

Broadview
www.broadview.com.cn

案例的选择与评估
类模板特化
标准库及Qt对字符串的处理
国际化与区域文化
C++的iostream

Qt的流
隐式共享与d-pointer技术
Qt容器与迭代器
多线程与可重入
信号与槽
Graphics/View框架

Model/View框架
Qt中的命令模式
Qt中的抽象工厂模式
Qt中的观察者模式
Qt的元对象系统
智能指针

Qt中的 C++技术

· 张波 编著 ·

国内首本深入剖析Qt设计理念及源代码的书籍

- 研读最优秀的C++开源项目，学习如何使用C++
- 以**80个**形象、鲜活的例子诠释面向对象设计思想
- 系统阐述了**8种**设计模式
- 分析比较了Qt与**C++标准库**的设计理念
- 全面覆盖C++的各种**语言特性**

让我们从这本书开始，一起体验C++处理复杂问题的强大及其精妙吧！



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Qt中的C++技术

· 张波 编著 ·

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书剖析了开源开发框架 Qt 中的 C++ 技术，给读者提供一个优秀的案例，以学习 C++ 语言以及面向对象设计技术。该书讨论了以下内容：类模板特化技术；分析比较了 C++ 标准库、Qt 对字符串、数据输入/输出的处理思路；隐式共享与 d-pointer 技术；函子及其在 QTL (Qt Template Library) 中的应用，QTL 是如何使用模板特化技术优化 QList 性能的；如何在 C++ 程序中嵌入汇编代码，实现一个原子操作，以很小的开销实现线程间通信；信号与槽机制；Graphics/View 框架等。

软件学院或者计算机学院的学生，可将本书作为课程“C++ 程序设计”或者“面向对象软件设计”的参考书；上述课程的教师，可将本书的内容融入他们的主讲或者试验环节，作为相关实训课程的教材；软件行业的开发者，可将本书作为深入学习 C++ 设计与编程技术的案例教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Qt 中的 C++ 技术 / 张波编著. —北京：电子工业出版社，2012.7
ISBN 978-7-121-17159-8

I. ①Q… II. ①张… III. ①软件工具—程序设计 ②C 语言—程序设计
IV. ①TP311.56 ②TP312

中国版本图书馆 CIP 数据核字 (2012) 第 106582 号

责任编辑：孙学瑛

特约编辑：顾慧芳

印 刷：北京中新伟业印刷有限公司

装 订：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：19.25 字数：444 千字

印 次：2012 年 7 月第 1 次印刷

印 数：4000 册 定价：55.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线：(010) 88258888。

据 2011 年 12 月 Tiobe 网站 (www.tiobe.com) 的排名, 最流行的前 5 个编程语言依次是 Java、C、C++、C# 以及 Objective-C。排名的依据是熟练使用一种语言的人数, 与该语言相关的课程数量以及支持该语言的第三方供应商的数量。自 2001 年这个排名标准诞生以来, C++ 几乎总是处于第 3 名。虽然 Perl、Visual Basic 以及 PHP 也曾占据过这个位置, 但是它们只能在这个位置上维持几个月。

软件行业的骨架是由 C++ (及其兄弟 C) 搭建起来的。

- 三大桌面操作系统 Windows, MacOS 以及 Chrome OS 使用了 C++。
- 运行在苹果公司 iPhone, iPod, iTouch 以及 iPad 上的操作系统, Windows Mobile 和 Symbian OS 使用了 C++。
- 在关系数据库管理系统方面, 主流的产品 Oracle Database, MySQL, IBM DB2, Microsoft SQL Server, IBM Informix, SAP DB/MaxDB 都使用了 C++。
- Web 浏览器方面, 依据网络分析公司 StatCounter 2011 年 11 月发布的数据, 全球用户数量最多的前 5 款浏览器依次为微软的 IE, Google Chrome, Mozilla Firefox, Safari 以及 Opera, 它们全部使用了 C++。
- 流行的办公套件 Microsoft Office, Sun Open Office, Corel Office 也都使用了 C++, 其中 Corel Office 在开发过程中曾使用过 Java, 但由于速度太慢最终转回 C/C++。
- 甚至, 出乎一般人的预料, C++ 也被用来开发网站。Google 的网站采用 C++ (及汇编), eBay 和 Amazon 采用 C++ 与 Java, Facebook 采用 LAMP (Linux+Apache+Mysql+PHP) 外加 C++。

