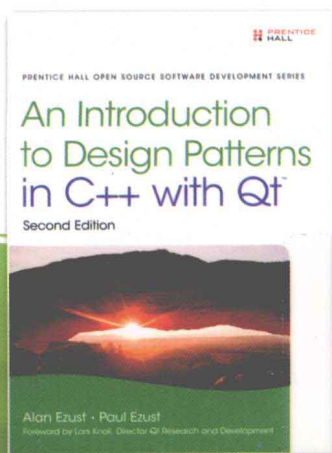


PEARSON

C++ Qt

设计模式(第二版)

An Introduction to Design Patterns in C++ with Qt
Second Edition



[美] Alan Ezust 著
Paul Ezust

闫锋欣 张学敏 张君施 等译
张德保 审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

C++ Qt 设计模式

(第二版)

An Introduction to Design Patterns in C++ with Qt
Second Edition

Alan Ezust
Paul Ezust 著

闫锋欣 张学敏 张君施 等译
张德保 审

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是美国萨福克大学已使用十余年的经典教程,利用跨平台开源软件开发框架 Qt 阐释了 C++ 和设计模式中的主要思想。全书共分四个部分:第一部分介绍 C++、UML、Qt、模型-视图、SQL、XML、设计模式等基础知识,目的是为零基础的 C++ 初学者铺垫一条学习面向对象编程的快捷之路;第二部分讲解内存访问、继承等重要的 C++ 特性,是前一部分的延伸和拓展;第三部分使用 Phonon 编写了一个多媒体播放器,展示了主要技术理念的应用方法;附录部分给出了 C++ 保留关键字、Debian 和 Qt 程序开发环境的配置等内容。每节的练习题和各章后面的复习题,既可作为课堂上的讨论题,也可进一步启发读者对于关键知识点的思考。

本书可作为软件开发人员学习 Qt 开发技术的参考书,也可作为从事 Qt 软件开发的研究人员和科技工作者的工具书。

Authorized translation from the English language edition, entitled *An Introduction to Design Pattern in C++ with Qt*, Second Edition, 9780132826457 by Alan Ezust, Paul Ezust, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2012 Alan and Paul Ezust.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2012.

本书简体中文版由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2012-1498

图书在版编目(CIP)数据

C++ Qt 设计模式:第2版/(美)艾朱斯特(Ezust, A.), (美)艾朱斯特(Ezust, P.)著;闫锋欣等译.

北京:电子工业出版社,2012.7

书名原文:An Introduction to Design Patterns in C++ with Qt

ISBN 978-7-121-16890-1

I. ①C... II. ①艾... ②艾... ③闫... III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 080514 号

策划编辑:许菊芳

责任编辑:许菊芳

印刷:北京京师印务有限公司

装订:

出版发行: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:31.75 字数:813 千字

印次:2012 年 7 月第 1 次印刷

定价:78.00 元

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 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

C++在1989年被标准化之前,就已经被广泛使用了很多年,这使得C++比当今流行的其他编程语言要相对成熟一些。C++是一种能够用于创建快速、高效且任务关键的系统的重要语言。C++同时也是最为灵活的编程语言之一,能够给开发人员提供多种编程风格,其中囊括了从高级的GUI代码到低级的设备驱动程序。

20世纪90年代的最初几年,C++是最流行和使用最为广泛的面向对象(Object-Oriented, OO)编程语言。许多计算机科学(Computer Science, CS)专业的学生都是借助C++来学习面向对象编程的。这是因为C++允许C程序员相对容易地转换到面向对象编程(Object-Oriented Programming, OOP),而在此之前,许多CS教授则一直在讲授C语言编程。

从1996年左右开始,Java超越C++成为学生们开启学习的一种面向对象语言。Java之所以会如此流行,是有许多原因的。

- 它比C++语言更简单。
- 有内建的垃圾回收机制,因此程序员无须关注令人恼火的内存释放工作。
- 开发工具箱中包含了一个标准的GUI类集合。
- 内建的String类支持Unicode。
- 支持多线程。
- 创建和“插入”Java档案(Java Archive, JAR)要比重新编译和链接库容易得多。
- 许多Web服务器提供Java API,能够容易地集成。
- Java程序是平台独立的(Wintel、Solaris、MacOS、Linux、*nix,等等)。

