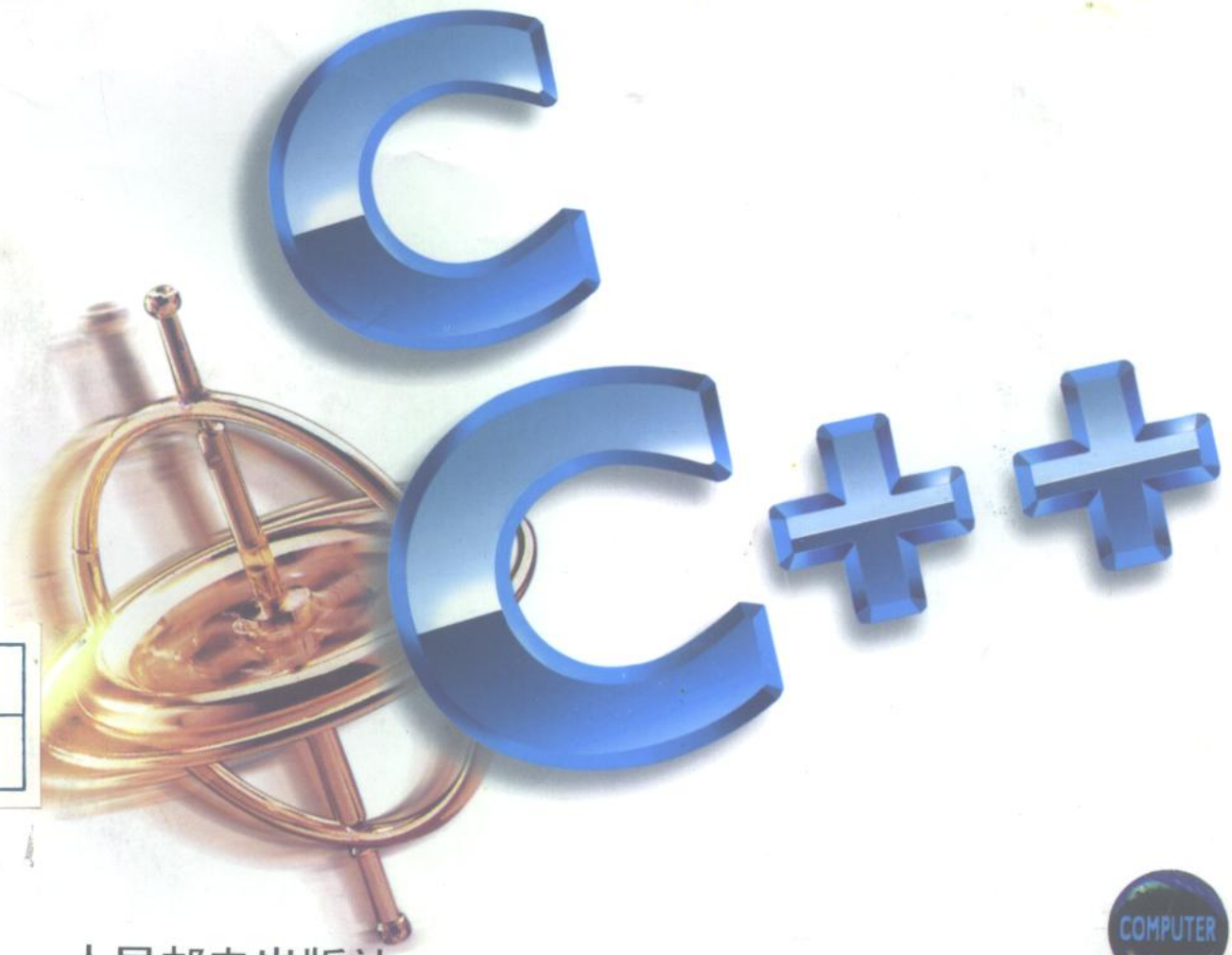
面向对象的 Windows 程序设计基础

廖湖声 编著
龙守谌 审



人民邮电出版社

312
H8/1

面向对象的 Windows 程序设计基础
人民邮电出版社

计 算 机 技 术 丛 书

面向对象的 Windows 程序设计基础

廖湖声 编著
龙守谌 审

人 民 邮 电 出 版 社

0030699

内 容 提 要

本书介绍了采用面向对象技术，利用 MFC 基本类库进行 Windows 程序设计的基本方法。全书分 10 章，分别是：Windows 系统概论，Windows 程序设计概论，面向对象程序设计和 MFC 基本类库，窗口输入消息处理，控制子窗口，菜单与对话框，图形设备接口，高级编程技术，面向对象程序设计技术的应用和通用基本类。

