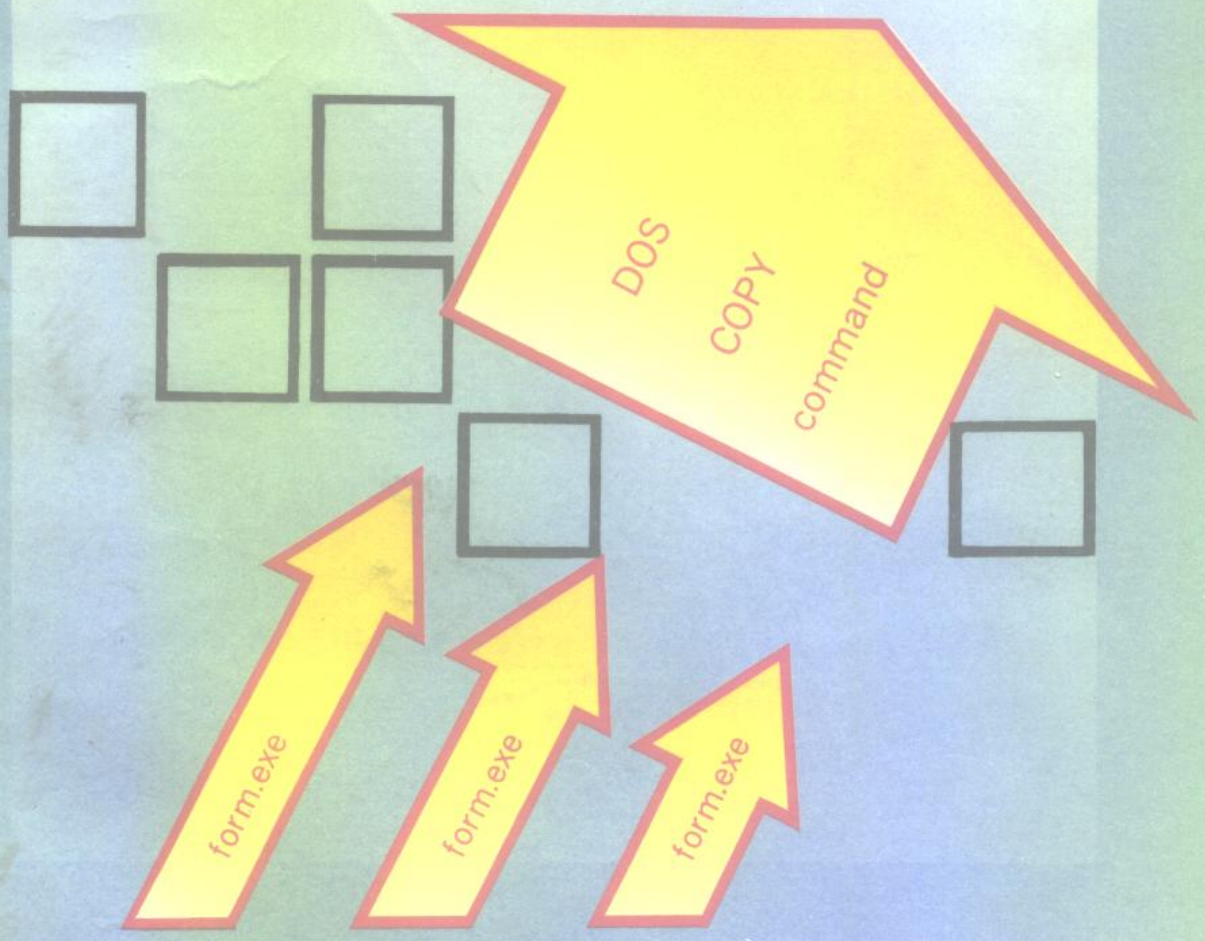
最新 Borland C++实用教程 2

Borland C++ 进阶

求 实 编著



科学出版社

2

最新 Borland C ++ 实用教程 2

Borland C ++ 进阶

求　实　编著

科学出版社

1994

（京）新登字 092 号

内 容 简 介

本书论述最新 Borland C++高级程序设计技术，包括 C++基础知识、高级知识、预处理技术、覆盖技术、嵌入式汇编语言、C++流、Container 类库，以及如何和 C++建立 Windows 应用程序，并给出了一些 Borland C++程序设计实例。对面向对象程序设计技术（Dos 下和 Windows 下）如继承窗口概念、菜单、对话框、弹出窗口、按钮、输入框等程序设计技术，此外，本书还细致地介绍了帮助用户找出影响程序执行效率的剖析器。

