

计算机专业人员书库



精通 Visual C++ 图像处理编程

(第3版)

周长发 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

本书各章的源程序代码



计算机专业人员书库

ZHOUZHANGFA'S SERIES
周长发 作品系列

精通 Visual C++ 图像处理编程 (第3版)

周长发 编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书深入介绍了在 Windows 95/98/Me/NT/2000/XP 环境下图像处理的理论和编程技术,并基于面向对象的程序设计方法,详细地讨论了图像处理算法的 Visual C++编程技巧。全书共分 9 章,主要包括图像基本操作、调色板技术、图像的特技显示与擦除、图像空间变换、颜色处理、图像频域变换、图像处理、图像编码与格式等的理论与编程技术。

本书是图像处理和多媒体编程的实用参考书,适合图像处理和多媒体编程人员阅读,也可以作为大专院校计算机及相关专业师生的教学参考书。

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

精通 Visual C++图像处理编程 / 周长发编著. —3 版. —北京:电子工业出版社, 2006.6

(计算机专业人员书库)

ISBN 7-121-02541-8

I. 精… II. 周… III. C 语言—图象处理—程序设计 IV. ①TP391.41 ②TP312

中国版本图书馆 CIP 数据核字(2006)第 041168 号

责任编辑:李 冰

印 刷:北京天宇星印刷厂

出版发行:电子工业出版社

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:29.75 字数:720 千字

印 次:2006 年 6 月第 1 次印刷

印 数:5000 册 定价:49.00 元 (含光盘 1 张)

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

图像是现代信息社会中最基本的信息。数字图像处理的研究对象是图像处理技术，这是计算机技术的重要分支和发展方向。数字图像处理是信息科学、工程学、医学、生理学、遥感技术、地学甚至社会科学等许多学科的重要基础，同时也是一种具有巨大的经济和社会效益的实用技术，广泛应用于军事、气象、交通、探矿、水利、农业甚至娱乐等领域。随着信息高速公路和数字地球等概念的提出和发展，数字图像处理技术已得到更加充分的发展，并在现代信息社会中占据了非常重要的地位。

数字图像处理既是一门实用的学科，又需要一定的理论基础支持。许多的实际编程涉及图像处理算法的编程实现。本书尽量将图像处理的理论和实际编程结合起来，使读者既能从中获得必要的理论知识，又能学到实际的编程技巧，并直接将书中介绍的技术用于实际编程。

数字图像处理是一个很大的话题，本书不可能详细介绍其中的所有细节。与前两版《精通 Visual C++ 图像处理编程》一样，本书仍然着重于实用性，大量的篇幅用于介绍图像处理的编程技术。相对于前两版，本书增加了大量的内容，介绍了图像处理的常用算法和图像格式，使得本书的内容更为全面。此外，这一版还将书中介绍的所有技术集成到一个通用的图像处理类库中，以方便读者直接在实际的图像编程中应用。

本书的适用对象

本书面向所有希望了解和掌握数字图像处理编程的开发人员。只要对图像处理感兴趣，无论是刚刚接触这一技术，还是已有一定的工作经验的读者，都能从书中找到自己所需要的内容。

本书的所有程序都是基于面向对象编程技术并用 Visual C++ 来实现的，因此对那些想提高自己面向对象设计和 Visual C++ 编程技巧的读者来说，也是一本很好的参考书。

本书假定读者具有一定的 Visual C++ 编程经验。如果读者不熟悉如何使用 Visual C++，建议还是先阅读一本 Visual C++ 编程的教科书之后再阅读本书。

本书的主要内容

本书在讲解了基本的图像处理理论和分析常用的图像处理技术的基础上，基于面向对象的编程技术，结合大量的示例，向读者展示了用 Visual C++ 进行图像处理编程的基本方法和技巧。