本书可供相关专业的大专院校师生和工程技术人员阅读，也可作为广大微机用户学习用 C 语言及 C++语言实现 Windows 程序设计的参考读物。

JS350/21

计算机技术丛书

面向对象的 Windows 程序设计基础

廖湖声 编著

龙守谌 审

责任编辑 王亚明

*

人民邮电出版社出版发行

北京朝阳门内南竹杆胡同 111 号

北京顺义兴华印刷厂印刷

新华书店总店科技发行所经销

*

开本：787×1092 1/16　1996 年 2 月　第 一 版

印张：13　1996年2月　北京第1次印刷

字数：317 千字　印数：1-- 6 000册

ISBN7-115-05892-X/TP・252

定价：18.00 元

丛 书 前 言

世界上发达国家普遍重视发展以计算机和通信为核心的信息技术、信息产业和信息技术的应用，一些经济发达国家信息产业发展迅速。

当前，我国处于国民经济高速发展时期。与此相伴随，必将有信息技术、信息产业和信息技术应用的高速发展。各行各业将面临信息技术应用研究与发展的大课题以及信息化技术改造的大任务、大工程。

为了适应信息技术应用大众化的趋势，提高应用水平，我们组织编写、出版了这套“计算机技术丛书”。这套丛书以实用化、系列化、大众化为特点，介绍实用计算机技术。

这套丛书采取开放式选题框架，即选题面向我国不断发展着的计算机技术应用的实际需要和国际上的实用新技术，选题不断增添又保持前后有序。

这套丛书中有的著作还拟配合出版软件版本，用软盘形式向读者提供著作中介绍的软件，以使读者方便地使用软件。

我们希望广大读者为这套丛书的出版多提意见和建议。

前　言

Microsoft Windows 系统在 90 年代已经成为 IBM PC 及其兼容机在 MS－DOS 下运行基于图形的多任务多窗口环境标准。Windows 系统提供的丰富的图形界面环境，使得广大微机用户和软件开发者耳目一新。广大用户无不希望使用基于 Windows 的应用系统，软件开发者也热心于为用户提供基于 Windows 的应用软件和工具软件。不容置疑，Microsoft Windows 系统已经成为微机系统软件环境的主流。

然而，在广大软件开发者中，基于 Windows 的应用程序已经获得了一个易于使用、难以开发的名声。为了开发 Windows 应用程序，软件开发者必须全面了解 Windows 系统的工作原理及其庞大的应用程序接口(API)。由上千个 API 函数、三百多个 Windows 消息以及众多的数据结构与消息结构组成的应用程序接口足以使得不少程序设计者望而怯步。

Microsoft C/C++ 7.0 系统的推出为采用面向对象方法开发 Windows 应用程序提供了支持。该系统提供的 Microsoft Foundation Class 基本类库系统(简称 MFC)为 Windows 应用程序建立了基本框架，以基本类库的形式为程序设计提供了高级应用程序接口。MFC 系统为各种 Windows 实体提供了基本类，封装了相关的大多数 API 函数、数据结构和消息结构。在程序设计中处理各种 Windows 实体时，程序开发者只要了解相应基本类提供的方法(成员函数的功能)，就能够找到实现方法。因此，可以认为学习 MFC 基本类库是掌握 Windows 程序开发技术的一条捷径。

本书力图以循序渐进的形式为读者介绍采用面向对象技术，利用 MFC 基本类库进行 Windows 程序设计的基本方法。本书在编写中尽量减少冗长无味的叙述，代之以简明的实例演示和详尽的程序注释。借助于各章举出的例题，读者有可能在几个小时内编制出自己的第一个 Windows 程序。随后逐步扩大程序功能，逐步掌握 Windows 编程和 MFC 应用的基本技巧。