通过将C++和Qt结合使用,也可以获得上述许多优点。

- Qt提供了一组更容易理解的GUI类,而且与Java的Swing类相比,其运行更快,看起来更舒服,使用起来也更加灵活。
- 信号和槽的使用要比Java中的(Action/Event/Key)Listener接口更容易。
- Qt拥有插件体系结构,这使得可以将代码加载到一个应用程序中而无须重新编译或者链接。
- Qt提供foreach机制,可以对集合进行迭代,其读写操作更为简单。

尽管Qt没有提供垃圾回收机制,但它提供的各种替代方法可用来避免在代码中直接进行堆对象的删除操作。

- 容器(参见6.8节)支持引用计数和写时复制。
- 父对象和子对象(参见8.2节)。
- QPointer、QSharedPointer和QWeakReference(参见19.11节)。
- 对象子类化(参见2.14节)。
- 栈对象(参见20.3节)。

用 Qt 开发 C++ 程序，基本能够像 Java 一样易用、易懂和方便（在某些时候甚至要超过）。Qt 在速度和效率方面则要明显优于 Java，这使得编写从处理密集型的服务器程序到高速图形处理密集型游戏程序都成为可能。

使用 Qt 来学习 C++ 的另外一个好处是，Qt 在开源项目中已经得到了广泛应用，因此可以学习大量有价值的开源代码，对其进行复用，甚至可以修改和改良这些代码。

如何使用本书

第一部分包含课程的主要内容：C++ 概述、UML、Qt 核心、QObject、窗件、模型-视图、SQL、XML 以及设计模式。这一部分的目的是为了尽量避免出现不断翻看前面内容的情况，编排时按照内容的简单程度和详细程度进行了排序，因此，即使是 C++ 初学者也不会在一开始就感到内容多到无法接受的程度。

出于完整性和用于参考的目的，第二部分涵盖了一些在第一部分所介绍内容的深层次扩展内容，它们虽然要“枯燥”一些，却是非常重要的 C++ 特性。当然，在阅读到这一部分时，会发现理解这些难点其实也很容易。

每一章的结尾都给出了一些练习题和复习题。在本书的 Web 站点上提供了其中大部分编程练习题的答案。对于那些复习题中的问题，如果答案在本章中无法找到，一般会指出其答案所在的位置。如果采用本书作为教程，那么这些问题可以由学生或老师在课堂上或者考试中进行提问。

关于格式的说明

本书中的每一个编程示例都是从实际工作的源代码中抽取的，其 Web 版本会提供每段代码是从哪个完整源文件中抽取的超链接。这可以让读者方便地试验这些例子。在 Web 版本中的文本和列表中出现的 Qt 库中的类名称，还提供了到其类文档页的超链接。

第二版前言

很难知道一本书什么时候才算“完成”，尤其当撰写的是有关一个还在持续发展中的项目(Qt)时。因此，像Qt一样，本书也处于从其有基本设想开始到现在的不断变化过程中。

自本书第一版出版后，奇趣科技公司(Trolltech)发布了Qt 4.2，这家公司后来被诺基亚公司收购，随后发布了Qt 4.7.3，而本书作者之一Alan也成为了一名诺基亚认证的Qt专家(Nokia Certified Qt Specialist)。

第二版中对每一章都做了改进。先后有许多人审阅过本书，而且我们也重写了不少的例子和注释内容，新增了一些练习题和复习题，以及一些讲义和解决方案。

本书的第一部分重新进行了编排，比以前更早地引入了Qt。我们认为，充分地利用Qt的静态简便函数写出一些基于对话框的简单应用程序，要比全面介绍各个类、函数等更好一些。文件流的引入也比以前要早，以便可以呈现一些更为有趣的例子和作业。

为了使用自第一版出版之后又新增加到Qt中的那些类，窗件、主窗口、模型和视图、反射、线程和数据库，这些章节都进行了重写。讲解模型和视图的那一章已经进行了修订，现在给出了委托、QXmlStreamReader、QStandardItemModel、QFileSystemModel、QColumnView、排序和过滤、剪切和粘贴以及拖动和放下的用法。

