



深入浅出

AutoCAD.NET二次开发

李冠亿 编

中国建筑工业出版社

深入浅出

AutoCAD. NET 二次开发

李冠亿 编

中国建筑工业出版社

图书在版编目 (CIP) 数据

深入浅出 AutoCAD. NET 二次开发/李冠亿编. —北京:
中国建筑工业出版社, 2012. 2
ISBN 978-7-112-14019-0

I. ①深… II. ①李… III. ①AutoCAD 软件 IV.
①TP391. 72

中国版本图书馆 CIP 数据核字 (2012) 第 013753 号

本书从程序设计及数据库等软件开发基础知识入手, 在详细介绍 AutoCAD 开发原理、开发环境、开发接口及基本功能的实现的同时, 给出了丰富的实例。本书还在对各种开发方式的技术研究的基础上, 对未来软件的开发模式、开发方向作了探讨。本书适合研发工程师作为工作参考书, 也适合初学者作为入门读物学习。

责任编辑: 咸大庆 武晓涛

责任设计: 李志立

责任校对: 王誉欣 关 健

深入浅出 AutoCAD. NET 二次开发

李冠亿 编

*

中国建筑工业出版社出版、发行 (北京西郊百万庄)
各地新华书店、建筑书店经销
霸州市顺浩图文科技发展有限公司制版
北京世知印务有限公司印刷

*

开本: 787×1092 毫米 1/16 印张: 30 字数: 745 千字
2012 年 5 月第一版 2012 年 5 月第一次印刷
定价: 78.00 元 (含光盘)
ISBN 978-7-112-14019-0
(22010)

版权所有 翻印必究

如有印装质量问题, 可寄本社退换

(邮政编码 100037)

前 言

AutoCAD 以普及的 DWG 文件格式，完善的图形绘制功能及强大的图形编辑处理功能在各行业计算机辅助设计中发挥着举足轻重的作用。对其进行二次开发可以继承其强大的图形数据处理接口，根据更深层次的需求扩展其绘图功能，以更高层次、高精度、高效率地进行计算机辅助设计。

AutoCAD 二次开发是在 AutoCAD 开放的 API 的基础上对其用户界面、业务逻辑以及数据库访问进行扩展。如今 CAD (Computer Aided Design) 软件的需求日益趋向信息化、集成化、协同化。在 .NET 环境对其进行二次开发可以灵活地结合其各种开发方式 (ObjectARX/ObjectDBX、AutoCAD ActiveX、AutoLisp)，不局限于开发语言以完全面向对象的编程方式结合强大的 .NET Framework 组件在最短的开发周期、最低的开发成本上创建便捷的数据处理机制及通用的数据库访问形式、用户界面绚丽、易于版本控制及 32 位系统与 64 位系统移植的跨平台跨组件跨服务的程序架构。

本书结构合理、图文并茂、讲解生动、实例丰富。力求循序渐进，从程序设计及数据库等软件开发基础知识入手，在详细介绍 AutoCAD 开发原理、开发环境、开发接口及基本功能的实现的同时结合丰富的实例进行讲解。在对各种开发方式的技术研究的基础上，对未来软件的开发模式和开发方向作了探讨。适合初学者、自学者入门，也适合研发工程师作为工具书、词典查询。

