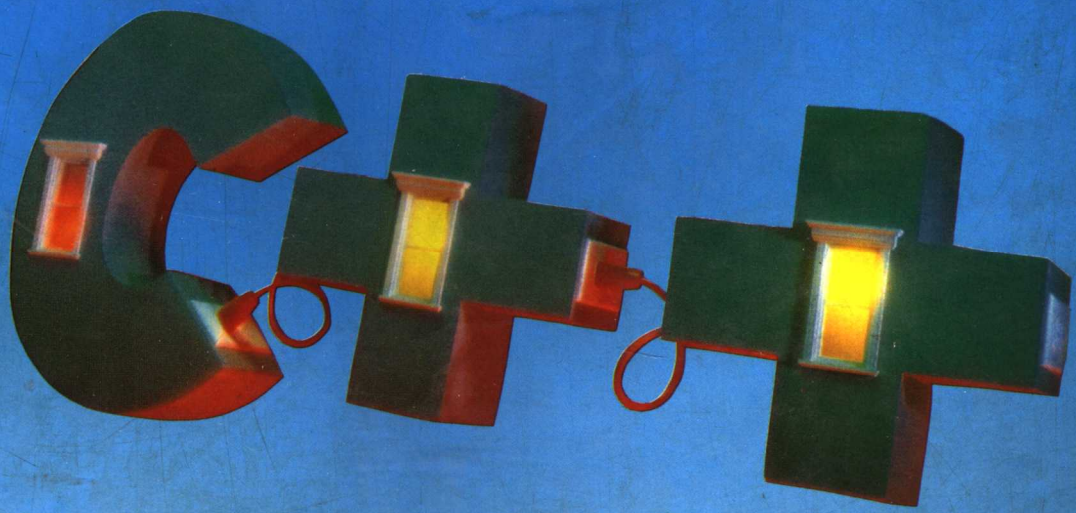
SAMS

北京科海培训中心

Borland C++3.1
编程指南

[美] Ted Faison 著
蒋维杜 吴志美
张新宇 李景淑 译
李 莉 审校



清华大学出版社

北京科海培训中心 SAMS

Borland C++3.1 编程指南

[美] Ted Faison 著

蒋维杜 吴志美
张新宇 李景淑 译
李 莉 审校

清华大学出版社

Borland C++ 3.1 Object-Oriented Programming

(Second Edition)

Ted Faison

(京)新登字 158 号

Borland C++ 3.1 编程指南

[美] **Ted Faison** 著

蒋维杜 吴志美

张新宇 李景淑 译

李 莉 审校

☆

清华大学出版社出版

北京 清华园

门头沟胶印厂印刷

新华书店总店科技发行所发行

☆

开本:787×1092 1/16 印张:50 字数:1216 千字

1993 年 12 月第 1 版 1993 年 12 月第 1 次印刷

印数:0001～5000 册

ISBN 7-302-01442-6/TP・565

定价:65.00 元

译者序

面向对象的编程法(OOP)是当前的热门课题。对许多人来讲,C++本身就是OOP的同义词,就象许多人下意识地将LISP编程法与人工智能联系起来一样。事实上,OOP并不是这种或那种语言的结果,而是其所用的特殊方法的结果。完全可能用Pascal、Ada、BASIC、汇编语言之类的语言开发OOP应用程序,尽管会更困难些。

本书以特殊的开发工具——Borland C++阐述了C++语言的OOP特点。由于当前Microsoft Windows编程方法正受到越来越多的关注,本书也涉及到这方面的有关问题,但只限于OOP框架内。读者将学会用OOP思想理解和思考Windows应用程序,使用与Microsoft Windows软件开发工具这种传统方法不同的开发方法。

本书是一本实用手册,因为其中有许多程序、工程实例。所有的例子程序和有编号的代码段都可以直接装入并编译,以验证要说明的各种不同的OOP性质。本书很大程度上考虑了功能和效率这些实际问题,读者应该具备C语言编程的经验,而且最少要有两年经验。有经验的用户将在每章尾部发现其感兴趣的材料。

引　言

20世纪80年代，C语言成为最主要、最通用的程序设计语言，使用C语言可以高效地写出可移植到各类计算机上的代码，从而使软件编制得更快，工程总体规模增大。随着工程规模的增大，软件复杂性也增加了，导致软件开发时间延长。如何缩短软件的研制时间，提高软件开发效率是许多公司面临的主要问题。AT&T公司扩充了ANSI C，开发C++语言，试图将面向对象程序设计的优点加入C语言，同时保留了使得C大受欢迎的许多特点，如简洁性、运行高效性等。

C++是为简化程序设计而开发的，因此，该语言本身要比其前身C语言更复杂。C++中增加特点的目标在于减少软件开发的困难程度。很显然，仅仅采用C++并不能保证能开发出更好或更简单的软件。为获得使用C++能带来的好处，还必须采用一种新的程序设计方法，通常称作面向对象的程序设计方法，简称OOP。

