

机械CAD 开发技术

林昌华 黄霞 杨岩 编著



国防工业出版社

National Defense Industry Press

机械 CAD 开发技术

林昌华 黄霞 杨岩 编著

国防工业出版社

· 北京 ·



内 容 简 介

本书介绍 AutoCAD 环境下进行二次开发的主要方法与关键技术,内容包括形、线型与图案、菜单与工具栏的定制与开发,AutoCAD 的 Visual LISP 集成开发环境和编写 Auto LISP 程序的技巧,对话框设计和开发技术在机械工程中的应用,并针对机械设计中的人机交互、设计计算、数据处理、参数化绘图列举了大量的应用实例。

本书适用于已掌握 AutoCAD 的基本操作,需深入了解 AutoCAD 以及对 AutoCAD 进行二次开发的工程技术人员,可作为大专院校工程类各专业的教材或教学参考书,也适宜作为广大工程制图技术员和机械工程师学习的教材及软件培训班的培训教材。

图书在版编目(CIP)数据

机械 CAD 开发技术 / 林昌华, 黄霞, 杨岩编著. —北京: 国防工业出版社, 2013. 8

ISBN 978-7-118-09108-3

I. ①机… II. ①林… ②黄… ③杨… III. ①机械设计—AutoCAD 软件—软件开发 IV. ①TH122

中国版本图书馆 CIP 数据核字(2013)第 208062 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 100048)

腾飞印务有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 12 $\frac{3}{4}$ 字数 288 千字

2013 年 8 月第 1 版第 1 次印刷 印数 1—2000 册 定价 35.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)88540777

发行邮购: (010)88540776

发行传真: (010)88540755

发行业务: (010)88540717

前 言

随着计算机应用技术的飞速发展,机械制造业数字化设计与制造已成为新形势下的必然趋势。尤其是在机械设计领域,CAD 技术的应用和推广,使得传统的机械设计手段和方法发生了根本的变化,利用 CAD 等现代设计方法来完成机械设计任务,已是各企业设计部门和机械设计人员的迫切要求。

CAD 技术的内容很广泛,包括数值计算与评价技术、参数化绘图与建模技术、信息与知识的处理技术以及可视化与仿真技术等。目前有关 CAD 技术方面的书籍很多,但大多限于对 CAD 技术的基本概念、原理、方法和相关应用软件功能的介绍,内容多而广,实际可操作性不强,特别是对于初学者很难上手。本书定位于机械 CAD 开发技术的基本应用和训练上,主要掌握机械零件设计过程中的人机交互、设计计算、数据处理、参数化绘图以及菜单管理,所涉及的软件为工程领域使用最为广泛的 AutoCAD,所涉及的编程语言为 AutoCAD 内嵌的 Auto LISP 语言,适用于具有一定机械产品设计知识并熟悉 AutoCAD 绘图软件的工程技术人员和机械类本专科学生。

AutoCAD 绘图软件具有丰富的绘图功能、强大的编辑功能和良好的用户界面,受到广大工程技术人员和大中专院校师生的普遍欢迎,使得它在机械、电子、船舶、建筑和服装等多个领域得到了极为广泛的应用。特别是该软件提供的各种编程工具和接口技术,为用户开发应用系统创造了十分便利的条件。内嵌的 Auto LISP 语言简单易学,程序编写和编辑调试快捷方便,初学者容易上手。更可贵的是 Auto LISP 语言所编程序的运行稳定性和兼容性极好,不用担心因版本升级而崩溃的问题,这是本书选用 AutoCAD 绘图软件作为机械 CAD 开发技术基础平台的主要原因。

本书内容分为 10 个章节:第 1 章介绍机械 CAD 技术的基本概念和基于 AutoCAD 系统二次开发的基本内容等;第 2、3 章介绍用户定制的一些内容,这些内容包括各种线型、图案以及图形符号的扩充和菜单技术;第 4、5、6 章比较详细地介绍了 Visual LISP 和 Auto LISP 语言开发工具、参数化绘图以及高级语言的接口技术等;第 7 章介绍对话框技术;第 8 章介绍机械 CAD 中的数据处理方法;第 9、10 章介绍机械 CAD 开发技术在机械产品设计中的应用。

与同类书相比,本书主要以结合实例的方式,由浅入深地进行系统的阐述。在内容安

排上做到了循序渐进,图文并茂,旨在突出易学实用的特点。所举实例内容简洁,通俗易懂,便于自学,以便初学者尽快入门。开发技术应用部分力求内容具体,方法可靠,便于读者系统学习和理解。