龙守谌教授审校了原稿，并提出了宝贵的意见。齐钢讲师参加了附录部分的编写。在这里也向在本书编写过程中给予作者帮助的同志们致谢。

欢迎读者对本书内容的不足之处给予批评指正。

作　者

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

试读结束，需要全本PDF请购买 www.ertongbook.com

第一章 Windows 系统概论

Microsoft 公司开发的 Windows 系统是一个基于图形的多任务、多窗口环境。自从 1983 年 11 月颁布初版 Windows 系统以来，经过几次更新，目前已经发展到 1993 年 5 月的 Windows NT 系统。

Windows 系统为应用系统提供了统一的用户界面，允许用户同时运行多个程序，并且能够以多种方式进行应用程序之间的数据交换与共享，使得广大微机用户耳目一新。Windows 系统的网络功能以及对应用多媒体技术的支持更为各行业微机应用软件系统的发展展示出美好的前景。因此，各软件厂商纷纷推出基于 Windows 的各种软件工具和应用系统，进一步推动了 Windows 系统的普及和发展。在国内计算机应用的各领域，中文技术资料比较齐全、汉化系统比较完备、以标准配置的 386 和 486 微机为工作平台的 Windows 3.1 系统及其开发工具目前最为普及。

对于应用软件开发人员来说，Windows 系统提供了设备无关的图形设备接口、内存管理和程序执行与调度功能，而文件系统的管理仍依靠 DOS 操作系统。应用系统的开发者可以利用 Microsoft 公司提供的开发工具 Windows SDK 和 Microsoft C 编译系统编制 Windows 应用程序。应用程序和 Windows 系统的接口是数百个应用程序接口函数(API 函数)和系统定义消息。然而，Windows 应用程序的风格和设计方法不同于传统的结构化程序设计，陌生的程序控制手段、众多的系统函数和系统定义消息曾使不少初学者望而生畏。但是，从另一角度来看，Windows 程序采用的消息驱动体系更接近于基于对象的程序系统，面向对象程序设计方法和 Windows 应用程序的结构设计自然吻合，模块划分和控制手段基本相同，可以认为采用面向对象技术是开发 Windows 应用系统的一条捷径。

在 Microsoft C/C++ 7.0 系统中为采用面向对象技术开发 Windows 应用程序提供了基本类库(Microsoft Foundation Class)。该系统中以对象类的形式为应用系统开发提供了直接使用窗口、图形设备等各种资源和基本数据结构的手段，通过合理地组织 Windows 系统的 API 函数和系统消息，以对象类及其成员函数代替 API 函数作为应用程序接口，在相当大的程度上简化了 Windows 应用系统开发技术。

1.1 统一的用户界面

在计算机软件工程中，要求应用程序的用户界面设计充分地满足使用者的需求。这种需求包括系统功能的完备性，更要求降低使用难度，迎合用户的心理，以确保系统的使用效率。然而，各种应用系统用户界面设计风格不同，尽管可能各有长处，但往往有可能对同类的处理提供不同的操作手段，从而迫使用户分别记住各种系统的操作方法。

Windows 系统为所有应用系统提供一种统一的图形用户界面。在系统中，所有程序都以

统一的窗口形式出现，提供统一的菜单格式。窗口中使用的滚动条、按钮、编辑框和列表框等各种操作对象也都采取统一的图形显示和统一的操作方法。在使用具有这种界面的应用系统时，面对用户的不再是单一的命令行输入，而是用各种图形表示的一个个对象，用户可以通过鼠标和键盘直接操作这些图形对象。用户将通过这种统一的用户界面使用各种 Windows 应用系统，从而增强了对计算机系统的运用能力。

对于程序设计者，Windows 系统以系统 API 函数和系统定义消息的形式为在应用系统中上述各种窗口的实现提供了支持。对于输入用的用户界面，开发者还可以利用开发工具 Microsoft SDK 的对话框编辑器 DLGEDIT 实现相关程序的自动生成。