本书按不同的图像处理技术分专题组织内容，共分为 9 章。

第 1 章 Visual C++ 编程概述

本章全面介绍了 Visual C++ 的技术概要和面向对象的技术特征，还讨论了编程风格问

题。本章推荐的是一种便于调试的实用编程风格。

第2章 图像基础

本章介绍了数字图像处理的一些基本概念,包括图像、调色板,以及设备相关位图和设备无关位图等。本章的示例程序 ShowDIB 说明了基本的位图编程方法。

第3章 面向对象的图像编程

本章是以后各章的基础,首先定义了一套位图处理的函数集,并在此基础上设计与操作设备无关位图的 CDib 类。本章示例程序 ViewDIB 实现的功能与上一章的 ShowDIB 完全相同,但 ViewDIB 是基于 CDib 类来编写的。比较这两个示例的异同有助于体会 CDib 类的强大功能。

第4章 图像的特技显示

本章主要介绍图像显示特技。在讨论了图像特技显示技术基础之后,本章详细介绍了扫描、移动、百叶窗、栅条、马赛克、渐显与渐隐和透明显示的实现方法,并在 CDib 类中实现这些功能。

第5章 图像变换

本章的主题是图像变换。首先介绍图像空间变换的理论基础,然后讨论了裁剪与合并、缩放、旋转及镜像等图像空间变换的编程技术,并将这些功能增加到 CDib 类中。

第6章 颜色处理

本章主要介绍颜色处理。在介绍颜色处理理论的基础上,探讨灰度化与伪彩色处理、颜色分离和滤色、颜色调整、亮度/对比度调整、Gamma 校正、颜色量化与减色、抖动以及颜色均值映射的实现技巧,并在 CDib 类中实现这些功能。

第7章 图像处理

本章主要介绍数字图像处理的理论与方法,是全书的核心。首先介绍数字图像处理的基础理论,包括卷积、滤波、正交变换及数字形态学理论等。然后介绍正交变换(傅里叶变换、余弦变换、沃尔什变换)、卷积、图像平滑与锐化、边缘增强、边缘检测(Sobel 边缘检测、Hough 边缘检测、均匀边缘检测、差异边缘检测和 Canny 边缘检测)、艺术效果(浮雕和油画)、腐蚀与膨胀、图像的开和闭、边缘提取及细化和去除随机噪声的编程方法。最后将所有这些功能增强到 CDib 类中。

第8章 图像格式

本章主要介绍图像编码和图像格式。8.1 节介绍了图像编码的基础理论,详细说明了 Huffman, LZW 和 RLE 编码的实现细节。随后的 4 节讨论了 PCX, GIF, TGA, JPEG 和 TIFF 文件格式的操作技术,并基于 CDib 类实现操作这些文件格式的 CPcx, CGif, CTga, CJpeg 和 CTif 类。

第9章 图像处理类库 ImageLib 及其应用实例

本章综合运用前几章的知识,以 CDib, CPcx, CGif, CJpeg, CTga, CTif 类为基础,以动态链接库的形式构建了一个可以直接用于实际图像处理编程的通用图像处理类库 ImageLib,并详细说明了用 ImageLib 编写一个功能较完整的图像处理程序 ImageBoard 的方法和过程。

本书附光盘 1 张,提供了各章的源程序代码(readme.txt 文件)介绍了使用光盘的方

法，以及光盘中各个目录与本书各章节的对应关系。

如何使用本书

本书提供了大量的实用代码和例程，涉及图像处理的各个方面。在阅读本书时，读者最好能准备好一台计算机，以便能随时尝试本书所提供的例程代码。

为了能够正确地使用本书提供的工具库和示例程序，读者的计算机系统应满足如下基本要求：

- CPU：奔腾 III 或更高
- 内存：不少于 64MB
- 操作系统：Windows 95/98/Me 或 Windows NT/2000/XP
- 编程环境：Visual C++ .NET（包含在 Visual Studio .NET 中）或 Visual C++ 6.0

第 3 版的新增内容

与第 2 版比较，第 3 版中新增加或者修改了如下内容：

- 在 CDib 类中增加了拷贝构造函数和赋值函数（3.3 节）。
- Gamma 校正（6.7 节）。
- 颜色均值映射（底片和曝光）（6.8 节）。
- 均匀边缘检测和差异边缘检测（7.6.2 节和 7.6.3 节）。
- Canny 边缘检测（7.6.4 节）。
- 艺术效果（浮雕和油画）（7.7 节）。
- 完全重写了 TIFF 格式的操作代码，改正了第 2 版中的一个缺陷，现在支持 TIFF 文件读和写操作（8.6 节）。
- 改正了在 Visual C++ 6.0 中编译 ImageBoard 项目时的一个问题，提供了一个 vcip.dsw，将所有的项目集成到一个工作区上。
- 校正了一些印刷和代码中的错误。

