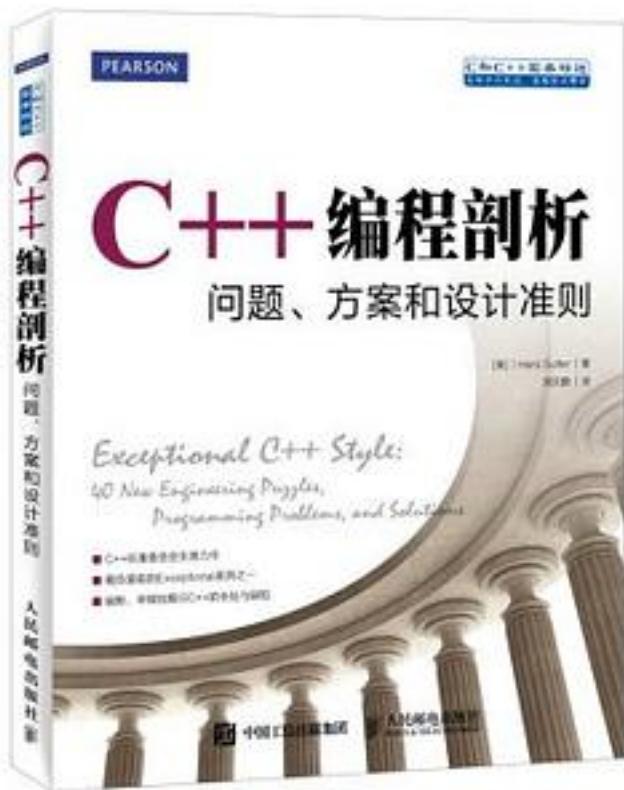


C++编程剖析



[C++编程剖析_下载链接1](#)

著者:[美]Herb Sutter

出版者:人民邮电出版社版

出版时间:2010-11-30

装帧:平装

isbn:9787115240996

软件“风格”所要讨论的主题是如何在开销与功能之间、优雅与可维护性之间、灵活性与过分灵活之间寻找完美的平衡点。在本书中，著名的C++大师Herb Sutter给出了40个编程问题，其目的是为了使读者不仅“知其然”，更要“知其所以然”，并帮助读者在软件开发中进行正确的决策。

本书是围绕实际问题及其解决方案展开论述的，对一些至关重要的C++细节和相互关系

提出了新的见解，为当今的关键C++编程技术（如泛型编程、STL、异常安全等）提供了新的策略。读者会在书中找到下列问题的答案。

可以从STL本身学习哪些库设计的知识？

如何避免削弱甚至损害模板代码的通用性？

为什么不应该对函数模板进行特化？正确的做法是什么？

异常安全如何超越try和catch语句？

什么情况下应当“泄漏”一个类的私有成分，怎么做？

如何让一个类不受版本变动的影响？

使用标准库容器的实际内存开销是多少？

如何使用const才能真正优化代码？

内联对程序性能有何影响？

为什么有些看似错误的代码实际上却能够通过编译且运行得很好？为什么我们要关心这种情况？

std::string的设计有什么问题？

本书使读者能够在设计、架构和编码的过程中保持良好的风格，从而编写出更健壮和更高效的C++软件。

作者介绍：

Herb Sutter ISO C++标准委员会主席，C/C++ Users Journal杂志特邀编辑和专栏作家，Microsoft Visual C++架构师，领导.NET环境下C++语言扩展的设计工作。除本书外，他还撰写了三本广受赞誉的图书：C++ Coding Standards、Exceptional C++和More Exceptional C++。

译者简介：

刘未鹏

毕业于南京大学软件学院，热爱C++，长期关注C++前沿技术，对C++有较深的研究。曾在《程序员》杂志上发表文章数篇，译有Imperfect C++（与荣耀合译）。个人blog为<http://blog.csdn.net/pongba>，邮箱pp_liu@msn.com。