本书由林昌华、黄霞、杨岩执笔编写,其中第1、2、3章由黄霞编写,第4、5、6章由杨岩编写,林昌华则担当了其余章节的编写和全书的统校工作。

由于作者水平有限,书中难免存在不足与疏漏之处,恳请广大读者不吝赐教,批评指正。

编 者

目 录

第 1 章 绪论	1	第 3 章 菜单及工具栏的开发	25
1.1 机械 CAD 技术概述	1	3.1 菜单及菜单文件	25
1.2 为什么要进行 CAD 软件的 二次开发	1	3.1.1 菜单文件的类型	26
1.3 AutoCAD 二次开发工具	2	3.1.2 菜单文件的调用过程	26
1.4 二次开发的主要内容	4	3.1.3 菜单文件的格式	27
练习题	4	3.1.4 菜单项的定义	28
第 2 章 库文件及其开发	5	3.1.5 菜单调用命令和 特殊字符	28
2.1 形文件的开发	5	3.1.6 菜单项定义举例	29
2.1.1 形的定义	5	3.2 菜单开发的一般方法	31
2.1.2 编译和加载形文件	8	3.2.1 利用局部菜单来建立 用户菜单	31
2.1.3 插入形	8	3.2.2 使用“自定义用户界面” 对话框来定义用户菜单	33
2.2 线型的开发	9	3.3 工具栏的创建	34
2.2.1 简单线型的定义	10	3.3.1 创建工具栏的步骤	35
2.2.2 复合线型的定义	11	3.3.2 编辑或创建按钮图像的 步骤	36
2.2.3 线型的开发方法	13	3.4 创建图像控件菜单	37
2.3 图案的开发	15	3.4.1 创建图像控件菜单并指定 图像控件幻灯片的步骤	37
2.3.1 图案的定义	15	3.4.2 激活图像控件菜单	38
2.3.2 图案文件及图案库的 创建	16	练习题	39
2.4 图块及其图形库的创建	18	第 4 章 Auto LISP 基础及 Visual LISP 开发环境	40
2.4.1 图块概述	18	4.1 Auto LISP 语言基础知识	40
2.4.2 图块的创建	18	4.1.1 Auto LISP 语言特点	40
2.4.3 图块的插入	21	4.1.2 Auto LISP 程序结构 形式	40
2.5 幻灯片及幻灯片库的创建	22		
2.5.1 幻灯片的创建与显示	22		
2.5.2 幻灯片库的建立	23		
练习题	24		

4.1.3	Auto LISP 的求值过程	42	6.1.1	Command 函数	77
4.1.4	Auto LISP 数据类型	43	6.1.2	实用几何函数	78
4.2	Auto LISP 程序文件的 加载及运行	45	6.1.3	对象捕捉函数	80
4.2.1	Auto LISP 程序的装入	46	6.2	交互输入函数	80
4.2.2	Auto LISP 程序的运行	46	6.2.1	GET 族输入函数	80
4.2.3	Auto LISP 程序的 自动装入	47	6.2.2	输入控制函数 initget	82
4.3	Visual LISP 开发环境	48	6.2.3	其他输入函数	82
4.3.1	Visual LISP 集成开发环境 (IDE)的特点	48	6.3	输出函数	83
4.3.2	Visual LISP 集成开发 环境窗口	48	6.3.1	用于屏幕和文件输出的 函数	84
4.3.3	Visual LISP 集成开发 环境的应用	49	6.3.2	只用于屏幕输出的 函数	84
4.3.4	编译及加载	55	6.4	文件操作函数	85
	练习题	58	6.4.1	打开文件函数 open	85
第 5 章	Auto LISP 基本函数	59	6.4.2	关闭文件函数 close	85
5.1	数值函数	59	6.4.3	文件操作函数综合 举例	86
5.1.1	基本算数函数	59	6.5	屏幕控制函数	87
5.1.2	三角函数	61	6.6	图形数据库操作函数	88
5.1.3	数据类型转换函数	62	6.6.1	选择集操作函数	88
5.2	赋值函数与求值函数	64	6.6.2	处理图元对象	91
5.3	表处理函数	65		练习题	95
5.4	字符串处理函数	67	第 7 章	人机交互界面设计	96
5.5	逻辑运算函数	67	7.1	对话框的控件	96
5.6	控制结构函数	71	7.1.1	按钮类控件	97
5.6.1	条件分支函数	71	7.1.2	选择类控件	97
5.6.2	循环结构函数	73	7.1.3	编辑类控件	98
5.6.3	顺序控制函数	74	7.1.4	列表类控件	99
5.7	函数的递归定义	75	7.1.5	框架类控件	99
	练习题	76	7.1.6	装饰说明类控件	100
第 6 章	Auto LISP 与 AutoCAD 的 通信	77	7.2	DCL 控件的属性	101
6.1	Auto LISP 的绘图功能	77	7.2.1	控件属性的类型	101
			7.2.2	预定义属性概览	102
			7.2.3	常用的四种属性	103
			7.3	设计对话框的步骤及原则	105
			7.3.1	设计对话框的一般	