C++ 的语言特性

C++ 的地位是由其鲜明的语言特性决定的。

(1) 它兼容 C, 意味着 C++ 项目可以复用过去 40 年来积累的 C 函数库以及 C 源代码。

(2) 对复杂系统的抽象表达能力。C++ 和其他面向对象编程语言一样, 使用“类”封装底层的数据和相关的操作, 使用“继承”描述基类和派生类的共性, 使用“多态性”描述不同子类的差异。一个复杂的系统总会被表示为一组具有清晰逻辑结构的类。为了解决复杂软件系统中名字冲突的问题, C++ 引入了函数名重载 (function overloading)、名字空间 (namespace) 机制。

(3) 执行速度快。C++ 和其他面向对象编程语言不同, 并不追求形式上的简单。为了提高程序执行速度, C++ 不惜将自己变得更加复杂。模板技术以及基于该技术的泛型编程思想

在编译阶段处理多态性问题，使得一个类模板或者函数模板能够处理各种类型。而其他编程语言采用虚函数与多态性解决类似的问题，相比之下，C++模板的执行速度更快。

(4) C++遵循“用时付费”原则。例如，有些编程语言将垃圾回收 (garbage collection) 作为语言内在的功能，无论一个程序是否需要，该机制在程序运行期间总会占用一定的计算资源。对于 C++，即使演化到最新的 C++0x11 标准，它仍然坚持不纳入垃圾回收机制，因为它认为需要该功能的程序可以采用第三方库，而不需要该功能的程序不应花费任何额外计算资源。C++程序被编译为可执行代码后可以直接运行，不像其他一些语言那样需要一个虚拟机来解释执行。在对速度要求极高的场合，我们可以在 C++程序中嵌入汇编代码。这些语言特性都能提高 C++程序的执行速度。

本书特点

然而，也正因为 C++的这些语言特性，使其成为所有面向对象语言中最复杂的一个。许多学习者抱怨 C++语言“难学难精”。为了降低学习难度，提高学习效率，C++学习者至少应该阅读两种类型的教材：

- 讲述“C++是什么”的教材，如 *Thinking in C++* 以及 *C++ Primer* 等。
- 讲述“如何使用 C++”的教材，如 *Effective C++* 以及 *The C++ Programming Language* (每章的 Advice 部分与 Part IV)。

阅读第一种类型的教材时，学习者通常不会遇到什么障碍，然而在阅读第二种类型的教材时，学习者有时会觉得其中一些 C++编程经验听起来有道理，但是在编程实践中并不一定能够灵活运用它们。有这种感觉并不奇怪，因为这些编程经验是 C++高手们从几十年的编程实践中总结出来的，要真正理解它们，需要读者也能以某种方式重演这些经验背后的编程实践。

一种最有效的方式就是选择一个优秀的 C++案例，研读它、模仿它，也就是说，C++学习者还应该阅读第三种类型的教材：案例教材。

目前有如下两种类型的案例教材。

(1) 偏重介绍如何使用某种开发框架 (如 MFC) 的教材。学习者看完这些教材，的确可以编写出一些具有图形界面的程序，但是无法体会如何使用 C++语言的特性来解决复杂的设计问题、性能问题。更值得注意的是，这些教材中的案例往往都是为了书籍的出版而特意设计的，并非来源于实际软件项目。其系统设计、代码质量未经同行审阅，可能会误导学习者。

(2) 深度剖析 C++标准模板库 (STL) 的教材。由于 STL 本身精巧的设计，阅读这类教材的确可以帮助学习者掌握 C++ (尤其是模板技术) 的精髓，但是这类教材抽象而深奥，令大多数学习者望而却步。

读者希望临摹如下这样的案例。

(1) 案例具有良好的设计、高质量的代码，已被成功地广泛使用。

(2) 案例不要涉及太多、太深其他领域的知识。

(3) 案例可以比较复杂（以展现 C++ 对复杂问题的处理能力）但是不要过于抽象，这样，读者可以通过形象、鲜活、具体的例子，体会 C++ 语言特性的应用。

(4) 案例应该全面覆盖 C++ 的各种语言特性。

寻找这样一个案例并非易事。在众多的开源项目中，我们最终选择了 Qt。Qt 是一个跨平台的 C++ 开发框架，它包含一个功能丰富的 C++ 类库以及一套简便易用的集成开发工具。Qt 所支持的平台不但包括 Linux，Windows 以及 Max OS X 等主流桌面操作系统，还包括诸如 Symbian，Maemo 以及 MeeGo 这样的嵌入式操作系统。

