

交互式CAD系统开发基础系列丛书  
交互式CAD系统开发基础系列丛书  
交互式CAD系统开发基础系列丛书

# 用Visual C++.NET 开发交互式CAD系统

黄国明 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

交互式 CAD 系统开发基础系列丛书

# 用 Visual C++.NET 开发交互式 CAD 系统

黄国明 编著



B1280712

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书从人机交互的角度,由浅入深、循序渐进地讲述了如何在 Visual C++.NET 环境下开发矢量图形系统。从向导生成的初始代码,分析了 MFC 框架体系结构,讨论了 MFC 程序的运行机制。结合具体程序开发,重点讨论了图元类、交互操作类的抽象、设计及管理方法,实现了矢量图形系统无级缩放、交互编辑、文档编辑、文档打印等基本功能。

本书注重整体设计,特别是图元、交互工具这几个类的划分非常清楚,每个类的抽象也非常严谨,读者可以非常容易地从图元、交互操作工具这两个方面扩展该图形系统,使之满足自己的需要。

本书适合于有限元、CAD、GIS 等领域从事软件开发的技术人员以及大专院校的师生阅读、参考。书中的代码在 Visual C++.NET 环境下编译,随书发行的光盘包含了每一个开发步骤的完整代码。

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

用 Visual C++.NET 开发交互式 CAD 系统/黄国明编著. —北京:电子工业出版社,2003.9

(交互式 CAD 系统开发基础系列丛书)

ISBN 7-5053-9162-3

I. 用… II. 黄… III. ①C 语言—程序设计②计算机辅助设计 IV. ①TP312②TP391.72

中国版本图书馆 CIP 数据核字(2003)第 083150 号

责任编辑:王昌铭

印刷:北京增富印刷有限公司

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:28 字数:714 千字 附光盘 1 张

版次:2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

印数:5000 册 定价:43.00 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。质量投诉请发邮件至 [zltts@phei.com.cn](mailto:zltts@phei.com.cn),盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

# 前 言

随着计算机技术的不断提高,计算机在科学计算方面的需求不断增加,其领域包括计算机辅助设计、计算流体力学、有限元分析等。这类应用的特点是数据量大,采用传统的方式准备数据效率非常低,而且容易出错。科学计算可视化与交互式建模是随这类应用而发展起来的,现已成为最活跃的研究领域之一。

目前市面上计算机图形编程方面的书不少,但大多讲解编程语言在图形编程方面的能力,罗列绘图函数的功能,孤立地实现各种图形元的绘制,而对坐标体系、图形无级缩放、图形复制、粘贴等必须的功能则很少涉及。

本书从人机交互的角度,讲解图形系统设计方法。它的特点是注重整体设计,特别是图元类、交互工具类的划分非常清楚、层次合理,每个类的抽象也非常严谨。这样,读者可以非常容易地扩展该图形系统,使之满足自己的需要。

## 本书概要

本书共 18 章,根据内容,大体可以分为以下 6 个方面。

**Visual C++.NET 环境部分:**第 1 章到第 3 章,介绍了 Visual C++.NET 新增功能、开发环境以及 Visual C++.NET 应用程序调试技术。

**图形编程基础:**第 4 章和第 5 章,介绍了面向对象程序设计的基本知识和图形程序设计的基础,重点是 MFC 设备描述表、坐标空间、映射模式这几方面的内容。

**MFC 应用程序框架:**第 6 章和第 7 章。这两章从分析应用向导生成的初始程序代码入手,介绍 MFC 程序框架、主要的类及其之间的关系。

**交互图形系统基础:**第 8 章和第 9 章。这两章快速实现一个所见即所得的应用程序。第 8 章采用菜单命令绘制随机的矩形、椭圆和直线这 3 种图形。第 9 章采用鼠标消息来绘制这些图形。到第 9 章结束时,图形采用结构数组来管理。

**交互图形设计:**这是本书的核心,包括第 10 章到 16 章的内容。第 10 章是视图基类设计,目的是实现一个支持无级缩放、可滚动的视图类。这一章的重点是关于坐标转换的讨论。第 11 章是实现视图,其具体内容是从 10 章实现的视图基类派生出视图类,在这个类中实现滚动、图形缩放、图形漫游、坐标转换等功能。第 12 章设计了图形系统的数据结构,具体讨论了几种可能的保存图形对象的数据结构,并给出了选择图形结构的一般原则。第 13 章为文档对象设计,它的目的是把第 9 章中用结构表示的图形抽象成类。到第 13 章结束时,图形系统采用对象指针链表来管理图形对象。第 14 章是交互操作设计,主要设计一个类层次来实现不同图形的创建、编辑、拾取等操作。在这一章中,首先设计一个交互操作的基类,然后从基类派生出矩形工具类、圆工具类、圆弧工具类和多边形工具类。读者可以方便地扩展其他的工具类。第 15 章实现图形拾取,讨论了图形拾取机制和不同图元拾取的算法。第 16 章为图形编辑,包括两方面的内容。一是图形对象的编辑,比如图形的修改、移动等;二是标准文档编辑功能,比如复制、粘贴、剪切、删除等。

