

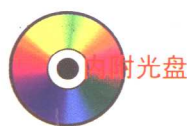
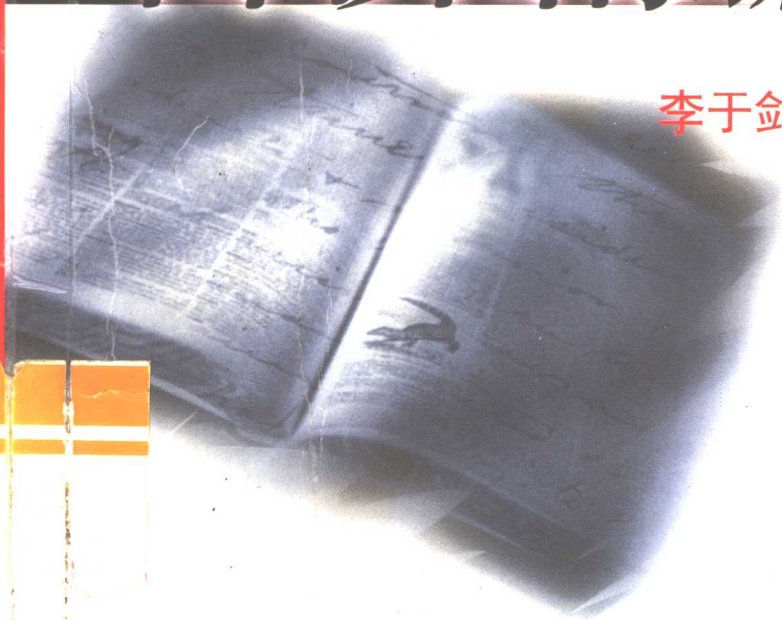


Visual C++

实践与提高

图形图像编程篇

李于剑 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

00102531

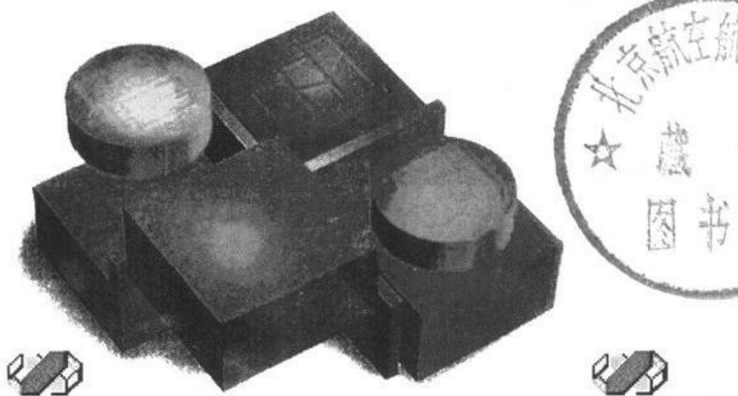
TP312C
179
V4

Visual C++ 实践与提高

—— 图形图像编程篇

李于剑 编著

 **Visual C++**



中国铁道出版社

(京)新登字 063 号

内 容 简 介

本书将向读者讲述如何用 Visual C++ 进行图形图像编程。本书共分为图形篇、图像篇、综合提高篇。图形篇以计算机绘图的基础知识为铺垫,介绍了各种图形元素的绘制方法,并结合实际代码实现了一个小型的绘图系统;图像篇通过实例向读者介绍了不同格式图像的处理方法,并给出了一些特效处理的技巧;综合提高篇综合应用了图形图像的知识,包括图形图像文件格式之间的转换、打印和打印预览、使用 OpenGL 和 DirectX 进行图形图像处理等内容。

本书内容结合实例进行讨论,有助于读者尽快掌握实践的方法。全书内容新颖、语言浅显、通俗易懂,适用于初、中级程序设计人员学习使用,也可同大中专院校师生阅读。

JS474/22

图书在版编目(CIP)数据

Visual C++ 实践与提高. 图形图像编程篇/李于剑编著. —北京:中国铁道出版社, 2001. 1
ISBN 7-113-04044-6

I. V… II. 李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 88294 号

书 名: Visual C++ 实践与提高——图形图像编程篇

作 者: 李于剑

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟

特邀编辑: 宁 夕

封面设计: 冯龙彬

印 刷: 化学工业出版社印刷厂

开 本: 787×1092 1/16 印张: 31.25 字数: 758 千

版 本: 2001 年 2 月第 1 版 2001 年 2 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04044-6/TP·502

