

Inside the C++ Object Model



[Inside the C++ Object Model_下载链接1](#)

著者:[美] Stanley B · Lippman

出版者:Addison-Wesley Professional

出版时间:1996-5-13

装帧:Paperback

isbn:9780201834543

Inside the C++ Object Model focuses on the underlying mechanisms that support object-oriented programming within C++: constructor semantics, temporary generation, support for encapsulation, inheritance, and "the virtuals"--virtual functions

and virtual inheritance. This book shows how your understanding the underlying implementation models can help you code more efficiently and with greater confidence. Lippman dispels the misinformation and myths about the overhead and complexity associated with C++, while pointing out areas in which costs and trade offs, sometimes hidden, do exist. He then explains how the various implementation models arose, points out areas in which they are likely to evolve, and why they are what they are. He covers the semantic implications of the C++ object model and how that model affects your programs. Highlights

- *Explores the program behavior implicit in the C++ Object Model's support of object-oriented programming.
- *Explains the basic implementation of the object-oriented features and the trade offs implicit in those features.
- *Examines the impact on performance in terms of program transformation.

Provides abundant program examples, diagrams, and performance measurements to relate object-oriented concepts to the underlying object model. If you are a C++ programmer who desires a fuller understanding of what is going on "under the hood," then Inside the C++ Object Model is for you! Get a value-added service! Try out all the examples from this book at www.codesaw.com. CodeSaw is a free online learning tool that allows you to experiment with live code from your book right in your browser.

0201834545B11102003

作者介绍:

Stanley B.

Lippman的工作是提供关于C++和面向对象的训练、咨询、设计和指导。Lippman在成为一名独立咨询顾问之前，曾经是迪士尼动化公司的首席软件设计师。他在AT&T Bell实验室的时候，领导了cfront 3.0版本和2.1版本的编译器开发组。他也是Bjarne Stroustrup领导的Bell实验室Foundation项目的成员之一，负责C++程序设计环境中的对象模型部分。Lippman还撰写了许多关于C++的论文。目前Lippman已受雇于微软公司，负责Visual C++项目。

译者侯捷：电脑技术自由作家，对于技术的钻研和发表，有独特的品性与坚持。作品涵盖著、译、评三方面，散见于各种刊特、媒体、网站论坛。

目录:

[Inside the C++ Object Model_下载链接1](#)

标签

C++

C/C++

programming

经典

计算机

编程

编译实现

计算机科学

评论

看这本书的时候真是各种爽啊！Lippman对于底层的对象模型讲的很深刻。读完这本书后，你就可以知道为什么有时候cast是会要有代价的，为什么C++在一些情况下效率是和C一样的，virtual table是怎么实现的，还有RTTI，还有很多问题Lippman都帮你做了解答。最后不得不提译者侯捷老师，他为本书插入了很多自己画的示意图和译注，真是为本书锦上添花，让我们理解本书轻松不少。

看完后一头雾水，需要消化段时间

看完c++ primer和这本书，才有点入门的感觉，这本书看完之后，自己对C++的熟悉程度仿佛又下降了点。写C++程序，一方面是写给程序员看，另一方面，是写给编译器看，或许C++的复杂性就来源于这一点吧。

我又不搞编译，读这本书干什么=。=

很久读的了，C++内部的实现机理讲解的很透彻。

读起来非常累的一本书,当时吃饭路上睡觉,游泳都在想的一本书.需要大量的思考,和比较扎实的计算机基础才能搞明白.

in-depth C++

没看effective系列,从过去的Java OO和杂七杂八的FP思维方式,补了Using and Understanding C pointers,数据结构课上深化容器的概念和知识,并且补了《程序员的自我修养》里面的链接部分,过了Stroupstrup的the C++ Programming Language的第一部分Tour系列,总算可以看我现在最关心的内存实现模型了。过了这关应该就可以自己尝试设计程序而不是简单的依葫芦画瓢。

从另外一个角度即实现这的角度来考虑C++

对EH解释得不甚了了 类内部的数据结构 过程结构 倒是很详细
Lipmann的语言功力不错 他的书浅显易懂 读完对C++的理解深了一些

差不多十一个小时读完,又涨知识了。

對這本書有特殊感情啊，呵呵。
這個作者是早期C++的參與者，內容沒得說。這是這個topic唯一的著作，你沒得選。可惜裡面錯誤比較多，而且不是typo，是技術性的。讀的時候需要非常小心。

了解C++对象内存模型的唯一书籍！必读。

对深入理解C++有绝对的帮助。不过感觉略有点啰嗦

helpful

进阶

深入C++的好书

复习完毕, 跟作者说的一样, 当回复读者的小品文看完了. 难度比当初看的时候小了很多...

神书, 把C++讲的很好

看完还是有点混, 还是得结合实践

[Inside the C++ Object Model_ 下载链接1](#)

书评

如果你跟我一样是个智商在250整点上的青年, 如果你也觉得看看技术书, 尤其是web青年都不待见的C++相关的技术书非常流弊, 来试试吧, 这本书最合适你了, 看了之后也不用跟着写几行代码, 顶多一把一把的薅头发, 反正不费电! 以前看书, 书里总得告儿我别这样别那样, 要这样要那样, ...