如何发表意见和获得技术支持

笔者欢迎读者提出宝贵意见。读者在阅读本书时遇到任何问题，或者发现了本书存在的错误和纰漏（限于笔者的能力，本书中出现错误、浅陋和陈旧之处在所难免），以及发表意见和建议，都请直率地告诉笔者。笔者会通过电子邮件尽力回复读者提出的问题。请通过如下电子邮件地址与笔者联系：

darin.zhou@gmail.com

致谢

感谢所有对本书的写作有过帮助的人。周建欣、林琳、黄家辉、邓优、李秀鹏、刘烁和何西涛等参加了本书有关内容的讨论和资料收集，周建欣、彭文、黄娅丽测试了本书的所有示例程序，辛平测试了图像工具库 ImageLib。

感谢电子工业出版社博文视点资讯有限公司为本书的出版付出的努力。

最后要感谢本书第 1 版和第 2 版的广大读者，他们的鼓励和批评是第 3 版得以出版的最大动力。实际上，这一版中的许多新增内容都是根据读者的反馈信息而增补的，许多问题也是热心读者所指出的。我在此向所有的读者表示衷心的感谢！我希望读者继续关心本书，也继续期待着读者的建议意见和批评指正。

周长发
2006 年

目 录

第 1 章 Visual C++编程概述	1
1.1 为什么用 C++来实现图像处理编程	1
1.2 Visual C++技术概要	2
1.2.1 Win32 编程	3
1.2.2 框架和文档—视结构	4
1.2.3 消息映射	5
1.2.4 Visual C++可视化编程	8
1.3 Visual C++ .NET 的新增特性	9
1.4 编程风格问题	14
1.5 本章小结	16
第 2 章 图像基础	17
2.1 数字图像的基本概念	17
2.2 调色板	20
2.2.1 调色板的概念	21
2.2.2 调色板操作	22
2.2.3 定义几个调色板函数	23
2.3 与设备相关位图 (DDB)	29
2.4 与设备无关位图 (DIB)	31
2.4.1 DIB 位图的结构	31
2.4.2 OS/2 DIB 格式	34
2.4.3 DIB 位图的操作	34
2.5 一个简单的位图示例: ShowDIB	47
2.6 本章小结	51
第 3 章 面向对象的图像编程	52
3.1 定义 DIB 处理函数集	52
3.1.1 Win32 SDK 中的 DIB API 函数	53
3.1.2 定义 DIB 处理函数集	55
3.2 CDib 类的设计目标	85
3.3 构造 CDib 类	87
3.4 CDib 类的编程示例: ViewDIB	107
3.5 本章小结	110

第 4 章 图像的特技显示	111
4.1 特技显示的技术基础	111
4.2 扫描	113
4.3 移动	118
4.4 百叶窗	124
4.5 栅条	125
4.6 马赛克	127
4.7 渐显与渐隐	129
4.8 透明显示	136
4.9 增强 CDib	139
4.10 显示特技示例: EffectShow	140
4.11 本章小结	143
第 5 章 图像变换	144
5.1 图像变换的理论基础	144
5.2 裁剪与合并	150
5.3 缩放	157
5.4 旋转	163
5.5 镜像	174
5.6 增强 CDib	178
5.7 图像变换示例: TransformShow	180
5.8 本章小结	183
第 6 章 颜色处理	184
6.1 颜色处理的理论基础	184
6.2 灰度化与伪彩色处理	189
6.3 颜色调整	193
6.4 亮度/对比度调整	201
6.5 颜色量化与减色	206
6.6 抖动	217
6.7 Gamma 校正	228
6.8 颜色均值映射	230
6.8.1 反转与底片	230
6.8.2 曝光	232
6.9 增强 CDib	234
6.10 颜色处理示例: ColorProcess	237
6.11 本章小结	239
第 7 章 图像处理	240
7.1 图像处理的基本方法	241