1.2 多 任 务

Windows 系统作为一种多任务系统允许同时运行多个应用程序，也允许同时运行一个应用程序的多个副本(程序实例)。使用者可以直接对各个程序的窗口进行操作，可以在各程序之间进行切换，并且可以通过动态数据交换（DDE）、动态链接库（DLL）和剪贴板(Clipboard)等手段在各个程序实例之间进行数据交换和实现数据共享。在这种多任务环境提供的空间里，使用者可以同时和多个应用系统进行对话，控制各个应用软件的执行与通信，获得与计算机系统中多个应用系统同时交流的能力。

Windows 系统的多任务功能和传统的分时操作系统不同，它的实现是依靠程序切换完成的，是一种非抢先多任务系统。一般情况下，只有在当前正在运行的程序处于等待用户输入的状态时，系统才能挂起该程序，转入其他程序的执行。Windows 系统以这种方式为各个程序提供了对 CPU 的共享。对于有特殊要求的应用系统，程序设计中可以依靠系统提供的专用系统函数和专用消息来实现程序切换。

另一方面，由于各程序之间共享计算机的资源，因此在应用程序的设计中，利用内存和输入输出设备时应考虑到其他应用程序的使用需求，保证整个 Windows 系统的工作效率。

1.3 消息驱动体系

在 Windows 系统中，消息传递是实现通信和控制的主要手段。整个系统以消息驱动的方式工作。系统中发生的用户输入操作、显示信息的改变、系统环境参数变化等所有事件都以系统定义消息的形式出现在相关的应用程序及其窗口。程序设计的主要任务就是为这些消息的处理设计代码。Windows 系统本身也根据这些消息对相关窗口的显示状态进行自动调整。

在各应用程序之间和各窗口之间的通信和控制也是利用消息完成的。发送者可以通过发消息要求接收者完成相应处理。这种消息传递可以使用系统定义消息，也可以使用用户自定义的消息。

Windows 系统中，有一个系统消息队列，用于保存系统中发生的所有消息。Windows 系统还要为每个应用程序的每个实例建立一个消息队列，依次保存发送给该程序实例的消息。在各

应用程序的主控部分，需要设置一个消息循环，利用一组系统函数从该程序实例的消息队列中依次读取和分析消息，并将它们发送给指定的窗口。

Windows 消息由消息标识(WORD message)、辅助参数(WORD wParam 和 LONG lParam)组成。消息标识表示消息种类，辅助参数用于表示该消息的其他属性。例如，消息 WM_PAINT 代表申请重画窗口中部分区域的要求；又如，WM_LBUTTONDOWN 是用户按下鼠标器左键时系统发送的消息，其辅助参数 lParam 中包含了光标位置，辅助参数 wParam 中表示被同时按下的其他键。

程序设计中需要为每个窗口设置一个窗口函数，负责处理该窗口收到的所有消息。在窗口函数中，将根据应用需求进行消息的识别和处理。开发者必须了解消息的定义、结构及其发生时间，以保证消息处理的正确无误。

按照这种消息驱动体系，各个应用程序在处理完一个消息之前，不可能进入其他消息的处理。当这种控制机制不能满足应用程序的设计要求时，则需要利用某些特殊的系统函数来发送不进队的消息或直接控制消息的传送。

1.4 内存管理

Windows 系统的内存管理突破了 DOS 系统内存 640KB 的限制。在配置 2MB 以上内存的基于 80386 以上处理器的微机上，386 增强模式下的虚拟内存管理可以直接使用 16MB 的内存空间，并且自动地在内存和硬盘之间进行页交换和重新装入。

Windows 系统采用了特殊的内存管理策略，将内存空间分为固定、可移动和可抛弃三个部分，并且提供了将共享数据使用的只读数据定义为程序资源的手段。在系统工作中，Windows 为每个应用程序的每个实例中使用的代码段、数据段和各种程序资源分配内存空间。同一应用程序的不同实例使用相同的代码段和资源，采用不同的数据段。为了提高内存空间的使用效率，Windows 系统一般要求分配的各个内存块可以移动，并且希望应用程序允许在必要时临时放弃其代码段和程序资源占用的内存空间，需要时再重新装入。