步骤	105
7.3.2 图形用户界面(GUI)	
的设计原则	105
7.3.3 预定义控件和控件	
组设计原则	106
7.3.4 设计对话框时应注意的	
几个问题	108
7.4 DCL 文件的编写方法	108
7.4.1 DCL 语法结构	109
7.4.2 对话框设计举例	109
7.5 对话框驱动程序设计	111
7.5.1 对话框驱动程序结构	111
7.5.2 对话框处理函数调用	
顺序	112
7.5.3 Auto LISP 对话框处理	
函数	113
7.5.4 对话框有效时不允	
许调用的函数	115
7.5.5 操作表达式和回调	
函数	116
7.6 对话框及其驱动程序设	
计应用范例	118
练习题	122
第 8 章 机械 CAD 中的数据处理	124
8.1 机械 CAD 数据类型及	
处理方法	124
8.1.1 数据程序化方式	124
8.1.2 数据文件方式	124
8.1.3 数据库方式	124
8.2 数表的程序化	125
8.2.1 一维数表程序化	125
8.2.2 一维数表的线性插值	126
8.2.3 一维数表的非线性	
插值	128
8.2.4 二维数表程序化	130
8.3 线图程序化	133

8.3.1 直线线图处理	133
8.3.2 曲线线图处理	135
8.4 数据文件化	139
8.4.1 一维数据文件建立格式与	
检索范例	139
8.4.2 二维数据文件建立格式与	
检索范例	141
8.5 数据库简介	143
练习题	144
第 9 章 机械设计编程及应用	146
9.1 机械 CAD 图形环境的	
设置	146
9.1.1 程序中设置图层、颜色、	
线型、线宽	146
9.1.2 程序中设置文字样式	146
9.1.3 程序中设置标注样式	146
9.2 机械设计计算程序实例	148
9.2.1 编程计算任意压力角对应	
的渐开线函数	148
9.2.2 根据运动规律,编制程序	
计算凸轮的基圆半径	148
9.2.3 渐开线标准齿轮传动	
重合度计算	149
9.2.4 齿轮传动变位系数和	
的计算	149
9.2.5 齿轮传动设计计算	
程序实例	150
9.3 机械设计参数化绘图	
程序实例	154
9.3.1 参数化绘制轴段图形	154
9.3.2 圆螺母参数化绘图	155
9.3.3 摆线轮工作廓线参	
数化绘图	157
9.3.4 渐开线直齿圆柱齿轮廓	
线参数化绘图	160

练习题	163
第 10 章 普通 V 带传动 CAD	
开发实例	164
10.1 普通 V 带传动设计计算	164
10.1.1 V 带传动设计计算	
步骤	164
10.1.2 V 带传动程序设计	
流程	165
10.1.3 主要程序变量	165
10.1.4 数据处理与检索	
说明	166
10.2 普通 V 带传动设计计算的	
程序设计	167
10.2.1 初始条件输入及其对	
话框界面设计	167
10.2.2 输入界面驱动程序	
设计	168

10.2.3 设计计算对话框界面	
设计	169
10.2.4 计算界面驱动及计	
算程序设计	173
10.3 普通 V 带传动带轮参数化	
绘图程序设计	182
10.3.1 实心式带轮参数化绘图	
程序设计	182
10.3.2 腹板式带轮参数化绘图	
程序设计	184
10.3.3 孔板式带轮参数化绘图	
程序设计	186
10.3.4 轮辐式带轮参数化绘图	
程序设计	188
10.3.5 参数化绘图对话框及主	
程序设计	192
10.4 用户管理菜单设计	194
练习题	194

第1章 绪 论

1.1 机械 CAD 技术概述

机械设计是根据使用要求确定产品的工作原理、运行方式、动力的传递、所用材料、结构形状以及技术要求等要素,并转化为图样和设计文件等具体的描述,它包含方案决策、概念设计、总体设计、结构设计、性能分析、装配过程仿真、工作过程仿真、产品信息数据管理等多方面的内容,是产品设计、制造、销售和使用整个生命周期中的首要环节。机械设计的过程实际上是“设计—评价—再设计”的反复循环和不断优化的过程。传统的机械设计工作量大、周期长。在产品更新换代速度越来越快的市场经济条件下,缩短产品的开发周期、提高产品质量、降低成本,已是各行各业增强市场竞争力和促进自身发展的迫切要求之一。因此,实现机械设计的快捷化、缩短设计周期、提高设计质量、降低产品成本就成为机械设计发展的当务之急。正是在这样的背景下,产生了计算机辅助机械设计,即机械 CAD(Computer Aided Design)。

