 | 111 |
| 7.2 词典总结 .....                    | 111 |
| 7.3 缺省设置的词典 .....                 | 112 |
| 7.4 定位器对象 .....                   | 113 |
| 7.5 图形变量 .....                    | 113 |
| 7.6 实体组 .....                     | 114 |
| 7.7 实体序列索引 .....                  | 116 |
| 7.8 空间索引和层索引 .....                | 117 |
| 7.9 扩展记录 .....                    | 118 |
| 7.10 总结 .....                     | 121 |
| <b>第8章 组织图形实体</b> .....           | 122 |
| 8.1 块容器 .....                     | 122 |
| 8.2 块表 .....                      | 122 |
| 8.3 布局块 .....                     | 125 |
| 8.4 外部引用 .....                    | 125 |

|                               |            |
|-------------------------------|------------|
| 8.5 块开始和块结束对象 .....           | 126        |
| 8.6 块容器的DXF表达 .....           | 126        |
| 8.7 总结 .....                  | 130        |
| <b>第9章 实体的共有属性 .....</b>      | <b>131</b> |
| 9.1 AcDbEntity的DXF顺序 .....    | 131        |
| 9.2 Layer属性 .....             | 132        |
| 9.3 Color属性 .....             | 132        |
| 9.4 Linetype属性 .....          | 139        |
| 9.5 Linetype Scale属性 .....    | 140        |
| 9.6 Lineweight属性 .....        | 141        |
| 9.7 Plot Style属性 .....        | 142        |
| 9.8 Visibility属性 .....        | 143        |
| 9.9 Hyperlinks属性 .....        | 143        |
| 9.10 Paperspace Flag属性 .....  | 144        |
| 9.11 Interactive-only属性 ..... | 144        |
| 9.12 总结 .....                 | 145        |
| <b>第10章 曲线和其他简单图形实体 .....</b> | <b>146</b> |
| 10.1 曲线的属性 .....              | 146        |
| 10.2 无边界的直线 .....             | 148        |
| 10.3 单边界直线 .....              | 149        |
| 10.4 有限长的直线 .....             | 149        |
| 10.5 圆 .....                  | 151        |
| 10.6 圆弧 .....                 | 152        |
| 10.7 椭圆和椭圆弧 .....             | 153        |
| 10.8 样条线 .....                | 155        |
| 10.9 3D面 .....                | 157        |
| 10.10 总结 .....                | 158        |
| <b>第11章 注释对象 .....</b>        | <b>159</b> |
| 11.1 点对象 .....                | 159        |
| 11.2 形状符 .....                | 160        |
| 11.3 单行文本 .....               | 162        |
| 11.4 段落文本 .....               | 165        |
| 11.5 特征控制框 .....              | 169        |
| 11.6 半定制注释对象 .....            | 171        |
| 11.7 总结 .....                 | 172        |

|                              |     |
|------------------------------|-----|
| <b>第12章 尺寸标注对象</b> .....     | 174 |
| 12.1 尺寸和样式 .....             | 174 |
| 12.2 尺寸比例 .....              | 175 |
| 12.3 计算尺寸文本字符串 .....         | 176 |
| 12.4 添加公差尺寸 .....            | 177 |
| 12.5 替代尺寸 .....              | 178 |
| 12.6 尺寸文本 .....              | 179 |
| 12.7 尺寸线和箭头 .....            | 180 |
| 12.8 定位尺寸文本 .....            | 182 |
| 12.9 AcDbDimension属性 .....   | 185 |
| 12.10 对齐的线性尺寸 .....          | 187 |
| 12.11 旋转的线性尺寸 .....          | 187 |
| 12.12 角度尺寸：第一部分 .....        | 188 |
| 12.13 角度尺寸：第二部分 .....        | 188 |
| 12.14 直径尺寸 .....             | 189 |
| 12.15 半径尺寸 .....             | 190 |
| 12.16 坐标尺寸 .....             | 190 |
| 12.17 总结 .....               | 191 |
| <br><b>第13章 线和面的集合</b> ..... | 192 |
| 13.1 引出线 .....               | 192 |
| 13.2 小型多义线 .....             | 194 |
| 13.3 大型多义线 .....             | 198 |
| 13.4 三维多义线 .....             | 200 |
| 13.5 多边形网格 .....             | 201 |
| 13.6 多义面网格 .....             | 203 |
| 13.7 多线 .....                | 204 |
| 13.8 总结 .....                | 208 |
| <br><b>第14章 影线与填充</b> .....  | 209 |
| 14.1 影线的定义元素 .....           | 209 |
| 14.2 轮廓和关联性 .....            | 210 |
| 14.3 填充与模式 .....             | 214 |
| 14.4 AutoCAD的边界描图器 .....     | 216 |
| 14.5 实体和轨迹 .....             | 216 |
| 14.6 封闭区 .....               | 217 |
| 14.7 总结 .....                | 218 |

