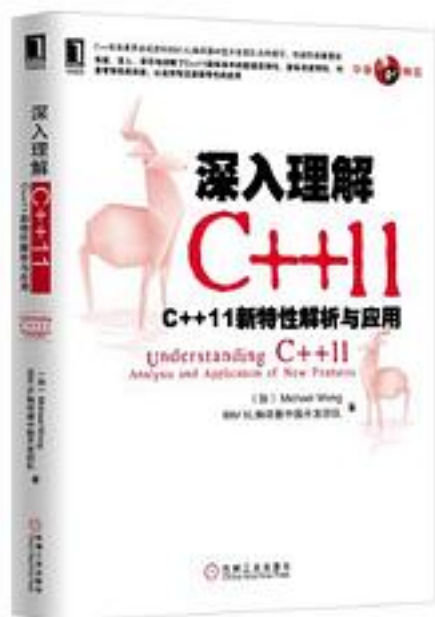


深入理解C++11



[深入理解C++11 下载链接1](#)

著者:Michael Wong

出版者:机械工业出版社

出版时间:2013-6

装帧:

isbn:9787111426608

《深入理解C++11:C++11新特性解析与应用》内容简介：国内首本全面深入解读C++11新标准的专著，由C++标准委员会代表和IBM XL编译器中国开发团队共同撰写。不仅详细阐述了C++11标准的设计原则，而且系统地讲解了C++11新标准中的所有新语言特性、新标准库特性、对原有特性的改进，以及如何应用所有这些新特性。

《深入理解C++11:C++11新特性解析与应用》一共8章：第1章从设计思维和应用范畴两个维度对C++11新标准中的所有特性进行了分类，呈现了C++11新特性的原貌；第2章讲解了在保证与C语言和旧版C++标准充分兼容的原则下增加的一些新特性；第3章讲解了

具有广泛可用性、能与其他已有的或者新增的特性结合起来使用的、具有普适性的一些新特性；第4章讲解了C++11新标准对原有一些语言特性的改进，这些特性不仅能让C++变得更强大，还能提升程序员编写代码的效率；第5章讲解了C++11在安全方面所做的改进，主要涵盖枚举类型安全和指针安全两个方面的内容；第6章讲解了为了进一步提升和挖掘C++程序性能和让C++能更好地适应各种新硬件的发展而设计的新特性，如多核、多线程、并行编程方面的新特性；第7章讲解了一些颠覆C++一贯设计思想的新特性，如lambda表达式等；第8章讲解了C++11为了解决C++编程中各种典型实际问题而做出的有效改进，如对Unicode的深入支持等。附录中则介绍了C++11标准与其他相关标准的兼容性和区别、C++11中弃用的特性、编译器对C++11的支持情况，以及学习C++11的相关资源。

作者介绍:

目录: 免责声明

序

前言

第1章 新标准的诞生

1.1 曙光：c++11标准的诞生

1.1.1 c++11/c++0x（以及c11/c1x）—新标准诞生

1.1.2 什么是c++11/c++0x

1.1.3 新c++语言的设计目标

1.2 今时今日的c++

1.2.1 c++的江湖地位

1.2.2 c++11语言变化的领域

1.3 c++11特性的分类

1.4 c++特性一览

1.4.1 稳定性与兼容性之间的抉择

1.4.2 更倾向于使用库而不是扩展语言来实现特性

1.4.3 更倾向于通用的而不是特殊的手段来实现特性

1.4.4 专家新手一概支持

1.4.5 增强类型的安全性

1.4.6 与硬件紧密合作

1.4.7 开发能够改变人们思维方式的特性

1.4.8 融入编程现实

1.5 本书的约定

1.5.1 关于一些术语的翻译

1.5.2 关于代码中的注释

1.5.3 关于本书中的代码示例与实验平台

第2章 保证稳定性和兼容性

2.1 保持与c99兼容

2.1.1 预定义宏

2.1.2 __func__ 预定义标识符

2.1.3 _pragma操作符

2.1.4 变长参数的宏定义以及__va_args__

2.1.5 宽窄字符串的连接

2.2 long long整型

2.3 扩展的整型

2.4 宏__cplusplus

2.5 静态断言

2.5.1 断言：运行时与预处理时

2.5.2 静态断言与static_assert

2.6 noexcept修饰符与noexcept操作符

- 2.7 快速初始化成员变量
- 2.8 非静态成员的sizeof
- 2.9 扩展的friend语法
- 2.10 final/override控制
- 2.11 模板函数的默认模板参数
- 2.12 外部模板
 - 2.12.1 为什么需要外部模板
 - 2.12.2 显式的实例化与外部模板的声明
- 2.13 局部和匿名类型作模板实参
- 2.14 本章小结

