

207

2009.11.03

2.22

高等学校教材

建筑 CAD 二次开发技术

吴 涛 编著



A1020235

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书主要讲述结合具体实例,运用 AutoLISP 以及 Visual LISP 开发环境、MAXScript 脚本语言二次开发建筑设计程序的方法和技巧。本书以讲“设计理念”为主,以“软件”为基础,具有建筑设计与软件应用相结合、内容深而广、可针对不同对象进行选择教学等特点。

本书是《建筑 CAD 应用技术》的姐妹篇。《建筑 CAD 应用技术》主要讲述 CAD 核心软件基础和 CAD 技术在建筑设计中的应用。

本书适合作为高等学校建筑学、建筑工程以相关专业本科、研究生教材,也适合广大建筑工程技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

建筑 CAD 二次开发技术/吴涛编著. —北京:电子工业出版社,2002.8

高等学校教材

ISBN 7-5053-7940-2

I. 建… II. 吴… III. 建筑设计:计算机辅助设计—高等学校—教材 IV. TU201.4

中国版本图书馆 CIP 数据核字(2002)第 062304 号

责任编辑:束传政 特约编辑:逢积仁

印 刷:北京市增富印刷有限责任公司

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:21.5 字数:528 千字 附光盘:1 张

版 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数:5 000 册 定价:29.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077



序

——来去匆匆的吴涛

骑着铁马，来去匆匆的吴涛，在一个偶然的机会有他相识。当时他大学毕业不久，是学地下工程的，但对我国刚刚起步的计算机技术满怀热情，刻苦学习，并紧密结合工程和教学实践，勤奋钻研。他酷爱建筑学和建筑艺术，努力把计算机辅助设计运用到建筑设计中。大概是 1994 年，正值我国建筑业蓬勃发展的时期，在一个建筑艺术要求较高的项目中，我们走到了一起，我用炭笔画了几幅徒手效果草图，请他用计算机制成正图。两天后，在屏幕前看到了他的制作，我兴奋，他竟能用奔腾 486 的计算机相当准确地把我的草图经建模、赋材、着色后存入计算机，再经我们合作调整视点、色调、光影、虚实处理，以及选择适当的材质，在很短的时间内完成多幅精美的效果图，顺利地完成了项目，并获得好评。这使我知道，他不仅计算机技术熟练，而建筑素养也很高。

近年来 CAD 技术在建筑设计及建筑工程中的应用已很普及，而各高校在 CAD 应用技术教学中大多是由教师自编零星教材或推荐各类参考书籍。在计算机技术快速发展的今天，教学内容随之不断更新，鉴于当前该学科学生掌握 CAD 技术的水准和程度很不一致，确实需要有一本既适应现状，又有扩展余地的教科书。吴涛在多年勤奋学习和实践过程中，积累了丰硕的经验。他编写的这套教材，具有建筑设计与软件基础相结合的特点；以核心软件和核心模块的概念进行教学时，能简化学习 CAD 技术的难度，更具有实用性：强调“设计概念”为主体，软件为基础，摆正了建筑设计创作和辅助设计的关系；内容由浅入深，循序渐进，深入浅出，并介绍了一些前沿技术，便于在教学中因材施教，适合作为建筑类学科教学用书，并可供建筑专业技术人员参考。

随着现代科学技术的进步，教材内容必将日新月异，我虽已“花甲有余”，当必擦亮眼镜，俨然以长者的角色向来去匆匆的吴涛说：

“真有自己的！”

“祝贺你！也许你会做得更好！”

二零零一年岁末

前 言

本书为建筑学、建筑工程及相关专业本科、研究生的教材。主要介绍运用 AutoLISP 以及 Visual LISP 开发环境、MAXScript 脚本语言二次开发建筑设计程序的方法和技巧。本书以讲“设计理念”为主，以“软件”为基础，具有建筑设计与软件应用相结合、内容深而广、可针对不同对象进行选择教学等特点。

本书是《建筑 CAD 应用技术》一书的姐妹篇。《建筑 CAD 应用技术》主要讲述 CAD 核心软件(包括 AutoCAD, 3D Studio VIZ, Lightscape, Photoshop 等)的基础和 CAD 技术在建筑设计中的应用。这两本书立足于 CAD 技术领域的前沿, 互相补充, 联系紧密。