关于多线程的章节全部重写了，以强调使用QtConcurrent算法而不直接用自己的代码来管理QThread的好处。

删除了动态窗体一章，取而代之的是给出如何手工地以及借助QtCreator/设计师来设计窗体的做法，给出了利用设计师做的窗体和用户代码相集成的各种技术。

媒体播放器的例子和练习都用Phonon进行了重写^①。

FileVisitor示例，最初是在Qt 3时代开发的，时至今日已失去了作用，现在有了QDir-Iterator。关于迭代的这一节是新写的，但不再会有一节专门用于很少用到的访问者模式，相反，将重点放在了迭代器模式。

为了发挥最新Qt组件的功能，对放入库中的一些类进行了简化或者改造。给出了一些新的智能和共享指针以及它们的用法。

书中讲解了更多的设计模式。对于每一种模式，或者提供一种实现方式(也许是作为练习)，或者在恰当的地方指出了实现或使用该模式的Qt类。

本书从各种不同的来源(包括诺基亚的官方Qt培训材料)借用了一些技巧提示和较佳的实践练习。这一版中的所有示例都已用Qt 4.7进行过测试。

① 如果打算使用这部分材料进行工作，要确保已安装libphonon-dev安装包。

致 谢

感谢在下列开源项目中的许多作者和贡献人员，感谢他们制作了这些自由的工具，感谢他们回答问题，还要感谢他们撰写了漂亮的文档。书中照搬了这些程序。

- jEdit^①
- Umbrello^②
- Firefox^③
- Doxygen^④
- dia^⑤
- imagemagick^⑥
- graphviz^⑦
- KDE^⑧, amarok^⑨, taglib^⑩
- docbook^⑪
- docbook-xsl^⑫
- xsltproc, xmllint, gnu xml libs^⑬
- subversion^⑭, git^⑮
- MoinMoin^⑯
- Bugzilla^⑰
- Apache httpd^⑱, ant^⑲, fop^⑳

① 参见<http://jedit.sourceforge.net>。

② 参见<http://uml.sourceforge.net/index.php>。

③ 参见<http://www.mozilla.org/products/firefox/>。

④ 参见<http://www.stack.nl/~dimitri/doxygen/>。

⑤ 参见<http://www.gnome.org/projects/dia/>。

⑥ 参见<http://www.imagemagick.org/script/index.php>。

⑦ 参见<http://www.research.att.com/sw/tools/graphviz/>。

⑧ 参见<http://www.kde.org/>。

⑨ 参见<http://amarok.kde.org/>。

⑩ 参见<http://developer.kde.org/~wheeler/taglib.html>。

⑪ 参见<http://www.docbook.org/>。

⑫ 参见<http://docbook.sourceforge.net/projects/xsl/>。

⑬ 参见<http://xmlsoft.org/XSLT/xsltproc2.html>。

⑭ 参见<http://subversion.tigris.org/>。

⑮ 参见<http://git-scm.com/>。

⑯ 参见<http://moinmoin.wikiwikiweb.de/>。

⑰ 参见<http://www.bugzilla.org/>。

⑱ 参见<http://httpd.apache.org/>。

⑲ 参见<http://ant.apache.org/>。

⑳ 参见<http://xmlgraphics.apache.org/fop/>。

- pidgin^①
- Python^②
- ReportLab PyRXP^③
- QtCreator^④
- mysql^⑤
- GNU Emacs^⑥
- Linux^⑦, gcc^⑧, gdb^⑨
- valgrind^⑩

感谢 Norman Walsh 和 Bob Stayton 开发了一个超级出版工具系统并撰写了它的文档。感谢 docbook 社区中给予帮助和指导的所有人员。

感谢@debian.org 的志愿者们，他们让测试工作始终保持最新并使其成为足够稳定、高效的开发平台。感谢 irc.freenode.net 带来了许多很好的想法。

感谢 Emily Ezust 提供了诸多文字加工技巧并让我们能够把 Qt 放在工作的第一位。感谢那些在文字上提供了输入和极有价值反馈信息的审稿人：Johan Thelin、Stephen Dewhurst、Hal Fulton、David Boddie、Andy Shaw 和 Jasmin Blanchette。感谢 Matthias Ettrich 的先见之明和大力支持。感谢 Trolls@Nokia 撰写了优秀的文档，创作了 QtCreator，在邮件组上回答了大量问题，并将 Qt 移植到了移动设备平台。