定 价: 51.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

随着计算机的技术日益普及和不断完善,计算机在图形图像方面的需求不断增加;同时,随着计算机技术的发展,对图形图像系统的应用环境和开发都提出了新的要求。正是由于以上原因,我们编写了本书,希望能够引导读者能够利用微软公司提供的 Visual C++ 开发工具进行图形图像编程。这样,读者不仅学习了如何使用 Visual C++ 的 AppWizard 和 MFC 开发 Windows 应用程序,又学习了图形图像编程的知识。

本书共分为三篇:图形篇、图像篇和综合提高篇。

第一篇为图形篇,分为 6 章进行讲述。

第 1 章介绍了有关计算机绘图方面的基础知识,首先介绍了 Windows 的 GDI 和设备环境的基本概念,着重介绍了 MFC 类库中所封装的设备环境类(CDC 及其派生类)以及 GDI 绘图对象类(CPen 类,CBrush 类和 CFont 类等)的定义和使用方法,最后,对图形编程中的坐标变换、映射模式和绘图模式进行详细的说明。

第 2 章首先介绍了各种图形元素(如直线、矩形、圆和圆弧等)的绘制方法,并详细介绍了绘制复杂曲线的思路和方法。然后对如何使用鼠标绘制各种图元的方法进行详细的说明。最后介绍了一些有关图形重画的知识以及如何在任意窗口内绘图的方法。以上的内容都可以通过本章最后一节的实例找到相应的实现方法。

第 3 章主要从整体上对图形应用程序编程思路进行了介绍。这一章涉及到了很多知识点,总结归纳了笔者多年来的经验,相信读者肯定会从中有很大收获。

第 4 章主要以一个小型图形系统为例(名为 VCad1.0),通过使用第 3 章中介绍的相关知识,着重介绍了如何实现小型绘图系统的绘制功能。

第 5 章仍以第 4 章中的小型图形系统作为实例,在第 4 章的基础之上对其功能进行扩展。介绍了对图元的编辑功能(如拾取、平移、旋转和镜向等)的实现方法。本章最后得到了功能相对比较完善的绘图系统 VCad2.0。

第 6 章是对第 5 章中的 VCad2.0 绘图系统的进一步完善,对其界面进行了美化,提高了绘图系统的交互性能。本章着重介绍了如何使用 VC 的资源编辑器进行添加工具栏资源和光标资源,如何创建自定义的工具栏,如何添加闪屏窗口(Splash Window),以及如何在程序运行当中改变光标的形状,以便提示当前系统的状态。本章最后得到了绘图系统的最终版本 VCad3.0。

第二篇为图像篇,分为 3 章进行讲述。

第 7 章是图像编程的基础,详细介绍了调色板和依赖于设备的位图——DDB 的创建与使用,概念清晰,内容丰富。

第 8 章首先介绍了 BMP 位图的格式和了有关图像编程的 Windows 的 API 函数,接着向读者推荐的一个 CDib 类,这个类的创建较为灵活,功能齐全。

第 9 章利用前面介绍的 CDib 类进行图像处理,包括显示处理、几何变换和颜色处理。涉及的范围较为广泛,实用性强。

第三篇为综合提高篇，分为 4 章进行讲述。

第 10 章向读者介绍了如何进行图形图像文件格式之间的转换，具体内容包括：将图形存为 DXF 文件，以便能够用 AutoCAD 读入；DDB 与 DIB 之间的转换；此外，还给出了一个在“打开文件”对话框中预览图形的例子。

第 11 章向读者介绍如何使用类库的打印和打印预显功能，并从打印文档和打印视的两个角度介绍了打印功能的实现。此外，本章还给出了位图打印和抓屏打印的方法。

第 12 章简明扼要地介绍使用 OpenGL 开发图形图像以及动画应用程序的方法，使读者能够快速入门，成为一名 OpenGL 高手。

第 13 章中简单地介绍了最新的 DirectX7.0 中的 DirectDraw 和 Direct3D 这两部分，使读者对使用 DirectX 进行图形图像编程有一个大概地了解。