理由有以下几条 一、书比较老了, 是在C++ 98标准出来之前写的
二、细节错误比较多, 侯捷的翻译导读里说他更正了不少, 可谁知道还有多少细节错误没发现; 而且, 我发现他的译文有些地方也有问题, 原文本来正确的, 却纠正成错的了
三、除非从事编译器方面的工作, 没必要死抠细...

其实完全可以用软件设计的视角来阅读这本书，在这本书中学到的不仅仅是语言，而是语言背后的东西——代码应该怎么写才会更有效率。软件设计中功能性的实现总是最低层次的，而软件背后的效率和设计的思想才更值得人去关注，linus反对用C++，因为用C完全能做到C++的效果，...

这本书写得很拗口，侯捷的翻译也有很多不符合习惯的用词，所以读起来颇为费力。
总结一下： 1. 虚函数的实现：
为每个带有虚函数的类，建立一个虚函数表，存放这个类的每个虚函数的地址。基类和派生类的虚函数表有着不同的内容。每个有虚函数的类的对象带...

4年前临近毕业的时候无意中从学校图书馆发现了它，从第一页开始就被深深吸引住了，如饥似渴，一连几天都铺在上面，虽然后面的内容大多没有看懂。后来工作之后的两年中，此书起码通读了不下3遍，每次都有不同的体会-----经典的书籍就是这样，随着时光的流逝，更加沉淀出它的芬芳...

第一代C++编译器开发主管所写。如果你想成为真正的C++高手，看这本书，他为你讲述了编译器在处理各种语法时在“后台”所做的事。对C++有较深入了解的读者会在读后有恍然大悟之感。侯杰翻译，质量相当不错，但内容太深，只适合对C++有较深了解的读者。

很好很强大，也非常深奥，比Effective系列深太多了。只有前5章勉强看懂。
这本书偏重的是C++内部实现，而effective则是讲C++的常用技术。虽然在技术上没有太大帮助，但绝对大大提高内功，会给你剖析了C++ class的实现机制。
建议想深入C++的人都读一读，会有一种醍醐灌顶的作用...

这本书买了好几年了。开始看的时候，感觉很深入艰涩，比较高难。这种感觉是因为初学之时注重表面上的“用法”。当渐渐发现这些“用法”非常复杂，难以掌握驾驭的时候，回头再看这本书，却又发现它讲的都是最简洁基本的内容。如果学习的过程中能不时从这本书里得到基本原理的启...

书里的最重要的部分还没有过时，对，我说的“最重要的部分”指的是对象模型。（其中提到的NRV技术目前的编译器基本都已经实现了，末尾的关于异常和模板的部分有充数的嫌疑）
书中关于数据成员的布局，非虚成员函数和虚函数的设计考量，在单继承，多继承/虚拟继承下都有详细的...

C++程序员晋级必读书，侯捷翻译的也相当出色，
正本书对C++的对象内存形式，继承的实现，多重继承，静态对象等做了详细描述。

如果我没记错的话，易筋经应该是一本内功秘籍；也就是说，它只会教你练内功，而不会教你练招式。看过武侠小说的都知道，同样的招式，内功强的人使出来是要完爆内功弱的。
我看完这本书后，对它的印象就大致如此。从本书中，你学不到什么编程技巧（招式），有的只是对c++这门语...

我是以日记的形式写下，写完发觉有些书评的性质，就发到此处。首先说《深度探索C++ 对象模型》是一本书，一本介绍C++
对象模型的书，这是一本独一无二的从编译器的解读来阐述C++
对象模型的书籍，因着编译器的设计者和学习者全球本来就少，加之深度研究...

刚看了几章，想说几句：侯捷先生的书似乎都跟Microsoft公司的产品或理念有关
这本书似乎也不例外 Stanley书里的例子一般都是以一些开源的软件来说明问题
比如比较效率一般用的是cc、NCC等编译器
而侯捷先生所举的例子大都是使用的Microsoft VC++ 开源的 外面是这样 里面是啥...

这本书非常适合用来装逼！人活着不为了装逼，那跟咸鱼又有什么区别！
整本书几乎没讲C++的任何语法，任何编程技巧，任何使用经验，说的内容就如标题所言：C++编译器说明书+virtual 关键字详解！
看这本书的时候，完全不需要写任何代码去验证。为什么？举个例子...

其实这本书的内容并不多，一些概念贯穿全文，只要理解了就能很顺畅地看完，一些性能测试的内容完全可以跳过，花两三天就能看完。上机写些简单的代码，切到汇编窗口单步调试，就能很直观地看到书里介绍的好多内容，对理解很有帮助。

个人是先看了本书的导读，然后对本书的2,3,4,5章节比较感兴趣，所以主要读了这几个章节。

通过对第二章的阅读，对C++里面的几个类型的构造函数有了一个更深入的了解，本书讲的很细，大有看了恍然大悟之感。

第三章主要是介绍了C++里面类中数据成员的内存布局，有丰...

做C++开发8, 9年了，虽然个人认为也积累了一些语言方面的开发经验，可是看到这本书以后，发现自己了解的内容还是太肤浅了，虽然这本书中介绍的内容并不能直接用于平时的项目开发，可是它能够让你对这门语言有一个更为深入的了解，同时也能够对面向对象的编程思想有了更为深入的...

看看书里的c++内存模型，虚表、多继承，难度不小。
由于c++复杂的特性，开发c++编译器的高人们八成是费了老力了，感谢一下他们。

[Inside the C++ Object Model 下载链接1](#)