使用 Qt 编写的 C++ 程序具有良好的跨平台特性，程序员几乎无须更改源代码，所编写的应用程序即可运行在各种操作系统中，这能大幅度缩短开发周期、降低开发成本。Qt 的 C++ 类库是完全面向对象的，该类库不但功能强大，而且设计精良、方便易用。这些优点使得 Qt 被 Adobe®，Boeing®，Google®，IBM，Motorola®，NASA，Skype® 等大型机构以及众多的中小公司采用。

Qt 类库非常复杂，仅本书剖析的两个子模块 QtCore 以及 QtGui 就有大约 22 万行代码，其复杂程度足以展现 C++ 对复杂问题的处理能力。该类库不但全面覆盖了 C++ 的各种语言特性，还用到了 MVC (Model-View-Control) 框架、隐式共享、信号与槽、命令模式、抽象工厂模式、观察者模式等精妙的设计技巧，是一个值得学习者临摹的案例。

本书内容

本书共 18 章。第 1 章讲述为什么会从众多的开源 C++ 项目中选择 Qt。读者可以借鉴其中的方法选择其他 C++ 案例，或者在学习其他编程语言时，使用其中的方法选择对应的案例。而且，读者还可以使用其中的工具 CppDepend 剖析其他软件的结构与质量。这一章还介绍了本书对术语、UML 类图方面的约定。在阅读后续章节前，读者应该首先阅读这一章。

本书不但剖析 Qt 的源代码，有的章节还涉及修改 Qt 的源代码，此时需要重新编译整个 Qt 库。第 2 章简要介绍 Qt，并讲述如何在 Visual Studio 2010 开发环境下安装、编译 Qt 库。Qt 库多处用到了类模板特化技术。考虑到一般的 C++ 教科书不会详细讲解这个话题，故第 3 章阐述该技术的概念和基本应用，第 6 章及第 9 章用到了该技术。

绝大多数 C++ 程序都会涉及字符串的处理，因而在 Qt 提供的所有功能中，我们首先在第 4 章讲述 Qt 对字符串的处理思路。和 C++ 标准库不同，Qt 不再区分单字节字符串、宽字节字符串，而是使用类 QString 表示所有类型的字符串，每个字符用类 QChar 表示，至少占用 2 个字节。这一章分析比较了 C++ 标准库、Qt 对字符串的两种处理思路。

数据的输入输出也是绝大多数 C++ 程序需要使用的功能。第 5 章~第 7 章分析比较 C++

标准库、Qt 对数据输入/输出的不同处理思路。Qt 的流框架功能丰富，易于使用但是可扩展性差。而 C++ 标准库的流框架具有良好的可扩展性，但是其本身的功能偏弱，不便于使用。

如果一个类的某些对象的数据成员具有完全相同的取值，我们可以令它们共享一块内存以节省空间。只有当程序需要修改其中某个对象时，我们再为其分配新的内存。这种技术被称为隐式共享，在 Qt 库中被广泛使用。第 8 章介绍该技术，并剖析了 QString 的部分源代码以演示该技术的具体实现。除此之外，这一章还介绍了 d-pointer 技术，该技术将一个类的所有数据成员分离出来，定义在另外一个类中，并在原先的类中定义一个指针，指向另外一个类。这种技术能维持 Qt 库的二进制数据兼容性、提高 Qt 库的编译速度。

起初开发 Qt 时，C++ 的标准模板库 STL 尚未成为业界标准，部分编译器根本没有配备 STL。为了实现良好的跨平台特性，Qt 的设计者干脆开发了一个类似于 STL 的模块，将其集成在 Qt 软件体系中，本文将其称为 QTL (Qt Template Library)。Qt 其他模块大量使用了 QTL，使其成为 Qt 软件体系的基石。正如设计精良的 STL 能够引来人们为其出书立著一样，QTL 的设计也有可圈可点的特点。第 9 章介绍函数的概念及其在 QTL 中的应用，以及 QTL 是如何使用模板特化技术优化 QList 性能的。

目前的 C++ 标准并不支持并发处理，但是并发处理的确可以提高程序性能、改善用户体验。因此，Qt 封装了不同操作系统的细节，向程序员提供了一组和平台无关的类来处理并发问题。第 10 章介绍如何使用 QThread 创建一个线程，如何使用互斥体 QMutex，信号量 QSemaphore 及条件量 QWaitCondition 实现线程同步。以上这些技术运行速度慢、开销大。这一章以多线程环境下 singleton 模式的实现为例，讨论了如何在 C++ 程序中嵌入汇编代码，实现一个原子操作，并巧妙地利用这个原子操作，以很小的开销、非常快的速度同步多个线程。