感谢 Prentice Hall 出版社的编辑和制作人员，感谢他们一丝不苟地阅读了本书并帮我们找出全书中分散的各个错误。

感谢 Chris Craig、Chris Gaal 以及 ics.com 的其他良师益友们提供的反馈和技术支持。感谢 ics.com 打造了一支可以让人乐于合作的 Qt 专家团队。

最后，还要感谢萨福克大学，它是整个项目的巨大源动力。也要感谢那些自 2003 年秋天以来在 CMPSC 331/608 中使用本书各个版本的那些学生们和许多大学中的同事们，他们以各式各样的形式在他们的班级中使用了我们的书，并为我们提供了一系列宝贵的 bug 报告和反馈。

写作缘由

在萨福克大学，我们抵制潮流并继续使用 C++ 讲授面向对象编程 (OOP)。许多年以来，我们在 CS1-CS2 (计算机专业) 的课程体系中使用标准的一学期制 OOP/C++ 课程作为必修课程，用数据结构和算法课程作为核心课程。我们的 CS1-CS2 课程体系已基于 C 编程语言多年，目前使用的是 Java。在面向对象编程课程中，学生们开发并掌握 C++ 语言课程、标准模板库 (Standard Template Library, STL) 的实质内容并理解面向对象中诸如封装、重构和工具开

① 参见<http://www.pidgin.im/>。

② 参见<http://www.python.org>。

③ 参见<http://www.reportlab.org/pyrxp.html>。

④ 参见<http://qt.nokia.com/products/developer-tools/>。

⑤ 参见<http://www.mysql.com/>。

⑥ 参见<http://www.gnu.org/software/emacs/emacs.html>。

⑦ 参见<http://www.kernel.org/>。

⑧ 参见<http://gcc.gnu.org/>。

⑨ 参见<http://www.gnu.org/software/gdb/gdb.html>。

⑩ 参见<http://valgrind.org/>。

发这样的关键概念。然而，我们发现 STL 是一个经常让学生们感到无所适从并让他们在低层次的编程结构和模板问题上花费过多时间的库。此外，所用到的命名约定也不是我们所喜欢和鼓励的编程风格类型。此外，STL 也没有提供用来编写带 GUI 的应用程序的基本内容，所以必须使用另外一种框架。

2003 年夏，当时的 Qt 版本还是 3.2，我们决定写一本书，期望它能够从更高层面上学习 OOP/C++。我们希望使用跨平台的 Qt 框架来提供大量 GUI 编程的内容，并引进和使用一些重要的设计模式。当 Qt 4.0 发布后，我们做了大量的重写，以适应 Qt 所引入的代码更新和改良。

本书首先是作为一本要在大学课堂中使用的教科书，但我们以一种可扩展的方式来完成这件事，并为具有不同背景的广大读者提供了大量的有用信息：已经有 C 语言或者另一种过程语言经验并希望了解面向对象和 GUI 编程的人们，没有 C 语言背景但熟悉 Basic、Java、Python 或另一种编程语言且希望了解 C++ 的人们。本书第一部分的目的在于让所有用户熟悉基本的 C++ 元素、面向对象的概念、UML 以及核心的 Qt 类。

我们坚信，只有将想法付诸实践才能对其有最好的理解，对设计模式来说尤其如此。许多 Qt 类和代码示例，都是在文献[Gamma95]中描述的较为流行的设计模式的具体实现。对于每个所讨论的设计模式，书中都给出了该示例的源代码，并包含了一些练习，以激发读者去复用、提炼和扩展那些代码。

库的复用不仅需要理解库，还需要理解模块化的软件、链接器以及库的设计。书中体现了我们的实践(作者的和学生的)，并从许多在线讨论社区中提取了大量有实际应用价值的经验。我们发现，这些经验帮助学生们解决了他们在课堂上使用本书前一版时遇到的大部分问题。

在萨福克大学 2003-2004 学年到 2010-2011 学年的每学期中，OOP/C++ 课程都用本书作为教材，获得了越来越可喜的成果，并从学生那里获得了非常宝贵的意见。在这门课程的早期(2003 年之前)，学生们通常需要为编程项目编写数千行的代码。相比之下，随着现在对代码复用的重视和健壮工具库的探索，学生的编程项目只需较少的代码。在我们看来，这个过程更为有趣，并能提供更多、更有价值的学习体验。