机械 CAD 是指将计算机技术运用到机械设计的整个过程中,利用计算机硬件、软件系统辅助设计人员对产品和工程进行分析计算、几何建模、仿真与试验、优化设计、绘制图形、工程数据库的管理、生成技术文件等的方法和技术。机械 CAD 技术给传统的机械设计方法和思路带来了极大的冲击,可以说是机械设计的一次重大革命。

计算机具有快速高效的计算处理能力、大的信息存储量和图形显示仿真能力,在产品的设计过程中既可以减轻设计人员的脑力和体力劳动,又可以为设计人员改进优化设计方法、设计参数提供便捷有效的技术手段。

机械 CAD 除了能完成传统的设计计算、结构设计和图样绘制外,也能轻松完成一些传统设计不能实现的工作,如设计参数的优化设计、零部件结构的有限元分析、机器以及机构的三维实体建模装配和运动模拟仿真等。现有的一些机械 CAD 系统的功能不仅仅局限于机械产品的设计,同时还包括系统与其他技术的结合,如与数据库技术结合,可将产品的信息进行存储管理;大多数 CAD 系统都提供了二次开发的平台,为用户扩大系统的功能提供了基础条件。目前,机械 CAD 技术已经广泛地应用于机械设计和制造的各个环节中,为缩短产品的开发生产周期、发展制造业信息化和提高企业的竞争力奠定了基础。

1.2 为什么要进行 CAD 软件的二次开发

随着计算机在机械工程领域的广泛应用,CAD 软件的需求量与日俱增,尤其是适合于机械零部件设计、分析和制造的 CAD/CAE/CAM 专用软件。目前市场上有许多商品化的 CAD 软件,如 AutoCAD、UG、CATIA、Pro/E 等。在这些软件中,除少数是针对本

行业的使用条件开发的专用应用软件外,绝大多数都是由软件公司开发的通用性 CAD 支撑软件,虽然这些软件各种功能强大、通用性极强,但其专业性较差。在复杂 CAD 问题或特殊用途的设计中,完全依据原有软件的功能往往难以解决问题,因此,根据本单位的实际需求或客户的特殊用途进行软件的用户化定制和二次开发,扩充其实用的功能,往往能够大大提高企业的生产效率和技术水平。

1.3 AutoCAD 二次开发工具

当前 AutoCAD 所采用的二次开发工具,即编程接口主要有 ActiveX Automation、VBA (Visual Basic for Applications)、Auto LISP 和 Visual LISP、ObjectARX 和 .NET。每种开发工具各有其特点,用户使用何种开发工具或接口类型主要由应用程序的需要和编程经验决定。

1. ActiveX Automation

ActiveX Automation 是 Microsoft 基于 COM(零部件对象模型)体系结构开发的一项技术。用户可以用它来自定义 AutoCAD,与其他应用程序共享图形数据以及自动完成任务。

用户可以从用作 Automation 控制程序的任意应用程序中创建和操作 AutoCAD 对象。因此,Automation 使编制跨应用程序执行的宏成为可能,而 Auto LISP 中就没有这种功能。

AutoCAD 通过 Automation 显示 AutoCAD 对象模型描述的可编程对象。这些可编程对象可由其他应用程序创建、编辑和操作。可以访问 AutoCAD 对象模型的应用程序是 Automation 控制程序,使用 Automation 操作另一个应用程序的最常用工具是 Visual Basic for Applications (VBA)。在很多 Microsoft Office 应用程序中,VBA 都是作为其中的一个部件。用户可以使用这些应用程序或其他 Automation 控制程序(例如 Visual Basic、.NET 和 Delphi)来运行 AutoCAD。

在 AutoCAD 中使用 ActiveX 接口具有以下两个优点:

(1) 可以在多种编程环境中编程访问 AutoCAD 图形。在 ActiveX Automation 出现之前,开发者只能用 Auto LISP 或 C++ 接口访问 AutoCAD 图形。