本书由孙福辉负责编写了第 1、2、3、4、5、6 章，李于剑负责编写了 7、8、9 章，孙蔚负责编写本书的第 10、11、12、13 章。参加本书编写的还有刘百合、宋卫堂、高飞、杨蕴石、刘清英、李永刚、赵伟、王涛、张春雷、王勇、李琪、薛立龙、王辉、董立、高明辉、张宝艳、郑春阳、张瑞武。杨继国、李艳霞、杨放为本书搜集了资料。李庸、宋清荣审阅了本书的初稿并对本书的取材提出了宝贵的意见。本书的录入工作由王宝负责。

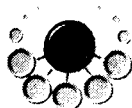
由于时间仓促，加之水平有限，不当之处在所难免，请读者批评指正。

编者

2000 年 12 月

目 录

第 1 章 图形基础编程	1
1.1 图形编程的基础知识	1
1.1.1 图形设备接口 GDI	2
1.1.2 设备环境 DC	2
1.1.3 设备环境类	2
1.1.4 GDI 对象	8
1.1.5 使用库存 GDI 对象	9
1.1.6 DC 与 GDI 设备之间的关系	10
1.2 在视图内绘图	10
1.2.1 OnDraw 成员函数	10
1.2.2 Windows 设备环境	11
1.3 与绘图相关的 GDI 对象类	11
1.3.1 CPen 类	12
1.3.2 CBrush 类	17
1.3.3 CFont 类	20
1.3.4 CRgn 类	23
1.3.5 CPalette 类	28
1.3.6 CBitmap 类	29
1.4 坐标变换和映射模式	29
1.4.1 Windows 映射模式简介	29
1.4.2 如何设置映射模式	30
1.4.3 逻辑坐标和设备坐标的转换	33
1.4.4 窗口和视口	36
1.5 如何设置绘图模式	37
1.6 本章小结	39
第 2 章 各种图形元素的绘制	41
2.1 图形元素绘制的实现方法	41
2.1.1 绘制点、直线	41
2.1.2 绘制矩形	43
2.1.3 绘制简单曲线	44
2.1.4 绘制复杂曲线	48
2.1.5 绘制文本	48
2.2 使用鼠标绘图	53



2.2.1	如何响应鼠标消息	53
2.2.2	如何捕获和设置鼠标状态	55
2.2.3	用鼠标绘制各种基本图元	60
2.2.4	实现绘图的拖曳显示	66
2.3	图形的重画	68
2.3.1	全部重画	68
2.3.2	局部重画	69
2.3.3	如何解决闪屏现象	70
2.4	应用实例	70
2.4.1	实例 1: 绘制三次曲线	70
2.4.2	实例 2: 实现曲线的选取功能	75
2.4.3	实例 3: 在对话框中实现各种图形的绘制	78
2.5	本章小结	85
第 3 章 图形应用程序的设计思路		87
3.1	面向对象的图形应用程序设计	87
3.1.1	派生和继承	87
3.1.2	函数覆盖与重载	89
3.1.3	虚函数机制	93
3.1.4	多态性的应用	94
3.2	图形类的设计	95
3.2.1	抽象基类	95
3.2.2	类的层次性	95
3.2.3	图形类的构造和析构函数	96
3.2.4	基类 CEntity	98
3.2.5	从 CObject 类派生	101
3.2.6	一个派生类 CLine	102
3.3	数据结构和数学模型	103
3.3.1	数据结构	103
3.3.2	数学模型	107
3.4	内存管理	108
3.4.1	框架内存分配	108
3.4.2	堆内存分配	109
3.4.3	内存分配的应用实例	110
3.5	数据存储和文件管理	111
3.5.1	文件存储格式	111
3.5.2	序列化和反序列化	112
3.5.3	应用程序的不同版本	114



