<<C++高级编程>>

图书基本信息

书名：<<C++高级编程>>

13位ISBN编号：9787302298977

10位ISBN编号：7302298971

出版时间：2012-10

出版时间：清华大学出版社

作者：(比)格莱戈尔(Gregoire, M.),(美)索尔特(Solter, N.A.),(美)凯乐普(Kleper, S.J.)著

页数：933

字数：1603000

译者：侯普秀,郑思遥

<<C++高级编程>>

内容概要

精通c++语言最新版本：c++11

c++是当今最流行的高级程序设计语言之一，常用于编写性能卓越的企业级面向对象程序，如游戏或大型商业软件。
但一个无法规避的事实是：c++语法纷繁复杂，学习难度较大。
如何才能化繁为简，全面系统地快速掌握c++知识呢？

c++高级编程(第2版)
将为您提供完美答案。
这本权威书籍在大量实例的引导下，解密c++中鲜为人知的特性，揭示最新版本c++11带来的显著变化，并探讨有助于提高代码质量和编程效率的编程方法、可重用设计模式和良好编程风格。
通过阅读本书，您将能得心应手地开发出优秀的c++11程序。

主要内容
提供详尽的代码范例，读者可随手在自己的代码中使用这些代码
全面介绍c++和stl技术，包括该语言不寻常和怪异的方面
展示应用c++语言高级特性的最佳实践，包括操作符重载、内存管理、制作模板和编写多线程代码
讨论编写跨语言和跨平台代码的技术
讲述代码重用的重要性以及编写易读c++代码的微妙之处

作者简介

作者：（比利时）格莱戈尔（Marc Gregoire）（美国）索尔特（Nicholas A.Solter） （美国）凯乐普（Scott J.Kleper） 译者：侯普秀 郑思遥 格莱戈尔是一名软件工程师。
他毕业于比利时鲁文的天主教大学业，获得计算机科学工程硕士学位。
之后，他在该大学获得人工智能的优等硕士学位。
完成学业后，他开始为大型软件咨询公司OrdinaBelgium工作。
他曾在Siemens和Nokia Siemens Networks为大型电信运营商提供有关在Solaris上运行关键2G和3G软件的咨询服务。
这份工作要求与来自南美、美国、欧洲、中东、非洲和亚洲的国际团队合作。
Marc目前在Nikon Metrology任职，负责开发3D扫描软件。
他的主要技术专长是C／C++，特别是Microsoft VC++和MFC框架。
除了C／C++之外，Marc还喜欢C#，并且会用PHP创建网页。
除了在Windows上开发的主要兴趣之外，他还擅长在Linux平台上开发24x7运行的c++程序；例如EIB家庭自动化监控软件。
2007年4月，因为在Visual c++方面的专业才能他获得了年度Microsoft MVP称号。
Marc还是CodeGum论坛的活跃分子（id为Marc G），并且为CodeGum撰写了一些文章和FAQ条目。
他还编写了一些自由软件和共享软件，并通过他的网站www.nuonsoft.com发布。
他还在www.nuonsoft.com／blog／维护了一个博客。
索尔特是一名计算机程序员，开发的软件范围很广，包括系统软件、游戏、网络服务和其他很多类型。
他在Sun Microsystem的高可用集群上所做的工作获得了3项专利，还就此在国际并行和分布式处理会议上发表了一篇技术论文。
在Sun的时候，他还喜欢参与OpenSolaris，他还是OpenSolaris Bible（Wiley，2009）的第一作者。
现在重新从事Web开发，他很开心地再次和Scott一起在Context Optional工作。
Nick在斯坦福大学学习计算机科学，他在这所大学获得了本科和理科硕士学位，他的主要研究领域是计算机系统。
他曾在富勒顿社区大学讲授了一年的C++课程。
Nick和他的妻子和两个孩子生活在美丽的科罗拉多，他在科罗拉多享受着雪上运动的乐趣。
凯乐普在小学就开始了他的编程生涯，那时他在Tandy TRS—80上用BASIC语言编写探险游戏。
作为他所在高中的Mac迷，Scott转向了更高级的语言，并且发布了一些屡获殊荣的共享软件。
Scott加入了斯坦福大学，并且在这所大学获得了本科和计算机科学的理学硕士学位，主要研究领域是人机交互。
在上大学的时候，Scott是一门涉及编程入门、面向对象设计、数据结构、GUI框架和小组项目的课程的助教。
他之后在斯坦福的一门课程采用这本书作为课本。
毕业后，Scott是几家公司创始团队中的首席工程师。
2006年，Scott与他人合伙创建了ContextOptional，Inc.，这是一家提供社会营销技术的市场领先的供应商。
在工作之余，Scott还热衷于在线购物、阅读和弹吉他。