本书内容安排如下：

第 1 章“概述”主要介绍 AutoCAD 软件的功能用途以及开发方式。

第 2 章“AutoCAD 操作应用”简单介绍 AutoCAD 软件的操作及应用。

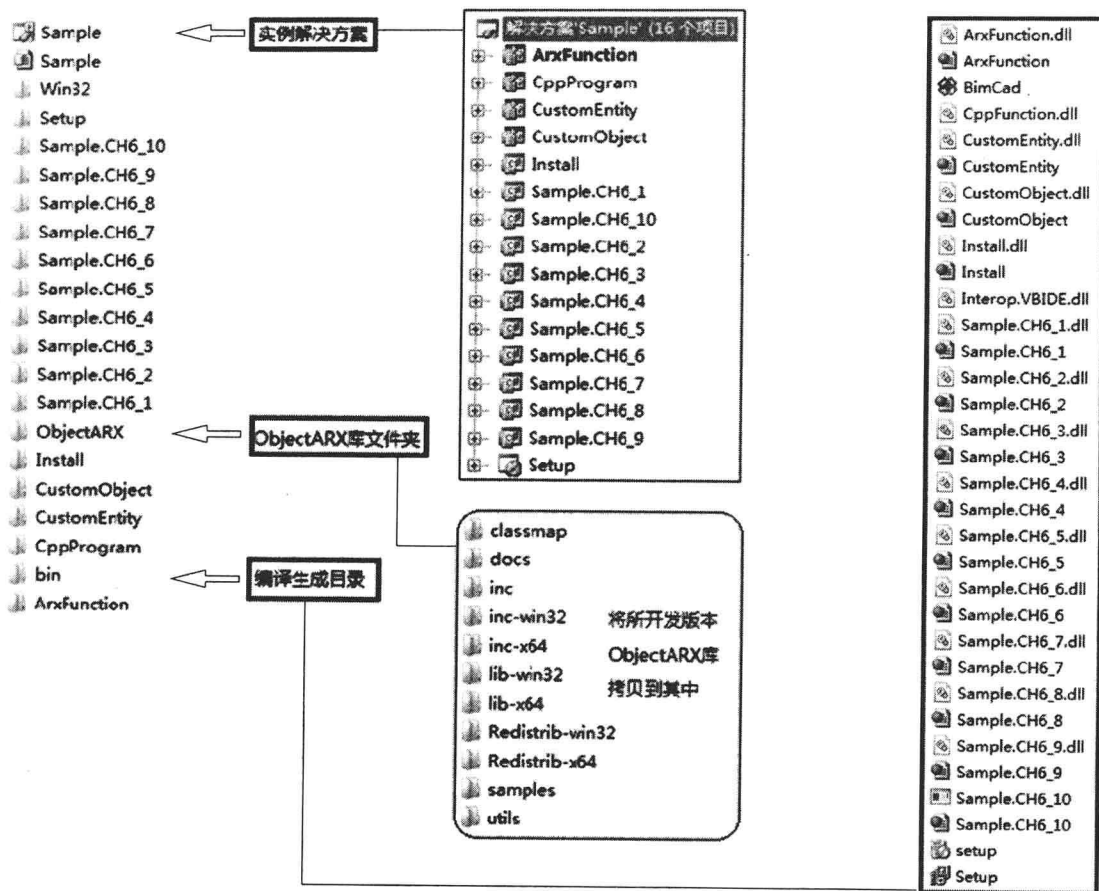
第 3 章“程序设计基础”主要介绍程序设计的基础知识。

第 4 章“数据库基础”主要介绍数据库相关的基础知识。

第 5 章“AutoCAD.NET 二次开发”详细介绍 AutoCAD.NET APIs 及 .NET 环境下结合 AutoCAD.NET、ObjectARX/ObjectDBX、AutoCAD ActiveX、AutoLisp 开发的基本方法以及开发环境的配置，程序的编译、调试及部署。

第 6 章“开发实例”详细介绍 AutoCAD 二次开发的实现方法以及开发思路。

本书实例的源代码可以在随书附送光盘中拷贝，光盘内容如下（所有实例在 WIN32 系统+VS2008+AutoCAD2010-2012 测试通过）：



由于作者水平及时间有限，本书难免存在不足与错误。读者发现任何问题或 AutoCAD 开发上的困难、需求和心得体会欢迎登录数字建筑网 (<http://www.BimCad.org>) 论坛进行交流。

希望这本书能对您了解 AutoCAD 二次开发技术有所帮助。

特别感谢：谢圆平总监、张平总工程师、张继峰博士、武鹏、张涛、王哲峰、宋跃芳、徐帅、郭思明、张树强、潘厚佳等在此书写作期间给予的帮助与支持，以及中国建筑工业出版社武晓涛编辑在此书出版期间付出的辛勤劳动。

李冠亿
2011-11-05
于北京

目 录

第 1 章 概述	1
1.1 AutoCAD 功能特点	1
1.2 AutoCAD 二次开发的意义及内容	3
1.3 AutoCAD 二次开发方式	4
1.3.1 ADS	4
1.3.2 ObjectARX/ObjectDBX	4
1.3.3 AutoLISP/Visual LISP	5
1.3.4 VBA/AutoCAD ActiveX Automation (COM)	6
1.3.5 AutoCAD.NET	8
1.4 开发方式对比	9
第 2 章 AutoCAD 操作应用	11
2.1 基本界面	11
2.2 基本功能	13
2.2.1 绘图	13
2.2.2 标注	15
2.2.3 编辑	15
2.2.4 参数化	15
第 3 章 程序设计基础	17
3.1 .NET 简介	17
3.2 C# 简介	22
3.2.1 开发环境	22
3.2.2 关键字	25
3.2.3 运算符、表达式	25
3.2.4 数据类型	27
3.2.5 基本语句	30
3.3 C++/CLI 简介	36
3.3.1 项目配置	36
3.3.2 数据类型	38
3.4 数据结构	43
3.4.1 线性结构	43
3.4.2 非线性结构	44
3.4.3 排序	47
3.4.4 查找	50
3.5 算法	51

第4章 数据库基础	53
4.1 数据模型	53
4.2 关系型数据库	53
4.2.1 二维表	53
4.2.2 关系代数	54
4.3 SQL 语言	56
4.4 数据库设计	57
4.5 DWG 数据库	58
第5章 AutoCAD. NET 二次开发	59
5.1 AutoCAD. NET 二次开发环境及配置	59
5.1.1 项目配置	59
5.1.2 编译、调试	64
5.1.3 程序部署	67
5.1.4* 混合开发	73
5.2 AutoCAD. NET API	88
5.2.1 Autodesk. AutoCAD. ApplicationServices (应用程序服务)	89
5.2.2 Autodesk. AutoCAD. DatabaseServices (数据库服务)	98
5.2.3 Autodesk. AutoCAD. EditorInput (用户交互)	269
5.2.4 Autodesk. AutoCAD. Geometry (几何)	292
5.2.5 Autodesk. AutoCAD. BoundaryRepresentation (边界表示)	300
5.2.6 Autodesk. AutoCAD. Runtime (运行时)	306
5.2.7 Autodesk. AutoCAD. Windows、Autodesk. Windows (窗体界面)	313
第6章 开发实例	323
6.1 数据库	323
6.1.1 添加块定义	323
6.1.2 添加实体到模型空间	325
6.1.3 添加、删除图层	326
6.1.4 添加对象到命名对象词典	330
6.1.5 写块克隆	331
6.1.6 设置当前视图	333
6.1.7* 自定义数据库对象	337
6.2 实体对象	346
6.2.1 创建简单实体	346
6.2.2 创建三维实体	349
6.2.3 添加普通图块	352
6.2.4 添加带属性的块	354
6.2.5 实体变形	358
6.2.6 扩展数据	360
6.2.7 注释比例	363
6.2.8 改变显示次序	366
6.2.9 驱动动态块	368
6.2.10 添加实体到组	372

