www.ncpress.com.cn
新世纪书局
PEARSON

技术经典
著作大系
信息科学

（美）Andrew Koenig，Barbara E. Moo /编著
张 明 /译

Accelerated C++ 中文版

通过示例进行编程实践

Accelerated C++：
Practical Programming by Example

参与C++演化与变革的两位大师，呈现给您世界最好的C++入门书籍

科学出版社

Accelerated C++中文版

——通过示例进行编程实践

Accelerated C++: Practical Programming by Example

(美) Andrew Koenig, Barbara E.Moo 编著
张 明 译

科学出版社

图字：01-2012-3453 号

内 容 简 介

本书系统介绍 C++程序设计，是美国斯坦福大学的经典教材。从使用 C++标准库中的高级抽象开始，使读者很快掌握编程方法。每一章都有很经典独特的例子以及非常到位的讲解，覆盖了 C++更多领域的内容，从标准库容器、泛型算法的使用，到类的设计、泛型算法的设计，本书都进行了详细的讲解。

本书作者有丰富的 C++开发、研究和教学经验，内容由浅入深，讲解精炼巧妙。无论是刚入门的新手还是有经验的 C++开发人员都能从本书中受益。

图书在版编目（CIP）数据

Accelerated C++：通过示例进行编程实践：中文版/(美)克尼格(Koenig, A.)，(美)莫欧(Moo, B. E.)编著；张明译. —北京：科学出版社，2012.6

书名原文：Accelerated C++：Practical Programming by Example

ISBN 978-7-03-034187-7

Ⅰ. ①A… Ⅱ. ①克… ②莫… ③张… Ⅲ. ①C 语言－程序设计－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2012）第 082890 号

责任编辑：周晓娟 何 武 陈 洁 / 责任校对：刘雪连
责任印刷：华 程 / 封面设计：王楠楠

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

中国科技出版传媒集团新世纪书局策划

三河市李旗庄少明印装厂

中国科技出版传媒集团新世纪书局发行 各地新华书店经销

*

2012 年 7 月 第 一 版 开本：16 开

2012 年 7 月第一次印刷 印张：22

字数：535 000

定价：69.00 元

（如有印装质量问题，我社负责调换）

前　言

讲述一种全新的C++编程方法

也许您也希望迅速掌握实用C++程序的编写方法，因此我们将在本书里首先解释C++中最有用的部分。显而易见，我们这么做有着很强的针对性，但同时也意味着即使C++是在C语言的基础上诞生的，但我们并不会以有关C语言的教学作为开始，而是一开始就使用高级数据结构，后面再对这些数据结构所依赖的基础进行解释。这个方法可以让您立即着手编写非常地道的C++程序。

从另外一个角度讲，我们还会使用非常特别的方法：集中注意力解决问题，而不是专门去探究语言与库的特征。当然，我们也会对这些特征进行解释，但这样做的目的是为程序提供支持，而并非用程序作为演示特征的工具。

因为本书并不只是讲解语言特征，主要目的是讲述C++程序设计，所以如果您已经掌握了一定的C++基础并希望以更加自然高效的风格使用这门语言，那么本书对您来说非常有用。C++的初学者通常会非常注重有关语言技巧的学习，但是常常不懂得如何运用它们去解决日常生活中遇到的问题。

无论对于初学者还是熟练的程序员，我们提供的方法都非常有用

在过去的每一个暑假中，我们都会在斯坦福大学开设为期一周的C++强化课程。刚刚开始的时候，我们采用了传统的教学方法：假定学生已经掌握C语言，于是我们在教学中从类的定义方法开始，然后再系统地过渡到语言的其他方面。我们发现，在开始之后的两三天之内，学生们会感到困惑并表现出挫折感，但是当他们掌握的知识能够让他们编写出实用的程序之后，这种感觉就会自然消失了。一旦到达了这种程度，他们就可以很轻松地继续学习下去。

当我们接触到一种能够用全新标准库提供足够支持的C++系统环境时，就会对课程进行彻底的更新。在新课程中，我们从开始就使用标准库，同时将注意力集中在实用程序的编写上，除此之外，只有在学生掌握了足以让他们高效使用各种语言细节的知识以后，我们才会对语言的细节进行深入探讨。

