<<More Exceptional C++>>

图书基本信息

书名：<<More Exceptional C++（中文版）>>

13位ISBN编号：9787121125928

10位ISBN编号：7121125927

出版时间：2011-1

出版时间：电子工业出版社

作者：（美）舒特　著，於春景　译

页数：331

译者：於春景

前言

与大师面对面（译序）　　小提琴家穆特录制过一张唱片，收录的是贝多芬最伟大的两部小提琴作品。
聆听那张唱片，你仿佛听见贝多芬在向你倾诉他对音乐艺术的理解和感悟，为你解答小提琴音乐创作的每一个疑问和困惑。
为了向音乐爱好者推荐贝多芬的那两部名作，穆特为她的那张名碟加上了一个恰如其分的标题——“与贝多芬面对面”。
合上More Exceptional C++的瞬间，我的第一反应是为这本名著也加上一个与之类似的副标题，然后，郑重地推荐给你。
是的，本书奉献给你的是又一位大师苦心孤诣的结晶。
不同的是，这位大师来自你所关注的C++程序设计领域。
对每一位C++爱好者或C++专业程序员来说，Herb Sutter的名字应该不会让人感到陌生。
作为ISO/ANSI C++标准委员会的委员，Herb Sutter不仅是C++程序设计领域公认的专家，还是深受程序员喜爱的技术讲师和作家。
Herb Sutter在互联网上主持的Guru of the Week专栏广受欢迎，几乎成为每一位C++程序员的网上必读物。
本书就是Guru of the Week的最新精华总结。
本书的主要特点可以反映在我为它所加的副标题中。
作为身经百战的专业程序员，而且长期从事程序设计的普及教育工作，Herb Sutter最清楚程序员在提升技术功力的过程中的所想所为。
他既了解初学者的疑问和困惑，也对专业程序员日常工作中遭遇到的陷阱和易犯的错误了如指掌。
因而，在本书中，Herb Sutter采用了独具匠心的“提问/解答”的方式来指导你学习C++的语言特性；在本书的每个专题中，Herb Sutter都合理地设想出你的疑问和困惑，又有如神助地猜到了你的（可能是错误的）解答，然后给你以指点并呈现出最佳方案，最后，还提炼出解决类似问题的一般性原则。
读了这本书，你仿佛和Herb Sutter这位大师面对面地进行了一场对话，亲历了他对你的指导和点拨。
关于本书的另一个特点，我想指出的是，作为C++标准委员会的委员，Herb Sutter在本书中强调了C++语言的最新标准和最新特性，强调了泛型程序设计和标准库的使用。
在本书的所有示例中，Herb Sutter为我们提供的是清新的C++编程风格和纯正的现代C++代码和范例。
本书适合的读者对象是中高级程序员，但这并不是说本书的内容高不可攀。
作者并不是在讲述高深的语言特性和设计技巧，更没有对基础话题避而不谈；相反，有关基础知识的介绍、回顾和深化散见于本书的很多条款之中。
只不过，这些条款的内容涉及的只是C++特性中的细节，它没有对C++的每一个特性、或每个特性中的每一个方面做全面的介绍。
尽管如此，只要具备基本的C++功底和一定的程序设计经验，你完全可以理解和消化本书的所有内容。
由于作者进行了巧妙的组织和精心的选材，本书每一个条款的内容都独立而完整，它可以让你在深入学习C++语言特性时事半功倍。
因而，我相信，无论是有一定基础的C++爱好者，还是身经百战的专业程序员，本书都将为你带来收获；在你的C++程序设计生涯中，它将是你的案头不可或缺的C++专著之一。
致谢　　能够翻译完这本书，我首先要感谢我的妻子妞妞和儿子斗斗，是你们给了我工作的动力，长期以来我都未能陪伴在你们身边，你们不但给了我宽容，还依然给我不变的爱和支持。
感谢yeka，是你促成了我和华中科技大学出版社的愉快合作，并给了我最直接的帮助。
感谢肖翔对译稿进行的认真审校，他在审阅过程中的真知灼见令我受益匪浅。
感谢moonsea，即使是在我工作的时候，你也会不时地扰乱我的心绪，迫使我无法继续工作—— 正因为这样，我才得以不时地逃离枯燥的键盘和屏幕，偷得一刻闲暇。
感谢作者Herb Sutter，在翻译本书的过程中，你对我的每一次请教都给予了耐心的解答—— 和大师的