0.1　为什么使用OOP方法？

几年前，计算机科学研究者发现：不管使用何种语言，程序员都可能书写和调试几乎完全等量的代码。工作量大体相等，但结果却有所不同。编写100行C语言程序，与编写100行汇编语言程序一样困难，但同样长度的C代码能做更多的事情。考虑到这一点，研究者们试图开发一种更高层次的语言，使单个程序员的能力加倍，从而减少项目开发时间和费用。

在70年代，“对象”这一概念在程序设计语言的研究者中间流传开来。一个对象是一组代码和数据的集合，它可以模拟自然的或抽象的实体，对象作为有效的程序设计项有两个主要原因：它们代表了普遍使用的物体的直接抽象，而且，对用户掩蔽了实现的复杂性。最早研制出的对象，是与计算机联系非常密切的项，如整数、数组、栈等。一些语言（如Smalltalk）是做为传统语言开发出来的，这些 语言中的任何成分都定义为对象。

面向对象的程序设计方法把重点放在对象之间的关系，而不是实现的细节上。关系是对象之间联系的纽带，通常是通过族谱形成的，新的对象类型就是通过族谱由其它对象类型生成的。把一个对象的实现细节掩藏起来，可以使用户把注意力集中在对象与系统其他部分之间的联系上，而不是对象行为的实现过程上。这种区别是很重要的，它表明了与早期的“命令”语言（如C语言）的根本脱离，在那些命令型语言中，函数和函数调用是程序设计的中心。

在C++中，语言本身的对象很少，设计对象的职责和重任由用户完成。Borland C++包含一些对象类型，不过，要实际应用该语言还需要设计更多的对象。设计出一组组相互联系的对象，就能使OOP的作用大大发挥出来。这些组通常称为类的等级。设计类的等级是OOP的中心。

0.2 本书结构

本书描述了基于 Borland C++3.1，使用 Borland 公司提供的对象类型来设计对象等级的方法。在此之前，首先介绍了 C++的特征。C++的主要 OOP 特征在各章分别介绍。

该书分为两部分。第一部分，“用 Borland C++进行面向对象的程序设计”，从总体上描述了 C++语言，侧重于 Borland C++和使之成为面向对象语言的特点。本书并无意作为 Borland C++的完整参考，只想告诉读者怎样从面向对象的角度来运用语言的特点。

第二部分，“开发 Windows 和 DOS 应用程序”，该部分从第 8 章开始，展示了怎样使用 Borland C++的类库和实用开发框架。90 年代以来，人们认为应该用图形界面代替其他任何界面，而 Windows 是 DOS 机器上最主要的图形用户界面。预见到这点，Borland 公司已为其 C++3.0 编译器开发了两种实用框架，以便于书写 Windows 或 DOS 程序。全书中大量的内容放在例程上(配套磁盘上有源代码)。每一个例子都调试过，可以编译和立即执行。

该书的结构有些不寻常。尽管表面上是直叙的，实际上有时需要查阅后来的章节。这是因为本书选择系统地描述 C++，而不是循序渐进的方法。这使得在本书中查阅信息很方便。例如，第 2 章有关析构函数的部分包括了有关析构函数的所有信息，也提到了虚析构函数，尽管虚析构函数在第 5 章才详细描述，这种表达顺序与大多其他 C++书不同，这样做利一定大于弊。

0.3 本书的描述

本书从总体上论述 C++编程方法，侧重于 Borland C++3.1。当某个具体方面与被推荐的 ANSI C++标准通用时，常常指的是 C++，而不是 Borland C++。因为 Borland C++实质上是 AT&T C++公开版本 2.1 的超集，声明这一区别是很有必要的。

第 1 章，“基础知识”，总结了 Borland C++的主要结构而没有作任何正式的定义或说明，着重描述 C++与 ANSI C 不同的方面。尽管 C 程序可用 C++编译器编译，C++并不能使用与 C 完全相同的技术。本章还指出 C++废弃了的一些 C 的特征。

第 2 章，“对象和类”，是 ANSI C 的面向对象扩展的真正开始。它介绍了对象和类的新概念。本章描述了怎样使用代码和数据来建立对象，对象是怎样使用的，有什么性质。

第 3 章，“继承性”，说明了对象是怎样从其他对象建立起来的，而不是乱构造的。继承使对象能够继承父类的性质，从而减少了完成任务所需的编码量和调试量。继承性还使得类可以象黑匣子一样重复使用，增加了程序员的编程效率。本章讨论了继承和多重继承两种情况。

第 4 章，“重载”，讲述了函数和操作符重载。有经验的程序员看到这儿也许会打呵欠，不过，即便如此也不能“跳”过这一章，因为重载允许不同的类使用统一的符号形式表示概念上相似的动作，这是一个很重要的性质，是 C++提供给程序员的一个简化措施，帮助他们更好地管理大型工程。