Qt 使用信号与槽机制进行对象间的通信。第 11 章阐述 Qt 为什么使用信号与槽机制而不是传统的回调函数进行对象间的通信。信号与槽机制大幅降低了各对象之间的耦合度，各对象的代码可以被独立地开发、测试，也更容易被复用。这种机制是 Qt 开发框架区别于其他框架的典型特征。

Qt 的 Graphics/View 框架被用来存放、显示二维图形元素，处理那些对图形元素进行操作的交互命令。第 12 章介绍 Qt 图形系统的基本知识，Graphics/View 框架如何处理图形元素，程序员如何定义新的派生类，以配合该框架实现动画效果。通过学习本章，读者将体会到虚函数的作用：一个开发框架实现基本、通用的功能。通过重载开发框架提供的虚函数，应用程序的代码能够和框架中的代码协同工作，达到复用框架代码的目的。

模型-视图-控制器架构 (Model-View-Control structure, MVC) 是一种流行的软件体系架构，它将数据及其显示分离开来，使用模型存取数据，使用视图显示数据，使用控制器处理用户的交互命令。第 13 章详细阐述 Qt 如何实现这个框架。该框架涉及 Qt 库的 36 个类，结构复杂、设计精良。这一章的篇幅最长，读者将从本章体会到一个复杂面向对象系统背后的设计理念。

Qt 库用到多种设计模式。第 14 章~第 16 章阐述 Qt 如何实现命令模式 (command pattern)、抽象工厂模式 (abstract factory pattern) 以及观察者模式 (observer pattern)。对于每一种模式, 我们先简要介绍它的一般定义, 再讨论 Qt 使用哪些类/机制实现这种模式, 最后给出一个例子演示该模式的作用。

Qt 库的另外一个显著特征是构建了一个比标准 C++ 的运行时类型信息 RTTI (Run-time Type Information) 系统功能更强大的元对象系统 (meta-object system)。第 17 章回顾了标准 C++ 中的 RTTI 系统, 讨论了它的局限性, 阐述了 Qt 的元对象系统以及如何使用该系统来获取对象的运行时信息。

C++ 的指针是一把双刃剑: 一方面, 它可被用来构建复杂的数据结构。但是另一方面, 对指针的错误使用将导致内存泄漏以及野指针等问题。第 18 章介绍 Qt 提供的一些智能指针。这些智能指针对普通 C++ 指针进行封装, 能够有效解决上述问题。

本书虽然侧重于剖析 Qt, 但不是系统讲授如何使用 Qt 开发 GUI 应用程序。然而, 本书的确详细阐述了 Qt 的一些基本组成部分, 比如输入/输出流、隐式共享、信号与槽、Graphics/View 框架、Model/View 框架等, 理解这些组成部分可以帮助读者尽快掌握 Qt 的使用。

本书读者对象

本书面向以下读者: 软件学院或者计算机学院的学生, 将本书作为学习“C++ 程序设计”或者“面向对象软件设计”课程的参考书; 上述课程的教师, 将本书的内容融入他们的主讲或者试验环节; 软件行业的开发者, 与本书一起领略 Qt 的精妙。

笔者由衷地感谢与敬佩 Qt 的创始人 Haavard Nord 以及 Eirik Chambe-Eng。正是他们为 C++ 世界带来了如此精彩的一个开发框架, 并且无私地公开了整个框架的源代码, 允许成千上万的 C++ 程序员以 GPL 授权方式自由地使用、修改、发布这个库。另外, 本书源于南开大学以及南开大学软件学院资助的一个教学改革项目, 在此一并表示感谢。最后, 感谢董颖、于洋等人对本书草稿的校对。