(2) 更易于与其他 Windows 应用程序(例如 Microsoft Excel 和 Microsoft Word)共享数据。

2. AutoCAD VBA

Microsoft Visual Basic for Applications (VBA) 是一个基于对象的编程环境,能提供丰富的开发功能。VBA 和 VB (Visual Basic) 的主要区别在于:VBA 与 AutoCAD 在同一进程空间运行,提供的是具有 AutoCAD 智能的、非常快速的编程环境。

用 AutoCAD VBA 开发将通过 AutoCAD ActiveX Automation 接口向 AutoCAD 发送消息。AutoCAD VBA 允许 Visual Basic 环境与 AutoCAD 同时运行,并通过 ActiveX Automation 接口提供 AutoCAD 的编程控制。这样就把 AutoCAD、ActiveX Automation 和 VBA 链接在一起,提供了一个功能非常强大的接口。它不仅能控制 AutoCAD 对象,也能向其他应用程序发送数据或从中检索数据。

将 VBA 集成到 AutoCAD,为自定义 AutoCAD 提供了便于使用的可视工具。例如,用户可以创建一个应用程序,用于自动提取属性信息,把结果直接插入 Excel 电子数据表并执行所需的任意数据转换。

AutoCAD 中的 VBA 编程由三个要素定义。第一个是 AutoCAD 本身,它提供了全面的对象,包括 AutoCAD 图元、数据和命令。AutoCAD 是一个具有多层次接口的开放式应用程序。要有效地使用 VBA,必须非常熟悉 AutoCAD 的编程特性。但是,VBA 的基于对象的方法和 Auto LISP 大不一样。

第二个要素是 AutoCAD ActiveX Automation 接口,它与 AutoCAD 对象进行消息传递(通信)。用 VBA 编程需要对 ActiveX Automation 有基本的了解。

第三个要素是 VBA 本身。它有自己的一套对象、关键字和常量等的集合,用于提供程序流、控制、调试和执行。AutoCAD VBA 中包括 Microsoft 关于 VBA 的扩展帮助系统。

AutoCAD ActiveX/VBA 接口的优点多于其他 AutoCAD API 环境的优点:

(1) 速度 用 VBA 在进程内运行,ActiveX 应用程序的速度比 Auto LISP 应用程序快。

(2) 易用 编程语言和开发环境易于使用并且随 AutoCAD 一起安装。

(3) Windows 交互操作性 ActiveX 和 VBA 是为与其他 Windows 应用程序一起使用而设计的,为应用程序之间的信息交流提供了绝佳的途径。

(4) 快速生成原型 VBA 的快速接口开发为原型应用程序提供了优良的环境,即使最终使用另一种语言开发那些应用程序。

3. Auto LISP 和 Visual LISP

Auto LISP 基于简单易学而又功能强大的 LISP 编程语言。由于 AutoCAD 具有内置 LISP 解释器,因此用户可以在命令提示下输入 Auto LISP 代码,或从外部文件加载 Auto LISP 代码。Visual LISP (VLISP) 是为加速 Auto LISP 程序开发而设计的软件工具。

Auto LISP 通过 Visual LISP (VLISP) 进一步得到增强,VLISP 提供了一个集成开发环境 (IDE),其中包含编译器、调试器和其他提高生产效率的开发工具。VLISP 添加了更多的功能,并对语言进行了扩展以与使用 ActiveX 的对象进行交互。VLISP 也允许 Auto LISP 通过对象反应器对事件进行响应。

Auto LISP 应用程序可以通过多种方式与 AutoCAD 交互。这些程序能够提示用户输入、直接访问内置 AutoCAD 命令,以及修改或创建图形数据库中的对象。通过创建 Auto LISP 程序,可以向 AutoCAD 添加专用命令。实际上,某些标准 AutoCAD 命令是 Auto LISP 应用程序。

Visual LISP 为 Auto LISP 应用程序提供三种文件格式选项:

(1) 读取 LSP 文件 (.lsp) ——包含 Auto LISP 程序代码的 ASCII 文本文件。

(2) 读取 FAS 文件 (.fas) ——单个 LSP 程序文件的二进制编译版本。

(3) 读取 VLX 文件 (.vlx) ——一个或多个 LSP 文件和/或对话框控制语言 (DCL) 文件的编译集。

由于 AutoCAD 能够直接读取 Auto LISP 代码,因此无需编译。Visual LISP 提供了一个集成开发环境,用户可以选择进行试验:在命令提示下输入代码后可立即看到结果。