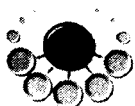
3.5.4 数据交换文件	116
3.6 系统交互界面环境设计	116
3.6.1 界面环境设计的原则	116
3.6.2 面向对象的消息响应机制	119
3.6.3 元素的拾取与选择方式	120
3.7 交互图形的辅助工具	122
3.7.1 栅格	122
3.7.2 动态导航	123
3.7.3 橡皮线	123
3.8 本章小结	123
第 4 章 小型图形系统的实现——图形的绘制	125
4.1 本书小型图形软件简介	125
4.1.1 传统图形软件的实现方法	125
4.1.2 本书图形软件的实现方法	126
4.2 图形软件的初步实现	126
4.2.1 建立工程 VCad (Visual CAD 的缩写)	126
4.2.2 加入命令和响应消息	127
4.2.3 软件基本变量及环境设置	129
4.3 位置类 Position	131
4.3.1 关于 Cpoint	131
4.3.2 基础类 Position 的实现	132
4.3.3 Position (世界坐标) 和 CPoint (屏幕坐标) 的关系	137
4.4 基类 CEntity 和 CCommand	139
4.4.1 图形基类 CEntity	139
4.4.2 命令基类 CCommand	142
4.5 直线的绘制	143
4.5.1 设置图形元素的绘制环境	143
4.5.2 直线类 CLine	145
4.5.3 创建直线命令类 CCreateLine	148
4.5.4 实现直线的绘制	152
4.5.5 关于重画问题	155
4.5.6 橡皮线功能的测试	156
4.6 矩形、圆和圆弧的绘制	158
4.7 文件存储及内存管理	158
4.7.1 文档的“脏”标记	158
4.7.2 释放图元链表的内存空间	159
4.7.3 文件的存储和加载	160
4.8 本章小结	163



第5章 小型图形系统——图元编辑和缩放显示	165
5.1 给图元类添加编辑成员函数	165
5.1.1 在基类中添加成员函数	165
5.1.2 在图元类添加成员函数	166
5.2 图元的包围盒计算	166
5.2.1 直线的包围盒计算	166
5.2.2 矩形和圆的包围盒计算	166
5.2.3 圆弧的包围盒的计算	167
5.3 图元的拾取算法	168
5.3.1 拾取的精度	168
5.3.2 直线的拾取	169
5.3.3 矩形的拾取	170
5.3.4 圆的拾取	171
5.3.5 圆弧的拾取	171
5.4 实现图元的选取	173
5.4.1 添加选择集	173
5.4.2 选择集的实现	174
5.4.3 图元的交互选取	177
5.5 图元的编辑算法	179
5.5.1 图元的移动	180
5.5.2 图元的旋转	181
5.5.3 图元的镜像	181
5.5.4 图元的删除	182
5.6 实现图元的交互编辑	182
5.6.1 添加编辑命令类	182
5.6.2 平移类的实现	183
5.6.3 旋转类的实现	187
5.6.4 镜像类的实现	193
5.6.5 图元的交互编辑	198
5.7 实现不同的显示方式	200
5.7.1 移动画面	201
5.7.2 键盘缩放画面	201
5.7.3 画面的局部缩放	202
5.7.4 满屏显示	203
5.8 本章小结	204
第6章 小型图形软件系统——美化软件界面	205
6.1 添加工具栏	205



6.1.1 添加绘图和编辑工具栏以及视图工具栏	205
6.1.2 添加属性工具栏	209
6.1.3 设置工具栏的位置	219
6.2 修改状态条	221
6.3 设置不同状态的光标	222
6.3.1 绘制状态时的光标	222
6.3.2 编辑状态时的光标	223
6.3.3 高亮提示时的光标	223
6.4 添加启动时的闪烁窗口	224
6.5 实现从 Explorer 启动和拖放	225
6.5.1 程序注册	226
6.5.2 在文档上双击鼠标左键	227
6.5.3 允许拖放	227
6.6 本章小节	228
第 7 章 图像编程基础	229
7.1 图像基本概念和图像编码	229
7.1.1 位图与矢量图	229
7.1.2 单色图像	230
7.1.3 灰度图像	230
7.1.4 彩色图像	230
7.2 调色板的基本概念和应用	231
7.2.1 调色板的原理	231
7.2.2 调色板的创建	232
7.2.3 调色板使用方法	235
7.2.4 MFC 中与系统调色板有关的消息	235
7.2.5 调色板的应用实例	237
7.3 依赖于设备的位图(DDB)和类 CBitmap	241
7.3.1 位图的基本概念	241
7.3.2 DDB 的创建	242
7.3.3 DDB 的用途和显示	247
7.3.4 位图的部分显示	248
7.4 本章小结	252
第 8 章 DIB 位图及 CDib 类	253
8.1 与设备无关的位图 DIB	253
8.1.1 Windows DIB 位图的结构	253
8.1.2 OS/2 DIB 结构	257
8.1.3 DIB 位图的操作	258