7.2	正交变换	248
7.2.1	傅里叶变换	248
7.2.2	离散余弦变换	263
7.2.3	沃尔什变换	270
7.3	卷积	280
7.4	平滑与锐化	284
7.5	边缘增强	286
7.6	边缘检测	290
7.6.1	Sobel 边缘检测与 Hough 边缘检测	290
7.6.2	均匀边缘检测	293
7.6.3	差异边缘检测	295
7.6.4	Canny 边缘检测	298
7.7	艺术效果	304
7.7.1	浮雕	304
7.7.2	油画	307
7.8	腐蚀与膨胀	309
7.9	开与闭	314
7.10	边缘提取	315
7.11	细化	318
7.12	去除随机噪声	322
7.13	增强 CDib	325
7.14	图像处理示例: ImageProcess	329
7.15	本章小结	333
第 8 章	图像格式	334
8.1	图像编码	334
8.1.1	数据冗余	334
8.1.2	数据编码	336
8.2	PCX	361
8.2.1	PCX 格式说明	361
8.2.2	实现操作 PCX 的类	364
8.3	GIF	366
8.3.1	GIF 文件存储结构	367
8.3.2	LZW 算法和 GIF 数据压缩	374
8.3.3	实现操作 GIF 的类	378
8.4	JPEG	381
8.4.1	JPEG 格式简介	381
8.4.2	实现操作 JPEG 的类	382
8.5	TGA	384
8.5.1	TGA 格式说明	384

8.5.2	实现操作 TGA 的类	392
8.6	TIFF	393
8.6.1	TIFF 格式说明	393
8.6.2	实现操作 TIFF 的类	403
8.7	图像格式示例: ViewImage	405
8.8	本章小结	407
第 9 章	图像处理类库 ImageLib 及其应用实例	408
9.1	创建图像处理类库 ImageLib	408
9.2	基于 ImageLib 实现图像处理程序 ImageBoard	410
9.2.1	界面设计	412
9.2.2	程序结构	413
9.2.3	工具条与状态条	416
9.2.4	橡皮筋矩形选块或画图	429
9.2.5	画曲线和写字	440
9.2.6	选块的拖曳	448
9.2.7	剪贴板操作	451
9.2.8	文件操作	457
9.2.9	关于进一步开发的建议	462

Visual C++编程概述

本章要点

- 为什么用 C++ 来实现图像处理编程
- Visual C++ 技术概要
- Visual C++.NET 的新增特性
- 编程风格问题

本书讨论 Visual C++ 的图像处理编程问题。现在已经出现了许多的现代编程语言，比如 Java 和 C#，它们都源于 C++，并且具有许多 C++ 不具备的优秀特性。读者可能会问，为什么仍然采用 C++ 来实现图像编程，而不采用这些新的编程语言？本书第 1 章的第 1.1 节首先回答这一问题。

1.1 为什么用 C++ 来实现图像处理编程

采用 C++ 来进行图像编程的主要原因是，与 Java 和 C# 等现代编程语言相比，C++ 在程序运行的效率、内存使用的可控性和编程的灵活性上具有优势。

图像处理需要进行大量的图像数据运算，经常使用复杂、费时的算法，因此图像处理程序的运行效率非常重要。C++ 代码被编译成汇编语言，可以直接在处理器上运行，效率很高。而 Java 被编译成字节码，C# 被编译成中间语言，都是不能直接在处理器上执行的，必须经过 Java 虚拟机或者 .NET 通用语言运行时的 JIT 编译之后才能执行，因此效率较低。因此，从运行效率的角度来看，采用 C++ 进行图像编程更合适。

对庞大的图像数据进行处理需要使用大量的内存，而计算机的物理内存往往是有限的，因此需要有效地控制内存的操作。C++ 直接控制内存的分配和释放，这虽然很繁琐，而且容易出错，但却能有效地控制内存的使用。Java 和 C# 引进了垃圾收集机制，将开发人员从内存管理的繁杂任务中解放出来，不再需要直接控制内存的分配和释放。但是，无效内存的收集和释放只能周期性地地进行，难以达到有效地使用内存。同样的运算，使用 Java 和 C# 所需的内存比使用 C++ 所需的内存要多，即内存的使用率较低。所以，从内存的使用效率来看，采用 C++ 进行图像处理编程更合适。