结果非常戏剧化：一天之后，学生就能在课堂上编写出像样的程序，这在以前的课程中需要花费近一星期。他们的挫折感也随之消失得无影无踪。

抽象

我们所采用的方法之所以适用，是因为C++（还有我们对它的了解）已经渐渐变得成熟了，这种成熟可以让我们忽略许多早期C++程序和程序员所依赖的低层次概念。

前 言

能够忽略细节的能力是成熟技术所具有的特征之一。例如，早期的汽车经常会出现故障，因此每个司机都被动地变成了业余的技工。那时，在不知道如何解决驾驶中突发问题的情况下，人们是不敢贸然驾车出门的。而今天的司机无需掌握详细的机械工程知识就可以轻松地驾驶汽车运送各种东西。当然，他们也可能会出于其他目的去仔细学习机械工程学方面的知识，但这是另外一回事。

我们将抽象定义为可供选择的忽略（将注意力集中在与手头上的任务相关的概念，而忽略其他微小的方面），我们认为这是现代程序设计中最为重要的方法。成功编写一个程序的关键在于是否清楚应该对哪些问题给予重视，应该忽略哪些部分。每种程序设计语言都提供了能够让我们创造有用抽象的工具，而每位出色的程序员都应该懂得如何使用这些工具。

我们认为抽象非常实用，因此本书中随处都会见到抽象。当然，通常情况下，我们并不会直接称其为抽象，实际上它们是以各种各样的形式出现的。例如，我们提到了函数、数据结构、类以及继承——这些都属于抽象。我们不仅仅只是提及它们，还会在本书中频繁地使用它们。

如果我们能够很好地设计和选择抽象，那么就有理由相信，即使在不了解其全部细节的情况下，也可以正确使用它们。我们不必成为机械工程师就能驾驶汽车，同样，在使用C++之前也不必一定要了解C++如何工作的细节。

本书的覆盖范围

如果您在看待C++程序设计时怀着一种非常严谨的态度，那么就有必要掌握我们在本书中所介绍的全部知识点——尽管书中并没有介绍您所需的全部知识。

虽然这种说法听上去有些绕口，但并不矛盾。没有哪一本厚度相同的书能够包含所有您要了解的C++知识，因为不同的程序员和应用所需的知识也不尽相同。因此，任何一本讲述C++知识的书籍——例如Stroustrup编著的《*The C++ Programming Language*》（Addison-Wesley，2000）——都会不可避免地介绍许多您无需了解的内容。这是因为即使您不需要这些知识，并不代表其他人也没有这种需要。

另一方面，C++的许多部分是非常重要的。因此，如果想实现更高的效率，就必须切实地掌握这些知识。我们会将注意力集中在这些重要的部分上。仅仅应用本书中介绍的信息编写各种实用程序是完全有可能的。事实上，本书的一位审阅人（一个用C++编写大型商业系统的程序员负责人）告诉我们，这本书基本上涵盖了他在工作中所需的所有工具。

有了这些工具，我们就可以编写地道的C++程序了——而不只是编写具有C语言或其他任何语言风格的程序。掌握本书中介绍的知识以后，我们就可以非常轻松地继续学习自己希望获得的知识，同时还能获得一种科学的学习方法。在业余的望远镜制造者团体中流传着这样一种说法，就是相对于一开始就制造6英寸的镜片，先制造一个3英寸的镜片再造出6英寸的镜片会更容易——对C++来说也是这个道理。

在本书中，我们仅仅介绍了标准C++而忽略了其他专门知识的延伸。这种做法有其特有的优势，也就是说我们教您编写的程序可以在任何环境下运行。尽管如此，这也表明我们不会探讨如何编写在窗口环境中运行的程序，因为这样的程序会不可避免地与特定环境或特定厂商紧密联系在一起，如果您想编写只在特定环境下运行的程序，就必须通过其他途径学习具体的编写方法——但千万不要就此放弃对本书的阅读！这是因为，我们在书中介绍的方法是通用的，以后您可以在任何环境中使用在这里学到的全部知识。当然，如果您想了解有关GUI应用原理，那么应该继续阅读一些关于这方面的书籍——但首先应阅读本书。

对熟练的C和C++程序员的提醒

在学习一门全新的程序设计语言时，我们可能会不自觉地使用自己已经了解的语言风格来编写程序。我们的教学方法从一开始就使用了来自C++标准库的高级抽象，这样就可以避免这种不自觉的行为了。对于那些熟练的C或C++程序员来说，会从这种方法中得到一些好消息和一些坏消息——实际上，好消息和坏消息都是同样的消息。