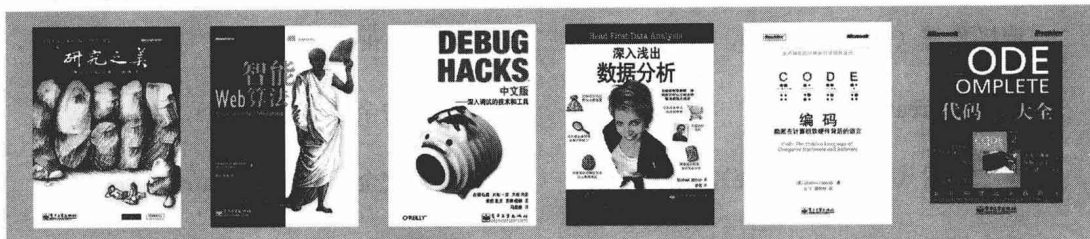
虽然笔者付出了近两年时间剖析 Qt、撰写此书, 但由于 C++ 以及 Qt 的复杂, 本书未能涵盖 Qt 的所有技术亮点。对本书的批评与建议请发送至 zhangbo_nk@163.com。关于本书的勘误、讨论以及其他信息, 请访问笔者个人博客 blog.csdn.net/zhangbo_nk。

张 波

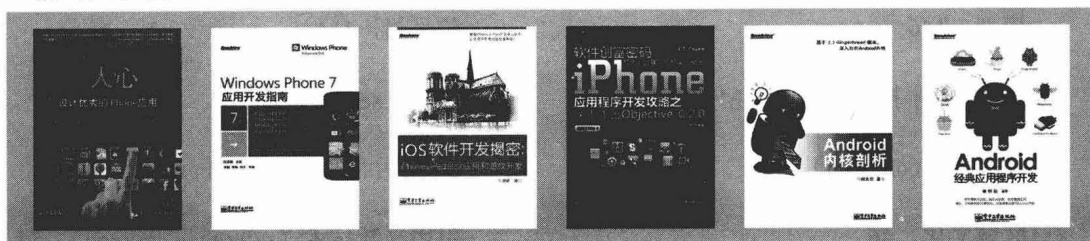
2012 年 3 月 1 日于南开大学

博文视点精品图书展台

专业典藏



移动开发



物联网

云计算



数据库

Web 开发



程序设计

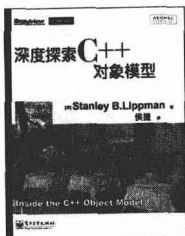


办公精品

网络营销



博文视点本季最新最热图书



《深度探索C++对象模型》

【美】Stanley B. Lippman 著 侯捷译

定价：69.00元

内容简介：作者 Lippman 参与设计了全世界第一套 C++ 编译程序 cfront，这本书就是一位伟大的 C++ 编译程序设计者向你阐述他如何处理各种 explicit（明确出现于 C++ 程序代码中）和 implicit（隐藏于程序代码背后）的 C++ 语意。

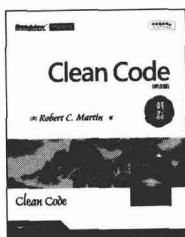


《研究之美》

【美】D.E. Knuth 著 高博译

定价：49.00元

内容简介：本书是计算机科学大师、“算法分析之父”高德纳（Donald E. Knuth）在 20 世纪 70 年代旅居挪威时撰写的适用于计算机科学的一种全新基础数学结构的情景小品。全书以一对追求自由精神生活的青年男女为主人公，展开了一段对于这种全新结构的发现和构造的对白。

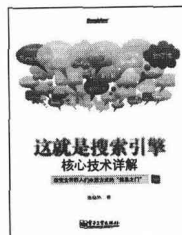


《Clean Code (评注版)》

【美】Robert C. Martin 著 韩磊评注

定价：79.00元

内容简介：软件质量，不但依赖架构及项目管理，而且与代码质量紧密相关。这一点，无论是敏捷开发流派还是传统开发流派，都不得不承认。本书提出一种观念：代码质量与其整洁度成正比。干净的代码，既在质量上较为可靠，也为后期维护、升级奠定了良好的基础。本书作者给出了一系列行之有效的整洁代码操作实践。



《这就是搜索引擎：核心技术详解》

张俊林 著

定价：45.00元

内容简介：本书的最大特点是内容新颖全面而又通俗易懂。对于实际搜索引擎所涉及的各种核心技术都有全面细致的介绍，除了作为搜索引擎核心的网络爬虫、索引系统、排序系统、链接分析及用户分析外，还包括网页反作弊、缓存管理、网页去重技术等实际搜索引擎必须关注的技术，同时用相当大的篇幅讲解了云计算与云存储的核心技术原理。



《Android经典应用程序开发》

韩超 编著

定价：59.00元

内容简介：本书内容理论联系实际，系统化组织。全书由 8 章组成，涵盖了 Android 经典应用程序开发的各个方面，知识量巨大，技术要点清晰，结构紧凑而精炼。拥有本书后，结合 Android 公开的参考文档和示例代码，读者将可以基本掌握在 Android 经典应用层开发领域的所有方面。