6.2.11 实体边界	374
6.2.12 获取子实体 GSMarker (图形系统标记)	376
6.2.13* 自定义实体	378
6.3 用户交互	390
6.3.1 输出消息	390
6.3.2 输入数据	392
6.3.3 自定义用户界面	397
6.3.4 面板	403
6.3.5 带式菜单	404
6.3.6 快捷菜单	409
6.4 事件	410
6.4.1 对象删除事件	410
6.4.2 文档双击事件	411
6.5 即时绘图	413
6.5.1 直线即时绘图	413
6.5.2 多段线即时绘图	417
6.6 规则重定义	423
6.6.1 显示重定义	423
6.6.2 夹点重定义	425
6.7 数据处理	433
6.7.1 写 Excel 文件	433
6.7.2 LINQ 数据检索	436
6.8* 混合开发	439
6.8.1* AutoCAD.NET 调用 ObjectARX 程序	439
6.8.2* AutoCAD.NET 调用 AutoCAD ActiveX 组件	442
6.8.3* AutoCAD.NET 与 AutoLisp 相互调用	444
6.9 程序部署	446
6.9.1 注册表部署	446
6.9.2 配置文件部署	452
6.10* Windows 程序与 AutoCAD 应用程序交互	454
附录	457
附录 A: AutoCAD 版本对应的注册表信息	457
附录 B: ObjectARX 库与 AutoCAD.NET 程序集对应关系	457
附录 C: AutoCAD 常用操作快捷键	458
附录 D: AutoCAD 系统变量	459
附录 E: 命名规范	470
参考文献	471

注：“*”内容为非完全使用 AutoCAD.NET 方式开发部分。

第 1 章 概 述

CAD (Computer Aided Design) 计算机辅助设计, 是计算机技术的一个重要的应用领域。AutoCAD 是美国 Autodesk 公司开发的交互式绘图软件, 用于二维及三维设计、绘图的系统工具。用户可以创建、浏览、管理、打印、输出、共享及精确复用富含信息的设计图形。

Autodesk 公司成立于 1982 年 1 月, 公司在近二十年的发展历程中, 不断丰富和完善 AutoCAD 系统, 并连续推出各个新版本, 使 AutoCAD 由一个功能非常有限的绘图软件发展到了现在功能强大、性能稳定、市场占有率位居世界第一的 CAD 系统, 在城市规划、建筑、测绘、机械、电子、造船、汽车等许多行业中得到了广泛的应用。

AutoCAD 更新升级的速度非常快, 每个新版本都会加入一些更新更强大的功能。AutoCAD2000 增添了 AutoCAD 设计中心 (ADC)、多文档设计环境 (MDE)、Internet 驱动、新的对象捕捉功能、增强的标注功能以及局部打开和局部加载的功能; AutoCAD2005 新增了图纸集功能和用于显示可能会在图形生命周期中修改的数据的字段对象; AutoCAD2006 新增了动态块功能, 动态块中定义了一些自定义特性, 可用于在位调整块, 而无须重新定义该块或插入另一个块; AutoCAD2008 增添了自动缩放注释及多重引线, AutoCAD2010 增强了三维功能, 更大的突破是参数化绘图功能, 可以在图纸上建立对象之间的几何关系及约束关系。AutoCAD 系统在不断完善, 功能在不断增强, 见图 1.0 AutoCAD 版本。

虽然 AutoCAD 本身的功能集已经足以协助用户完成各种设计工作, 但用户还可以通过 Autodesk 以及数千家软件开发商开发的五千多种应用软件把 AutoCAD 改造成为满足各专业领域要求的专用设计工具。AutoCAD 荣登全球绘图软件的龙头宝座, 主要是因为它具有开放的体系结构, 它允许用户和开发者采用高级编程语言对其进行扩充和修改 (即二次开发), 能最大限度地满足用户的特殊要求。

1.1 AutoCAD 功能特点

1. 具有完善的图形绘制功能。
2. 具有强大的图形编辑功能。
3. 可以采用多种方式进行二次开发或用户定制。
4. 可以进行多种图形格式的转换, 具有较强的数据交换能力。
5. 支持多种硬件设备。
6. 支持多种操作平台。
7. 具有通用性、易用性, 适用于各类用户。