**交互图形系统优化设计：**第 17 章和第 18 章。第 17 章是界面设计，主要讨论了工具箱、对话框和状态栏坐标显示的设计。第 18 章是文档打印，涉及到文档基本打印程序设计方面的内容，包括分页、打印页眉、页脚。

## 本书光盘

本书从第 8 章起开发一个完整的图形程序，每一章都在前一章代码的基础上实现新的功能。光盘上把不同章节中的代码分别存放，同时，光盘上还包含了环境和安装方面的说明。

## 对读者的要求

阅读本书要求读者有 C++ 方面的基础知识，但不要求读者有窗口系统编程的知识。本书对 MFC 类函数没有做过多的解释，读者可以参考 Visual C++.NET 的在线帮助。如果读者没有 Visual C++.NET 环境，本书开发的实例在 Visual C++6.0 环境下也可编译。

本书从策划到写作一直得到了苏金明的帮助并提出了宝贵的意见。本书的第 4 章和第 5 章的 1, 2, 3, 7 节由刘兵编写。

编著者 2003 年 6 月

E-mail:hgm@sc.cei.gov.cn

# 目 录

|                           |      |
|---------------------------|------|
| 第 1 章 Visual C++.NET 新增功能 | (1)  |
| 1.1 .NET 框架与托管代码          | (1)  |
| 1.1.1 .NET 框架             | (1)  |
| 1.1.2 公共语言运行时             | (1)  |
| 1.1.3 托管代码与非托管代码          | (2)  |
| 1.1.4 .NET 框架类库           | (2)  |
| 1.2 Visual C++.NET 新名词    | (3)  |
| 1.2.1 解决方案                | (3)  |
| 1.2.2 起始页                 | (4)  |
| 1.3 开发环境变化                | (4)  |
| 第 2 章 Visual C++.NET 开发环境 | (6)  |
| 2.1 新版概述                  | (6)  |
| 2.2 开发环境的菜单功能             | (8)  |
| 2.2.1 文件菜单                | (8)  |
| 2.2.2 编辑菜单                | (9)  |
| 2.2.3 视图菜单                | (9)  |
| 2.2.4 项目菜单                | (11) |
| 2.2.5 生成菜单                | (12) |
| 2.2.6 调试菜单                | (12) |
| 2.2.7 工具菜单                | (13) |
| 2.2.8 窗口菜单                | (14) |
| 2.2.9 帮助菜单                | (14) |
| 2.3 方案导航区                 | (15) |
| 2.3.1 方案浏览区               | (16) |
| 2.3.2 类视图                 | (18) |
| 2.3.3 资源视图                | (19) |
| 2.4 界面设置                  | (20) |
| 2.5 创建应用程序                | (21) |
| 2.5.1 使用向导创建应用程序          | (21) |
| 2.5.2 添加代码                | (26) |
| 2.5.3 添加类                 | (26) |
| 2.5.4 添加变量                | (27) |
| 2.5.5 添加函数                | (28) |
| 2.5.6 添加消息映射函数            | (30) |

|                            |      |
|----------------------------|------|
| <b>第 3 章 程序调试</b>          | (32) |
| 3.1 Visual C++.NET 调试器     | (32) |
| 3.1.1 Visual C++.NET 调试器概览 | (32) |
| 3.1.2 调试环境的建立              | (32) |
| 3.1.3 如何设置断点               | (34) |
| 3.1.4 控制程序的运行              | (36) |
| 3.1.5 查看工具的使用              | (37) |
| 3.2 高级调试技术                 | (41) |
| 3.2.1 MFC 调试机制             | (41) |
| 3.2.2 TRACE 宏的利用           | (42) |
| 3.2.3 ASSERT 宏的利用          | (42) |
| 3.2.4 ASSERT_VALID 宏的利用    | (43) |
| 3.2.5 内存漏洞的检查              | (43) |
| <b>第 4 章 面向对象的程序设计</b>     | (45) |
| 4.1 面向对象的程序设计              | (45) |
| 4.1.1 面向对象的概念              | (45) |
| 4.1.2 数据封装                 | (46) |
| 4.1.3 继承                   | (46) |
| 4.1.4 多态性                  | (47) |
| 4.2 类和对象                   | (47) |
| 4.2.1 类                    | (48) |
| 4.2.2 构造函数和析构函数            | (50) |
| 4.2.3 重载构造函数               | (52) |
| 4.2.4 类的指针                 | (53) |
| 4.3 类的继承性                  | (54) |
| 4.3.1 继承性                  | (54) |
| 4.3.2 继承和构造函数、析构函数         | (56) |
| 4.3.3 多重继承                 | (58) |
| 4.4 类的多态性                  | (60) |
| 4.4.1 指向基类的指针              | (60) |
| 4.4.2 虚函数                  | (61) |
| 4.4.3 纯虚函数及抽象类             | (62) |
| <b>第 5 章 MFC 图形编程基础</b>    | (65) |
| 5.1 图形设备接口                 | (65) |
| 5.2 设备描述表                  | (66) |
| 5.2.1 设备描述表类型              | (66) |
| 5.2.2 MFC 设备描述表类           | (67) |
| 5.2.3 基类: CDC              | (67) |
| 5.2.4 用类 CPaintDC 绘图       | (68) |
| 5.2.5 用类 CClientDC 管理用户区   | (68) |