建筑 CAD 是近年来新兴的、发展迅速的建筑工程领域计算机应用技术, 内容包括利用 CAD 技术进行工程图纸的设计与绘制, 建筑内外空间三维效果的预示, 利用分析软件进行可行性研究, 利用影像合成、三维建筑浏览动画及虚拟现实等手段为建筑设计进行全方位的技术支持。

本教材的主要特点如下:

1. 针对目前相关书籍中存在的内容选择片面、深度欠缺、CAD 技术与建筑设计脱钩的现象, 提出了独特的观点, 具有建筑设计与软件应用相结合、内容深而广, 可针对不同对象进行选择教学等特点。

2. 建筑 CAD 技术的使用者主要从事规划、建筑设计、室内设计的工程师、建筑师和规划师等, 这些专业人员不可能花费大量的时间去学习软件。同时, 建筑设计领域的 CAD 软件种类繁多, 多数又非实际意义上的建筑专业软件, 而且在建筑设计中的应用只局限于某个软件的某个功能模块, 甚至是某个模块的部分功能。鉴于此, 本套教材提出了“核心软件”和“核心模块”的概念, 简化了学习 CAD 的难度。

3. CAD 技术无论多先进, 依然是辅助设计技术, 永远替代不了“人”这一主要因素, 盲目提高 CAD 技术的地位, 将对设计创作带来“变异”和“束缚”的影响。因此, 本套教材具有“设计理念”为主, 以“软件”为基础的特色。

4. “图纸是工程的语言”, 这句话是不言而喻的, 因此建筑图的设计绘制依然是 CAD 教学的重要内容和基础。但在 CAD 技术迅猛发展的今天, 建筑空间的表达不局限于“图”这一单一媒体, 了解和掌握前沿发展技术, 如多媒体表达、虚拟现实技术

等也十分必要。

5. CAD 技术已成为建筑设计人员不可缺少的有效工具。随着 CAD 技术在建筑设计领域的不断深入, 广大的工程设计人员已不能满足只会应用 CAD 技术。掌握一定的 CAD 软件研发知识, 能开发一些专业性较强的设计、制图、计算程序, 也是工程设计人员要求掌握的必备知识。因此, 本套书的第 2 分册专题讲述了建筑 CAD 二次开发技术, 针对建筑设计的专业性, 较为全面地介绍了以 AutoCAD、3D Studio VIZ 软件为平台, 二次开发和定制建筑 CAD 系统的主要概念和方法。

很多人为本套教材的出版付出了艰苦的劳动。东南大学的王文卿教授审阅了第 1 分册——《建筑 CAD 应用技术》书稿, 并为本套教材撰写了序言; 解放军理工大学的许宏发副教授审阅了第 2 分册——《建筑 CAD 二次开发技术》书稿, 并对书中大量的程序进行了试运行, 提出了许多修改建议; 解放军理工大学的陈志龙教授、汤桦先生、赵希涛先生对书稿提出了宝贵意见; 都迷芝女士、曹世霞女士为本书的文字打印、插图编排做了大量工作, 编者在此深表谢意!

由于编者水平有限, 错误和疏漏之处在所难免, 恳请广大读者和同行批评指正。

编 者

目 录

第 1 章 绪论	1
1.1 软件技术基础知识	2
1.2 我国建筑 CAD 软件发展现状和研究方向	13
1.3 建筑 CAD 系统二次开发的主要内容和工具	18
1.4 本书的组织结构和约定	24
1.5 小结	25
第 2 章 以 AutoCAD 为平台定制建筑 CAD 系统	27
2.1 定制专业样板图形	28
2.2 设计专业宏命令	32
2.3 定制标准库文件系统	42
2.4 定制专业菜单系统	49
2.5 小结	65
第 3 章 AutoLISP 编程及 VisualLISP 开发环境	69
3.1 AutoLISP 语言及 Visual LISP 开发环境基础知识	70
3.2 AutoLISP 常用函数及应用	81
3.3 运用 AutoLISP 编辑图形数据库	100
3.4 Visual LISP 开发环境应用	107
3.5 小结	120
第 4 章 建筑设计制图的 AutoLISP 程序范例	123
4.1 常用建筑设计 AutoLISP 程序	124
4.2 墙体设计 AutoLISP 程序	136
4.3 楼梯设计 AutoLISP 程序	158
4.4 小结	168
第 5 章 运用 DCL、AutoLISP 设计管理对话框	171
5.1 DCL 定义对话框基本知识	172
5.2 DCL 控件及其相关属性	180
5.3 用 AutoLISP 控制对话框	193
5.4 建筑专业对话框设计范例	205
5.5 小结	223
第 6 章 运用 MAXScript 开发建筑程序	225
6.1 MAXScript 语言的基础知识	226
6.2 MAXScript 语言表达式	235
6.3 MAXScript 语言的相关组成及函数	245
6.4 运用 MAXScript 定制用户操作界面	260
6.5 小结	279