|                      |     |
|----------------------|-----|
| <b>第15章 位图图像</b>     | 219 |
| 15.1 图像定义            | 219 |
| 15.2 图像实体            | 220 |
| 15.3 图像定义到实体的链接      | 223 |
| 15.4 光栅图像的全局设置       | 223 |
| 15.5 总结              | 224 |
| <b>第16章 块插入和属位对象</b> | 225 |
| 16.1 块引用             | 225 |
| 16.2 过滤插入            | 228 |
| 16.3 属性对象            | 230 |
| 16.4 总结              | 233 |
| <b>第17章 布局对象</b>     | 234 |
| 17.1 绘图设置            | 234 |
| 17.2 布局对象            | 241 |
| 17.3 抽象视图            | 244 |
| 17.4 视窗专用的绘图辅助工具     | 248 |
| 17.5 视窗到模型空间         | 249 |
| 17.6 总结              | 254 |
| <b>第18章 定制对象</b>     | 255 |
| 18.1 基本问题            | 255 |
| 18.2 理想的定制对象         | 256 |
| 18.3 代理和复苏           | 257 |
| 18.4 DXF文件中的类        | 258 |
| 18.5 代理对象和实体         | 259 |
| 18.6 代理图形            | 261 |
| 18.7 总结              | 266 |
| <b>第19章 嵌入式对象</b>    | 267 |
| 19.1 体、固体和区域         | 268 |
| 19.2 ACIS数据          | 268 |
| 19.3 OLE数据           | 271 |
| 19.4 VBA工程           | 273 |
| 19.5 结束语             | 274 |
| <b>附录A 图形的预览图像</b>   | 275 |
| <b>附录B 其他资源</b>      | 277 |

## 第1章 了解AutoCAD数据库

### 内容简介

理解图形

使用对象和容器

识别对象

理解类和层级结构

对象属性

在开始浏览图形数据库中的特殊对象之前，需要了解图形数据的一般概念。本章给出了有关AutoCAD图形数据库内核的基本信息。

在本章的介绍过程中，会解释一些术语，虽然其中的许多术语，如容器或对象，读者已经从日常生活中了解到了，但当它们用于描述计算机数据时，就具有了特殊的含义，尤其在描述计算机图形时，其含义就更特殊了，所以有必要精确地定义这些术语。

**说明：**在有关计算机绘图或编程的书中，读者常会发现不同的作者采用不同的术语去描述同一件事情。

本书尽量采用AutoCAD文档中使用的术语。由于诸多原因，更主要是历史上的原因，本书的术语并不总是反映当前的业界标准，例如，在AutoCAD中，属性（attribute）是一个特殊的对象类型，而通常所说的属性指的是对象的特征。

本章将介绍计算机图形的基本结构、浏览储存在图形文件中的数据以及区分它们的方法。此外，还将介绍图形数据的内部结构和图形数据之间的引用方式。另外，还将讲解对象之间是如何被识别和链接的。

AutoCAD中一个重要的问题，即不同软件间如何进行图形转换，这也是本章的一个主题。一个图形数据库是一个完整的实体吗？换句话说，一个AutoCAD图形软件中的信息是否包含了所有在其他软件中构造同样图形的必要数据。此外，将遇到的另一个问题是：AutoCAD图形文件的文档（如本书中的）是否是完整无缺吗？本章将给出这些问题的答案。

### 1.1 图形中有什么？

在回答这个问题之前，让我们从最基本的问题开始：什么是图形？在早于CAD时代，图形就是用油墨在纸上绘制出线条，从而形成真实物体或想象物体的可视化显示。

计算机创造的“图形”仍然描述真实的或假想的物体，但它是通过对象的数学模型实现的。一辆汽车的CAD模型包括多个CAD模型，有车轮、轮胎和制动器。一个轮胎有一个几何描述，同时也具有制造特性，如材料和附加的功能信息，例如最大速度或充气的程度。

虽然人们依然可以把CAD模型输出到纸上，但这仅仅是CAD模型诸多用途中的一种。此外，可能会有该模型的多种不同视觉再现。输出到纸上的可能是模型的前视图或侧视图，可能显示出了制动器系统的细节，或是演示在交通堵塞时色彩丰富的汽车模型的样子。

### 1.1.1 AutoCAD图形或AutoCAD模型

可能读者会认为这些特征的CAD模型并不存在。在所有AutoCAD图形文件中, 99.9%仅仅包含了线和弧线, 与过去呈现在纸上的图形没有什么不同。

虽然对于大多数图形文件来说, 你是正确的, 但是绘图师利用图形去绘制轮胎时并没有告诉你纸面上的线条和真实轮胎之间的关系, 而AutoCAD图形数据库能够包含轮胎和制动器的数学模型, 正如它包含线条一样。

如果看得更细致些, 读者会发现CAD轮胎与CAD线条之间没有任何区别, 两者都是真实或抽象物体的抽象模型。CAD轮胎是真实轮胎的一个抽象模型, CAD线条通常是真实物体边界的抽象模型。