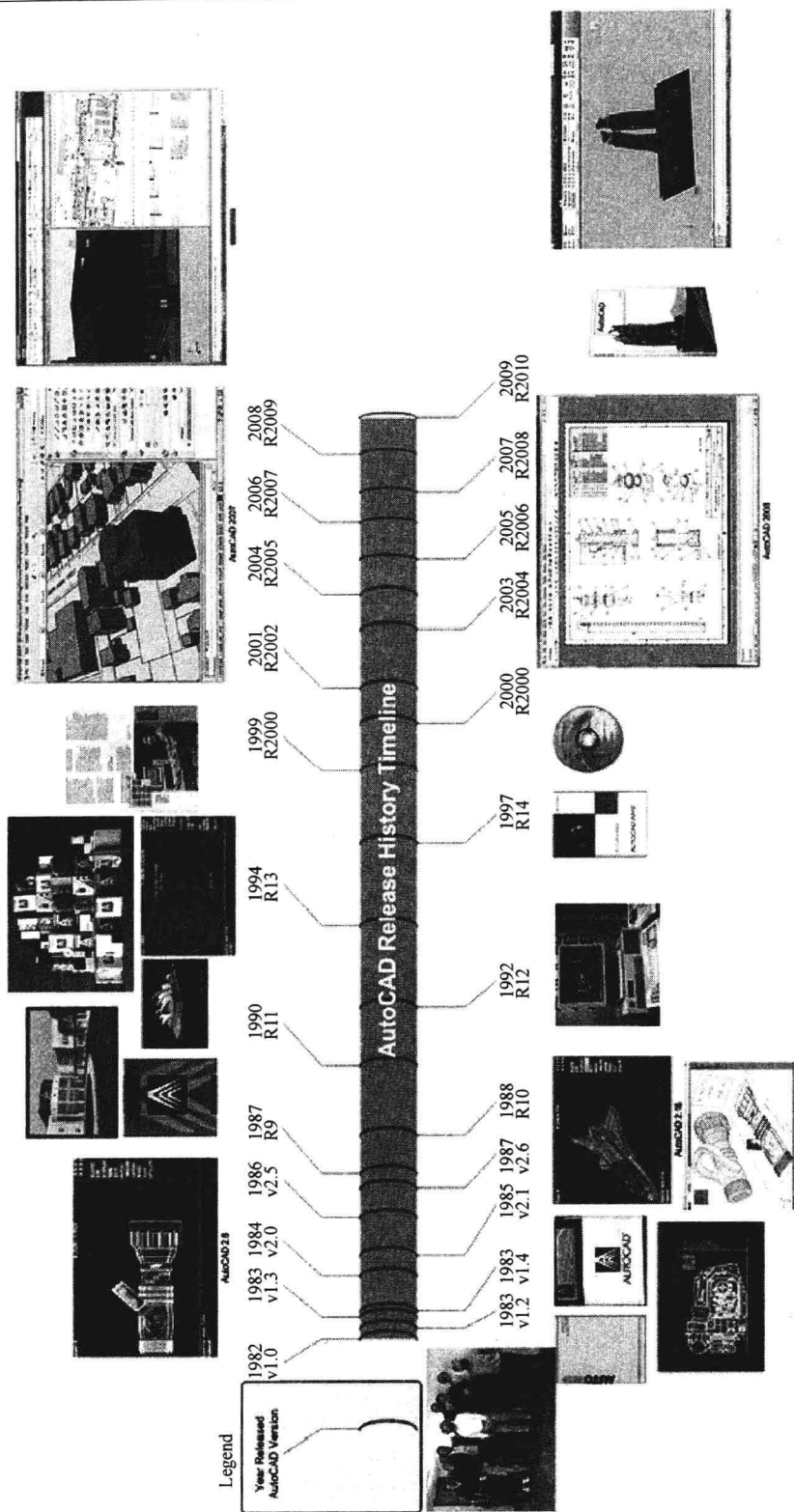


图 1.0 AutoCAD 版本

1.3 AutoCAD 二次开发方式

AutoCAD 第一版于 1982 年 11 月推出，其二次开发方式也在不断地涌现（见图 1.3-1 开发方式时序、图 1.3-2 开发方式原理）。

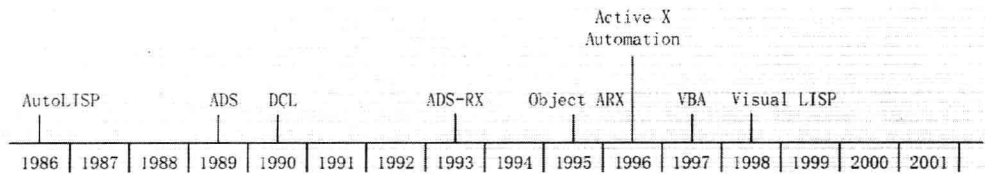


图 1.3-1 开发方式时序

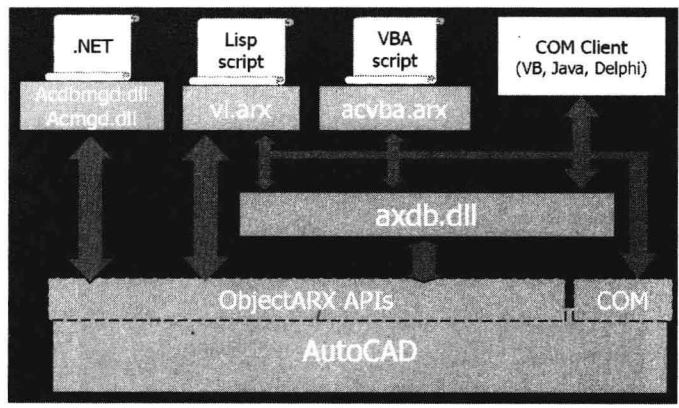


图 1.3-2 开发方式原理

1.3.1 ADS

ADS (AutoCAD Development System) 是 AutoCAD 的 C 语言开发系统，ADS 本质上是一组可以用 C 语言编写 AutoCAD 应用程序的头文件和目标库，直接利用用户熟悉的各种流行的 C 语言编译器，将应用程序编译成可执行的文件并在 AutoCAD 环境下运行，这种可以在 AutoCAD 环境中直接运行的可执行文件叫做 ADS 应用程序。ADS 由于其速度快，又采用结构化的编程体系，因而很适合于高强度的数据处理，如二次开发的机械设计 CAD、工程分析 CAD、建筑结构 CAD、土木工程 CAD、化学工程 CAD、电气工程 CAD 等。

1.3.2 ObjectARX/ObjectDBX