附录 A AutoLISP 错误信息	281
A.1 用户程序错误	282
A.2 内部错误	286
附录 B DXF 组码表	289
附录 C AutoCAD 系统变量	295
附录 D 配套学习盘组成	331
参考文献	333

第 1 章

绪 论



本章主要内容

- 软件技术基础知识
- 我国建筑 CAD 软件发展现状和研究方向
- 建筑 CAD 系统二次开发的主要内容和工具
- 本书的组织结构和约定



CAD 技术是用计算机系统进行产品的方案设计、解析计算、判断优化、分析评估和详细设计的一门技术,其根本任务是为产品的开发和生成建立一个全局信息模型,它具有应用广泛、直观、高效、优质的特点。CAD 技术早先应用于机械和电子行业,随后进入建筑等工程设计行业并迅猛发展起来。

目前,CAD 技术已渗透到建筑设计领域的各个环节中,并成为设计人员不可缺少有效工具。随着 CAD 技术在建筑设计领域中的不断深入,广大的工程设计人员已不能满足只会运用 CAD 技术进行设计绘图。与此同时,通用 CAD 软件在处理一些专业性较强的问题时也表现出明显的缺陷。因此,工程设计人员掌握一定的 CAD 软件研发知识,能开发出一些专业性较强的设计、绘制、编辑等程序,对今后专业设计的高效性具有重要意义,这也是适应 CAD 技术进一步向网络化、集成化、专业化方向发展的需要。

本书作为《建筑 CAD 应用技术》的进阶教材,针对建筑设计的专业性,全面地介绍了以 AutoCAD、3D Studio VIZ 软件为平台,二次开发和定制建筑 CAD 系统的主要概念和方法。涉及的内容包括软件技术和建筑 CAD 软件研发的基础知识,以 AutoCAD 为平台定制建筑 CAD 系统,运用 AutoLISP 编程及 Visual LISP 开发环境设计建筑专业程序,运用 DCL、AutoLISP 设计管理交互对话框以及运用 3DS MAXScript 设计专业程序等。

本章作为本书的概论,主要包括:

- 软件技术基础知识
- 我国建筑 CAD 软件发展现状和研究方向
- 建筑 CAD 系统二次开发的主要内容和工具
- 本书的组织结构和约定

1.1 软件技术基础知识

计算机技术的飞速发展以及计算机技术与各学科日趋紧密的结合,对人才培养提出了新的要求,计算机知识和应用能力已成为工程人员知识能力结构的重要组成部分。虽然广大的建筑设计人员不必成为计算机专家,但了解和掌握一定的计算机软件技术,对今后的学习和工作将有很大的帮助。因此在讲述建筑 CAD 软件二次开发技术之前,有必要了解一定的软件技术及开发的基础知识,诸如计算机软硬件系统、程序语言、程序设计过程、数据结构、软件工程等。

1.1.1 计算机系统概述

由计算机的硬件和软件组成统一的整体称为计算机系统。计算机系统的硬件是构成计算机系统的全部硬件设备,如中央处理器(CPU)、内存存储器(RAM、ROM)、外存储器(软盘、硬盘、磁带等)以及各种输入设备(键盘、鼠标、显示器、打印机、绘图仪等)。为了充分发挥计算机硬件的使用效率,还需要配备计算机的软件,它是构成计算机系统的全部软件设备,如系统软件和应用软件。

计算机系统从 20 世纪 50 年代到现在约有 50 多年的历史,经历了 4 个时期

的发展:

- **早期阶段** 在计算机系统发展的早期,硬件经历了不断的变化,但软件的开发进度缓慢,成本也很高;软件仅为每一种用途分别设计,通用性很差,没有形成软件产品。

- **第2个时期** 随着以计算机为基础的系统日益增多,计算机软件库开始膨胀,软件维护成为那个时期发展的特点。

- **第3个时期** 这个时期从20世纪70年代初期开始,一直延续到80年代初期,在这个时期分布式系统(多台计算机、各机器并行执行和相互通信)极大地增加了计算机为基础系统的复杂性。硬件的迅速发展已超过软件提供的支持能力,“软件危机”日益严重,为了对付不断增长的软件危机,软件工程才得到了认真的对待。

- **第4个时期** 硬件以惊人的速度迅猛发展,为以计算机为基础的系统开辟了全新的应用领域,软件工程的发展已步入成熟阶段。

1.1.1.1 软件与软件分类

软件(Software)是所有程序、数据和文档的总称,它是控制硬件或有着具体应用目的的指令集,主要分为系统软件和应用软件。

系统软件是由计算机系统设计者或生产者提供的程序、使用及维护手册的总称,这些程序的作用在于扩充计算机功能,控制计算机高效率运行,为用户提供更多的方便。系统软件主要包括:操作系统,程序设计的各种语言处理程序(汇编器、解释器和编译器),数据库管理系统,计算机网络系统,诊断和故障处理程序,编辑程序和调试程序等。由于配置了各种系统软件,从而大大地改善了用户使用计算机的环境。微型计算机常用的操作系统有DOS、Windows系列、UNIX等。

应用软件则是用户利用计算机系统提供的功能为解决特定问题而编制的程序及其使用说明书和维护手册的总称。这类程序种类繁多,不同的应用领域、不同的部门需要编制不同的应用程序。为了减少重复劳动,提高程序的使用效率,应用软件也在逐步商业化,形成了各种软件包。如工程设计中常用的绘图软件包AutoCAD、工程三维可视化系统3D Studio VIZ等。

有时,系统软件和应用软件不能截然分开,如各种标准库,既可看成是应用软件,也可以看成计算机厂家提供的系统软件。对于一个使用计算机的工程技术人员来说,熟悉各种系统软件的目的是为了更好地发挥计算机的功能,更有效地从更高水平上开发应用软件,编制本专业的应用程序。

1.1.1.2 计算机语言概述

半个世纪以来,计算机硬件也经历了4个时期,即电子管、晶体管、集成电路和大规模集成电路到超大规模集成电路的换代等。现代计算机正向着巨型机、微型机、并行式、分布式、网络化和智能化几个方面发展,其相应的软件也逐步丰富和完善。随着计算机硬件的发展,计算机语言经历了机器语言、汇编语言、高级程序设计语言、面向对象的程序设计语言等过程。

(1) 机器语言

在计算机出现的初期,人们直接使用机器语言来编写程序,这种程序叫机器语言程序。它要求程序设计人员相当熟悉计算机的所有细节,如计算机的指令系统、存储容量、寄存器的类型和个数等。这样做工作量非常大,容易出错而且不易修正;同时程序依赖特定的机器类型,局限性很大,移植性很差。

所谓机器语言就是指机器的指令系统,它包含了每台计算机能够执行的基本操作,这些操作由指令形式(一个字节或多个字节的二进制代码形式)构成。它是惟一能裸机直接识别和执行的指令。

(2) 汇编语言

为了摆脱机器语言编程的困难,使计算机成为广大工程技术人员都能够使用的工具,同时也为了减轻程序设计人员编写和调试机器语言程序繁重的劳动,提高程序设计的效率,出现了以符号指令替代机器指令的编程方法。用符号语言编写的程序称为符号程序。它是使用以英文名称缩写的指令助记符,如“取数”用“LD”,“加法”用“ADD”等,比单纯用“0”和“1”的代码串容易记忆。在此基础上进一步扩充就成为汇编语言。用汇编语言编制程序比用机器指令代码方便得多,不仅易于检查和修正错误,而且指令、原始数据和结果的存放单元可由机器根据定位为指令自动分配。但是符号指令不能被计算机直接识别和执行,必须将它翻译成机器语言后才能执行,这个翻译工作是由汇编程序来完成的。