C++ 中大量地使用指针，使得编程的灵活性很高，便于程序员施展编程技巧来提高程序的效率。Java 完全地取消了指针，C# 极大地限制了指针的使用，这都限制了程序员施展技巧。当然，指针的使用同时也增加了程序出错的可能性。但是对于熟练的 C++ 程序员来说，可能更愿意使用指针这一利器来进行图像编程。

1.2 Visual C++ 技术概要

面向对象程序设计（Object-Oriented Programming, OOP）方法已出现近 30 年，在 20 世纪 90 年代已成为程序设计的主流方向。面向对象程序设计语言是现代程序开发的主要工具，任何一个优秀的应用程序开发人员都必须至少熟练掌握一种面向对象的编程语言。

程序包含两类基本元素，即数据和操作数据的指令集（称为代码）。传统的程序设计语言以设计代码为核心，程序设计实际上就是指定程序指令的先后次序，数据表示必须适应代码的设计。模块化程序设计方法将完成某一功能的指令集组成一个相对独立的程序模块（即函数或过程），使得程序的结构清晰，便于有效地维护，对程序设计技术有很大的促进。但由于结构化程序设计方法并不能保证各程序模块之间真正的相互独立，程序设计者在设计一个模块时很难完全排除其他模块的影响。随着程序规模的增大，各模块之间的相互影响导致一些难于测试，难以定位发现的错误，增加了程序开发和维护的困难。面向对象的程序设计方法就是在这种背景下出现和发展起来的。

面向对象程序设计方法主要以数据为中心，代码是围绕着需要处理的数据而设计的。面向对象的程序设计语言具有如下的主要特征。

1. 对象的类描述

面向对象的程序设计语言将程序描述的事物看成一个整体，称为对象（object）。事物的属性基本上可以分为两部分，即内部状态（性质）和对数据的操作方法及由此造成的对外部的影响。对象的数据用于描述内部状态，代码完成对数据的操作。因此，对象就是包含数据和代码的完全独立的实体。类（class）就是具有相同的属性的所有对象的逻辑原型，是对象的规则和设计。同一类对象具有相同的性质和方法，每一个具体的对象都是类的一个实体，创建对象就是把类实例化。

2. 封装性（Encapsulation）

封装性是 OOP 的核心技术，是面向对象程序设计语言将数据和处理数据的方法组合在类中，并具有模块化和信息隐藏的特征。类是一个独立的模块，类的内部状态描述数据对程序的其他部分是不可见的，类只向外界公布其具有 public 属性的数据和代码，并构成类与外界的接口。外界不能直接对类的内部状态进行修改，而只能通过这个接口将信息传递给类，并由类定义对内部数据进行操作的方法进行内部修正，外界不能决定这种修正的结果，只能得到类进行操作所做出的反应。封装性能防止类与外部的非法交互和访问，避免外界对对象内部状态的错误改变，确保类这一模块的真正独立，以保证程序的安全运行。

同时，由于程序的其他部分只能访问类的接口，因此只要保持类的接口不变，改变类的内部结构、工作方式和实现就不会对整个程序产生非预期的影响。所以，对类的内部做任何的优化都是安全的。

3. 多态性（Polymorphism）

不同的类或对象对外界传入的相同信息能够根据自身的性质做出不同的反应，这就是多态性。不同的类或对象都可以通过设计自己专有的处理外界传入信息的方法来实现多态性。在 OOP 中，多态性具有两方面的实际意义，一是具有相同名字的接口（具有 public 属性的公有函数）在不同类中能具有不同的意义和实现；二是具有同一名字的函数可以具

有不同的实现代码。在调用时,根据传入参数的不同而调用不同的代码,这叫做函数的重载。

4. 继承性 (Inheritance)