第3章 通用为本，专用为末

- 3.1 继承构造函数
- 3.2 委派构造函数
- 3.3 右值引用：移动语义和完美转发
 - 3.3.1 指针成员与拷贝构造
 - 3.3.2 移动语义
 - 3.3.3 左值、右值与右值引用
 - 3.3.4 std::move：强制转化为右值
 - 3.3.5 移动语义的一些其他问题
 - 3.3.6 完美转发
- 3.4 显式转换操作符
- 3.5 列表初始化
 - 3.5.1 初始化列表
 - 3.5.2 防止类型收窄
- 3.6 pod类型
- 3.7 非受限联合体
- 3.8 用户自定义字面量
- 3.9 内联名字空间
- 3.10 模板的别名
- 3.11 一般化的sfinae规则
- 3.12 本章小结

第4章 新手易学，老兵易用

- 4.1 右尖括号>的改进
- 4.2 auto类型推导
 - 4.2.1 静态类型、动态类型与类型推导
 - 4.2.2 auto的优势
 - 4.2.3 auto的使用细则
- 4.3 decltype
 - 4.3.1 typeid与decltype
 - 4.3.2 decltype的应用
 - 4.3.3 decltype推导四规则
 - 4.3.4 cv限制符的继承与冗余的符号
- 4.4 追踪返回类型
 - 4.4.1 追踪返回类型的引入
 - 4.4.2 使用追踪返回类型的函数
- 4.5 基于范围的for循环
- 4.6 本章小结

第5章 提高类型安全

- 5.1 强类型枚举
 - 5.1.1 枚举：分门别类与数值的名字
 - 5.1.2 有缺陷的枚举类型
 - 5.1.3 强类型枚举以及c++11对原有枚举类型的扩展
- 5.2 堆内存管理：智能指针与垃圾回收
 - 5.2.1 显式内存管理

- 5.2.2 c++11的智能指针
- 5.2.3 垃圾回收的分类
- 5.2.4 c++与垃圾回收
- 5.2.5 c++11与最小垃圾回收支持
- 5.2.6 垃圾回收的兼容性
- 5.3 本章小结
- 第6章 提高性能及操作硬件的能力
- 6.1 常量表达式
 - 6.1.1 运行时常量性与编译时常量性
 - 6.1.2 常量表达式函数
 - 6.1.3 常量表达式值
 - 6.1.4 常量表达式的其他应用
- 6.2 变长模板
 - 6.2.1 变长函数和变长的模板参数
 - 6.2.2 变长模板：模板参数包和函数参数包
 - 6.2.3 变长模板：进阶
- 6.3 原子类型与原子操作
 - 6.3.1 并行编程、多线程与c++11
 - 6.3.2 原子操作与c++11原子类型
 - 6.3.3 内存模型，顺序一致性与memory_order
- 6.4 线程局部存储
- 6.5 快速退出：quick_exit与at_quick_exit
- 6.6 本章小结
- 第7章 为改变思考方式而改变
- 7.1 指针空值—nullptr
 - 7.1.1 指针空值：从0到null，再到nullptr
 - 7.1.2 nullptr和nullptr_t
 - 7.1.3 一些关于nullptr规则的讨论
- 7.2 默认函数的控制
 - 7.2.1 类与默认函数
 - 7.2.2 “= default”与 “= deleted”
- 7.3 lambda函数
 - 7.3.1 lambda的一些历史
 - 7.3.2 c++11中的lambda函数
 - 7.3.3 lambda与仿函数
 - 7.3.4 lambda的基础使用
 - 7.3.5 关于lambda的一些问题及有趣的实验
 - 7.3.6 lambda与stl
 - 7.3.7 更多的一些关于lambda的讨论
- 7.4 本章小结
- 第8章 融入实际应用
- 8.1 对齐支持
 - 8.1.1 数据对齐
 - 8.1.2 c++11的alignof和alignas
- 8.2 通用属性
 - 8.2.1 语言扩展到通用属性
 - 8.2.2 c++11的通用属性
 - 8.2.3 预定义的通用属性
- 8.3 unicode支持
 - 8.3.1 字符集、编码和unicode
 - 8.3.2 c++11中的unicode支持
 - 8.3.3 关于unicode的库支持
- 8.4 原生字符串字面量
- 8.5 本章小结

- 附录a c++11对其他标准的不兼容项目
- 附录b 弃用的特性
- 附录c 编译器支持
- 附录d 相关资源
- • • • • [\(收起\)](#)

[深入理解C++11_下载链接1](#)

标签

- C++
- C++11
- C/C++
- 编程
- 计算机
- 编程语言
- 程序设计
- 新标准

评论

C++ Primer
第5版是从零讲C++，很厚很浪费时间，而这本是讲C++11新出来的特性，对于已经熟悉C++的人来说，很合适，能够快速了解C++11有哪些新特性，讲的也非常好，简洁易懂