程序设计人员在使用汇编语言编程时,仍必须熟悉计算机的硬件结构,因此程序设计仍然很繁琐和低效。但是正因为依赖于硬件结构,熟练的程序员可结合硬件的特点设计出高质量的程序,所以直至今日,汇编语言仍起着重要的作用,常用于对时间和空间效率要求较高的计算机系统核心程序和实时控制程序等。

(3) 高级语言(面向过程的程序设计语言)

高级语言是由表达各种意义的“词”和“数学公式”,按照一定语法规则组成的面向过程的语言。用高级语言进行程序设计比较接近人的习惯,编出的程序与具体的机器指令无关,可以独立于机器,通用性较强。用高级语言编写的源程序机器是不能直接识别和执行的。目前,世界上有许多的程序设计语言,比较流行的也有几十种,如用于科学与工程计算的 FORTRAN 语言,会话式的 BASIC 语言,教学和系统设计的 Pascal 语言,数据处理的 COBOL 语言,人工智能中使用的表处理 LISP 语言,结构化、模块化、可编译的通用 C 语言等。

本书重点讲述的 AutoLISP 实际就是人工智能中使用的表处理 LISP 语言的一种变种,它嵌套在绘图软件 AutoCAD 中,将 LISP 和 AutoCAD 有机结合,使之既具有一般高级编程语言的基本结构和功能,又具有一般高级语言所没有的强大图形处理功能,通过 AutoLISP 程序可以为 AutoCAD 增加专业性更强的命令集。

(4) 面向对象的程序设计语言

与过去面向过程程序设计比较,面向对象程序设计语言的最大特点在于面向的是对象而不是面向过程。所谓对象就是现实世界中的实体,如桌子、硬件、规律等,AutoCAD 软件中的图形,如直线、曲线、圆等也都作为对象考虑。可以将具有共同行为和特征的实体集合归纳为一类,即每个对象都是属于某一个类的对象,如 AutoCAD 图形中的所有块都组成了块集合,而模型空间集合包含了模型

空间中的所有图形对象（圆、直线、多义线等）。面向对象的程序设计是程序设计中的一种新思想，该思想认为程序是一种相互关联、同时又离散的对象集合，而面向对象程序设计语言是支持这种思想的高级程序设计语言。它主要具有封装性、继承性和多态性等特点。

- **封装性** 通过引入“类”的概念而产生。类把一定数据和关于这些数据的操作封装在一起，这个特点可以显著减少程序各模块之间不良影响，这在多人协作编程时作用更加明显。

- **继承性** 指程序的原有代码可以方便地移植到新的程序中，而且新程序在继承旧程序代码的同时可以增添自己新的内容。继承性大大提高了程序代码的重用率，使得系统开发过程具有更好的连续性，易于应付用户对软件不断发展的要求。

- **多态性** 指相似而实质不同的操作可以有相同的名称，如“和”的操作可以是“整数的和”，也可以是“矢量的和”。多态性使得程序设计与人的思维习惯更趋于一致。

用面向对象的程序设计语言编写的程序也不能直接为机器所识别和执行，必须编译成机器指令后才能运行。面向对象程序设计语言分为两类，一类是按照面向对象设计思想而构成的面向对象程序设计语言，称为纯的面向对象程序设计语言，如 Smalltalk、Eiffel（埃菲尔）等；另一类是在面向过程程序设计语言的基础上增加了对对象的成分构成混合性面向对象的程序设计语言，如 C++、VC++、Borland C++、Delphi 等，本书中所要讲述的集成式开发环境 Visual LISP 就是这类语言程序。

Visual LISP 与 Microsoft ActiveX、ObjectARX 以及 Microsoft Visual Basic 等都是面向对象程序设计语言。经 Autodesk 公司开发的 Visual LISP ActiveX 接口，使 AutoCAD 的对象模型在交叉应用整合方面具有更好的适应性。这将表明用户所开发的应用程序不仅兼容 AutoCAD 软件，而且与其他 ActiveX-Compliant 应用程序一样，通过联合数据库就可以方便地套用，从而解决多年来应用程序智能化弱及整合性差的问题。