有许多已出版的 C++ 书籍或者是讲授 C++ 的，或者是讲授 Qt 的，但我们发现，C++ 书籍会使用各种不同的编程风格，并且它们强调的一些主题在使用 Qt 时并不经常用到。我们已经看到的所有 Qt 书籍，都会假定读者已有了一定的 C++ 知识。相比之下，这本书会认为读者没有任何 C 或 C++ 的编程经验，而且它的例子和作业中涵盖了使用 Qt 类时需尽早了解的那些 C++ 语言特性。本书可以用作讲授 C++ 和设计模式的教材，其重点放在代码设计和复用上。

据我们所知，尚没有其他大学级 C++ 教科书中会包含 Qt 例子并提供复习题、练习，以及为教师提供教学讲义。一些代码示例、实验讲义以及其他有用的资料都可以从我们的 dist 目录下载(该 URL 可以在本书最后给出的“参考文献”中找到)。

关于作者

Alan Ezust 在蒙特利尔的麦吉尔大学 (McGill University) 获得计算机科学硕士专业学位。目前，他是 ICS 公司 (www.ics.com) 的一名诺基亚认证 Qt 专家。ICS 是一家专注于为 Qt 软件提供培训和专业服务的公司。Alan 曾在 Learnix 任教并从事课件开发，随后在 Objectivity 公司工作，在那里他讲授并撰写了 Python、UNIX、C、C++、Java、Perl、设计模式、Froglogic Squish 以及面向对象数据库方面的课程。

Paul Ezust (剑桥，马萨诸塞州) 在萨福克大学数学与计算机科学系主持工作已有 30 多年，他根据美国计算机学会 (Association for Computing Machinery, ACM) 的基本准则引领着计算机科学课程的发展。他从事过大量的外部咨询、合同制编程以及计算数学的研究。

本书原本只是对 Paul 制定和完善了 8 年的课程的探索，如今已经演变成一个完全转型的代表，也是一种完全不同的讲授面向对象编程的教学方法，这要感谢温和而又不懈施压的 Alan。自 2003 年以来，本书的各种演进版本以及中文版和开源版本，已经成功地应用于萨福克大学以及澳大利亚、中国、俄罗斯、南非、美国、欧盟各国以及中国台湾地区的大学中。这一版增加了一些新的主题并通过自第一版付梓以来在 Qt 中新增的框架和类对一些内容做了完善。

目 录

第一部分 设计模式与 Qt

第 1 章 C++简介.....2	2.12 转换.....63
1.1 C++概述.....2	2.13 const 成员函数.....65
1.2 C++简史.....2	2.14 子对象.....66
1.3 第一个 C++例子.....3	2.15 练习：类.....67
1.4 标准输入与输出.....5	2.16 复习题.....74
1.5 函数介绍.....7	第 3 章 Qt 简介.....78
1.6 qmake, 工程文件及 Makefile.....11	3.1 风格指南与命名约定.....79
1.7 获得在线帮助.....16	3.2 Qt 核心模块.....80
1.8 字符串.....16	3.3 Qt Creator, 用于 Qt 编程的集成 开发环境.....82
1.9 流.....18	3.4 练习：Qt 简介.....83
1.10 文件流.....20	3.5 复习题.....84
1.11 用于用户输入/输出的 Qt 对话框.....23	第 4 章 列表.....85
1.12 标志符, 类型和常量.....26	4.1 容器简介.....85
1.13 C++简单类型.....27	4.2 迭代器.....85
1.14 const 关键字.....36	4.3 关系.....90
1.15 指针与内存访问.....37	4.4 练习：关系.....91
1.16 引用变量.....41	4.5 复习题.....92
1.17 const*与*const.....42	第 5 章 函数.....94
1.18 复习题.....44	5.1 函数重载.....94
第 2 章 类与对象.....46	5.2 可选实参.....96
2.1 struct 简介.....46	5.3 运算符重载.....98
2.2 类定义.....47	5.4 按值传递参数.....101
2.3 成员访问限定符.....49	5.5 按引用传递参数.....103
2.4 封装.....51	5.6 const 引用.....105
2.5 UML 介绍.....51	5.7 函数返回值.....106
2.6 类的友元.....52	5.8 从函数返回引用.....106
2.7 构造函数.....53	5.9 对 const 重载.....107
2.8 析构函数.....55	5.10 inline 函数.....109
2.9 static 关键字.....56	5.11 带变长实参表的函数.....112
2.10 类的声明和定义.....59	
2.11 复制构造函数与赋值运算符.....60	