8.1.4 一个 DIB 位图的例子	267
8.2 CDib 类	275
8.2.1 Win32 SDK 的 DIB API 函数	275
8.2.2 构造 CDib 类	277
8.2.3 CDib 类的使用	294
8.3 本章小结	297
第 9 章 图像的显示、变换及颜色处理	299
9.1 图像显示处理	299
9.1.1 显示处理的基础	300
9.1.2 扫描	301
9.1.3 移动	306
9.1.4 百叶窗	311
9.1.5 栅条	314
9.1.6 马赛克	319
9.1.7 渐显	322
9.2 图像的几何变换处理	327
9.2.1 裁剪	328
9.2.2 平移	332
9.2.3 缩放	332
9.2.4 镜像	336
9.3 图像的颜色处理	340
9.3.1 图像的颜色处理理论	340
9.3.2 灰度化处理	340
9.3.3 亮度和对比度调整	343
9.4 本章小结	353
第 10 章 图形图像文件格式之间的转换	355
10.1 图形文件格式转换	355
10.1.1 常见图形文件格式	355
10.1.2 将自定义文件格式存为 DXF 格式	357
10.1.3 将图形文件存为图像文件	364
10.1.4 如何加载常见的图形文件格式	365
10.2 图像文件格式转换	365
10.2.1 常见图像文件格式	365
10.2.2 DDB 转换为 DIB	366
10.2.3 DIB 转换为 DDB	369
10.3 应用实例：在文件打开对话框中预览图形图像	370



10.3.1 生成预览位图——在内存位图中绘制图形	371
10.3.2 存取位图数据	373
10.3.3 在打开和保存文件对话框中显示位图	374
第 11 章 打印和打印预览	389
11.1 关于 MFC 的打印和打印预览	389
11.1.1 OnDraw 函数	389
11.1.2 OnPrint 函数	390
11.1.3 打印及打印预览过程	390
11.2 单页文档的打印	391
11.2.1 单页文档打印的简单实例	391
11.2.2 关于打印时的缩放	391
11.3 多页文档的打印	392
11.3.1 设置映射模式	392
11.3.2 计算可打印区域	392
11.3.3 分页打印	393
11.4 位图的打印	400
11.5 所见即所得的打印：抓屏打印	403
11.5.1 创建新的位图类	403
11.5.2 实现新的位图类	408
11.6 本章小结	410
第 12 章 使用 OpenGL 开发图形图像	413
12.1 OpenGL 简介	413
12.1.1 OpenGL 函数的分类	413
12.1.2 OpenGL 函数的功能	414
12.1.3 OpenGL 的数据类型和函数名	415
12.2 使用 VC++ 开发 OpenGL 应用程序	416
12.2.1 使用 GLU 和 AUX 函数	416
12.2.2 使用 MFC 和专用函数	420
12.2.3 两种开发方法之间的联系	427
12.3 基本绘制方法	427
12.3.1 清除窗口中的内容	427
12.3.2 绘制图原	428
12.3.3 设置颜色	432
12.3.4 设置图原的样式	435
12.4 变换	439
12.4.1 如何在二维平面显出三维空间	439
12.4.2 视点变换和模型变换	440



12.4.3	投影变换	444
12.4.4	视口变换	446
12.4.5	变换矩阵操作	447
12.5	光照与材质	447
12.5.1	现实世界与 OpenGL 光照	448
12.5.2	一个简单的光照实例	449
12.5.3	创建光源	451
12.5.4	多光源	453
12.5.5	选择光照模型	453
12.5.6	定义材质特性	454
12.5.7	漫射和环境反射	455
12.5.8	镜面反射	455
12.5.9	出射光	456
12.5.10	改变材质特性	456
12.6	图像处理	460
12.6.1	相关函数介绍	460
12.6.2	图像处理实例	462
12.6.3	基本知识	469
12.6.4	应用实例	470
12.7	本章小结	475
第 13 章 DirectX 简介		477
13.1	DirectDraw 简介	477
13.1.1	什么是 DirectX	477
13.1.2	DirectDraw 的优越性	478
13.1.3	DirectDraw 中的基本概念	478
13.1.4	DirectDraw 的体系结构	480
13.1.5	DirectDraw 的组成	480
13.2	Direct3D 简介	481
13.2.1	什么是 Direct3D	481
13.2.2	Direct3D 的体系结构	482
13.2.3	Direct3D 的组成	482
13.2.4	DirectDraw、Direct3D、和 Direct3D 接口的关系	484
13.2.5	Direct3D 的几何变换管道	484
13.2.6	Direct3D 的设备类型	485
13.3	本章小结	486