1.1.1.3 语言处理程序

用计算机语言编写的程序除了机器语言程序外，其他程序都不能为机器直接识别和执行，必须经过各种语言的处理程序或翻译器处理转换为机器语言程序后，才能为机器所识别和执行。由翻译所得到的结果称为目标程序，目标程序可以用机器语言表示，也可以用汇编语言或其他中间语言表示，而语言处理程序提供这种翻译功能。

- **汇编程序** 其主要功能是把汇编语言源程序翻译成机器能识别的目标程序，源程序通常是由 ACSII 码表示的符号指令串。

- **编译程序** 用高级语言编写的源程序可以通过编译的途径，将其翻译成语义上等价的的目标程序。如果目标程序是以汇编语言表示的，则必须再经汇编程序翻译成机器语言程序；如果目标程序是某种中间语言表示，则该目标程序可以由解释器解释执行或再经编译后执行。例如，PASAL P_编译器所产生的中间代码

叫 P_代码, 这种 P_代码程序可以解释执行, 也可以先编译后执行。

- **解释程序** 解释途径是把翻译和运行合并在一起, 这种方式并不形成整个目标程序, 而是按照源程序中语句的动态顺序逐句地进行解释并立即执行。完成这项工作的语言处理程序称为解释程序, 它所输出的是计算结果而不是目标程序。在编译和解释两种方式中, 解释器比编译器灵活, 因为解释器有对机器的控制权, 能够在发现错误时停止源程序执行并指出错误; 但就速度而言, 编译器是绝对优先的。

1.1.2 设计计算机程序与软件

计算机软件都由程序模块构成。无论计算机所完成的工作多么繁杂或精细, 必须是按照人们预先编写好的程序进行工作。

1.1.2.1 程序设计过程

一个较为完整的程序设计过程可分为以下三个步骤:

- 分析问题, 构造模型
- 算法设计, 过程描述
- 程序编写, 测试运行

简单地可归结为“模型—算法—编码”三个阶段。

(1) 分析问题, 构造模型

每一个程序都应有明确的功能要求, 即程序应解决问题的内容、性质和规模等。在程序设计开始时, 必须分析程序提供的功能应能解决哪些问题, 对理论上可遵循的公式、方法进行适当的修正, 考虑量化和积累效应, 使公式化内容适应于计算机的计算模式。对于非数值类的数据处理, 应该通过分析问题来构造模型。实际上, 许多应用领域的问题不能直接在计算机上解决, 需要经过简化后建立相应的模型, 才能上机运算。各种分析问题、建立模型的方法已成为应用领域中引人注目的研究课题, 有很多专著及文献可供参考。

(2) 算法设计, 过程描述

模型一旦建立起来, 也就是确定了程序应该解决什么问题, 那么在编写出程序之前, 首先要为它寻找一个“如何做”的算法, 然后按算法编出程序。**算法定义:** 一个算法表示一个计算过程, 它应该是明确的、确定的和有限的, 它本身以能被执行的基本指令形式给出。一般说来, 运算结果的精确度及运算速度均与算法有着密切的关系。

进行算法设计的同时, 还要进行数据设计, 包括选择数据类型、考虑数据的组织和存放形式, 即数据结构。数据结构直接影响数据的存取速度, 也影响处理数据的效率。随着计算机应用领域的日益广泛, 计算机算法愈来愈多地涉及非数值程序设计, 其中基本技术就是查找和排序。算法给出计算机在“如何做”的过程中所包含的明确步骤, 这种处理过程必须准确地描述出来。描述的方法很多, 如书面语言形式、流程图形式和类程序设计语言的形式等。有了处理过程的描述就可以较为方便地编写程序了。

数据结构是一个重要概念，数据结构主要研究数据的组织方法以及响应的操作算法。它不仅是一般程序设计的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础，因此在计算机科学中占有重要地位。对于相当多的程序设计问题来说，理清数据内在关系可以获得对问题的正确观察角度，看清问题的结构，甚至是解法的结构。数据结构包括逻辑结构和物理结构两个方面。逻辑结构是对数据元素之间逻辑关系的描述，它可用一个数据元素的集合和定义在此集合上的一些关系来表示；物理结构是数据结构在计算机中的表示或实现的形式，又称为储存结构。