## 1.2 对象和容器

AutoCAD图形不过是任意抽象模型或物体的容器, 这种对象的集合被称为图形数据库。

图形数据库在许多方面类似于读者已知或已经接触过的其他类型的数据库。正如用于组织地址簿或CD集的数据库, 图形数据库中包含由行(记录)和列(字段)组成的表。

图形数据库中的记录或对象可以是普通的直线或弧线, 但也同样可能, 并且将越来越可能是汽车和轮胎、墙壁和门、书和作者。

AutoCAD的图形数据库几乎可以容纳任何对象。当某对象遵循一定的准则时, 它就可以显示出来, 例如, 一个AutoCAD对象必须具有颜色属性。

AutoCAD图形数据库的创建者可以自由地往库中添加任意对象, 但把对象添加在何处却不是随意的。每一个对象必须放在形成图形数据库的大量容器中的唯一的一个容器内。

### 1.2.1 符号表

不具有视觉描述, 但属于某种预定对象类型(类)的对象必须收录进符号表容器中。符号表容器由9个数据库表组成。下面列出了其中的八个表, 第6章中将给出详细描述。

- 尺寸样式表(AcDbDimStyleTable, DIMSTYLE)定义了一系列用于标注尺寸的属性, 如箭头类型或文本对齐。
- 层表(AcDbLayerTable, LAYER)是为层对象而设计的, 用于组织图形实体。
- 线型表(AcDbLinetypeTable, LTYPE)包含线条的种类, 用于规定图形实体显示或绘制的方式。
- 注册的应用程序表(AcDbRegAppTable, APPID)包含储存扩展对象数据的第三方应用程序的名称。
- 文本样式表(AcDbTextStyleTable, STYLE)定义了用于文本对象的一系列属性, 例如字体和字符取向。
- 用户坐标系表(AcDbUCSTable, UCS)列出了用户可能激活并使用的笛卡尔坐标系。
- 视图表(AcDbViewportTable, VPORT)规定了平铺视窗的命名集。
- 视图表(AcDbViewTable, VIEW)包含图形视图的说明。用户可以调用视图显示或打印图形。

除了上述符号表外，还有特殊的块表（AcDbBlockTable, BLOCK\_RECORD）。块表由块容器和所有图形实体组成，有关块表的描述见第8章。

与dBase或Microsoft Access的数据库表一样，每一个符号表只能包含一个特殊类型的对象。符号表内的对象被称为符号表记录（Symbol table records）。符号表中的符号表记录总是同类的对象，例如，层表中的所有记录是层记录对象；相反，层记录对象不能在层表以外的地方出现。

一个符号表内的所有记录都具有“name”字段，作为表记录的标识符或关键字。这类类似于数据库表中的主关键字。关键字的值是一个字符串并且在一个符号表内必须唯一，以防止数据重复并可维持数据的完整性。这种方法与公司雇员记录文件中的社会安全号类似。同一个符号表中的关键字段标识符是唯一的，因此，层表与线型表能够容纳关键字段是STANDARD的记录。但是，同一个符号表中的两个记录不能采用相同的关键字段值。这种单记录—单关键字准则有一个例外：AcDbViewportTable允许具有重复关键字，在第6章中解释了这种特殊用法。

一个符号表关键字内单个字符的大小写无关，因此，符号表可以包含Standard关键字或STANDARD关键字，而不是两者都有。

除了关键字段，符号表中的记录还有其他的一些字段，如层表记录包含储存与该层相关的颜色、线型、锁定状态以及其他属性的字段（参见第6章中有关介绍）。

## 1.2.2 理解“词典”

符号表或任何其他采用固定记录的数据库表都是不可改变的。符号表中的所有记录必须是同类的，例如，不能将线型记录与层记录放在同一个表中。

然而，在图形数据库中我们不得不处理大量不同种类的对象或记录，我们必须能够把弧线对象和直线对象放入共同的容器中。因此，比起符号表来说，图形数据库包含了更加灵活的对象容器，这些容器被称为“词典”。

每一个词典都是任意对象的一个容器，词典中的每一个对象被称为一个记录。与符号表有所不同，一个词典可以包含不同种类的对象，也可以包含其他的词典。把完备的词典放入另一个词典中，就可以创建词典的树状层级结构。树的底部是“根”词典，“根”词典不包含于任何其他的词典。在一个AutoCAD图形数据库中，每个词典若不是包含在“根”词典中（直接地或通过一系列的中间词典），就是被链接到特殊的图形实体上，如圆。

与符号表一样，每个词典中的每个记录都有一个关键字，词典中的关键字是唯一的。但是与符号表记录有所不同的是，这里的字符是大小写相关的。没有视觉描述特性以及不是字符表记录的所有对象都必须录入词典。词典或符号表的外部没有不具有视觉描述特性的对象。

读者可能会发出疑问，为什么要区分符号表和词典呢？对于这一点，除了符号表不应该仅仅是另一个词典外，没有任何技术上的原因。符号表用于图形数据库比词典更长久。一段时间以前，Autodesk只是决定把这两者分开，然而，直到现在它们之间仍然保持着这种差别。

## 1.2.3 将实体组合成块

现在读者已经知道没有视觉特性的对象到那里去了，但那些具有图形描述的对象被放到那里了呢？这些具有图形描述的对象被称为实体，它们常常是大多数人所感兴趣的。