这些消息就是在我们介绍C++时，您可能会觉得很奇怪——因为您所掌握的知识对自己的学习来说好像用处并不是很大。一开始，您需要学习的知识会比意料中的多很多（这是坏消息），但学习效率也会比预期高很多（这是好消息）。特别是如果您已经对C++有所了解，那么，之前您学习的很可能是C编程，这就意味着您的C++程序风格是建立在C语言基础上的。这种做法本身没有什么错误，但是，由于我们采用的方法非常与众不同，以至于我们认为会让您看到自己从未见过的C++的另一面。

当然，许多语法细节都很相似，但它们仅仅是细节而已。我们处理重要概念时所采用的顺序很可能会与您之前接触过的完全不同。例如，一直到本书第10章我们才会提到指针和数组，除此之外，我们甚至根本不会对您可能已经很熟悉的printf和malloc进行讨论。另一方面，我们会在第1章就开始讨论标准库的string类。我们所宣称的要采用一种全新的方法是实实在在的！

前 言

本书的结构

您可能会发现，将本书分成两部分考虑会更方便一点。第一部分是从开头到第7章，在这部分内容中，我们会将注意力集中在使用标准库抽象的程序上；第二部分从第8章开始，我们会在这部分定义属于自己的抽象。

介绍库本身就是一种不同寻常的想法，但我们认为这是一种很合适的做法。C++语言的许多部分（尤其是其中比较难的部分）在很大程度上是为了体现库作者的利益而存在的，库的使用者根本无需了解这部分的语言。因此，我们一直到本书的第二部分才会提及这些内容，因此，我们很快就能开始编写实用的C++程序了。但如果使用另一种更为传统的方法，可能还要等很久才能开始进行程序的编写。

掌握库的使用方法之后，就可以开始了解比较低端的工具并且尝试用这些工具来编写自己的库了，顺便提一下，库就是在这些工具的基础之上建立起来的。另外，对于如何使一个库变得更加有用以及何时应该避免全部重写新代码这两点上，您也会获得一个感性的认识。

虽然本书的厚度比不上其他许多与C++有关的书籍，但是，对于每一个重要的概念，我们至少会使用两次以上，而关键概念的使用次数会更多。因此，我们在本书的许多部分中会提到其他部分。在提及其他部分时，我们采用诸如39.4.3节这样的形式——如果这本书有这么多小节的话。

本节的每一章（除了最后一章）都会有一个称为“小结”的内容。编写这部分内容主要有两个目的：可以让您加深对本章所述概念的记忆，同时，它们也涵盖了一些额外的相关知识——我们认为您早晚需要了解这些知识。建议您在初次阅读时跳过这些内容，在日后需要时再阅读它。

本书的两个附录从细节层次上概述并解释了语言和库的重要组成部分，我们希望这些内容会对您编写程序有所帮助。

充分利用本书

每一本有关程序设计的著作都会包含程序，本书也不例外。为了理解程序的工作方式，必须在计算机上运行它们。计算机已经非常普及了，而且新型计算机也层出不穷——也就是说，当您读到这句话时，我们提到的任何关于它们的信息都可能是不准确的。因此，如果您还不知道如何编译和运行C++程序，请访问站点http://www.acceleratedcpp.com并参阅我们发布的信息。我们会不断更新这个站点中的内容，添加

一些关于C++程序运行技巧的信息和建议。我们在这个网站中还提供了一些机器可读的示例程序版本和一些您可能感兴趣的其他信息。

致谢

我们谨对以下人员表示我们衷心的谢意，没有他们就不可能有本书。在很大程度上，本书的成功要归功于我们的审核人员：Robert Berger、Dag Brück、Adam Buchsbaum、Stephen Clamage、Jon Kalb、Jeffrey Oldham、David Slayton、Bjarne Stroustrup、Albert Tenbusch、Bruce Tetelman和Clovis Tondo。Addison-Wesley的许多工作人员参与了本书的出版工作，我们知道的有Tyrrell Albaugh、Bunny Ames、Mike Hendrickson、Deborah Lafferty、Cathy Ohala以及Simone Payment等。Alexander Tsiris审校了13.2.2节中的希腊词源。最后，以高级编程作为教学起步的想法已经在我们的脑海中萦绕了好几年，这是受数以百计参加过我们课程的学生以及成千上万参与我们讨论的人们的激励而产生的。

Andrew Koenig　Barbara E.Moo
Gillette, New Jersey
2000年6月

目　录

目 录