给我感受最深的就是细致。用了很多代码描述特定的需求是从什么样的问题中产生的（比如在移动语义之前，连拷贝构造都要先介绍一遍），这样的需求在C++11之前需要怎样的“奇技淫巧”来解决，而在C++11中提供了怎样的标准语法来简单、漂亮的解决，即使资历较浅如我者，也能通过这样的方式理解C++11，也更深刻地理解了C++98以及编程本身。不愧是编译器的团队，对涉及编译器特性的地方，比如RVO、内存模型都有讲解（内存模型讲得很细）。虽然是团队作品，但保持了文字的规范（编辑错误还是略有）、叙述的耐心。非常用心，力荐！我略感遗憾的是对新标准库特性介绍得好像就很不完整了，这本书题材非常好，希望有C++14或C++17的升级版，把新标准库特性也系统介绍一下。

写的确实非常好，写编译器的人就是厉害啊。

前100页很不错...

擦，这都能卖书，还卖这么贵！

第100本，虽然不多，加油加油

找不到合适的书，算是不错了，支持一下！

内容不全，也并没有太多的深入。标题党

一般般，基本是标准的一些罗列和演示，直接看cppreference其实就差不多了，就当存个档。说深入理解也严重了。有些特性没实现，也不常用。C++11也算是C++最具革命性的一个改进了。

C++11太强大了，primer已经不够用了。解释还是不错的，但当我翻阅C++标准时候才发现很多都是直接摘抄标准文档。

介绍C++11特性。中规中矩吧。

IBM中国的程序员中文不好，或者我的中文不好

不错的书

参考有余，深入难当，不过学点lambda的语法糖足够

如果你想快速了解c++11有哪些特性，该书可以帮助你。c++11标准库添加的东西比较多，该书甚少涉及，需要通过额外的资料来了解

总体来说还不错，适合快速概览C++11新特性。

作者的名字里看到了本科同学。把几个特性都简单的过了一遍，但是讲得不深，比如智能指针部分只是给了几个样例程序。建议配合 Effective Modern C++ 阅读。

C++11的入门书，标准库基本没有涉及。

本书系统的讲解了C++11新增的功能，但不包括C++11新增库（例如thread）这部分内容。

算不上深入~~

书评

刚看了3分钟，就在书本正文第二页，发现了一个错误：ISO/IEC 14882，被写成了：IOS/IEC 15882 情何以堪呀，这作者是难道是做苹果 IOS 开发的，帮导师干活？幸亏是先下载 pdf 来看，也没亏本

正如老头说的11算是新的一门语言.大家都要了解.从vs2010起就断断续续了解0x,了解着新的特性,有高兴,也有感觉更多的坑.所以,作为一个不大可能写出个排山倒海库的半吊子c++程序猿，心情决定着我重点了解大的,对性能,工作有利的完美特性.新容器,右值引用,lambda,thread,并行. 对...

现在真是什么书都敢把书名叫做“深入理解XXX”。觉得这本书叫做《C++11 初探》之类的更合适，因为基本是在罗列C++11的局部特性，然后放上一些提案中或者Wikipedia中代码的例子，就凑成了一本书。并没有很多深入的探讨。不过作为第一次专门看C++11相关的书的我来说，还是学到...

6.3.3 explains memory orders which are newly supported by c++11. The explanation may be the most readable text about memory order in Chinese. Crystally clear! In my idea, c++11 memory model is as same as Java. If you are confused about the concept of mem...

C++11特性的基本介绍，对于std::move的介绍没我想象中的深度，不过Lambda函数还是学到以前不知道的知识，要看完全版的话看C++ Standard Library 11就够了。其他的只是简单的举例，离深入理解还有一段距离。

很好的一本书，写的通俗易懂。覆盖也比较全，适合阅读，而且不需要花很长的时间。看这本书，结合实际，会越看越快。很多技术知识点的背景也都有介绍，会从根源去介绍，读起来也就比较好理解，而且会把编程与性能结合起来，比如运行期改如何，编译器该如何，介绍的很通俗，易...

对C++11期待很久了。

本书选取大部分C++11中新增和修订（相对于C++98来说）的特性作为主题，并按照一定的相关性将这些特性划分成章节，然后逐个地对这些特性进行讲解，这对于有C++经验的人来讲，是一种非常好的组织方式和成书方法，这样，你只需要将本书中的内容吃透，基本上可...

书籍其实不错，对C++11的不少特性都有详细的描述（照书中说法应该还有下册，期待一下~），作者都是一线的编译器开发者，所以内容上还是有不少干货的，当然，说其鞭辟入里还不至于，但是用以扩充视野、查漏补缺绝对没有问题，推荐~~~更难得的是书籍作者基本都是国人，更应该好好...

[深入理解C++11 下载链接1](#)