直接对话让我如沐春风。
最后，感谢互联网上所有给予我帮助和鼓励的朋友们，lostmouse（我在专业论坛上的网名）希望，这本译作能成为我对你们最好的回馈；我还希望，你们能一如既往地帮助我，指出我在这本译作中留下的每一个疏漏和错误。
感谢你们！
於春景（lostmouse） 2002年5月于深圳蛇口 序 怎样才能成为专家？
在我涉足过的所有领域，答案都一样： 1. 掌握基础知识。
2. 将相同的内容再学习一遍，但这一次，请将你的注意力集中在细节上——这些细节的重要性，你头一次可能并没有认识到。
如果挑选了合适的细节来学习，并且彻底掌握了它们，进而达到不再为之困惑的程度，你就离成为专家为期不远了。
然而，除非已经成为了专家，你又怎么知道该挑选哪些细节来学习呢？
如果有人已经为你挑选了合适的细节，你就会学得更快，并且乐在其中。
举个例子，我曾经参加过一次摄影学习班，授课的是一位很不错的摄影师，名叫Fred Picker。
他告诉我们，摄影中仅有的两个困难环节是：照相机该摆放在哪儿、何时该按快门。
然后，他花了学习班的大部分时间教我们有关曝光、加工和冲印的技术细节——只有完全了解了这些细节，我们才能够很好地掌握摄影；而只有很好地掌握了摄影，我们去关注那两个“困难的”环节才有意义。
学习C++编程的细节，有一个特别引人入胜的方式，即：尽力去回答有关C++编程的问题。
例如： f（a++）；和f（a）；++a；的效果一样吗？
可以用迭代器去改变set的内容吗？
假设你正在使用一个名为v的vector，它占用的内存数量已经增长到让你 担心的程度。
于是你会想到去清除这个vector，将内存返还给系统。
调用v.clear（）可以完成这一使命吗？
你可能已经猜到，这些表面上看起来显而易见的问题，其答案一定是No——否则我也不会把它们提出来——但你知道答案为什么是No吗？
你确信？
本书回答了这些问题。
此外，它还回答了其他许多精心挑选的问题，这些问题针对的都是看起来很平常的程序。
没有多少书籍具有本书这样的特色——当然，它的前任Exceptional C++除外。
有许多自封“高级”的C++书籍，其实它们中的大多数要么只是针对专项主题进行了讨论——如果你只是想精通那些特定主题，而不是想深入研究日常编程中遇到的问题，那些书还不错——要么只是用“高级”这个词来引诱读者而已。
一旦透彻理解了这些问题和它们的答案，你在编程时就不必劳神于细节，你就尽可以将注意力集中在真正需要尽力解决的问题上。
Andrew Koenig 2001年6月 前 言 古希腊哲学家苏格拉底通过向学生提问进行教学——他用那些精心构思的问题来指导学生，启发他们从已知得出结论；让他们认识到自己正在学习的事物如何相互联系，这些事物与他们已有的知识又如何联系。
这种教学法如此出名，以致于我们今天把它专称为“苏格拉底问答法”。
如果让我们也从学生的角度来看问题，苏格拉底的教学法将引导我们，促使我们思考，帮助我们联系并应用现有的知识去获取新知。
本书如同它的前任Exceptional C++ [Sutter00]一样，借鉴了苏格拉底的教学法。
它假设你目前正身处C++产品软件开发的某个领域，采用“提问/解答”的形式指导你如何有效地使用标准C++语言及其标准库——尤其是，如何运用现代C++中有效的软件工程理论来解决问题。
这些问题大多直接取自于我和其他人在编写产品级C++代码时积累的经验，以所谓的“疑问”和“难题”的形式呈现给你。
“疑问”的目的在于帮助你从现有的知识和刚刚学到的知识中得出结论，并展现它们之间如何相互关