Windows 是一种图形操作系统，它把所有的东西都作为图形在屏幕上进行显示，甚至把文本也作为图形进行显示。因此，用户就不会奇怪为什么 Windows 拥有大量的图形处理函数库，即我们通常所说的 GDI 函数库。另外，Windows 的设备无关性，使用户不可能事先知道系统所连接的设备类型，也不可能对显示性能进行假设，所以绘制屏幕的功能必须通过设备上下文 DC 来间接实现。

Visual C++ 的 MFC 封装了许多与设备环境（DC）相关的类，通过这些类使得用户可以非常容易地对设备环境（DC）进行处理。这些类不仅包含 DC 自身，而且还有字体、画笔和画刷等绘图工具。

具体的讲，进行图形程序编程就是针对 Windows 自带的图形设备接口函数库和设备环境（DC）进行操作。通过调用这些函数库中的函数，来完成绘制工作。

本章作为图形编程的基础，主要介绍了如下内容：

- 理解计算机绘图的一些基本概念；
- 什么是图形设备接口（GDI）；
- 理解设备环境 DC 及其相关的 MFC 类；
- 常用的 GDI 对象及其相关的 MFC 类介绍；
- 计算机绘图中坐标转换的基本知识。

1.1 图形编程的基础知识

Windows 图形编程主要是利用图形设备接口（GDI）中的相关函数实现的。通过确定设备环境（DC）的“状态”，以确定图形的颜色，尺寸等属性。为了使用 GDI 和 DC 来绘图必须完成以下工作：

- 确定 GDI 绘图对象，如画笔、画刷和字体等。
- 确定绘制时的缩放尺寸的映射模式。
- 确定其他细节，如文本的对齐参数，多边形的填充状态等。



1.1.1 图形设备接口 GDI

前面提到的图形设备接口 (GDI)，表示的是一个抽象的接口。换句话说，也就是相当于一个关于图形显示的函数库。通过该接口可以实现对图形的颜色、线条的粗细等属性的控制。我们的程序可以通过调用这些 GDI 函数和硬件打交道的，从而实现了设备无关性。也就是说，对于 Windows 编程不允许直接访问显示硬件，而必须通过和特定窗口相关联的“设备环境”跟显示硬件进行通讯。因此，各种 GDI 函数会自动参考被称为“设备环境”的数据结构进行绘制工作。下面的一节中将详细讨论什么是设备环境。

1.1.2 设备环境 DC

设备环境 DC (Device Context 又称设备上下文，也称设备描述表) 实际上就是一个关于如何绘制图形的方法的集合，它不仅可以绘制各种图形，而且还可以确定在应用窗口中绘制图形的方式，即确定绘图模式和映射模式。用户在绘图之前，必须获取绘图窗口区域的一个设备环境 DC。接着才能进行 GDI 函数的调用，执行适合于设备环境 DC 的命令。获取 DC 时，用户不必关心大多数的属性，因为 Windows 初始化了一套完整的属性和对象集合，用户可以使用它们渲染显示。同时，为了创建自己应用程序的特定显示，还可以更改这些属性和对象。为了便于理解，请看下面的例子：Windows 缺省的 DC 包括了一个黑色画笔，任何所绘制的线条都是黑色的。如果用户想绘制其他颜色的线条，必须用另一种所需颜色的画笔代替缺省的画笔。新颜色的画笔可在程序中建立并写入设备描述表。在 Windows 编程术语中，给 DC 提供新对象的操作被称为将绘图对象选取到 DC 中。另外，应该注意的是，Windows 的设备环境是 GDI 的关键元素，它代表了不同的物理设备。分为 4 种类型：显示器型，打印机型，内存型和信息型。每种类型的设备环境都有各自的特定用途，详见表 1-1。

表 1-1 设备环境的类型和用途

设 备 环 境	用 途
显示器型	支持视频显示器上的绘图操作
打印机型	支持打印机和绘图仪上的绘图操作
内存型	支持位图上的绘图操作
信息型	支持设备数据的访问

1.1.3 设备环境类

MFC (Microsoft 基本类库) 4.21 版中包含了一些设备环境类，其中基类 CDC 包含了绘图所需要的所有成员函数 (包括部分虚函数)，并且除了 CMetaFileDC 类之外，其所有的派生类均只有构造函数和析构函数的定义有所不同。在 MFC 中，提出这些派生类的目的是为了在不同的显示设备上进行显示。