5.12 练习：加密.....	113	9.6 窗件的布局.....	205
5.13 复习题.....	115	9.7 设计师和代码的集成.....	210
第 6 章 继承与多态.....	116	9.8 练习：输入窗体.....	215
6.1 简单派生.....	116	9.9 事件循环：重访.....	216
6.2 具有多态性的派生.....	121	9.10 绘制事件和画图.....	222
6.3 抽象基类的派生.....	127	9.11 复习题.....	224
6.4 继承设计.....	130	第 10 章 主窗口和动作.....	225
6.5 重载，隐藏与重写.....	132	10.1 QAction, QMenu 和 QMenuBar.....	225
6.6 构造函数，析构函数与复制赋值 运算符.....	133	10.2 区域和 QDockWidget.....	232
6.7 处理命令行实参.....	137	10.3 QSettings：保存和恢复应用 程序的状态.....	234
6.8 容器.....	141	10.4 剪贴板和数据传输操作.....	236
6.9 托管容器，组合与聚合.....	142	10.5 命令模式.....	237
6.10 指针容器.....	145	10.6 tr() 和国际化.....	243
6.11 复习题.....	159	10.7 练习：主窗口和动作.....	244
第 7 章 库与设计模式.....	163	10.8 复习题.....	244
7.1 建立并复用库.....	164	第 11 章 范型和容器.....	246
7.2 练习：安装库.....	169	11.1 范型与模板.....	246
7.3 框架与组件.....	171	11.2 范型，算法和运算符.....	250
7.4 设计模式.....	172	11.3 有序映射示例.....	252
7.5 复习题.....	178	11.4 函数指针和仿函数.....	255
第 8 章 QObject, QApplication, 信号和槽.....	179	11.5 享元模式：隐式共享类.....	257
8.1 值和对象.....	181	11.6 练习：范型.....	260
8.2 组合模式：父对象和子对象.....	182	11.7 复习题.....	261
8.3 QApplication 和事件循环.....	187	第 12 章 元对象，属性和反射编程.....	262
8.4 Q_OBJECT 和 moc 一览表.....	188	12.1 QMetaObject——元对象模式.....	262
8.5 信号和槽.....	189	12.2 类型识别和 qobject_cast.....	263
8.6 QObject 的生命周期.....	190	12.3 Q_PROPERTY 宏——描述 QObject 的属性.....	265
8.7 QTestLib.....	191	12.4 QVariant 类：属性访问.....	267
8.8 练习：QObject, QApplication, 信号和槽.....	194	12.5 动态属性.....	270
8.9 复习题.....	194	12.6 元类型，声明和注册.....	273
第 9 章 窗件和设计师.....	195	12.7 invokeMethod().....	275
9.1 窗件的分类.....	195	12.8 练习：反射.....	275
9.2 设计师简介.....	197	12.9 复习题.....	276
9.3 对话框.....	199	第 13 章 模型和视图.....	277
9.4 窗体的布局.....	201	13.1 模型-视图-控制器(MVC).....	277
9.5 图标，图像和资源.....	202	13.2 Qt 模型和视图.....	278
		13.3 表格模型.....	287

