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

電子工業出版社
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

電子工業出版社
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。

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 zlts@phei.com.cn，盗版侵权举报请发邮件至 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

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。

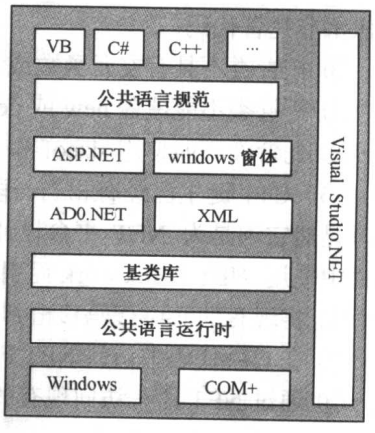

图 1-1　.NET Framework 体系结构

.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

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关紧要的，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