|              |                                          |              |
|--------------|------------------------------------------|--------------|
| 5.3          | 图形对象 .....                               | (70)         |
| 5.3.1        | 笔: 类 CPen .....                          | (70)         |
| 5.3.2        | 刷子: 类 CBrush .....                       | (72)         |
| 5.3.3        | 字体: 类 CFont .....                        | (74)         |
| 5.3.4        | 区域: 类 CRgn .....                         | (76)         |
| 5.4          | 坐标空间 .....                               | (77)         |
| 5.4.1        | 物理设备坐标空间 .....                           | (78)         |
| 5.4.2        | 设备坐标空间 .....                             | (78)         |
| 5.4.3        | 逻辑坐标空间 .....                             | (79)         |
| 5.5          | 映射模式 .....                               | (81)         |
| 5.5.1        | 坐标映射 .....                               | (81)         |
| 5.5.2        | MM_TEXT 映射模式 .....                       | (82)         |
| 5.5.3        | MM_LOENGLISH 和 MM_HIENGLISH 映射模式 .....   | (82)         |
| 5.5.4        | MM_LOMETRIC 和 MM_HIMETRIC 映射模式 .....     | (84)         |
| 5.5.5        | MM_TWIPS 映射模式 .....                      | (85)         |
| 5.5.6        | MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式 ..... | (85)         |
| 5.5.7        | 映射模式示例 .....                             | (86)         |
| 5.6          | 窗口与视口函数 .....                            | (90)         |
| 5.6.1        | 窗口与视口原点 .....                            | (90)         |
| 5.6.2        | 设定窗口与视口范围 .....                          | (92)         |
| 5.7          | 矢量图形和正文 .....                            | (93)         |
| 5.7.1        | 矢量图形绘图方式 .....                           | (93)         |
| 5.7.2        | 绘制点 .....                                | (94)         |
| 5.7.3        | 绘制线和多边线 .....                            | (95)         |
| 5.7.4        | 绘制矩形 .....                               | (99)         |
| 5.7.5        | 绘制区域 .....                               | (101)        |
| 5.7.6        | 绘制正文 .....                               | (104)        |
| <b>第 6 章</b> | <b>MFC 生成的骨干程序 .....</b>                 | <b>(107)</b> |
| 6.1          | VisDraw 最终版本介绍 .....                     | (107)        |
| 6.1.1        | VisDraw 界面 .....                         | (107)        |
| 6.1.2        | VisDraw 功能 .....                         | (108)        |
| 6.1.3        | VisDraw 开发步骤 .....                       | (109)        |
| 6.2          | 使用 AppWizard 生成 VisDraw 框架 .....         | (110)        |
| 6.2.1        | 创建 VisDraw 项目 .....                      | (111)        |
| 6.2.2        | 指定应用程序类型 .....                           | (113)        |
| 6.2.3        | 指定复合文档选项 .....                           | (114)        |
| 6.2.4        | 指定文档模板字符串 .....                          | (115)        |
| 6.2.5        | 指定数据库选项 .....                            | (116)        |
| 6.2.6        | 指定应用程序外观 .....                           | (116)        |
| 6.2.7        | 指定应用程序的附加支持 .....                        | (118)        |