1.1.3.1 CDC 类中常用的成员函数

表 1-2 介绍了在基类 CDC 中一些比较有用的成员函数。

表 1-2 CDC 类中一些有用的函数

函 数	说 明
Arc()	绘制椭圆弧
BitBlt()	把位图从一个 DC 拷贝到另一个 DC
Draw3dRect()	绘制三维矩形
DrawDragRect()	绘制用鼠标拖曳的矩形
DrawEdge()	绘制矩形的边缘
DrawIcon()	绘制图标
Ellipse()	绘制椭圆
FillRect()	用给定画刷的颜色填充矩形
FillRgn()	用给定画刷的颜色填充区域
FillSolidRed()	用给定颜色填充矩形
FloodFill()	用当前画刷的颜色填充区域
FrameRect()	绘制矩形边界
FrameRgn()	绘制区域边界
GetBkColor()	获取背景颜色
GetCurrentBitmap()	获取所选位图的指针
GetCurrentBrush()	获取所选画刷的指针
GetCurrentFont()	获取所选字体的指针
GetCurrentPalette()	获取所选调色板的指针
GetCurrentPen()	获取所选画笔的指针
GetCurrentPosition()	获取画笔的当前位置
GetDeviceCaps()	获取显示设备能力的信息
GetMapMode()	获取当前设置映射模式
GetPixel()	获取给定像素的 RGB 颜色值
GetPolyFillMode()	获取多边形填充模式
GetTextColor()	获取文本颜色
GetTextExtent()	获取文本的宽度和长度
GetTextMetrics()	获取当前字体的信息
GetWindow()	获取 DC 窗口的指针
GrayString()	绘制灰色文本
LineTo()	绘制线条
MoveTo()	设置当前画笔的位置
Pie()	绘制饼块
Polygon()	绘制多边形



续上表

函 数	说 明
Polyline()	绘制一组线条
RealizePalette()	将逻辑调色板映射到系统调色板
Rectangle()	绘制矩形
RoundRect()	绘制圆角矩形
SelectObject()	选取 GDI 绘图对象
SelectPalette()	选取逻辑调色板
SelectStockObject()	选取库存（预定义）图形对象
SetBkColor()	设置背景颜色
SetMapMode()	设置映射模式
SetPixel()	把像素设定为给定颜色
SetTextColor()	设置文本颜色
StretchBlt()	把位图从一个 DC 拷贝到另一个 DC，并根据需要扩展或压缩位图
TextOut()	绘制文本串

若用户（或应用程序框架）需要构造派生的设备环境类对象，就可以将 CDC 指针传给诸如 OnDraw 之类的函数。对于用屏幕进行显示来说，常用的 CDC 派生类有 CClientDC 和 CWindowDC，而对于用其他的设备进行显示来说（如打印机或内存缓冲区），则可以直接构造一个基类 CDC 的对象。

1.1.3.2 CDC 类的派生类的功能及其之间的区别

CDC 各派生类各有特点，并可以完成不同的功能，表 1-3 介绍了各派生类的主要功能。

表 1-3 CDC 类的派生类简介

派生类名称	说 明
CClientDC	这是一个设备描述表，提供对窗口客户区域的图形访问。在窗口中画图时可使用此类 DC，但对 WM_PAINT Windows 消息除外。
CMetaFileDC	这个设备描述表代表 Windows 元文件，它包含一系列命令已重新产生图像。想要创建独立于设备的文件时可使用此类 DC，用户可以回放这种文件来创建图像。
CPaintDC	这是创建响应 WM_PAINT Windows 消息的设备描述表。应用程序可以使用此 DC 更新 Windows 显示，通常在 MFC 应用程序的 OnPaint() 函数中使用。
CWindowDC	可以提供在整个窗口（包括客户区和非客户区）中画图的设备描述表

那么，这几种派生的 DC 类之间到底存在着什么区别呢？下面我们就通过比较，具体的介绍这几种 DC 类之间存在的差别：

● CWindowDC 类与 CPaintDC 和 CClientDC 类的区别

CWindowDC 类与 CPaintDC 和 CClientDC 类的区别的一个方面是：用 CPaintDC 类和