这使 Auto LISP 语言容易试验,而不管用户的编程经验如何。

4. ObjectARX

ObjectARX 技术为设计软件应用程序提供了共享智能化对象数据的基础。用户可以运行第三方 ObjectARX 应用程序,也可以自己开发。

ObjectARX®(AutoCAD 运行时扩展)是开发 AutoCAD 应用程序的编译语言编程环境。ObjectARX 编程环境包括许多动态链接库(DLL),它们和 AutoCAD 运行在同一地址空间,并直接操作 AutoCAD 内核数据结构和代码。这些库利用 AutoCAD 的开放式体系结构,提供对 AutoCAD 数据库结构、图形系统和 AutoCAD 几何图形引擎的直接访问权限,以扩展 AutoCAD 在运行时的类和功能。另外,也可以使用 DLL 来创建新命令,这些新命令的操作方式与 AutoCAD 的基本命令操作方式相同。

ObjectARX 库可以与其他 AutoCAD 编程接口(例如 Auto LISP 或 VBA)结合使用,从而启用跨 API 集成。

5. NET

通过 Microsoft .NET Framework,用户可以使用编程语言(如 VB.NET 和 C#)创建与 AutoCAD 进行互操作的应用程序。

.NET Framework 是由 Microsoft 开发的中性语言编程环境。除了运行时环境之外,Framework 还提供了便于开发基于 Windows 和基于 Web 的、安全的可互操作的应用程序类库。.NET 是微软新推出的开发平台,具有众多优点。基于 .NET 平台对 AutoCAD 进行二次开发,可充分利用 .NET 的各种优势,在保证功能强大的前提下大大提高开发速度。

AutoCAD 的强大生命力在于它的通用性、多种工业标准和开放的体系结构。其通用性使得它在机械、电子、航空、船舶、建筑、服装等领域得到了极为广泛的应用。但是,不同的行业标准使得各领域在使用 AutoCAD 的过程中均需根据自身特点进行定制或开发。AutoCAD 的各种开发工具可满足广大用户的需求。

本书主要介绍 AutoCAD 的定制及如何利用 Auto LISP 和 Visual LISP 进行机械 CAD 二次开发的方法技术。

1.4 二次开发的主要内容

- (1) 定制开发专用库文件,以减少设计绘图时的重复性工作。
- (2) 编写各种可完成用户自定义功能的函数和程序,以实现快捷的科学计算、数据处理、数据传递和参数化绘图。
- (3) 定制符合用户要求的各种菜单及菜单文件,以方便管理和调用。
- (4) 开发符合用户定义的对话框和界面,以实现人机交互。

练 习 题

1. 目前你了解的机械 CAD 支撑软件主要有哪些? 各具有何种功能特点?
2. 机械 CAD 可以辅助设计人员完成哪些工作?
3. 机械 CAD 软件开发包含哪些内容?

第2章 库文件及其开发

AutoCAD 最显著的特点是开放式结构,其提供的二次开发工具主要分为两大类:内部定制工具和二次开发工具。所谓定制就是按照 AutoCAD 提供的方法和文件格式,根据用户的具体要求,通过文字编辑系统所支持的 ASCII 码文件建立新的功能库,如设置符合企业标准和工作需要的图形符号库、字体、线型或填充图案库等。

2.1 形文件的开发

形是一种能用直线、圆弧和圆来定义的特殊实体,用户可以利用形的功能定义各种简单的图形符号,建立符号库。形可方便地被插入图形中,并按需要指定比例系数及旋转角度,以获得不同的位置和大小。

在 AutoCAD 中,形从定义到插入图形符号,需经以下几个步骤:

- 按规定格式进行定义形,即用文本编辑器或字处理器建立形文件;
- 对已建立的形文件进行编译,生成“.SHX”文件;
- 加载编译后的形;
- 把形插入所需的图形环境中。

这一过程可用表 2-1 来表示。

表 2-1 形从定义到调用的过程

内 容	定义形文件	编译形文件	装入形文件	调用形文件
工 具	文本编辑程序	COMPILE 命令	LOAD 命令	SHAPE 命令
结 果	得到形的源文件(*.SHP)	得到形的目标文件(*.SHX)	形文件中定义的形可被加载	在当前作业中插入形

调用一个形与调用一个块(BLOCK)在形式上有些相似,但 AutoCAD 系统对二者的定义是完全不同的。块比形易学、易用且通用性强。块在本次绘图作业内部定义,形由形文件在作业外部支持。调用一个形只是将形码(名)和变换参数(插入点、比例和转角)记录于图形文件。组成形的矢量仅读取到缓冲区,并存入图形文件。而块无论是否被调用,在定义时就占用了图形文件的一些存储空间。因此,在进行二次开发时,一般将常用的符号、字体等定义为形。这样,既可显著地节省存储空间,也可为多个图形文件所共用。