第 5 章，“多态性”，C++最大力推荐的一个特点。本章详述了多态性，说明了通过虚函数的应用，简化程序设计的具体方法。本章还讲述了虚函数的优缺点，包括对其运行时间特点

的解释。

第 6 章,“流”,讲述输入和输出(I/O)。任何程序必须能输出结果才会有用。因此必须有输出信息的方式。总的说来,程序既需输入也需输出。本章从新的 C++结构——流的角度描述了输入和输出。I/O 流对文件和硬设备是一致的。该概念也适用于对内存的操作。

第 7 章,“包容类库”,是 Borland C++3.1 独有的,不能适用于其他编译器。本章描述了 Borland 装配的包容类库,并以应用例子说明。该类库对任一程序设计工程都是基本的,必须象学 Borland C++一样学习它。类的重用是使 C++多产的一个重要特征。本章不仅说明了怎样直接使用包容类库,还介绍了如何利用它们作为基类来建立自定义的类。本章详细介绍了基于对象和基于模板的包容类的情况。

第 8 章,“用于 Windows 程序设计的类”,从 C++程序员的角度介绍了微软公司的 Windows,并假定读者对作为图形用户界面和编程环境的 Windows 是熟悉的。本章讲述了在 Windows 环境下设计和开发类的方法,每个类都有简单的示例实用程序(配套磁盘上有相应的代码)。现有的有关 Windows 程序设计的文献大多只讲述了对 C 程序员非常有效的技术,对面向对象的程序设计来说还不够。在 Borland C++中,用类来简化工作量,并掩藏 Windows 程序设计通常会遇到的许多困难。

第 9 章,“完整的窗口程序”,将前述章节的知识连起来,设计编制了一个小的窗口应用程序。许多 Windows 的结构都有说明,包括消息框、对话模式、写字板界面,使用打印机,装 DLLS(动态连接库)等等。大范围地使用对象可使任务相对简单,甚至更受欢迎。

第 10 章,“对象窗口类库”,从对象窗口库(OWL)着手,研究了 Borland 实用框架。本章讲述了重用基本 OWL 类来按要求改制典型 Windows 程序不同部分的方法。本章采用的方法是低层次的,重点集中在单个类或窗口对象上,而不是应用。还将涉及传统控制、持久对象、图象闪动等方面。为理解本章和下章的内容,需对 OWL 有基本了解。

第 11 章,“OWL 应用程序”,与第 10 章相比,采用了高层次的方法进行 OWL 编程。从 OWL 类中派生出新类来支持一些一般应用需要,如状态行、弹出菜单、工具板、编辑窗等。从 Windows 程序设计的观点来看,第 10、11 章包含了本书中最有意义的内容。

第 12 章,“Turbo Vision 类”,处理另一类 Borland 实用框架,Turbo Vision(TV)。为改变 TV 应用程序的基本部分,从 TV 已有的类中派生出了一低层次的类,如状态行、菜单条、桌面等。本章末尾详细介绍了固定的 TV 对象。理解本章和第 13 章,需对 TV 有基本了解。

第 13 章,“Turbo Vision 应用程序”,使用第 12 章设计的类和例子建立了不同的 TV 应用程序。这些应用程序强调了 TV 的一些重要特点,包括上下文有关帮助、性质检查和编辑窗口。使用本章的指导,可以开发复杂的应用程序,很好地用上鼠标,可以指定颜色,多层重叠的窗口,热键和集成帮助设施。本章还展示了如何为适应需求,通过派生类来改变 TV 的基本特征,而不必考虑事件管理、图形方式等内在的细节。

0.4 软硬件需求

学编程方法可以不需要计算机,不过,有一台计算机将大有帮助。为掌握本书的内容,不仅需要研究不同例子的源代码,还得试着做些改动,并编译。由此,需要下面一些软硬件:

- IBM—PC—AT 机或其兼容机

- MS—DOS 3.31 或更高版本
- 与 Microsoft 兼容的鼠标
- EGA,VGA 或更好的图形适配器
- Borland C++3.1 版本
- Borland 实用框架(用于第 10 至 13 章)
- Windows3.1 或更高版本

第 8 章到第 13 章的示例工程文件是用 Borland C++3.1 版本编写的。如果用以前的版本,也许不能成功地使用这些工作文件。

微软公司用于 Windows 的软件开发的工具(SDK)并不一定需要。如果有 Turbo C++,或 Borland C++ 3.1 版本以前的版本,仍可以编译 1 至 6 章的代码,不过第 7 章的包容类将不能用——除非改变代码。

第 8 到 11 章的 Windows 代码需要 Borland C++3.1。如果用 Borland C++3.0 和 Windows3.0,需要对源代码或工程文件做修改。

目　录

第二部分 开发 Windows 和 DOS 应用程序