《疯狂Java讲义 (第2版)》

李刚 编著

定价：109.00元

内容简介：本书是《疯狂 Java 讲义》的第 2 版，第 2 版保持了第 1 版系统、全面、讲解浅显、细致的特性，全面介绍了新增的 Java 7 的新特性。

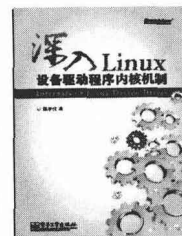


《专业嵌入式软件开发——全面走向高质量编程》

李云 著

定价：108.00元

内容简介：本书分为 6 篇。硬件篇就嵌入式软件开发所需掌握的处理器概念进行了介绍。工具篇对 make、gcc 编译器、binutils 工具集、ld 链接器和 gdb 调试器进行了讲解，其中对 make 这一嵌入式开发环境的全能管家进行了精辟的介绍，致力于帮助读者成为 Makefile 方面的专家。



《深入Linux设备驱动程序内核机制》

陈学松 著

定价：98.00元

内容简介：这是一本系统阐述 Linux 设备驱动程序技术内幕的专业书籍，它的侧重点不是讨论如何在 Linux 系统下编写设备驱动程序，而是要告诉读者隐藏在那些设备驱动程序背后的那些内核机制及原理。作者通过对 Linux 内核源码抽丝剥茧般的解读，再辅之以精心设计的大量图片，使读者在阅读完本书后对驱动程序前台所展现出来的那些行为特点变得豁然开朗。



《从Oracle到DB2开发——从容转身》

王飞 李小明 李睿明 成玫论 编著

定价：69.00元

内容简介：本书是“舞动 DB2”系列的第二本，分为三大部分，共 8 章。第一部分从开发者遇到的问题进行分析，提出在 Oracle 兼容模式下开发数据应用的新思路，并讲述了从 Oracle 迁移到 DB2 的方法、工具及案例。第二部分讲述了 DB2 开发工具方面的知识，以及如何更有效地开发存储过程、用户自定义函数和触发器。



《剑指Offer：名企面试官精讲典型编程题》

何海涛 著

定价：45.00元

内容简介：本书剖析了 50 个典型的程序员面试题，从基础知识、代码质量、解题思路、优化效率和综合能力五个方面系统整理了影响面试的 5 个要点。全书分为 7 章，主要包括面试的流程，讨论面试流程中每一环节需要注意的问题；面试需要的基础知识，从编程语言、数据结构及算法三方面总结了程序员面试的知识点；

欢迎投稿：

投稿邮箱：jsj@phei.com.cn
editor@broadview.com.cn

读者信箱：market@broadview.com.cn

电话：010-51260888

更多信息请关注：

博文视点官方网站：
http://www.broadview.com.cn

博文视点官方微博：
http://t.sina.com.cn/broadviewbj

九载耕耘 奠定专业地位

以书为证 彰显卓越品质

博文视点诚邀精锐作者加盟

《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、管辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

九年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巅。

英雄帖

江湖风云起,代有才人出。

IT界群雄并起,逐鹿中原。

博文视点诚邀天下技术英豪加入,

指点江山,激扬文字

传播信息技术,分享IT心得

● 专业的作者服务 ●

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

善待作者——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯,并会参照作者实际的工作、生活节奏,量身制定写作计划,确保合作顺利进行。

提升作者——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



联系我们

博文视点官网: <http://www.broadview.com.cn>

新浪官方微博: <http://weibo.com/broadviewbj>

投稿电话: 010-51260888 88254368

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

腾讯官方微博: <http://t.qq.com/bowenshidian>

投稿邮箱: jsj@phei.com.cn

关于本书用纸的温馨提示

亲爱的读者朋友: 您所拿到的这本书使用的是**环保轻型纸!**

环保轻型纸在制造过程中添加化学漂白剂较少,颜色更接近于自然状态,具有纸质轻柔、光反射率低、保护读者视力等优点,其成本略高于胶版纸。为您带来更好的阅读体验并与读者共同支持环保,我们在没有提高图书定价的前提下,使用这种纸张。愿我们共同分享纸质图书的阅读乐趣!