ObjectARX (AutoCAD Runtime eXtension) 是一种崭新的开发 AutoCAD 应用程序的工具，以 C++ 为编程语言，采用先进的面向对象的编程原理，提供可与 AutoCAD 直接交互的开发环境，能使用户方便快捷地开发出高效简洁的 AutoCAD 应用程序。ObjectARX 能够对 AutoCAD 的所有事务进行完整的、先进的、面向对象的设计与开发，并且开发的应用程序速度更快、集成度更高、稳定性更强。ObjectARX 从本质上讲，是一种

特定的 C++ 编程环境 (见图 1.3-3 ObjectARX 开发环境), 包括一组动态链接库 (DLL), 这些库与 AutoCAD 在同一地址空间运行并能直接利用 AutoCAD 核心数据结构和代码, 库中包含一组通用工具, 使得二次开发者可以充分利用 AutoCAD 的开放结构, 直接访问 AutoCAD 数据库结构、图形系统以及 CAD 几何造型核心, 以便能在运行期间实时扩展 AutoCAD 的功能, 创建能全面享受 AutoCAD 固有命令的新命令。ObjectARX 的核心是两组关键的 API, 即 AcDb (AutoCAD 数据库) 和 AcEd (AutoCAD 编译器), 另外还有其他的一些重要库组件, 如 AcRX (AutoCAD 实时扩展)、AcGi (AutoCAD 图形接口)、AcGe (AutoCAD 几何库)、ADSRX (AutoCAD 开发系统实时扩展)。ObjectARX 还可以按需要加载应用程序; 使用 ObjectARX 进行应用开发还可以在同一水平上与 Windows 系统集成, 并与其他 Windows 应用程序实现交互操作。