在应用程序设计中，需要确定程序的代码段、数据段和各资源是否允许被移动和重新装入，以求既满足应用程序的工作需求，又不影响多任务的整个 Windows 系统的工作效率。应用系统开发中，需要注意程序的内存模式以及可移动的内存块对程序设计的限制和影响，及时释放内存资源，最大程度地满足多任务处理的需求。

1.5 图形设备接口

在应用系统中经常使用各种图形输出和显示设备，如各种分辨率的视频显示器、打印机和绘图仪等等。为了方便涉及输出和显示设备的程序设计，Windows 系统以系统函数的形式提供了丰富的、设备无关的图形设备接口(GDI)。通过各种 GDI 函数的调用，应用程序可以完成各种图形的输出，而不必关心具体的输出和显示设备及其特殊的使用要求。

在 Windows 系统的软件包中提供了各种图形输出和显示设备的驱动程序。对于一个具体的输出设备，只要装入相应的驱动程序，就可以通过图形设备接口使用该设备。

对于 Windows GDI 函数的使用者，可以把窗口看作是一个使用基于像素的坐标系统。开发者可以在此基础上，设置图形和图像输出的各种工具和输出方式，通过建立某种映射关系构造出虚拟坐标系统，以方便各种图形设备的使用。

1.6 数据交换与共享

Windows 系统为各个应用程序之间的数据交换和数据共享提供了多种支持。

动态数据交换(DDE)是一种数据交换的消息协议，通过建立应用程序之间的数据链，实现数据的自动传送。系统为 DDE 的使用提供了一组系统定义消息。应用程序中将通过这些消息的发送和处理实现数据交换。

动态链接库(DLL)是实现代码和资源共享的一种工具。和 DOS 系统下使用的静态链接库(LIB)不同，DLL 库中的函数出现在运行中的多个应用程序时，内存中仅保存它的一个副本。再者，DLL 库中使用的资源和数据将被各应用程序共享，从而节省了内存空间。Windows 系统本身的各程序模块也是以动态链接库的形式提供的。

剪贴板(Clipboard)是用于数据交换的另一工具。利用 Windows 系统提供的工具，用户和应用程序可以向剪贴板写入数据，也可以从中读出数据，从而实现应用程序之间的数据交换。

第二章 Windows 程序设计概论

在开始学习 Windows 程序设计时，习惯于 DOS 应用程序开发的程序员往往很不适应 Windows 应用程序的设计方法。本章将结合一些 C 语言的程序实例介绍 Windows 应用程序开发的基本概念和程序结构，熟悉使用 C 语言进行 Windows 应用程序设计的读者可以直接阅读第三章，学习使用 C++语言和 MFC 基本类库的程序设计方法。