<<C++高级编程>>

书籍目录

章节摘录

版权页： 插图： 提供整洁的接口 为了避免在接口中遗漏功能，某些程序员走向另一个极端：他们包含了可以想到的所有功能。
使用这个接口的程序员总是可以找到完成任务的方法。
遗憾的是，这个接口可能非常混乱，他们无法指出如何实现这个接口。
不要在接口中提供不必要的功能，保持接口的简单整洁。
乍看上去，这个指导方针与前面避免遗漏必要功能的策略相违背。
为了避免遗漏功能而包含所有想象得到的接口尽管是一个策略，但是并不是一个健全的策略。
应该包含必要的功能并省略不必要甚至起反作用的接口。
再次考虑汽车示例。
开汽车只需要使用几个组件：方向盘、刹车、油门踏板、换挡、后视镜、里程计以及仪表板上的一些其他仪表。
现在想象一下，如果汽车的仪表板与飞机的驾驶员座舱类似，具有上百个仪表、控制杆、监控器以及按钮，这没法用！
由于接口比较简单，开汽车比开飞机容易多了：您不需要关心海拔高度、与控制塔通话或者控制飞机中众多的组件（例如机翼、发动机以及起落装置）。
此外，从发展的观点来看，比较小的库容易维护。
如果您试图让每个人都很愉快，那么就应该留出更多的空间来容纳错误，如果您的实现非常复杂以至于纠缠不清，哪怕一个错误也能让库变得无效。
遗憾的是，设计简洁接口的思想看起来很好，但是实际上非常困难。
这个规则基本上是主观的：您决定什么是必须的，什么不是。
当然，当您的判断出错时客户一定会通知您。
提供文档以及注释 无论接口多么便于使用，都应该提供使用文档。
如果不告诉程序员如何使用，不能期望他们会正确使用库。
应该将库或代码称为供其他程序员使用的产品。
产品应该带有说明其正确用法的文档。
提供接口文档有两种方法：接口自身内部的注释以及外部的文档。
您应该尽量提供这两种文档。
大多数公开的APl只提供外部文档：在许多标准Unix以及Windows头文件中都缺少注释。
在Unix中，文档的形式通常是名为inan pages的在线手册。
在Windows中，集成开发环境通常附带文档。
虽然多数API以及库都取消了接口内的注释，但是我们认为这种形式的文档才是最重要的。
绝不应该给出一个只包含代码的“赤裸的”头文件。
即使注释与外部文档完全相同，具有友好注释的头文件也比只有代码的头文件看上去舒服，即使是最优秀的程序员也希望经常看到书面语言。
有些程序员使用工具将注释自动转换为文档，第5章详细讨论这一技术。
无论您提供注释、外部文档还是二者都提供，文档都应该描述库的行为而不是实现。
行为包括输入、输出、错误条件以及处理、预定用法以及|生能保障。
例如，描述生成单个随机数的调用的文档应该说明这个调用不需要参数，返回一个预先指定范围的整数，还应该列出当出现问题时可能抛出的所有异常。
文档不应该详细解释实际生成数字的线性同余算法，在接口注释中提供太多的实现细节可能是接口开发中最常见的错误。
适用于维护库的人员而不是客户的注释会破坏接口以及实现的良好分离，许多开发人员都见到过这种情况。
当然内部的实现也应该有文档记录，只是不要把它作为接口的一部分公开。
第5章详细讨论如何在代码中恰当地使用注释。

<<C++高级编程>>

编辑推荐

《C++高级编程(第2版)》并不是讲解语言的大量细节并给出少量真实世界的场景，而是教您如何在真实世界中使用C++。
《C++高级编程(第2版)》还会披露一些鲜为人知的特性，使用这些特性可以让编程更简单；还讲解了可重用的编码模式，模式是区分编程新手和专业程序员的标志。
编程书籍往往重点描述语言的语法，而不是语言在真实世界中的应用。
典型的C++教材在每一章中介绍了语言中的大部分知识，讲解语法并列举示例。
格莱戈尔、索尔特、凯乐普编著的《C++高级编程(第2版)》不遵循这个模式。

版权说明