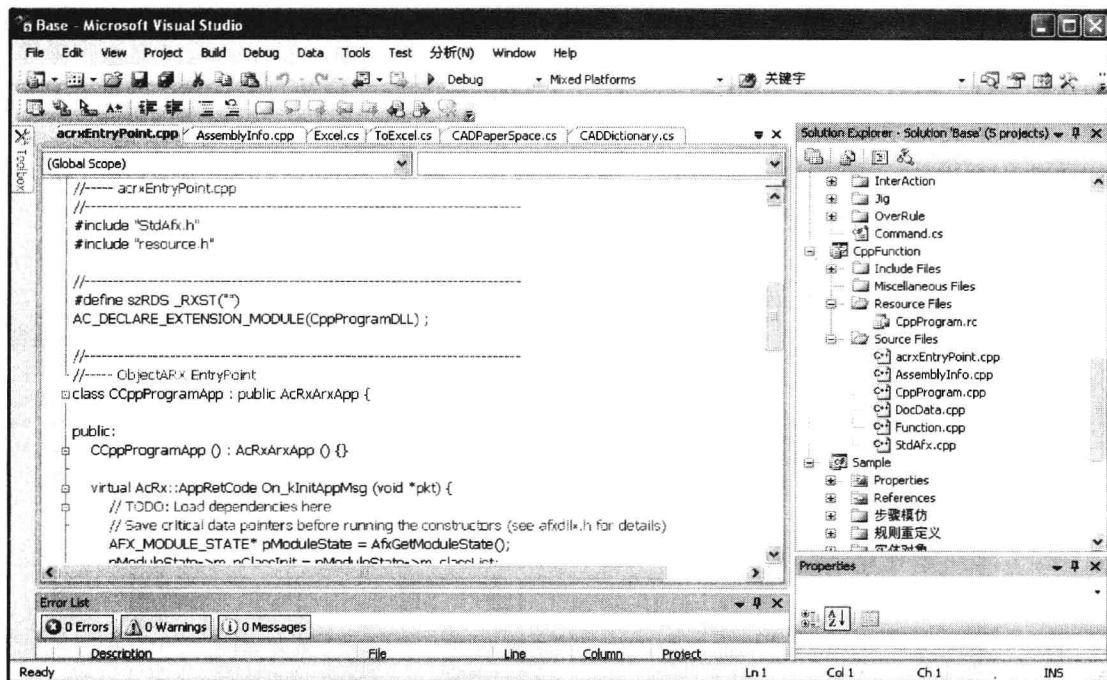


图 1.3-3 ObjectARX 开发环境

ObjectDBX (DataBase eXtension) 包含了一系列动态链接库 (DLL), 通过它们可以对 DWG 数据库进行扩展, 建立包含在 AutoCAD 图形文件中的自定义对象, 也可以使生成的应用程序在 AutoCAD 环境之外对 DWG 图形进行处理。ObjectDBX 库中含有使自定义类 (实体、非图形对象等) 可作为 AutoCAD 中的扩展对象或自定义实体的智能处理。

ObjectDBX 与 ObjectARX 最明显的差别就是 ObjectARX 需要 AutoCAD 作为宿主程序, 而 ObjectDBX 可以基于任意的宿主程序。ObjectDBX 的支持下使得 AutoCAD 二次开发中数据库操作、业务逻辑、用户界面的开发可以分离。

1.3.3 AutoLISP/Visual LISP

AutoLISP 基于简单易学而又功能强大的 LISP (List Processor) 编程语言。由于 Au-

toCAD 具有内置 LISP 解释器, AutoCAD 能够直接读取 AutoLISP 代码, 无需编译。因此用户可以在命令行中输入 AutoLISP 代码, 或从外部文件加载 AutoLISP 代码。AutoLISP 应用程序可以通过多种方法与 AutoCAD 交互。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 AutoLISP 程序, 可以向 AutoCAD 添加专用命令。实际上, 某些标准 AutoCAD 命令就是 AutoLISP 应用程序。其特点为:

1. AutoLISP 语言编写出来的程序可读性非常强。
2. 是 AutoCAD 内嵌语言, 调试非常方便, 与 AutoCAD 无缝集成。
3. 数据结构灵活多变, 且可以调用绝大部分 AutoCAD 的内置命令。

VisualLISP (VLISP) 是为加速 AutoLISP 程序开发而设计的软件工具 (见图 1.3-4 Visual LISP 开发环境)。AutoLISP 通过 VisualLISP (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, 用户在命令提示下输入代码后可立即看到结果。这使 AutoLISP 语言容易调试。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 AutoLISP 通过对象反应器对事件进行响应。Visual LISP 为 AutoLISP 应用程序提供三种文件格式选项:

1. LSP 文件 (.lsp): AutoLISP 程序代码的 ASCII 文本文件。
2. FAS 文件 (.fas): 单个 LSP 程序文件的二进制编译版本。
3. VLX 文件 (.vlx): 一个或多个 LSP 文件和/或对话框控制语言 (DCL) 文件的编译集合。

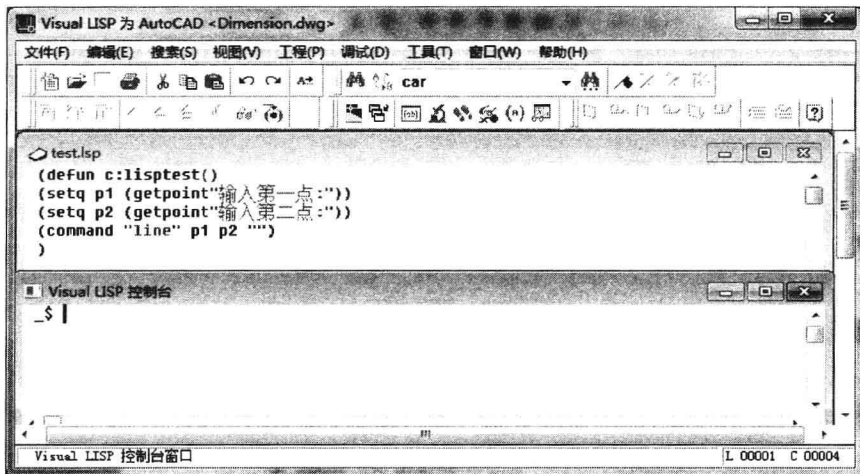


图 1.3-4 Visual LISP 开发环境

1.3.4 VBA/AutoCAD ActiveX Automation (COM)

VBA 即 Microsoft Office 中的 Visual Basic for Applications 开发环境 (见图 1.3-5 VBA 开发环境) 被集成到 AutoCAD 中。VBA 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。不过在 2012 以后的新版本中 AutoCAD 将不再支持 VBA 开发。