2.1.1 形的定义

定义形的文件称为形文件,它是一种扩展名为“.SHP”的 ASCII 文本文件;它的每行字符数不超过 128 个,否则会导致编译失败。AutoCAD 将忽略所有空行及分号右边的内容。建立或修改形文件可使用文本编辑器或字处理器。

形的定义具有一定的格式和规定,用户必须严格遵守。每个形的定义由一个标题行和若干描述行组成。

• 标题行

标题行以“*”开始,后面为说明形的编号、大小及名称,格式如下:

* Shapenumber,Defbytes,Shapename

其中:星号“*”为标识符,不能省略。

“Shapenumber”为形的编号,每个形定义必须有一个编号,且不能相同,其编号范围在 1~255 之间。

“Defbytes”表示形定义描述行所具有的字节数,其中包括形的描述结束符“0”所占用的字节,每个形的定义字节数不得超过 2000。

“Shapename”是形的名称,每个形必须有一形名,且这个形名必须大写,否则会被忽略。

• 描述行

标题行之后为描述行,它是用数字或字母描述形中所包含的线段、弧的大小及方向。数字和字母为一个一个字节,字节之间用逗号分开。描述行以“0”结束。每一个形的描述字节数不能超过 2000 个,包括结束符“0”。

• 描述码

形的定义有两种描述编码:一种是矢量长度与方向编码,另一种是特殊码。

矢量长度与方向编码:

描述一个直线矢量的长度和方向需用 3 个字符:第一个必须是 0,它表示后边两个字符是十六进制数,第二个字符代表矢量的长度,有效值为 1~F(1~15 个单位长),第三个字符代表矢量的方向,方向编码见图 2-1。

例 1 定义一个如图 2-2 所示的形,定义的代码如下:

* 210,6,DBOX

018,014,010,01C,016,0

定义中,标题行为“* 210,6,DBOX”,它说明该形的编号是 210,形定义描述行共 6 个字节,形的名称是 DBOX。在描述行“018,014,010,01C,016,0”中,共有 6 个字节,前 5 个十六进制数分别用来绘制形中的 5 条线,0 是结束符。

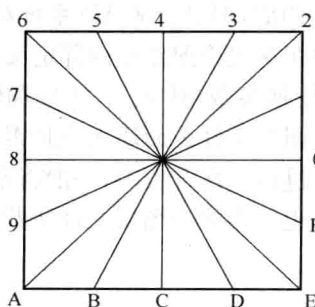


图 2-1 方向码

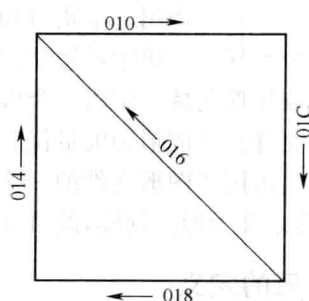


图 2-2 形图例 1

例 2 定义一个如图 2-3 所示的四角星图形,定义的代码如下:

* 121,9,SJX

017,019,013,015,01F,011,01B,01D,0

由两个形的图例可以看到,用方向码定义形,其矢量长度是沿着 X 或 Y 方向度量的,即沿斜线方向的 1 个单位长是指其在 X 或 Y 方向的投影长度为 1,而不是斜线的长度为 1。

由图 2-1 方向码可知,矢量长度与方向码有一定的局限性,因为它只有十六个方向,最大长度是 15,而且只能绘直线段,所绘的形总是连续性的。为弥补上述不足,AutoCAD 还提供了特殊码。

• 特殊码

为定义不同对象,如直线、圆弧以及描述各种状态,如抬笔、落笔和形定义结束等,AutoCAD 提供了一些特殊描述码。这些码是专用的,可以是十六进制,也可以是十进制。这些特殊码的含义和功能见表 2-2。