继承性指一个类可以派生出新的类。新类能继承原类定义的性质和方法,还能在原类定义的性质和方法之外加入自身定义的性质和方法。通过继承性,能形成类之间的层次结构。在上层中已经定义的性质和方法能被下层直接继承使用,下层就不需要重新定义,从而实现了代码的重复利用。这样,下层的类只需专注于自身的新特征描述,提高了程序设计的效率和程序组织的有效性。

不是由其他类派生的类称为基类。由其他类派生的类叫做该类的子类,该类叫做其子类的父类或超类。子类可以利用多态性特征来对继承而来的方法进行重载,使得具有相同名字的方法在子类和父类中实现不同的功能。

C++是运用最广泛的面向对象程序设计语言;Visual C++是一个具有集成、交互和可视化编程的C++实现,具备上述所有的OOP特征。

编写 Visual C++程序实际上就是一个构造类和把类实例化的过程。由于 Windows 95/98/Me/NT/2000/XP 是 PC 平台中应用最广泛的操作系统 (Microsoft 力图用一个叫做 Win32 的标准的 32 位应用程序接口来作为对这几个操作系统的共同开发接口,所以经常采用 Win32 来代表 Microsoft 的 32 位 Windows 操作系统),因此 Visual C++主要用于针对 Win32 的应用程序开发。

1.2.1 Win32 编程

Win32 具有抢先式多任务、多线程和线性寻址内存管理等特征,Win32 编程的基本要求包括:

- (1) 应用程序的执行独立于硬件设备;
- (2) 应用程序具有图形用户界面;
- (3) 能在 Windows 95/98/Me 和 Windows NT/2000/XP 之间透明移植,并可移植到支持 Windows NT/2000/XP 的 RISC 硬件平台;
- (4) 高性能的抢先式多任务和多线程管理;
- (5) 高级的多媒体支持;
- (6) 通过 OLE 2 技术实现多个应用程序的对象定位。

Microsoft 为进行 Win32 编程提供了一套名为 Win32 SDK 的应用程序编程接口,其中包括上千个 Win32 系统函数。Visual C++包括一套叫做 MFC (Microsoft Foundation Class Library) 的 C++类库,其中定义了进行 Win32 编程所需要的各种类。有的类封装了大部分的 Win32 SDK 中应用程序编程接口函数;有的类封装的则是应用程序本身的数据和操作;还有的类封装了 ActiveX、OLE 和 Internet 编程特性,以及 WinSock 网络特性和 DAO (Data Access Objects)、ODBC (Open Database Connectivity) 数据访问功能。Win32 SDK 和 MFC 是实现 Win32 编程的主要工具。

Visual C++的 AppWizard 工具能自动生成应用程序框架,该框架定义了应用程序的轮廓,并提供了用户接口的标准实现方法。运用 Visual C++的资源编辑器 (Resource Editor) 能直观地设计程序的用户界面,而 ClassWizard 能把用户界面和程序代码连接起来。程序

员要做的就是用 MFC 类实现框架中未完成的应用程序的特定功能部分。所以，使用 Visual C++ 可以实现 Win32 的可视化程序设计。

1.2.2 框架和文档—视结构

所谓框架（Framework），就是应用程序所应具备的软件模块按一定的结构所组成的集合。基于 MFC 的应用程序框架是定义了程序结构的 MFC 类库中类的集合，是 Visual C++ 编程的骨架。运用 MFC 应用程序框架能获得如下优点。

- 具有标准化的程序结构和用户接口：这对具有标准用户界面的 Win32 程序来说，可以极大地减轻程序员的负担，使程序员不必过多地考虑界面，可把主要精力放在程序设计上，以提高程序设计的效率。
- 框架产生的程序代码短，运行速度快，具有很大的灵活性：MFC 封装了 Win32 SDK 中的几乎所有函数，能实现 Win32 系统的任何功能。
- 强大的功能：除封装了大部分的 Win32 SDK 函数外，MFC 还提供了应用程序本身的数据和操作，以及 ActiveX, OLE, Internet, WinSock, DAO (Data Access Objects), ODBC (Open Database Connectivity) 等操作类。