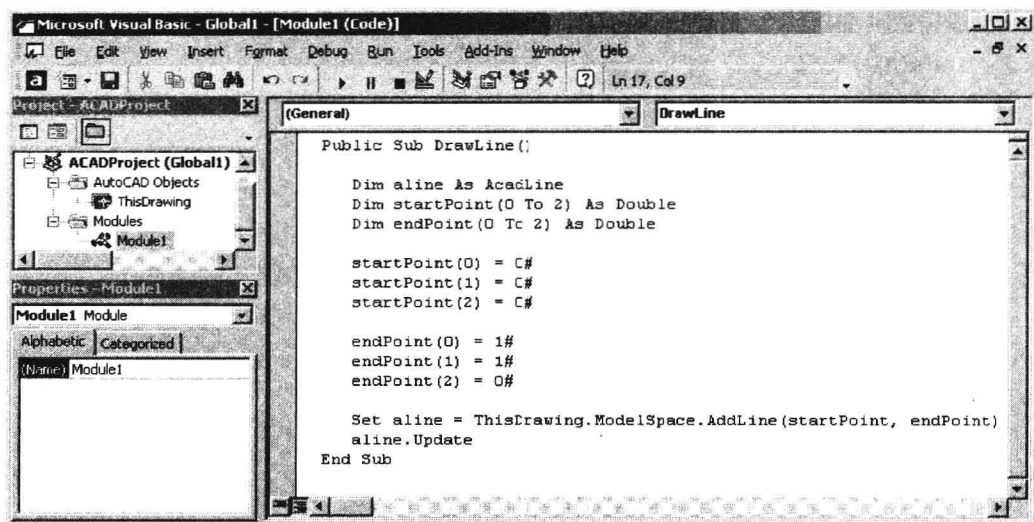


图 1.3-5 VBA 开发环境

AutoCAD ActiveX 是一种可供开发者通过编程手段来操纵 AutoCAD 应用程序的机制 (见图 1.3-6 AutoCAD ActiveX 原理)。是 Microsoft® 基于 COM (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。ActiveX 是由一系列的对象, 按一定的层次结构组成的, 每一个对象都代表了 AutoCAD 中的一个明确的功能。每一个对象本身包含着自己的属性和方法, 通过方法可以实现对象的一些操作。应用程序正是通过对 AutoCAD 暴露的各级对象的方法和属性的引用控制 AutoCAD 工作的。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 ActiveX Automation 编程的优点有:

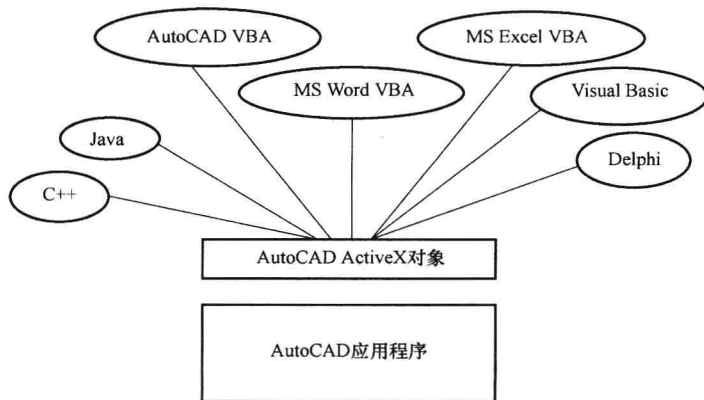


图 1.3-6 AutoCAD ActiveX 原理

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 AutoLISP 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。

1.3.5 AutoCAD.NET

从 AutoCAD2006 开始, Autodesk 为其开发增加了 .NET API。*.NET API 提供了一系列托管的外包类 (Managed Wrapper Class), 使开发人员可在 Microsoft .NET Framework 下, 使用任何支持 .NET 的语言, 如 VB.NET、C# 和 Managed C++ 等对 AutoCAD 进行二次开发。其优点是完全面向对象, 在拥有与 C++ 相匹配的强大功能的同时, 具有方便易用的特点, 是较理想的 AutoCAD 二次开发工具。

.NET API 与传统 ObjectARX 的区别主要源于在 .NET 环境下 (见图 1.3-7 AutoCAD.NET 开发环境) 开发应用程序与在 VC 环境下开发应用程序的区别。首先, 在 VC 环境下, 程序员需要自己管理内存的申请和释放, 而 .NET 采用了垃圾回收机制, 由 .NET 框架自行判断内存回收的时机并实行回收, 从而解决了令 C++ 程序员头痛的内存泄漏问题。也正是由于这个特点, 在 .NET 环境下不能像在 C++ 环境下那样利用析构函数释放其他的资源, 需要程序员在程序中显式地释放。在 .NET API 中, 主要通过 Dispose 函数进行资源的释放。