(3) 程序编写，测试运行

编写程序之前先选定一种程序设计语言。一个好的又适于具体问题的语言可以使程序的结构清晰、简洁，可以正确地记述待解决的问题，同时还可以准确地表示过程，以便数据的抽象化和模块化。

仅编制一个可运行的、结果正确的程序是不够的，应该努力提高程序设计的技术和改善程序的质量，一个高质量的程序应具备以下条件：

- 建立正确的数学模型和确立有效的计算方法
- 运行结果必须是正确的，且精度及各方面均有要求
- 程序本身具有良好的结构，逻辑清楚，易读易懂
- 程序运行时间尽可能短，同时尽可能合理地使用内存
- 便于检查、修正、移植和维护

程序要进行检查和调试，目的是查找和改正程序中存在的错误，使程序能顺利地运行。程序的调试是一项复杂而繁琐的工作，也是一项值得研究的内容。

1.1.2.2 软件开发方法

软件的开发过程是从对用户的需求开始进行分析，经过设计和实现，到最终产生可以正确执行的代码和有关使用和维护的各种文档。完成整个过程的方法就称为软件开发方法。目前软件开发方法有传统的非形式化开发方法、严格的形式化开发方法和逐步从非形式化向形式化过渡的开发方法等三种类型。

• **传统的非形式化开发方法** 它的基本思想是引入一些非形式化的图形和文字记号，如数据流图、结构图、结构化语句和判定表等，提供一定的设计原则，协助开发人员按照一定的步骤，比较明确和简练地书写程序代码。

• **严格的形式化开发方法** 它的基本思想是对系统建立一个数学模型，研究模型和提供一种基于数学的或形式语义学的软件规格说明语言，用这种语言严格地描述所开发的软件功能，并由计算机完全机械化地将它转换成可执行的代码。

• **逐步从非形式化向形式化过渡的开发方法** 这是一种介于形式化和非形式化之间的开发方法，即从非形式化规格说明向形式化源代码过渡的方法。过渡过程中的各种中间文档用非形式化语言和形式化语言的混合体来描述；随着设计的深化，文档中形式化部分逐渐增多，而非形式化部分逐渐减少，直至完全变成形式化的可执行源代码。

1.1.2.3 软件开发环境

软件开发环境 (Software Development Enviroment, 简称 SDE) 是一组相关的软件工具集合, 它们组织在一起支持某种软件开发方法或者与某种软件开发模型相适应。软件工具指这样一类的软件, 它们用来帮助开发、测试、分析和维护另一类计算机软件及其文档编制。从用户的角度出发, 软件开发环境中还应有会话系统 (人机界面) 和环境数据库等。

- **人机界面** 指软件开发环境与用户之间一个统一的交互式对话系统。多窗口屏幕显示、弹出式菜单驱动方式和鼠标控制等新型软硬件技术支持的一个友好用户接口, 是软件开发环境的重要标志。用户通过人机界面向计算机输入各种数据和命令, 以此操纵和控制计算机为其服务; 而计算机则通过人机界面及时地将计算结果、处理的情况和控制状态等显示出来, 供用户了解并等待或指导用户下一步的行动。图 1-1 显示的是 AutoCAD 软件的 VBA 开发环境, 它提供了一个比较友好的、功能强大的人机界面。