MFC 框架的核心是文档—视结构（Document-View Architecture），这是一个很有用，但又往往较难入门的功能。简单地说，文档—视结构就是将数据和对数据的观察或数据的表现（显示）相分离。文档仅处理数据的实际读、写操作，视则是显示和处理数据的窗口，视可以操作文档中的数据。

1. 框架结构

MFC 框架的基本结构包括应用程序对象、主框架窗口、文档及视等，框架通过命令和信息将它们结合在一起，共同对用户的操作做出响应。

图 1-1 表示了 MFC 框架结构和各对象之间的关系。应用程序管理着一个（SDI）或多个（MDI）文档模板，每个文档模板管理着一个或多个文档。用户通过主框架窗口中的视来观察和处理数据。

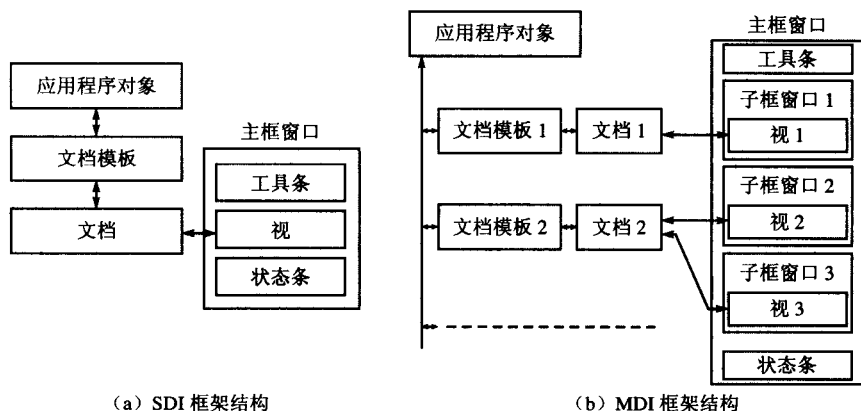


图 1-1 MFC 框架结构

2. 应用程序对象

这是由 CWinApp 派生的应用程序类对象。一个应用程序有且仅有一个应用程序对象。

它负责应用程序实例的初始化和进程结束时的资源清除，以及创建和管理应用程序所支持的所有文档模板。

任何 Windows 应用程序都包含一个 `WinMain()` 函数。框架应用程序也一样，只是框架中的 `WinMain()` 函数由 MFC 类库提供并被隐藏，在框架应用程序启动时调用。应用程序对象的 `InitInstance()` 函数就是由 `WinMain()` 调用的。

3. 主框窗口

这是应用程序的主窗口。MFC 框架定义了两种基本的主框窗口类，即单文档接口（Single Document Interface, SDI）主框窗口类 `CFrameWnd` 和多文档接口（Multiple Document Interface, MDI）主框窗口类 `CMDIFrameWnd`。应用程序的主框窗口应从其中之一派生出来。对 SDI，视是主框窗口的子窗口；对 MDI，必须从 `CMDIChildWnd` 派生出主框窗口的子窗口，视是该子窗口的子窗口。

4. 文档

文档类由 `CDocument` 类派生而来，文档类对象由框架生成的 File 菜单中的 New 或 Open 命令创建，并指定应用程序数据的实际读、写操作。如果想让应用程序支持 OLE 功能，则应从 `COleDocument` 类派生文档类。

文档由应用程序对象创建和维护的文档模板（Document Template）所创建。框架分别使用 `CSingleDocTemplate` 和 `CMultiDocTemplate` 文档模板来管理 SDI 和 MDI，前者可以创建和存储一种类型的文档，后者保存了一种类型的多个文档的列表。

5. 视

视类从 `CView` 或其子类（`CEditView`，`CFormView`，`CRecordView`，`CScrollView` 等）派生而来，是显示和观察文档数据的窗口类。视类定义了用户以什么方式见到文档的数据，以及如何与其进行交互，一个文档数据可能有多个视。

6. 文档—视结构

框架在响应它生成的标准用户接口 File 菜单中的 New 和 Open 命令时创建文档—视结构，其创建次序如下：

（1）在程序启动时，`WinMain()` 函数调用应用程序对象的 `InitInstance()` 函数，并在其中创建文档模板。

（2）程序运行过程中，用户选取了 File 菜单中的 New 或 Open 菜单项，框架将调用 `CWinApp::OnFileNew()` 或 `CWinApp::OnFileOpen()` 函数，并使用已创建的文档模板创建文档。