|                                    |       |
|------------------------------------|-------|
| 6.2.8 查看并为应用程序指定基类 .....           | (119) |
| 6.3 AppWizard 生成的文件 .....          | (120) |
| 6.3.1 自述文件 .....                   | (121) |
| 6.3.2 项目文件 .....                   | (121) |
| 6.3.3 应用程序源文件和头文件 .....            | (121) |
| 6.3.4 资源文件 .....                   | (121) |
| 6.3.5 预编译的头文件 .....                | (122) |
| 6.3.6 帮助文件 .....                   | (122) |
| 6.4 运行 VisDraw .....               | (122) |
| 6.4.1 编译 VisDraw 程序 .....          | (122) |
| 6.4.2 运行 VisDraw 应用程序 .....        | (123) |
| 6.4.3 VisDraw 的功能 .....            | (123) |
| <b>第 7 章 MFC 应用程序框架</b> .....      | (126) |
| 7.1 MFC 应用程序框架概述 .....             | (126) |
| 7.1.1 封装 .....                     | (126) |
| 7.1.2 继承 .....                     | (126) |
| 7.1.3 虚拟函数和动态约束 .....              | (127) |
| 7.1.4 MFC 的宏观框架体系 .....            | (127) |
| 7.2 VisDraw 的类和文件 .....            | (128) |
| 7.2.1 浏览 VisDraw 类和文件 .....        | (128) |
| 7.2.2 VisDraw 类的层次结构 .....         | (129) |
| 7.2.3 VisDraw 宏和全局函数 .....         | (130) |
| 7.2.4 CObject 类特性 .....            | (132) |
| 7.3 VisDraw 的重要组成部分及其之间的相互关系 ..... | (134) |
| 7.3.1 应用程序对象 .....                 | (135) |
| 7.3.2 主框架窗口对象 .....                | (137) |
| 7.3.3 文档对象 .....                   | (140) |
| 7.3.4 视图对象 .....                   | (141) |
| 7.3.5 文档/视图结构 .....                | (144) |
| 7.4 VisDraw 的运行机制 .....            | (145) |
| 7.4.1 调用 CWinApp 类构造函数 .....       | (146) |
| 7.4.2 WinMain 接收控制 .....           | (147) |
| 7.4.3 进入消息循环 .....                 | (149) |
| 7.5 文档模板的意义 .....                  | (152) |
| 7.6 VisDraw 的消息映射 .....            | (154) |
| 7.6.1 MFC 处理的三类消息 .....            | (154) |
| 7.6.2 MFC 消息映射的实现方法 .....          | (155) |
| 7.6.3 常用的消息映射宏 .....               | (155) |
| 7.6.4 VisDraw 的消息映射 .....          | (156) |

|                       |       |
|-----------------------|-------|
| <b>第 8 章 菜单</b>       | (158) |
| 8.1 菜单资源              | (158) |
| 8.2 菜单属性              | (159) |
| 8.3 菜单助记符             | (160) |
| 8.4 菜单快捷键表            | (161) |
| 8.4.1 为菜单命令添加快捷键      | (161) |
| 8.4.2 创建快捷键表项         | (161) |
| 8.4.3 设置快捷键属性         | (162) |
| 8.4.4 添加快捷键消息处理       | (162) |
| 8.5 为 VisDraw 添加菜单    | (163) |
| 8.5.1 添加顶层菜单          | (164) |
| 8.5.2 添加子菜单           | (164) |
| 8.6 添加命令处理函数          | (165) |
| 8.7 消息映射与命令路径         | (167) |
| 8.7.1 Windows 消息分类    | (167) |
| 8.7.2 消息映射            | (168) |
| 8.8 实现消息处理函数          | (170) |
| 8.8.1 添加视图类成员变量和函数    | (170) |
| 8.8.2 测试 VisDraw 应用程序 | (173) |
| <b>第 9 章 所见即所得绘图</b>  | (174) |
| 9.1 采用鼠标绘图            | (174) |
| 9.1.1 鼠标消息            | (174) |
| 9.1.2 非客户区鼠标消息        | (176) |
| 9.1.3 绘图过程中的鼠标消息      | (176) |
| 9.1.4 捕获鼠标和设置鼠标状态     | (178) |
| 9.1.5 实现鼠标绘图的橡皮筋效果    | (179) |
| 9.1.6 添加鼠标消息处理函数      | (180) |
| 9.2 实现鼠标绘图            | (182) |
| 9.2.1 定义视图类数据成员       | (182) |
| 9.2.2 实现鼠标消息函数        | (183) |
| 9.3 运行 VisDraw 应用程序   | (187) |
| 9.3.1 修改代码            | (187) |
| 9.3.2 VisDraw 目前存在的问题 | (189) |
| 9.4 VisDraw 的初步改进     | (189) |
| 9.4.1 修改成员变量          | (190) |
| 9.4.2 修改鼠标消息处理函数      | (191) |
| 9.4.3 实现 OnDraw 函数    | (193) |
| 9.4.4 运行 VisDraw 应用程序 | (194) |
| <b>第 10 章 视图基类设计</b>  | (196) |
| 10.1 屏幕滚动             | (196) |