本书可供软件专业的程序设计、开发人员和高等院校计算机专业师生参考。

最新 Borland C++实用教程 2

Borland C++进阶

求 实 编著

责任编辑 杨家福

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

北京外国语大学印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994 年 5 月第 一 版 开本：787×1092 1/16

1994 年 5 月第一次印刷 印张：27 1/4

印数：1—5 000 字数：622 400

ISBN 7-03-004043-0/TP・341

定价：26.00 元

前　言

本书是"Borland C++ 实用教程"之二，主要讨论用 Borland C++ 设计 Windows 应用程序的方法和技巧。我们知道，作为一种方便、实用的 C／C++ 语言编译程序，Borland C++ 可以构造运行于传统 DOS 环境下的 C／C++ 语言程序。但是，随着 MS Windows 窗口图形环境的推出，Borland C++ 的主要目的已转向开发基于 Windows 的应用程序。原因是，Borland C++ 以对象作为程序单元，而 Windows 应用程序也采用面向对象的风格。例如，Windows 中的窗口就是一个对象。其次，用 Borland C++ 集成环境开发 Windows 应用程序比用 MS C／C++ 7.0 更方便、更有效。

本书按开发 Windows 应用程序的一般步骤进行组织，而且假定读者已经熟悉 Borland C++ DOS 程序设计的基本方法 (这也是"Borland C++ 实用教程"之一的主要内容)。对于初学 Windows 程序设计的人，可能会因一开始就要接触到许多全新的概念而不知所措。为此，本书的前两章用一些简单的 Windows 程序实例引入这些概念，并在此基础上讲述 Windows 应用程序的组成、开发步骤和构造方法。第三章到第五章则讨论设计 Windows 程序时所要考虑的界面对象，如窗口、菜单和对话框等。有了这些知识，读者就可以尝试着用 Borland C++ 开发简单的 Windows 应用程序了。但是，这几章的内容还没有充分利用 Borland C++对 Windows 的支持，即 Borland C++ 中固有的专用于 Windows 的类体系。这正是第七章到第十五章的主题。这几章着重讨论 ObjectWindows 程序设计，内容涉及界面、窗口、模块、对话框、控制、多文档界面 (MDI) 及流等预定义的对象。应尽量利用这些对象进行编程，而不是自己重复开发代码来实现现有的功能，因为它们可以极大地减少开发时间和编码的复杂性。在已编制出程序代码后，可利用第十六章的内容在集成环境或命令行上编译和连接各程序模块，并利用第十七章的内容加入资源，以生成可执行的 Windows 应用程序。

"动态链接库 (DLL)"是迄今为止尚未涉及到的一个陌生概念，因为它是 Windows 特有的。作为相对独立的一部分，第十八章给出了对 DLL 的完整解释。接下来的两章是一些程序实例；读者可以模仿这些实例的程序结构和设计风格。最后四章讨论如何调试和剖析用 Borland C++ 开发出的 Windows 程序。

值得指出的是，Windows 是一个复杂的窗口图形环境，限于篇幅，本书无法涉及它的全部内容，如 Windows 内存管理、图形设备接口 (GDI)、动态数据交换 (DDE) 等。因为这些内容是特定于 Windows 的，与本书的主题非直接相关，有关这些内容，请读者参考其它相关书籍。

前两章由李波、朱志勇、朱尽染执笔，第三到第六章由谢宇、阎继忠、何晓珊执笔，接下来的九章由何大庆、覃社庭、魏泱、何国辉、王勇、韩亮执笔，第十六到第十八章由魏宁、李梅执笔，第十九章到第二十章由刘秀琳、徐伟、张荣军执笔。最后四章由张勇、欧阳琼执笔。诗颖、白琴为本书的编排付出了辛苦的劳动。在此谨对以上同志表示感谢。

目　　录

第一部分　ObjectWindows 初步

第二部分 使用 ObjectWindows

第三部分 程序实例

第一部分　ObjectWindows 初步

第一章 继 承 窗 口

本章重点放在面向对象编程上。读者通过学习本章可以了解到 Windows 应用程序的构成及运行机制，同时还能看到如何利用 ObjectWindows 编写多任务协同运行的面向对象程序。