13.4 树模型	295	第 16 章 更多的设计模式	335
13.5 智能指针	298	16.1 创建模式	335
13.6 练习：模型和视图	300	16.2 备忘录模式	342
13.7 复习题	301	16.3 Façade 模式	347
第 14 章 验证和正则表达式	302	16.4 复习题	352
14.1 输入掩码	302	第 17 章 并发	353
14.2 验证器	304	17.1 QProcess 和进程控制	353
14.3 正则表达式	306	17.2 QThread 和 QtConcurrent	363
14.4 正则表达式验证	313	17.3 练习：QThread 和 QtConcurrent	374
14.5 子类化 QValidator	314	17.4 复习题	375
14.6 练习：验证和正则表达式	316	第 18 章 数据库编程	376
14.7 复习题	317	18.1 QSqlDatabase：从 Qt 连接 SQL	377
第 15 章 XML 解析	318	18.2 查询和结果集	381
15.1 Qt XML 解析器	320	18.3 数据库模型	382
15.2 SAX 解析	321	18.4 复习题	383
15.3 XML，树结构和 DOM	325		
15.4 XML 流	332		
15.5 复习题	334		

第二部分 C++语言规范

第 19 章 类型与表达式	386	第 20 章 作用域与存储类	416
19.1 运算符	386	20.1 声明与定义	416
19.2 语句与控制结构	389	20.2 标志符的作用域	417
19.3 逻辑表达式的求值	394	20.3 存储类	423
19.4 枚举	395	20.4 命名空间	426
19.5 有符号整型类型与无符号 整型类型	396	20.5 复习题	430
19.6 标准表达式转换	398	第 21 章 内存访问	431
19.7 显式转换	400	21.1 指针误用	431
19.8 用 ANSI C++ 类型转换进行 更安全的类型转换	401	21.2 带有堆内存的更多指针误用	433
19.9 重载特殊的运算符	405	21.3 内存访问小结	435
19.10 运行时类型识别	410	21.4 数组简介	435
19.11 成员选择运算符	412	21.5 指针的算术运算	436
19.12 练习：类型与表达式	413	21.6 数组，函数与返回值	437
19.13 复习题	415	21.7 不同类型的数组	439
		21.8 有效的指针操作	439
		21.9 数组与内存	441

21.10 练习：内存访问	441	22.3 多重继承	448
21.11 复习题	442	22.4 public, protected 和 private 派生	453
第 22 章 继承详解	443	22.5 复习题	454
22.1 虚指针和虚表	443		
22.2 多态与虚析构函数	445		

第三部分 编 程 作 业

第 23 章 MP3 自动点唱机作业	456	23.4 源选择器	459
23.1 Phonon/MultiMediaKit 配置	457	23.5 各播放列表数据库	460
23.2 播放列表	457	23.6 星号评分	460
23.3 多种类型的播放列表	458	23.7 排序，过滤和播放列表编辑	460
附录 A C++的保留关键字	461	附录 D Alan 的 Debian 程序员快速 指南	480
附录 B 标准头文件	462	附录 E C++/Qt 配置	485
附录 C 开发工具	463	参考文献	491

第一部分 设计模式与 Qt

- 第 1 章 C++简介
- 第 2 章 类与对象
- 第 3 章 Qt 简介
- 第 4 章 列表
- 第 5 章 函数
- 第 6 章 继承与多态
- 第 7 章 库与设计模式
- 第 8 章 QObject, QApplication, 信号和槽
- 第 9 章 窗件和设计师
- 第 10 章 主窗口和动作
- 第 11 章 范型和容器
- 第 12 章 元对象, 属性和反射编程
- 第 13 章 模型和视图
- 第 14 章 验证和正则表达式
- 第 15 章 XML 解析
- 第 16 章 更多的设计模式
- 第 17 章 并发
- 第 18 章 数据库编程

第1章 C++简介

本章介绍 C++ 编程语言。将给出一些基本的概念，如关键字、常量、标志符、声明、基本类型以及类型转换。还将给出 C++ 的历史、演变过程以及它与 C 语言的关系。也会介绍几个标准库和 Qt 类。

1.1 C++概述

C++最初是在 C 中添加了一系列的预处理器宏，作为 C 的扩展而编写的，它被称为“带类的 C”^①。经过多年的演变和优化，C++在 C 的基础上添加了许多高级特性，比如强类型化、数据抽象、引用、运算符重载、函数重载以及对面向对象编程的大量支持。

C++保留了使 C 语言流行和成功的主要特性：速度、效率以及广泛的表达能力，这种表达能力使得程序员能够在从最低层（例如直接的操作系统调用和位操作）到最高层（例如操作包含大而复杂的对象的容器）的多个层次上进行编程。