目录: 泛型编程与C++标准库 1

第1条 vector的使用 2

第2条 字符串格式化的“动物庄园”之一：sprintf 9

第3条 字符串格式化的“动物庄园”之二：标准的（或极度优雅的）替代方案 14

第4条 标准库成员函数 23

第5条 泛型性的风味之一：基础 26

第6条 泛型性的风味之二：够“泛”了吗 30

第7条 为什么不特化函数模板 36
第8条 友元模板 42
第9条 导出限制之一：基础 51
第10条 导出限制之二：相互影响，可用性问题以及准则 58
异常安全问题及相关技术 67
第11条 try和catch 68
第12条 异常安全性：值得吗 72
第13条 对异常规格的实际考虑 75
类的设计、继承和多态 83
第14条 顺序，顺序 84
第15条 访问权限的使用 88
第16条（几乎）私有 93
第17条 封装 101
第18条 虚拟 110
第19条 对派生类施加规则 118
内存和资源管理 129
第20条 内存中的容器之一：内存管理的层次 130
第21条 内存中的容器之二：它到底有多大 133
第22条 进行new操作，也许会抛出异常之一：new的方方面面 140
第23条 进行new操作，也许会抛出异常之二：内存管理中的实际问题 148
优化和效率 155
第24条 常量优化 156
第25条 再论内联 161
第26条 数据格式和效率之一：什么时候压缩是真正重要的 168
第27条 数据格式和效率之二：（甚至更少的）位操纵 172
陷阱、缺陷和谜题 179
第28条 不是关键字的关键字（或者：另一种注释） 180
第29条 这是初始化吗 186
第30条 要么double要么彻底完蛋 191
第31条 狂乱的代码 194
第32条 小小的拼写错误？鬼画符似的语言以及其他奇形怪状的东西 199
第33条 操作符，无处不在的操作符 202
风格案例研究 207
第34条 索引表 208
第35条 泛型回调 218
第36条 构造式union 226
第37条 分解std::string之一：概观std::string 242
第38条 分解std::string之二：重构std::string 247
第39条 分解std::string之三：给std::string瘦身 255
第40条 分解std::string之四：再论std::string 259
参考文献 267
索引 271
· · · · · (收起)

[C++编程剖析](#) [下载链接1](#)

标签

C++

编程

计算机

C/C++

计算机科学

编程语言

软件工程

重点

评论

已经读了无数本C++书了，得出的结论是：想练C++，还是去盗google的base库吧 XD

类似像这样的书都需要边看边实践 有些地方实践领悟起来会需要一些时间
千万不可为了求快 不求甚解就这么过去了 ==

翻译得不错~

相比之下这本书里的内容有点边缘了（不少内容其实在其他书里也有涉及，只是这里相对更加细节化），而且有些items感觉就是Sutter用来凑数的...

Herb Sutter大神的书，豆瓣网页本书作者信息有误

终于做了个了解。
Sutter老大化身话痨，絮絮叨叨的细节，这本书不适合初学者，“C++从入门到放弃系列”

很深很细节。会讲一些比较冷僻的特性，然后来龙去脉优缺点说半天，最后说建议千万不要用。。。什么泛型模板资源管理异常处理，通通是我的软肋，硬着头皮看吧，看一点是一点。看到最后终于还是有种跪了的感觉。

工作当中很多关于设计的争论其实在语言演进的过程当中无数次发生过，所以要成为一个合格的软件开发，还是要归本溯源，探寻语言的设计初衷。没有最好的语言，只有最合适的。这句话同样适用于设计，读一本探寻开发语言本质的书，从来都不会是一桩亏本买卖！

跟泛型编程相关的章节都没看（因为看不懂）。不过没关系，反正在不久的将来我就会重新把它读一遍的。

const 原来对优化没用，纠正了我的误解。模板函数重载解析的条目加深了我的理解。尽量用非虚接口，虚函数尽量私有，析构要么公用虚要么保护非虚，讲解非常清楚。总的来说，我更喜欢exceptional系列这种风格，而不是effective系列的风格

[C++编程剖析 下载链接1](#)

书评

这本书不适合初学者，作者在序中已经说了，要先看看the c++ programing language 或者c++ primer 这本书 编程风格要先看看effective c++
书里面一共有40个问题，分成七个部分，每个问题分析的很透彻，最后还有几个准则。书的翻译质量非常好，译者是一个研究生...

本书的深度非常高，正如 Herb Sutter所说的：

我期望你已经掌握了C++基础知识，如果你还没有，可以从一本介绍性和概览性的C++好书开始学习。像Bjarne Stroustrup的《The C++ Programming Language》)或Stan Lippman和Josee Lajoie合著的《C++Primer》这样的经典著作都是不错的...

最后的这一版 Style 的觉得还行，比起Exceptional C++

没那么枯燥很多，不过总之看起来好像总是没那么有激情一样，我也还是觉得中规中矩~~~~~

看一遍还是值得的。如果要深入里面的某些东西，可以有更加好的专著，而且从这本书里面也得不到太多的东西了，所以最好的就是把书看完...

[C++编程剖析 下载链接1](#)