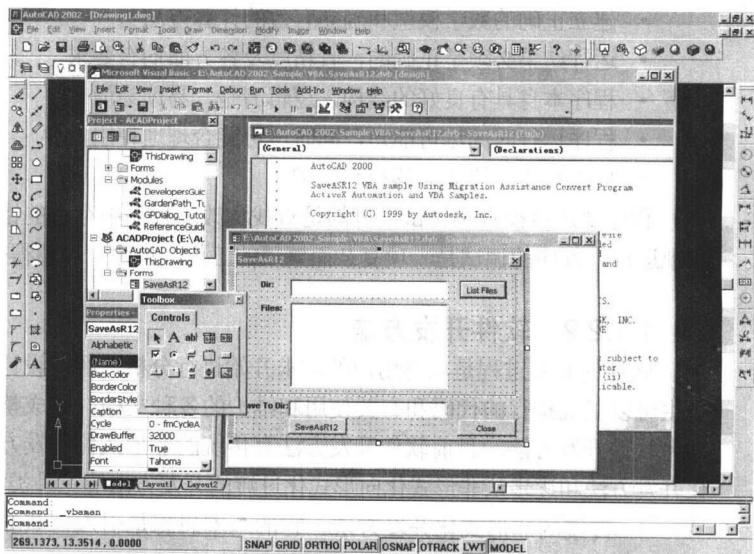


图 1-1 AutoCAD 软件的 VBA 开发环境

- **工具集** 它是软件开发环境的主要成分, 为软件开发服务, 用以提高软件的质量和软件的开发率。例如编辑工具、编译工具、调试和维护工具、分析工具等。这些工具在开发环境中不是封闭和分离的, 而是以综合、一致和整体连贯的形态来支持软件开发, 并且是与某种软件开发方法或者某种软件开发模型相适应的。

- **软件环境数据库** 软件环境数据库也称为软件工程数据库或者项目数据库。它是软件开发、维护过程中项目数据的存储设施, 是软件开发环境的核心部分, 是多个软件工具之间连接统一的媒介。工具从数据库中获得输入信息, 经过加工将结果再放回数据库中, 而其他的工具又可以从数据库取出这些信息获得输入。所有的工具均使用同一类型的数据对象。所以环境数据库把软件环境中的

各个工具有机地结合起来，对软件的开发和维护提供了一个连接完整的支持。

1.1.3 软件工程概述

“软件工程”一词是 1968 年北大西洋公约组织 (NATO) 在联邦德国召开的一次会议上首次提出的。它的中心思想是把软件当作一种工业产品，而不是某种个体或小作坊的神秘技巧，要求“采用工程化的原理与方法对软件进行计划、开发和维护”。这样做的目的，不仅是为了实现预期的速度和经费完成软件的生产计划，也是为了提高软件的生产效率和可靠性。软件工程是从技术（方法和工具）和管理两个方面，研究如何更好地开发和维护计算机软件的一门新兴学科。

1.1.3.1 软件工程的概念和目标

软件工程的目的是能够成功地完成软件开发项目。而一个成功的项目，其目标是生产一个可接受的产品。一个软件产品应该是系统中所有程序及能够使用和改变这些程序的文档集合。一个软件开发项目是软件生产者通过生产软件产品而换取用户支付开发费用的努力结果。这个过程包括从帮助用户定义问题、澄清需求，到软件产品是否满足用户需求的整体评价。因此，一个成功的软件开发项目，应该是在有限的时间和成本内提交一个用户和开发者都满意的软件产品。

但是在软件项目开发过程中，往往存在许多问题。如成本失控、计划超期、开发效率不高、软件质量低下、后期不易维护，以及用户经常改变其需求而造成系统不能再满足其新的需求，使系统无法运行或在开发过程中终止和废弃等。这就需要在软件项目开发过程中采用工程化的观点和方法，解决这一系列的问题。

这里给出软件工程的定义：软件工程是对需求、计算机技术、人员及其技能、时间、成本和其他资源的管理，并籍此在一个满足开发者需求的过程中形成一个满足用户需求的软件产品。事实上，软件工程是应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则和方法创建软件，以达到改善生产效率、提高质量、降低成本的目的。

1.1.3.2 软件工程研究内容

软件工程所研究的内容主要归纳为以下四个方面：

- **软件的规范和标准** 软件工程的生产对象是软件产品，要使这种产品具有很好的工效，又要具有良好的经济指标，就必须有一组规范和标准，用来限制软件开发中的自由度，告诉人们应该怎样完成开发任务，以保证软件产品具有高度的可靠性、一致性、可维护性以及较高的劳动生产率和较低的成本等。一般来说，不同的国家、不同的软件公司都具有不同软件规范和标准。目前，随着国际标准组织 ISO 9000 系列标准的不断推进，世界软件行业也正在向其靠近。

- **软件开发技术和方法** 从软件工程诞生以来，许多从事软件的学者和专家，一直致力于软件的理论、方法和技术的研究，例如支持软件整个生命周期的模型研究，支持软件开发过程中各阶段的方法与技术的研究等。

- **软件工具和软件开发环境** 软件工具所指的范围很广，按其功能大致可