2.1 Windows 应用程序的基本结构

2.1.1 程序开发过程

在 Microsoft C/C++7.0 环境下，从源程序产生一个 Windows 环境下的应用程序的过

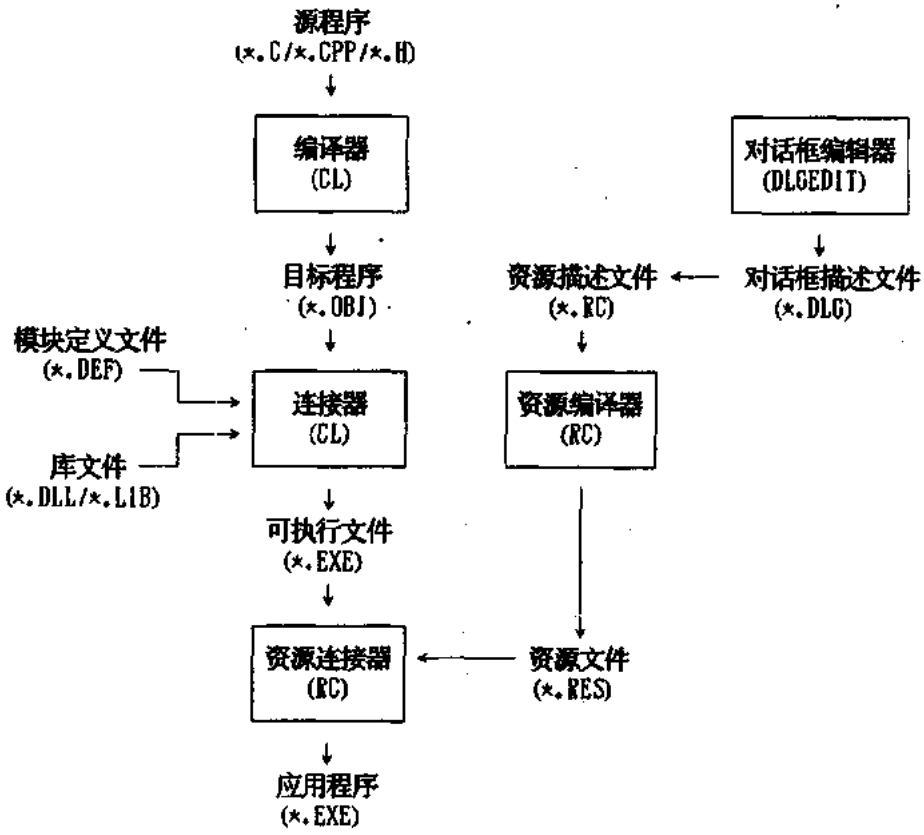


图 2.1 Windows 应用程序的产生过程

程如图 2.1 所示。

如图所示，用 C 语言或 C++语言编制的源程序经过编译连接器 CL 的处理产生以 OBJ 为文件扩展名的目标程序；随后，CL 程序中的连接器根据程序员开发的模块定义文件将目标程序和必要的库文件相连接，产生以 EXE 为扩展名的可执行文件；但是，这种可执行文件在 Windows 环境中可能还是不可执行的，利用资源编译器 RC 中的资源连接器把该文件和相关的资源文件连接起来将产生可执行的应用程序。

以 RES 为扩展名的资源文件是由资源编译器 RC 编译处理程序员编制的资源描述文件时产生的。在资源描述文件的编制中可以直接包含以 DLG 为扩展名的对话框描述文件，这种对话框描述文件是利用 Windows SDK 开发工具所提供的对话框编辑器 DLGEDIT 产生的。

综上所述，在开发一个 Windows 应用程序时，程序员应编制以下几种文件：

(1) 以 C 为扩展名的 C 程序文件或以 CPP 为扩展名的 C++ 程序文件，以及以 H 为扩展名的头文件。

(2) 以 RC 为扩展名的资源描述文件。

(3) 以 DEF 为扩展名的模块定义文件。

(4) 利用对话框编辑器 DLGEDIT 生成的对话框描述文件。

2.1.2 一个窗口的程序结构

通常情况下，Windows 应用程序的程序结构是按照窗口进行划分的，每个窗口的相关数据结构和处理函数组成一个模块。从面向对象方法的角度来看，就是把每个窗口看作是一个对象。

一个窗口的程序设计一般包括以下几步：

(1) 窗口类的登记

(2) 窗口的创建

(3) 窗口函数的设计

Windows 系统中为窗口的属性设置提供了一个结构 WNDCLASS，通过设置该结构的各分量定义一个窗口类，则使用该结构生成的窗口将具有相同的属性。在生成一个窗口之前需要定义一个窗口类变量，并设置各种属性：

```
WNDCLASS ws;
ws.lpfnWndProc     = 此类窗口的窗口函数名
ws.lpszClassName   = 窗口类的名字
...
RegisterClass( &ws );
```

结构 WNDCLASS 中的分量很多，最重要的是窗口类名 lpszClassName 和窗口函数名 lpfnWndProc。前者用于标识此类窗口，后者规定了此类窗口中负责消息处理的窗口函数。系统函数 RegisterClass 的调用将完成窗口类的登记。

在应用程序中，需要使用窗口时可以通过系统函数 CreateWindow 的调用创建窗口：

```
hWnd = CreateWindow(窗口类名,
                    窗口标题，窗口类型，
                    窗口左上角坐标 X，Y，
                    窗口宽度 CX，高度 CY，
```


试读结束，需要全本PDF请购买 www.ertongbook.com