1.1 何谓 Windows 应用程序

Windows 应用程序是一种特殊的程序类型，它具有下述特点:

(1) 必须是一种专门的可执行文件格式。

(2) 仅在 Windows 下运行。

(3) 一般在屏幕上的一个矩形窗口内运行。

(4) 应用程序的显示和功能执行遵循标准的用户界面原则。

(5) 可以同时与其它的 Windows 应用程序或非 Windows 应用程序一起运行，当然还包括同一程序的其它实例。

(6) 能够与其它 Windows 应用程序通信和共享数据。

1.1.1 Windows 的优点

对用户和开发人员来说，Windows 具有许多的优点。

对用户说来，这些优点包括如下几方面:

(1) 标准且可预断的操作。

如果知道某一应用程序的使用方法，那么便能够使用所有的应用程序。因为 Windows 应用程序的界面是标准化的，它们具有一致性。

(2) 不须为每一应用程序建立各种设备的驱动程序，因为 Windows 为其所支持的全部外设都提供了驱动程序。

(3) 互相协作与通信。

(4) 多任务: 能够同时运行多个应用程序。

(5) 能访问更多的内存。

对于 80286，80386 及 i486 机器，Windows 支持保护模式；在 80386 和 i486 机器上，Windows 还支持虚拟存储。

对开发人员说来，这些优点包括如下几方面:

(1) 设备无关的图形保证图形应用程序可以运行于多种标准显示适配器。

(2) 对于各种类型的打印机、显示器和定点设备 (如鼠标和跟踪球) 的直接支持。

(3) 丰富的图形例程库。

(4) 为大型程序提供更多的内存。

(5) 支持菜单、图标、位图等等。

1.1.2 需求

Windows 提供上述诸多优点必须有一定的硬件条件作基础。一般情况下，Windows 应用程序要求较快的 CPU (相对于 DOS 应用程序而言)，较好的图形硬件 (如 VGA 卡及显示器) 和较多的内存。

对于常规使用，有一台 80286 机器，内带至少 2MB 内存，Windows 就能运行得很好了。

1.2 面向对象的窗口编程

面向窗口环境编程要求熟悉许多新的概念。即使开发一个简单的 Windows 程序都可能令人望而却步。ObjectWindows 简化了 Windows 编程的过程，它使程序员能够把主要精力放在应用程序的功能上，而不是放在它的格式上。

借助 ObjectWindows，程序员可以使用对象来表示相当复杂的 Windows 程序单元。

ObjectWindows 的窗口对象封装了所有窗口需要的数据，能执行公共的窗口操作，并且能够响应 Windows 的公共消息和事件。ObjectWindows 的窗口类和应用程序类完全管理消息的处理。

1.2.1 一个较好的 Windows 接口

ObjectWindows 使用了 Turbo C++ 面向对象特征，把 Windows API 的各个部分都封装了起来，从而使程序员同 Windows 编程的细节隔离开来。其结果是，只须花少量的时间和精力便能编写 Windows 程序。特别是，ObjectWindows 提供了下述三个有益的特点:

(1) 窗口信息的封装。

(2) 诸多 Windows API 函数的抽象化。

(3) 自动消息响应。

1. 封装窗口信息

ObjectWindows 提供了定义 Windows 窗口、对话框及窗口控制的操作与数据存储的对象。在一个 ObjectWindows 应用程序中，一个界面对象便充当一个可见的 Windows 界面元素的代表。尽管它们在运行上是紧密配合的，但是有必要弄清两者之间的区别。

界面对象是与新的界面元素相关联的。直接调用界面对象的 Create 成员函数便能生成新的界面元素。Windows 调用时传递的界面对象数据成员定义了所要生成的窗口单元的物理属性。Windows 返回所生成界面元素的句柄和标识。界面对象用其数据成员 HWindow 来存放这个句柄。

由于对窗口操作的 Windows 函数需要获取被操作窗口的句柄，故把句柄作为一个数据成员能使其访问更为容易。类似地，数据成员可以用来存放某个特定窗口的画图工具或状态信息。

2. 抽象化的 Windows 函数