|               |                               |              |
|---------------|-------------------------------|--------------|
| 10.1.1        | VisDraw 目前的缺陷 .....           | (196)        |
| 10.1.2        | 滚动的基本理论 .....                 | (197)        |
| 10.1.3        | 实现滚动必须完成的任务 .....             | (198)        |
| 10.2          | 添加视图基类 .....                  | (199)        |
| 10.2.1        | 由向导生成的视图基类 .....              | (199)        |
| 10.2.2        | 添加成员变量 .....                  | (201)        |
| 10.3          | 设备坐标与逻辑坐标的转化 .....            | (202)        |
| 10.3.1        | 坐标映射过程 .....                  | (203)        |
| 10.3.2        | 映射模式 .....                    | (204)        |
| 10.3.3        | 为什么需要坐标转换 .....               | (205)        |
| 10.3.4        | 实现坐标转换函数 .....                | (206)        |
| 10.3.5        | 客户区中心逻辑坐标 .....               | (207)        |
| 10.4          | 滚动视图函数重载 .....                | (210)        |
| 10.4.1        | 设置滚动尺寸 .....                  | (210)        |
| 10.4.2        | 设定视图区中心 .....                 | (213)        |
| 10.4.3        | 得到滚动位置 .....                  | (215)        |
| 10.4.4        | 自动调整视图大小 .....                | (216)        |
| 10.5          | 实现视图滚动 .....                  | (217)        |
| 10.5.1        | 滚动条消息处理 .....                 | (217)        |
| 10.5.2        | 使用键盘滚动视图 .....                | (220)        |
| <b>第 11 章</b> | <b>实现视图 .....</b>             | <b>(223)</b> |
| 11.1          | 实现滚动功能 .....                  | (223)        |
| 11.1.1        | 修改视图基类 .....                  | (223)        |
| 11.1.2        | 设置滚动区域大小 .....                | (223)        |
| 11.1.3        | 设备坐标转化为逻辑坐标 .....             | (225)        |
| 11.2          | 建立实际坐标系 .....                 | (228)        |
| 11.2.1        | 建立坐标系 .....                   | (228)        |
| 11.2.2        | 实际坐标与逻辑坐标的转换 .....            | (229)        |
| 11.2.3        | 绘制网格线 .....                   | (231)        |
| <b>第 12 章</b> | <b>VisDraw 文档数据对象设计 .....</b> | <b>(236)</b> |
| 12.1          | 抽象图形元的设计 .....                | (236)        |
| 12.1.1        | 图形元的数据结构 .....                | (236)        |
| 12.1.2        | 添加图形元基类 .....                 | (237)        |
| 12.1.3        | 图形元基类的数据成员 .....              | (241)        |
| 12.1.4        | 图形元基类的成员函数 .....              | (243)        |
| 12.2          | 矩形图元类 .....                   | (245)        |
| 12.2.1        | 向导生成的矩形图元类 .....              | (245)        |
| 12.2.2        | 添加成员变量 .....                  | (245)        |
| 12.2.3        | 边界矩形盒的计算 .....                | (247)        |
| 12.2.4        | 实现图形绘制 .....                  | (248)        |

|               |                        |       |
|---------------|------------------------|-------|
| 12.3          | 点图元类 .....             | (250) |
| 12.3.1        | 向导生成的点图元类 .....        | (250) |
| 12.3.2        | 实现成员函数 .....           | (251) |
| 12.4          | 圆图元类 .....             | (254) |
| 12.4.1        | 向导生成的圆图元类 .....        | (254) |
| 12.4.2        | 添加成员变量和函数 .....        | (256) |
| 12.4.3        | 计算边界矩形 .....           | (258) |
| 12.5          | 圆弧图元类 .....            | (260) |
| 12.5.1        | 向导生成的圆弧图元类 .....       | (260) |
| 12.5.2        | 已知圆心计算圆弧其他参数 .....     | (261) |
| 12.5.3        | 由圆弧上三点计算圆弧参数 .....     | (263) |
| 12.5.4        | 绘制圆弧 .....             | (265) |
| 12.5.5        | 圆弧边界矩形的计算 .....        | (267) |
| 12.6          | 多边形图元类 .....           | (269) |
| 12.6.1        | 向导生成的多边形图元类 .....      | (270) |
| 12.6.2        | 添加成员变量 .....           | (271) |
| 12.6.3        | 绘制多边形图元 .....          | (274) |
| 12.6.4        | 计算多边形图元边界矩形 .....      | (275) |
| 12.6.5        | 添加多边形顶点 .....          | (277) |
| <b>第 13 章</b> | <b>文档设计</b> .....      | (279) |
| 13.1          | 文档/视图结构 .....          | (279) |
| 13.2          | 视图与文档之间通信 .....        | (280) |
| 13.3          | 数据结构设计 .....           | (281) |
| 13.3.1        | 为文档选择合适的数据结构 .....     | (282) |
| 13.3.2        | VisDrawDoc 的成员变量 ..... | (284) |
| 13.3.3        | VisDrawDoc 的成员函数 ..... | (286) |
| 13.3.4        | VisDrawDoc 的文档界面 ..... | (287) |
| 13.4          | 使用文档对象 .....           | (289) |
| 13.4.1        | 删除代码 .....             | (289) |
| 13.4.2        | 添加和修改代码 .....          | (291) |
| 13.5          | 运行 VisDraw .....       | (295) |
| 13.6          | 保存文档数据 .....           | (296) |
| 13.6.1        | 序列化和反序列化 .....         | (296) |
| 13.6.2        | 序列化机制 .....            | (297) |
| 13.6.3        | 图形元序列化 .....           | (298) |
| 13.6.4        | 在文档中序列化所有图形 .....      | (302) |
| 13.6.5        | VisDraw 程序版本控制 .....   | (305) |
| <b>第 14 章</b> | <b>交互操作设计</b> .....    | (306) |
| 14.1          | 图形交互问题 .....           | (306) |
| 14.1.1        | 与鼠标相关的函数 .....         | (306) |