（3）文档模板同时创建主框窗口（SDI）或子框窗口（MDI）。

（4）主框窗口（SDI）或子框窗口（MDI）创建文档对应的视。

文档—视结构中各对象的交互关系如图 1-2 所示。在文档—视结构中各对象创建以后，程序员应覆盖相应类的成员函数，以对各对象做具体的初始化。在视类中覆盖 `OnInitialUpdate` 是初始化视的最合适方法。覆盖文档类的 `OnNewDocument` 和 `OnOpenDocument` 成员函数可以为文档做特定的初始化。

1.2.3 消息映射

Windows 应用程序是消息驱动的，应用程序不能直接得到用户所做的操作事件，如鼠

标按键、键盘输入和窗口移动等。这些操作由操作系统管理，操作系统检测到操作事件后，便向相关的应用程序发送消息，应用程序响应这些消息来完成用户的操作。

1. 消息

Windows 中的消息是操作系统与应用程序之间、应用程序之间、应用程序各对象之间相互控制与信息传递的方式。

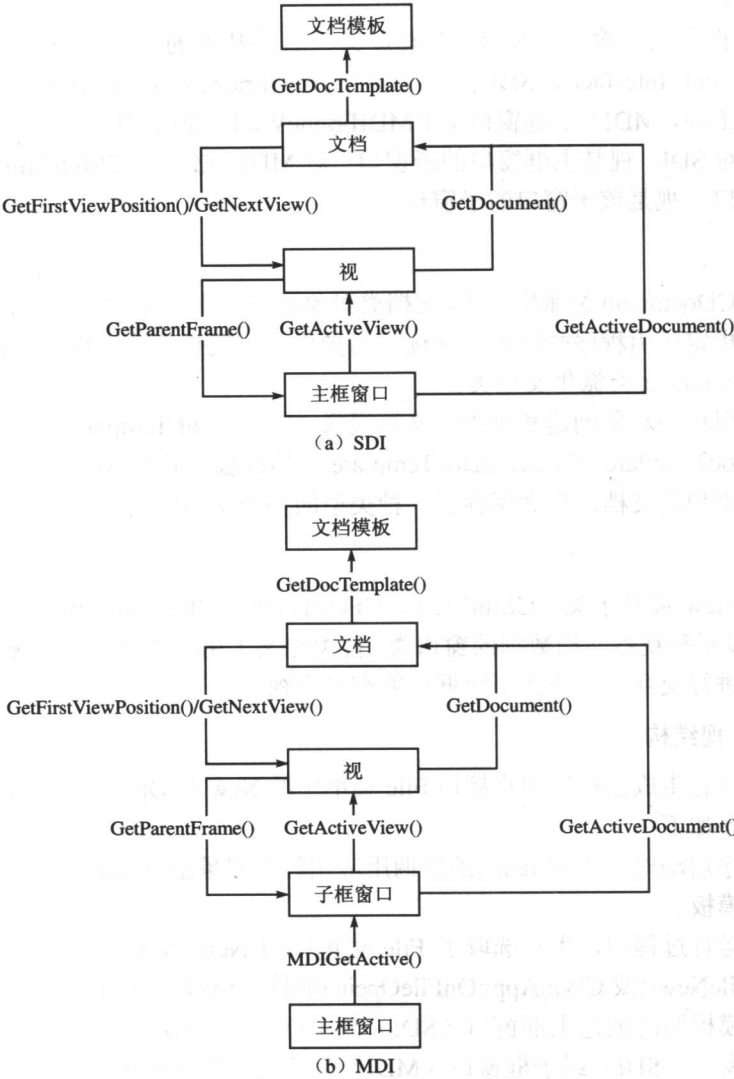


图 1-2 文档—视结构中各对象的交互关系

消息的基本格式是：

Message wParam lParam

Message 是消息名称；wParam 是与消息相关的 WORD 型参数；lParam 是消息相关的 LONG 型参数。