C++设计之初的基本原则是：添加到 C++中的任何功能，都不应导致不使用此功能的 C 语言代码的运行时开销^②。C++中存在许多高级特性，它们使程序员能够编写出可阅读的、可复用的、面向对象的程序，而使用这些特性不应导致编译器做额外的大量工作。不过，为了维持程序的功能和代码的可维护性，付出一些小代价（稍长的编译时间）还是值得的。有些特性存在运行时开销，但是被 C++编译器编译的 C 程序，应该与使用 C 编译器编译时运行得一样快。

1.2 C++简史

C++由 Bjarne Stroustrup 在 AT&T 公司 Bell 实验室工作时所设计，最终由 Bell 实验室打包并负责其市场化工作。1981 年，AT&T 公司内部开始出现最初的 C++ 版本，其后 C++ 根据用户的反馈逐步演化发展。

1986 年初，Stroustrup 撰写的图书 *The C++ Programming Language* 第一版发行。随着 1989 年 C++ 2.0 的发布，C++ 迅速成为一种严谨、实用的编程语言。同年，人们开始致力于制定 C++ 的国际标准。1997 年，美国国家标准化学会 (American National Standards Institute, ANSI) 的一个委员会完成并在内部公布了一个 C++ 语言的草案标准，名称为 *Draft Standard The C++ Language*, X3J16/97-14882, Information Technology Council (NSITC), Washington, DC.。

1998 年 6 月，参加过历时 9 年的 ANSI/ISO (International Standards Organization, 国际标准化组织) 工作的来自 20 个主要国家的代表一致接受了该草案标准。Stroustrup 撰写的 *The C++ Programming Language* 第三版于 1997 年出版，该书被公认为是权威的 C++ 参考手册。

对标准的后续完善是由 ISO 及国际电工委员会 (International Electrotechnical Commission,

① 参见 http://www.research.att.com/~bs/bs_faq.html#invention。

② 遗憾的是，异常处理打破了这一原则，如果启用会导致一些开销。这就是为什么许多库都不使用异常的原因。

IEC)负责的, IEC 是一家对电工学各个领域进行标准制订规范评估的国际机构。2005 年发布的 Technical Report 1(也称为“TR1”), 对 C++语言和标准库进行了大量扩充。2010 年, 负责 C++的国际标准工作组被命名为 ISO/IEC JTC1/SC22/WG21。C++草案标准 2010 版^①可从网络自由获得。C++0x 是“C++的下一个版本”的非官方名称, 有望于 2011 年定型^②。

1.3 第一个 C++例子

在整本书中, 都将通过代码例子来解释并演示重要的编程概念以及面向对象编程(Object Oriented Program, OOP)的思想。每一个例子的目标都是用最小的例子来扼要而有效地讲解概念和技术。示例 1.1 展示了 C++语法中的一些基本要素。

示例 1.1 src/early-examples/example0/fac.cpp

```
/* Computes and prints n! for a given n.
   Uses several basic elements of C++. */

#include <iostream>                                     1
int main() {                                           2
    using namespace std;                               3
    // Declarations of variables
    int factArg = 0 ;                                  4
    int fact(1) ;                                       5
    do {                                               6
        cout << "Factorial of: ";                     7
        cin >> factArg;                                8
        if ( factArg < 0 ) {
            cout << "No negative values, please!" << endl;
        }                                             9
    } while (factArg < 0) ;                             10
    int i = 2;
    while ( i <= factArg ) {                            11
        fact = fact * i;
        i = i + 1;
    }                                                  12
    cout << "The Factorial of " << factArg << " is: " << fact << endl;
    return 0;                                          13
}                                                      14
```

- 1 标准 C++库。在较老的 C++版本中, 可能会发现<iostream.h>。但是那个版本已经过时, 不提倡使用。
- 2 main 函数的开始, 它返回一个 int 值。
- 3 允许使用符号 cin, cout 和 endl, 而不必在每个符号前面加上“std::”。
- 4 C 语言风格的初始化语法。
- 5 C++语言风格的初始化语法。
- 6 do...while 循环的开始。

① 参见 <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2010/n3225.pdf>。

② 国际标准化组织 ISO/IEC 于 2011 年 8 月发布了 C++11 (先前被称作 C++0x) 编程语言标准。——译者注