MTS22/02

|               |                       |              |
|---------------|-----------------------|--------------|
| 14.1.2        | 捕捉鼠标输入 .....          | (307)        |
| 14.1.3        | 在屏幕上拖动图形 .....        | (308)        |
| 14.1.4        | 保存图形对象到文档 .....       | (308)        |
| 14.1.5        | 将图形以实际数据重画 .....      | (308)        |
| 14.1.6        | 图形对象的拾取 .....         | (308)        |
| 14.2          | VisDraw 交互操作的缺陷 ..... | (308)        |
| 14.3          | 交互工具框架设计 .....        | (310)        |
| 14.4          | 交互操作基类实现 .....        | (312)        |
| 14.4.1        | 向导生成的交互操作基类 .....     | (312)        |
| 14.4.2        | 添加成员变量 .....          | (313)        |
| 14.4.3        | 检索交互工具对象指针 .....      | (315)        |
| 14.5          | 矩形工具类 .....           | (318)        |
| 14.5.1        | 向导生成的矩形工具类 .....      | (319)        |
| 14.5.2        | 添加图形工具对象指针到链表 .....   | (319)        |
| 14.5.3        | 实现鼠标函数 .....          | (320)        |
| 14.5.4        | 修改视图类代码 .....         | (325)        |
| 14.5.5        | 运行 VisDraw .....      | (326)        |
| 14.6          | 圆工具设计 .....           | (327)        |
| 14.6.1        | 向导生成的圆工具类 .....       | (327)        |
| 14.6.2        | 实现鼠标函数 .....          | (328)        |
| 14.6.3        | 运行 VisDraw .....      | (331)        |
| 14.7          | 圆弧工具 .....            | (331)        |
| 14.7.1        | 向导生成的圆弧工具类 .....      | (332)        |
| 14.7.2        | 添加画弧工具 .....          | (333)        |
| 14.7.3        | 鼠标处理函数的框架结构 .....     | (334)        |
| 14.7.4        | 按下鼠标左键操作函数 .....      | (335)        |
| 14.7.5        | 画弧时的屏幕反馈信息 .....      | (337)        |
| 14.7.6        | 测试圆弧工具 .....          | (340)        |
| 14.8          | 多边形工具类 .....          | (340)        |
| 14.8.1        | 向导生成的多边形工具类 .....     | (341)        |
| 14.8.2        | 初始化多边形工具 .....        | (341)        |
| 14.8.3        | 实现鼠标函数 .....          | (342)        |
| 14.8.4        | 测试多边形工具 .....         | (344)        |
| <b>第 15 章</b> | <b>图形拾取 .....</b>     | <b>(346)</b> |
| 15.1          | 图形拾取功能和机制 .....       | (346)        |
| 15.1.1        | 图形拾取功能描述 .....        | (346)        |
| 15.1.2        | 图形拾取机制 .....          | (347)        |
| 15.2          | 图形拾取算法 .....          | (348)        |
| 15.2.1        | 边界矩形击中测试 .....        | (348)        |
| 15.2.2        | 图形元素拾取条件 .....        | (350)        |