第 1 章 案例的选择与评估	1
1.1 案例的初步选择	1
1.2 案例的定量评估	3
1.3 其他案例	5
1.4 基本约定	6
1.5 关于类图的约定	8
第 2 章 Qt 概述	11
2.1 Qt 版权	13
2.2 Qt 库的编译	14
2.3 开发环境的设置	16
2.4 主控台的输入与输出	18
2.5 Qt 风格的编程规范	19
2.6 与 Qt 及 C++ 相关的文献资源	21
第 3 章 类模板特化	24
3.1 类模板特化	24
3.2 Traits 技术	27
3.3 类型分类 (Type Classification) 技术	28
3.4 降低代码膨胀	30
第 4 章 标准库及 Qt 对字符串的处理	32
4.1 字符及其编码	32
4.2 标准库的类模板 <code>basic_string</code>	34
4.3 Qt 的类 <code>QString</code>	37
第 5 章 国际化与区域文化	41
5.1 区域文化	41
5.2 <code>facet</code>	44
5.3 类 <code>locale</code> 的实现	50
5.4 类模板 <code>facet</code> 的实现	52

5.5 派生新的 facet 类	53
第 6 章 C++的 iostream	56
6.1 C 语言的 scanf/printf 函数组	56
6.2 iostream 的总体结构	57
6.3 字符特征的描述	61
6.4 模板特化后的总体结构	64
6.5 文件流	65
6.6 字符串流	71
6.7 流缓冲区	73
6.8 二进制文件的处理	76
6.9 用户自定义类型的输入和输出	77
第 7 章 Qt 的流	79
7.1 文件系统及底层文件操作	80
7.2 类 QTextStream	81
7.3 类 QDataStream	83
7.4 类 QLocale	86
7.5 iostream 和 Qt 流类的比较	87
第 8 章 隐式共享与 d-pointer 技术	88
8.1 隐式共享	88
8.2 d-pointer 在隐式共享中的应用	90
8.3 二进制代码兼容	92
8.4 d-pointer 模式的实现	96
8.5 QObject 中的 d-pointer	98
第 9 章 Qt 容器与迭代器	101
9.1 QTL 概述	101
9.2 QTL 容器和 QDataStream 的无缝连接	107
9.3 类型分类技术在 QList 中的应用	109
9.4 函子的应用——相关词词典	112
第 10 章 多线程与可重入	115
10.1 创建一个线程	116
10.2 线程间同步	116

10.3	线程安全与可重入	121
10.4	多线程环境下的 singleton 模式	122
第 11 章	信号与槽 (Signals and Slots)	134
11.1	对象树 (QObject Tree)	134
11.2	信号与槽机制	135
11.3	信号与槽的应用例子	140
第 12 章	Graphics/View 框架	145
12.1	Qt 图形系统介绍	145
12.2	Graphics/View 框架	146
12.3	例子——相撞的老鼠	147
第 13 章	Model/View 框架	152
13.1	Model/View 框架总体架构	154
13.2	模型 (Models)	157
13.3	视图 (Views)	179
13.4	选择操作	180
13.5	委托 (Delegates)	185
13.6	代理模型 (Proxy Models)	188
13.7	便利视图类	199
第 14 章	Qt 中的命令模式	206
14.1	Qt 的 Undo Framework	206
14.2	使用 Undo Framework 的一个例子	211
第 15 章	Qt 中的抽象工厂模式	215
15.1	抽象工厂模式简介	215
15.2	QTextCodec 及其子类的定义	217
15.3	界面风格	220
第 16 章	Qt 中的观察者模式	235
16.1	事件处理机制	236
16.2	事件滤波器	237
16.3	一个简单的例子——图像浏览器	239
16.4	一个有趣的例子——鼠标手势	240

第 17 章 Qt 的元对象系统	252
17.1 C++ RTTI (Run-time Type Information)	252
17.2 Qt 的元对象系统	258
第 18 章 智能指针	263
18.1 QPointer	263
18.2 QSharedPointer	275
参考文献	282
索引	284