联。
“难题”则向你展示如何去分析C++设计和编程上的论题——某些是常见的论题，某些并不常见；某些是浅显的论题，某些则比较深奥；还有一些论题之所以拿来讨论，是因为——唔，仅仅因为——它们很有趣。
本书覆盖C++的方方面面。
但我没有说它触及C++的每一个细节——那将需要更多的篇幅——我是在说，本书提取了C++语言和标准库构件中的广泛素材，向你展示看似毫不相关的个体如何被综合利用起来，构成解决常见问题的新颖方案。
它还展示了这些素材中看似毫不相关的那些部分自身是如何相互关联的——即使有时你不希望它们有如此的关联；以及，应当如何处理这些关联。
你将在此找到关于模板与名字空间、异常与继承、健壮类的设计与设计模式、泛型程序设计与宏的使用技巧等内容——这些内容绝不是杂乱地堆砌在一起，而是作为连贯的条款，向你展示现代C++中这些组成部分之间的相互联系。
何为“More”？
More Exceptional C++起步于Exceptional C++驻足之处。
本书继承了前任的传统：它提供了新的内容，这些内容被组织为短小的条款，形成有主题的章节。
前一本书的读者会在此发现一些熟悉的章节和主题，如异常安全、泛型编程、内存管理技术等，但如今它们包含了新的内容。
这两本书在结构和主题而非内容上有重叠之处。
More Exceptional C++还有何不同之处？
本书特别强调了泛型编程技术以及如何有效地使用C++标准库，并涉及了如traits和predicates这样的重要技术。
有几个条款还深入分析了使用标准容器和算法时应该牢记的要点——这其中的许多要点，我在别的地方还没见到它们被提及过。
此外，一个新的章节和两个附录集中讨论了单线程和多线程环境下的优化议题——对于编写产品级代码的软件开发商来说，这些议题在目前比以往任何时候都更具实用价值。
本书的大多数条款最初出现于互联网和杂志专栏，尤其是Guru of the Week的GotW专题31到62，以及我曾为C/C++ Users Journal、Dr.Dobb’s Journal、以前的C++ Report和其他出版物撰写的印刷版专栏和文章中。
自最初版本出现以来，本书的内容历经大量的修订、增补、校正和更新，因而本书（连同它在www.gotw.ca上不可缺少的勘误表）可以被认为是那些原始材料的最新正式版本。
你应该知道的　　我认为你已经了解了C++的基础知识。
如果不是这样，建议你从一本好的C++入门和概念性的书籍开始，最好选择一本经典的大部头著作，如Bjarne Stroustrup的The C++ Programming Language [Stroustrup00]，或者是Stan Lippman与Jos é e Lajoie合著的C++ Primer第三版 [Lippman98]。
然后，一定要挑选一本指导编程风格的书，例如Scott Meyers经典的Effective C++套书[Meyers96][Meyers97]。
我发现这套书有基于浏览器阅读的CD版本[Meyers99]，十分方便好用。
如何阅读本书　　本书的每个条款以一个难题或疑问的形式呈现，它带有一条介绍性的标题，类似下面这样：　　条款#：条款的题目　难度：X　　条款的题目和难度等级提示你将要面对的是何种难题。
注意，难度等级是我的主观评断，我只是猜想大多数人碰到每个问题时会觉得有多难，所以你很可能会发现一个难度为“7”的问题对你来说比某个难度为“5”的问题还容易。
自从写作Exceptional C++以来，我不时收到一些电子邮件，说“某某条款比你说的要容易（难）”。
面对同一条款，不同的人认为“更容易”或“更难”是很自然的。
难度等级随人而定；对你来说，任何条款的实际难度真正取决于你的知识和经验，它对别人来说会相对更容易或更难。