|               |                   |       |
|---------------|-------------------|-------|
| 15.2.3        | 点的拾取 .....        | (351) |
| 15.2.4        | 矩形和直线的拾取 .....    | (352) |
| 15.2.5        | 圆的拾取 .....        | (354) |
| 15.2.6        | 圆弧的拾取 .....       | (355) |
| 15.2.7        | 多边形的拾取 .....      | (356) |
| 15.3          | 实现图元拾取 .....      | (357) |
| 15.3.1        | 单击图形拾取判断 .....    | (357) |
| 15.3.2        | 添加选择集 .....       | (358) |
| 15.4          | 拾取图元显示策略 .....    | (360) |
| 15.4.1        | 图形对象的关键点 .....    | (360) |
| 15.4.2        | 矩形类的关键点 .....     | (363) |
| 15.4.3        | 圆类的关键点 .....      | (364) |
| 15.4.4        | 圆弧类的关键点 .....     | (365) |
| 15.4.5        | 多边形类的关键点 .....    | (367) |
| 15.4.6        | 点图元的关键点 .....     | (368) |
| 15.4.7        | 图元关键点的绘制 .....    | (369) |
| 15.5          | 交互图形拾取 .....      | (371) |
| 15.5.1        | 向导添加的选择工具 .....   | (371) |
| 15.5.2        | 拾取操作 .....        | (372) |
| 15.5.3        | 点选操作 .....        | (373) |
| 15.5.4        | 窗口拾取 .....        | (374) |
| <b>第 16 章</b> | <b>图形编辑</b> ..... | (378) |
| 16.1          | 修改图形 .....        | (378) |
| 16.1.1        | 关键点击中测试 .....     | (378) |
| 16.1.2        | 修改关键点坐标 .....     | (379) |
| 16.1.3        | 矩形关键点坐标的修改 .....  | (380) |
| 16.1.4        | 圆图形关键点的修改 .....   | (382) |
| 16.1.5        | 圆弧关键点坐标的修改 .....  | (382) |
| 16.1.6        | 多边形关键点坐标的修改 ..... | (383) |
| 16.1.7        | 修改图形操作 .....      | (384) |
| 16.2          | 移动图形 .....        | (389) |
| 16.2.1        | 矩形图元的平移 .....     | (390) |
| 16.2.2        | 圆的平移 .....        | (391) |
| 16.2.3        | 多边形平移 .....       | (392) |
| 16.2.4        | 点的平移 .....        | (393) |
| 16.3          | 标准编辑 .....        | (393) |
| 16.3.1        | Windows 剪贴板 ..... | (394) |
| 16.3.2        | 复制数据 .....        | (394) |
| 16.3.3        | 粘贴数据 .....        | (397) |
| 16.3.4        | 删除 .....          | (399) |

|                                              |       |
|----------------------------------------------|-------|
| 16.3.5 剪切 .....                              | (400) |
| 16.3.6 全选 .....                              | (401) |
| <b>第 17 章 界面设计</b> .....                     | (402) |
| 17.1 添加工具栏 .....                             | (402) |
| 17.1.1 工具栏的可视化设计 .....                       | (402) |
| 17.1.2 创建工具栏 .....                           | (404) |
| 17.1.3 工具栏的隐藏/显示 .....                       | (408) |
| 17.1.4 命令更新 .....                            | (409) |
| 17.2 状态栏 .....                               | (410) |
| 17.3 对话框与控件 .....                            | (412) |
| 17.3.1 对话框的基本概念 .....                        | (413) |
| 17.3.2 控件的基本概念 .....                         | (413) |
| 17.3.3 对话框模板的设计 .....                        | (414) |
| 17.3.4 对话框类设计 .....                          | (414) |
| 17.3.5 对话框的调用 .....                          | (417) |
| 17.4 弹出式菜单 .....                             | (419) |
| <b>第 18 章 文档打印</b> .....                     | (421) |
| 18.1 打印设计 .....                              | (421) |
| 18.1.1 MFC 打印体系结构 .....                      | (421) |
| 18.1.2 采用 MFC 进行打印程序设计 .....                 | (422) |
| 18.2 改变映射模式 .....                            | (423) |
| 18.3 对文档编写页码 .....                           | (423) |
| 18.3.1 计算可打印区域 .....                         | (424) |
| 18.3.2 MFC 在何处中断页面的打印 .....                  | (425) |
| 18.4 纵向打印和横向打印 .....                         | (427) |
| 18.5 添加页眉和页脚 .....                           | (428) |
| 18.5.1 添加 PrintHeader 和 PrintFooter 函数 ..... | (428) |
| 18.5.2 添加页眉 .....                            | (429) |
| 18.5.3 添加页脚 .....                            | (430) |
| <b>参考文献</b> .....                            | (432) |

# 第 1 章 Visual C++.NET 新增功能

Microsoft 推出的 Visual C++ 和 Microsoft 基本类 (Microsoft Foundation Class, MFC) 库成功地将面向对象和事件驱动编程概念联系起来, 使编写 Windows 应用程序的过程变得简单、方便, 而且代码量小。新版本的 Visual Studio 选择了与 Microsoft 的 .NET Framework 同时发布, .NET Framework 为创建 Web 应用程序和单机程序提供了一个强大的、语言中立的开发环境。新版本中, Visual C++ 升级为 Visual C++.NET。

本章首先适当地介绍一下 .NET Framework, 然后研究 Visual C++ (现在为 Visual C++.NET) 有哪些变化, 这将决定您今后如何使用 Visual C++ 创建应用程序; 最后介绍为开发 MFC 应用程序, Visual C++.NET 所新增的功能和变化。

本章内容:

- ◆ .NET 框架
- ◆ Visual C++.NET 新名词
- ◆ 开发环境的变化

## 1.1 .NET 框架与托管代码

### 1.1.1 .NET 框架

Microsoft 为开发人员提供了 .NET Framework, 它是一组服务、类及数据类型, 能提高开发人员的开发效率, 并且能够更加容易地使用 Windows 操作系统所提供的一组底层功能。 .NET Framework 使开发人员能够集中精力去更好地实现应用程序的功能, 而不必担心具体的管理细节。

.NET Framework 的体系结构如图 1-1 所示。

.NET Framework 由彼此独立又相互关联的两部分组成: 公共语言运行时 (CLR) 和基类库。CLR 是它为我们提供的服务, 基类库是它实现的功能。 .NET 的大部分特性, 如垃圾收集、版本控制、线程管理等, 都使用了 CLR 提供的服务。

### 1.1.2 公共语言运行时

当为 .NET Framework 编译源代码时, 得到的目标代码并不能够直接在处理器上运行。为 .NET 进行编译实际上是在为 CLR 进行编译, 因为编译得到的代码并不是 CPU 能够识别的代码, 而是一种叫做“微软中间语言 (MSIL, 简称 IL)”的新语言的代码。MSIL 为一个虚拟的 CPU 定义了一套处理指令, 已

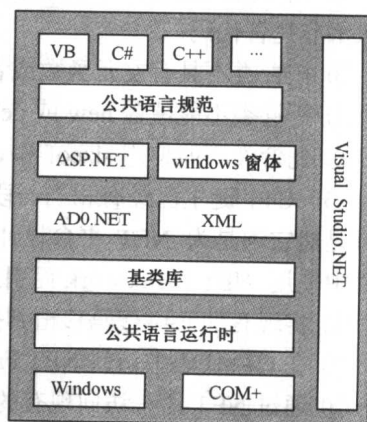


图 1-1 .NET Framework 体系结构



编译为 IL 的代码必须进一步编译成本机的机器指令后才能在 CPU 上运行。CLR 提供了一个实时编译器，用来把 IL 代码编译成本机机器代码。虽然这个过程进行了大量的优化，但还是没有直接把源代码编译成机器指令效率高。

在 CLR 控制下运行的代码称为托管代码，因为 CLR 定义了代码开发语言所必须遵守的规则。CLR 具有如下功能：

- (1) 通过类型安全检查、内存管理、无用单元回收以及异常处理来管理运行的代码。
- (2) 提供能够提高类型安全的通用类型系统，从而使代码更加稳定。
- (3) 提供访问系统资源的功能，包括 Windows API 和 COM InterOp 服务。
- (4) 提供跨语言支持，包括统一的异常处理和跨语言的调试。

### 1.1.3 托管代码与非托管代码

为 CLR 编写以及使用 CLR 提供的服务的代码为“托管代码”，那些未使用 CLR 服务的代码叫“非托管代码”。

目前 C++ 是惟一的既能够生成托管代码，又能够生成非托管代码的 .NET 语言。在面对 .NET 平台时，将面对一个关键的决定，就是开发托管代码，还是开发非托管代码或者把这两者混合起来。这可由下面的一些因素来进行选择。

(1) 非托管代码总是比托管代码迅速，因为它不具备任何与 CLR 服务相关的系统开销，比如垃圾收集和引用跟踪。运行速度对于很多应用程序来说是无紧要的，但是对某些应用程序（比如系统应用程序、工具、游戏等）是非常重要的。

(2) 非托管代码可以直接调用 Win32 API 函数和现有的 C++ 代码，而不必再花时间进行从托管代码到非托管代码的转换。

(3) 非托管代码比托管代码有更大的确定性。垃圾收集器的确切运行时间是不可预知的，这可能给非托管资源比如数据库连接、文件句柄和通信端口带来潜在的问题。

(4) 对托管类有以下几个特殊限制：

- ① 托管类只能从托管类派生，这就意味着从托管类派生的类总是托管的；
- ② 托管类只能支持单实现继承，如果编写的代码要利用多重继承并想保持它，那么需要保留非托管代码；

③ 托管类不具有友元函数或者友元类；

④ 托管类不能覆盖 new 或 delete 运算符；

⑤ 托管类不能使用 sizeof 和 offsetof 运算符。

(5) C++ 诞生已经有相当一段时间，在人们还没有开始设想 .NET 的时候就已经用于开发，显然它不是为 .NET 平台设计的。托管的概念就是为了允许我们在现有的和将来的 C++ 项目中使用 .NET Framework 而引入的，它适合下面的开发需求：

- ① 将现有的 C++ 代码移植到托管环境中；
- ② 在 C++ 代码中访问 .NET Framework 类；
- ③ 通过 .NET 语言访问现有的 C++ 代码。

### 1.1.4 .NET 框架类库

C++ 程序员都比较熟悉 MFC (Microsoft Foundation Class) 和活动模板库 (Active Template Library, ATL)。 .NET Framework 类与 MFC 相似，但是，利用 .NET Framework 类，开发人