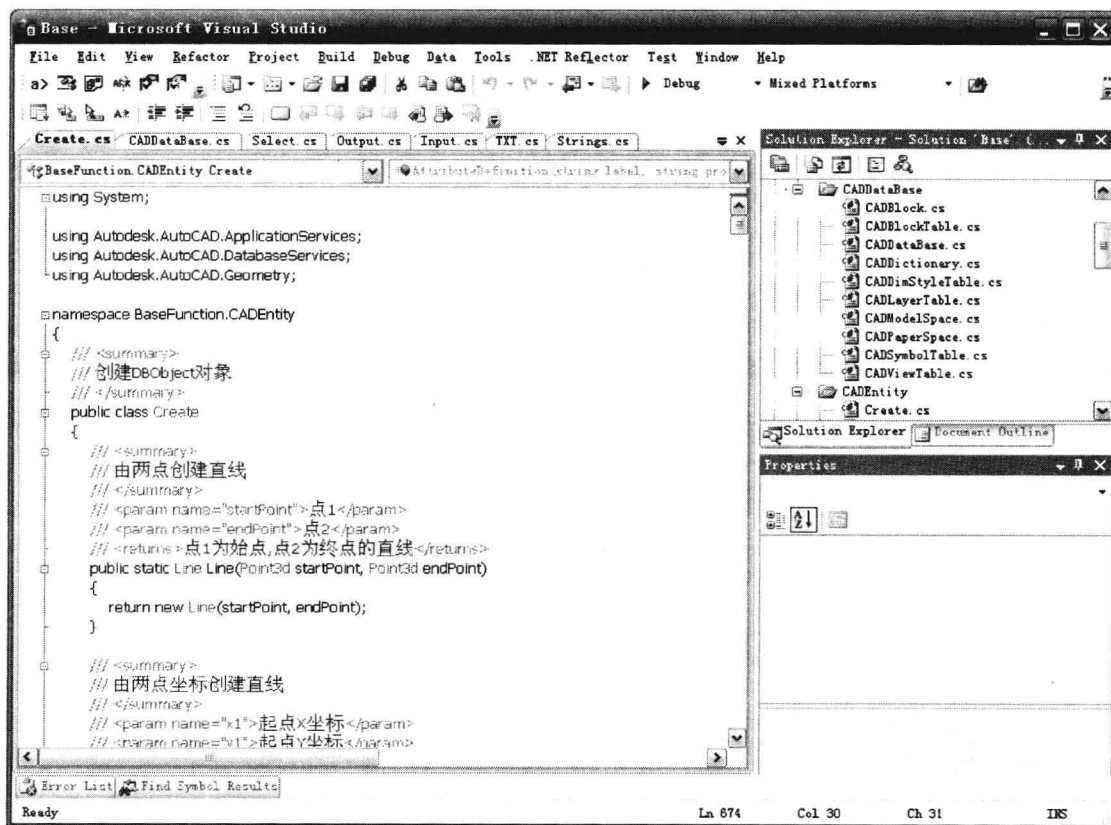


图 1.3-7 AutoCAD.NET 开发环境

其次，ObjectARX 中的各种反应器（Reactor）在 .NET API 中由外包类映射为各种事件（Event），可通过定义这些事件的响应函数来响应 AutoCAD 的各种操作。同时对于错误信息的处理也从函数返回值改变为通常异常来处理，使其更好地兼容 .NET。由于 VB.NET、C# 等语言都是完全面向对象的，没有全局函数的概念，所以 .NET API 将 ObjectARX 下的全局函数封装为 .NET API 下的某些对象或对象的属性，如 ObjectARX 下与用户交互的系列全局函数被封装为 CommandLinePrompt 类。

1.4 开发方式对比

1. 开发方式对应的编程语言见图 1.4-1 所示 AutoCAD 开发方式-编程语言对应关系。

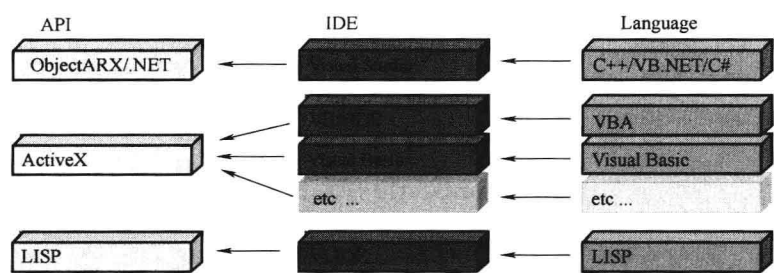


图 1.4-1 AutoCAD 开发方式-编程语言对应关系

2. 开发方式能力对比见图 1.4-2 AutoCAD 开发方式能力范围及图 1.4-3 AutoCAD 开发方式能力。

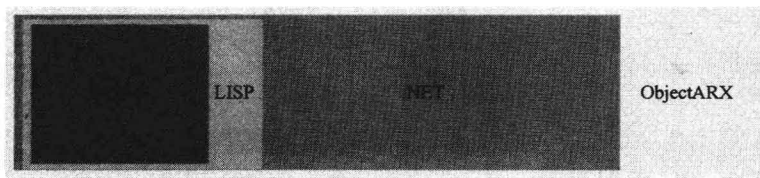


图 1.4-2 AutoCAD 开发方式能力范围

	LISP	NET	ObjectARX
API操作能力	No	No	Yes
高级用户界面	No	No	Yes
底层事件	No	No	Yes
高级API访问	No	No	Yes
完善的异常处理	No	No	Yes
64位/UNICODE支持	Yes	Tricky	No
垃圾回收	Yes	Yes	No
混合语言开发	No	No	Difficult
混合API's交互性	Limited	Limited	Difficult
语法简洁	Yes	Yes	No
自定义实体	No	No	Yes
性能	No	No	Yes++

图 1.4-3 AutoCAD 开发方式能力