第 1 章 案例的选择与评估	1
第 2 章 Qt 概述	11
代码段 2-1, 使用 Qt 进行主控台输入与输出, 取自 z:\examples\qt_console\main.cpp	19
第 3 章 类模板特化	24
代码段 3-1, 类模板 Stack, 引自 z:\examples\template_specialization	24
代码段 3-2, 完全特化的类模板 Stack, 引自 examples\template_specialization	25
代码段 3-3, 部分特化的类模板 Stack, 引自 examples\template_specialization	25
代码段 3-4, 使用 traits 技术封装 float 及 double 类型的特征, 取自 z:\examples\float_traits\main.cpp	27
代码段 3-5, 类型分类技术, 取自 z:\examples\type_classification\main.cpp	29
代码段 3-6, 应用类模板特化降低代码膨胀, 摘自 examples\reduce_code_bloat	30
代码段 3-7, 应用类模板特化降低代码膨胀, 摘自 z:\examples\reduce_code_bloat	31
第 4 章 标准库及 Qt 对字符串的处理	32
代码段 4-1, 类模板 basic_string 的构造函数	35
代码段 4-2, 类模板 basic_string 常用构造函数的使用, 取自 z:\examples\basic_string_demo\main.cpp	36
代码段 4-3, basic_string 对象和字符串的比较	36
代码段 4-4, 字符串的不同存放方式, 摘自 z:\examples\qstring_demo\main.cpp	39
代码段 4-5, QString 的字符编码转换功能, 摘自 z:\examples\qstring_merit\main.cpp	40
第 5 章 国际化与区域文化	41
代码段 5-1, 类模板 time_get 的使用方法, 取自 z:\examples\locale_time_get\main.cpp	47
代码段 5-2, 类模板 time_put 的成员函数 put 的用法, 取自 z:\examples\locale_time_ put\main.cpp	48
代码段 5-3, 类模板 codecvt 的成员函数 in 的功能, 取自 z:\examples\locale_codecvt\ main.cpp	50
代码段 5-4, 类 locale 以及 facet 的实现框架, 取自 VS 2010 安装目录 crt\src\xlocale	51
代码段 5-5, 类 locale::id 的作用	53
代码段 5-6, 创建新的 facet 子类, 取自 z:\examples\locale_unit\main.cpp	54

第 6 章 C++的 iostream	56
代码段 6-1, 应用类模板 char_traits 实现大小写不敏感的字符串类 ci_string, 取自 z:\examples\ci_string\ci_string\main.cpp	63
代码段 6-2, 对文件流进行读取以及写入操作, 取自 z:\examples\ fstream_demo\main.cpp	68
代码段 6-3, 具有多种格式设置的流, 取自 Z:\examples\share_streambuf	74
代码段 6-4, 流缓冲区的复制, z:\examples\copy_streambuf\main.cpp	75
代码段 6-5, 二进制文件的读取, 摘自 z:\examples\process_binary\main.cpp	77
代码段 6-6, 直接操作二进制文件对应的流缓冲区, 摘自 z:\examples\ process_binary\main.cpp	77
代码段 6-7, 用户自定义类型的输入和输出, 摘自 z:\examples\ overloaded_io\main.cpp	78
第 7 章 Qt 的流	79
代码段 7-1, 使用 QFileInfo 以及 QDir 获取各驱动器下的子目录名, 取自 z:\examples\QDir_demo\main.cpp	81
代码段 7-2, 使用 QFile 操作一个文件, 取自 z:\examples\QFile_read\main.cpp	81
代码段 7-3, 使用 QTextCodec 转换编码方案, 摘自 z:\examples\ QTextStream_demo\main.cpp	82
代码段 7-4, 使用 QDataStream 输出二进制数据, 摘自 z:\examples\ QDataStream_usage\main.cpp	84
代码段 7-5, 使用 QDataStream 读取二进制数据, 摘自 z:\examples\ QDataStream_usage\main.cpp	84
代码段 7-6, 重载运算符以使 QDataStream 支持新的数据类型, 取自 z:\examples\QDataStream_demo\main.cpp	85
代码段 7-7, QLocale 的使用, 摘自 z:\examples\qlocale_set\main.cpp	86
第 8 章 隐式共享与 d-pointer 技术	88
代码段 8-1, 采用隐式共享技术的 QString::toCaseFolded(), 取自 src\ corelib\tools\qstring.cpp	89
代码段 8-2, QString 的复制构造函数, 摘自 src\corelib\tools\qstring.h	90
代码段 8-3, 类 Matrix 的传统定义方式, 摘自 z:\examples\d_pointer\matrix\main.cpp	90
代码段 8-4, 应用 d-pointer 模式的类 Matrix, 摘自 z:\examples\d_pointer\matrix_with_d_pointer\main.cpp	91
代码段 8-5, Qt 4.5 版本中类 QLocale 的定义, 摘自 S:\corelib\tools\qlocale.h	93
代码段 8-6, Qt 4.5 版本中类 QLocalePrivate 的定义, 摘自 S:\corelib\tools\qlocale_p.h	93