但大多数情况下你会发现，难度分级是一种不错的经验方法，它指引给你的和你所认为的大致相当。
　　你可能打算从头至尾阅读整本书。
这很好，但不一定非如此不可。
你可能会集中阅读一个章节中的所有条款，因为你对那个章节的议题特别感兴趣。
这也不错。
书中有一些条款被我称为“短系列”，因为它们涉及的是相关的问题，你会看到这些条款用“之一”、“之二”等来标识。
除了这些“短系列”外，其余的条款都是很独立的。
在本书的条款中还包含很多交叉索引，有些索引还参考到Exceptional C++，你可以遵循这些索引随意跳转阅读。
我唯一要告诉你的是，制作“短系列”是为了让它们成为一组以便于连续阅读，除此之外，如何阅读，选择权在你。
　　名字空间、typename、URL引用以及其他约定　　我在本书中提出了不少建议，但我不会指引你去做一些连我自己都没做过的事。
这包括整本书里我在我自己的示例代码中所做的那些事。
我也会遵循程序设计的现有习惯和现代风格，即使有时候这样做不会对事情带来本质上的差异。
　　关于这一点，说说名字空间。
在本书的代码示例中，如果你在一个例子中看到了一个文件范围内的using指令，又在几页或几个条款后另外的例子中看到了一个函数范围内的using指令，这其实没有什么更深层的含义，只是说明在那些特定情况下，这样做是合理的，而且从美学的角度来看，也让我感到美观。
至于名字空间的基本知识，请参阅条款40。
在书写代码时，如果想强调我所指的是标准中的东西，我会用std：：来修饰标准库名称。
这一点确立后，我往往会转而使用不带修饰的名称。
　　再说说模板参数的声明。
我时常碰到一些人，他们认为写class而不写typename是过时的做法，即使这二者没有功能上的不同、而且C++标准本身也在到处使用class。
纯粹出于代码书写风格方面的考虑，并且为了强调本书所讨论的是当今现代的C++，在声明模板参数时，我也已经转到使用typename而不使用class。
唯一的例外是条款33中的一处，在那儿我直接引用了标准中的代码——　标准用的是class，我就随它去了。
　　除非我明确地称某段代码是“完整程序”，否则一般不是。
请记住，这些示例通常只是代码片段或者只是程序的一部分，它们不会就这样孤立地通过编译。
为了用我所提供的程序片段构成完整程序，你一般还得做一些显而易见的添加工作。
　　最后，说说URL。
互联网上，事事在变。
特别是，你无法掌控的那些事物在变。
这样一来，在印刷书籍上随意发布URL就成为了一种真正的痛苦：不用等到一本书在你的书桌上躺上五年，在它还没送到印刷厂之前，那些URL可能就已经过时了。
本书中，当我引用其他人的文章或网站时，我是通过我自己的网站www.gotw.ca上的URL来实现的。
这个网站我可以自己控制，它包含的只不过是直接指向实际网页的重定向链接。
如果你发现印刷在本书中的某个链接不再可用，请发电子邮件告诉我，我会更新这个链接，让它指向新的网页位置（如果我能够重新找到这个网页），或者标示这个网页已经不再存在（如果我无法找到这个网页）。
无论哪种方式，本书的URL将保持最新——尽管在当今互联网世界里，印刷传媒是这样举步维艰。
唉！
　　致谢　　深切感谢丛书编辑Bjarne Stroustrup，还有Debbie Lafferty、Tyrrell Albaugh、Chanda Leary-Coutu、Charles Leddy、Curt Johnson，以及Addison-Wesley出版社的其他成员，感谢他们在这个

项目中的鼎力协助和坚持不懈。
很难想象还能找到比他们更棒的人一起共事，他们的热情和协作使这本书完全达到了我预想的目标。
另外值得感谢和称赞的是审阅过本书的许多专家。
对本书的许多内容，他们毫无保留地提出了深刻的见解和犀利的批评，而这些见解和批评是那样一针见血。
正是因为他们的努力，你手中的这本书比初稿更完整、更易于理解、更实用。
特别感谢（大致以我收到审阅意见的顺序）Scott Meyers、Jan Christiaan van Winkel、Steve DewHurst、Dennis Mancl、Jim Hyslop、Steve Clamage、Kevlin Henney、Andrew Koenig、Patrick McKillen，以及一些不知名的审阅者。
书中遗留的所有错误、疏忽和歧义都是因为我，而非他们。
最后，将所有的感谢献给我的家人和朋友——无论是在这本书的写作和出版期间，还是其他任何时候，他们都一直陪伴在我身边。
Herb Sutter　　2001年6月于多伦多

<<More Exceptional C++>>

内容概要

对C++程序员来说，ISO／ANSI C++标准的问世标志着一个新纪元的开始。
C++标准为程序设计提供了很多新的便利和可能，但要想在如此众多的信息中挖掘到其中的精髓，现实世界中的程序员缺少足够的时间。
本书针对一定的主题，为程序员提供了简明扼要的指导，从而将学习时间和疑惑减至最少。
本书内容包括泛型程序设计与C++标准库、优化与性能、异常安全议题及技术、继承与多态、内存及资源管理、自由函数与宏等。
　　本书内容全面丰富，论述翔实清晰，作者权威且经验丰富，是C++程序员的必备读物。

作者简介

Herb Sutter

Herb Sutter是公认的C++软件开发专家，他同时定期地在世界各地的会议上应邀座淡 作为130多篇技术文章的作者，Herb同时担任ISO／ANSI C++标准委员会的秘书、C／C++Users杂志的特约编辑和专栏作家。

以及C++Report的前主编一在主要的C++屠言的Internet新闻组comp，1ang，c++，moderated上，发表有Herb广为流传的“ C++Guru of the Week ”系列，该新闻组白1995年成訌以来，一直山Herb担任主持，联系作者，清垂洶

<<More Exceptional C++>>

书籍目录

版权说明