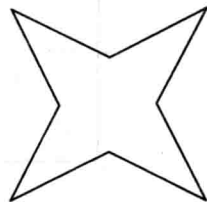


图 2-3 形图例 2

表 2-2 特殊码及其功能

十六进制	十进制	含 义 及 功 能
000	0	形定义的结束码
001	1	激活绘图模式(落笔),默认状态下该值有效
002	2	关闭绘图模式(抬笔)
003	3	用下一个字节除矢量长度(缩小)
004	4	用下一个字节乘矢量长度(放大)
005	5	将当前位置压入栈,保存当前的坐标位置,以便再恢复到此位置。位置栈只能存放 4 个位置,且在形定义结束前必须弹出栈中所有内容
006	6	将栈中内容弹出到当前位置,即恢复由 005 码保存的原先位置
007	7	画出由下一个字节给出的字形,所指下一字节应为所引用形的编号
008	8	由该码后的两个字节(X,Y)给出的位移增量确定下一点的位置
009	9	由该码后的多个(X,Y)给出的位移增量确定一系列点的位置,位移增量(X,Y)后由字节(0,0)结束
00A	10	由下两个字节定义八分弧,相应的描述码为:“10,(半径,-0SC)”。其中半径为 1~255 的整数,“-”表示将沿顺时针方向绘制圆弧,无此符号则表示沿逆时针方向绘弧;S 为起始八分弧的符号,其值为 0~7,C 是该八分弧的跨度,该值是一个 0~7 的值,若为 0 则意味着有八个八分弧或一整圆。八分圆弧的方向编号见图 2-4
00B	11	用于画那些起始位置和终了位置为非八分圆角边界的弧。该定义使用五个字节:“11,起点偏移,终点偏移,高位半径,低位半径,-0SC” 起点和终点偏移表示弧的开始处和结束处离八分弧边界的距离。而高位半径则为半径值的高 8 位标记,若半径值小于 255 个单位值,该值将为零;低位半径为半径值的低八位。偏移量计算公式为:(实际角度-相应八分圆弧角度)×256/45。“-0SC”含义同 00A
00C	12	由圆弧端点(X,Y)位移量和圆弧凸度定义弧,该定义为:“12,X 位移值,Y 位移值,凸度”。其中 X,Y 位移值为圆弧另一端点相对于当前点的位移,取值范围为-127~127 之间;凸度=2×127×弦高/弦长。半圆的凸度值为 127,直线凸度值为 0
00D	13	由一系列的圆弧端点位移量和各圆弧的凸度绘多个圆弧。格式为:“13,(X1,Y1,凸度 1,X2,Y2,凸度 2,...),(0,0)”
00E	14	适用于垂直标注的文本字体,使下一个字符画在前一个字符的下面

例3 创建如图2-5所示的符号库,包括位置度、圆柱度、面轮廓度、粗糙度和基准符号的形,其代码定义如下:

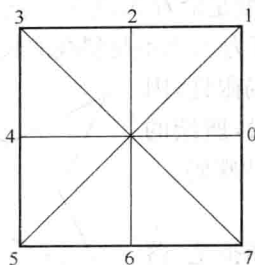


图2-4 八分圆弧的方向编号

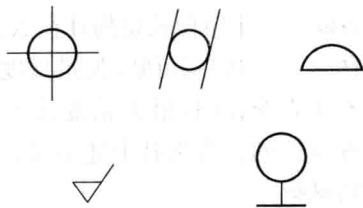


图2-5 基准符号图形

* 211,17,WZD

2,010,1,10,(1,000),2,010,1,048,2,8,(2,2),1,04C,0

* 212,27,YZD

3,8,2,8,(1,0),1,8,(2,8),2,8,(4,0),1,8,(-2,-8),2,8,(1,4),
1,10,2,-000,0

* 213,6,MLKD

12,4,0,-102,048,0

* 214,23,CCD

3,8,2,8,(-3,4),1,9,(3,-4),(5,8),(0,0),2,8,(-8,-4),1,8,(5,0),0

* 215,12,JZFH

2,018,1,020,2,018,1,014,10,(1,-060),0

上述代码分别定义了五个形,可在文字编辑器中编辑,并将该定义的形代码保存为形文件 FH. SHP。分别用到了抬笔、落笔、缩放、八分之圆弧和圆弧凸度等功能,其中面轮廓度定义中的“4,0”表示右端点的位置坐标,“-102”的“-”表示从左端点开始顺时针画弧,凸度= $2 \times 127 \times \text{弦高} / \text{弦长} = 2 \times 127 \times 1.6 / 4 = 102$ 。

2.1.2 编译和加载形文件

所定义形的源文件必须要经过编译、加载后才能调用,编译形文件的命令为 COMPILE,在 AutoCAD 图形环境下键入该命令,将弹出选择形或字体文件对话框,通过对话框选择形源文件即可完成编译。编译后得到的形文件与源文件同名,扩展名为 . SHX。譬如:编译上述的形源文件“FH. SHP”后,将产生形编译文件“FH. SHX”。

如果形的源文件定义有错,则 AutoCAD 会拒绝编译,并给出相应提示。

加载形文件的命令为 LOAD,键入该命令后,选择编译后产生的形文件,即可完成加载。

2.1.3 插入形

对于加载后的形,可通过 SHAPE 命令随时插入绘图工作区。执行过程如下:

命令:shape