

图书基本信息

书名：《C++编程思想》

13位ISBN编号：9787111091622

10位ISBN编号：7111091620

出版时间：2002-1-1

出版社：机械工业出版社

作者：埃克尔

页数：801

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《C++编程思想》

内容概要

《C++编程思想(英文版第2版)》是丛书中的一本，丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色：有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

海报：

书籍目录

Preface

1: Introduction to Objects

2: Making & Using Objects

3: The C in C++

4: Data Abstraction

5: Hiding the Implementation

6: Initialization & Cleanup

7: Function Overloading & Default

8: Constants

9: Inline Functions

10: Name Control

11: References & the Copy-Constructor

12: Operator

13: Dynamic Object Creation

14: Inheritance & Composition

15: Polymorphism & Virtual Functions

16: Introduction to Templates

A: Coding Style

B: Programming Guidelines

C: Recommended Reading

Index

编辑推荐

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。《C++编程思想》（英文版第2版）是书库中的一本，在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C.Berkley、C.M.U.等世界名牌大学采用。

精彩短评

- 1、Eckel的Thinking in C++ (机械工业)
- 2、经典。读了中文，又读了英文，可惜中文翻译太差。
- 3、C++圣经
- 4、这本书都说很值得一看,英语好点的可以看英语不太好的也可以把它当做阅读理解来看哦~~
- 5、可以用来学习英语
- 6、传说中的C++变成死相~~不知道这本书怎么翻译出来的~~直接怀疑用金山快译出来稍微改一下就卖了~真的想学C++可以去看它的英文版或者C++ primer的中文版
- 7、为数不多完整读过的C++原版书籍
- 8、@designer
老实说,此书中译版确是业内烂翻译的典范...
君不见各大网站对其恶评如潮?
甚至译者也在china-pub表示了确实仅仅是其老师署个名而已...
翻译的烂也罢了,至少弄个勘误的网把重大的错误纠正一下吧...
但是完全木有,而且貌似最近还弄了个合订本. 尚没接触过,不晓得有没改进.
技术书有个错误神马的着实难免,但是如本书这般经典却被翻译得晦涩难懂和纰漏百出也算奇葩了...
所以您说的“翻译的已经非常好了”;...
实在不敢苟同.
- 9、没仔细读过。。所以吃亏了。
- 10、第一次读的时候看的很慢，不知所云。这次重读，阅读速度极快，理解也透彻，技术上果然更上一层楼。
- 11、匆匆翻过
- 12、好书，毋庸置疑。
- 13、讲的比primer要深，但同时对基础的知识讲的也很清楚
- 14、几个月前，决定找一个C++的书好好学一下，基于对Bruce Eckle的盲目崇拜，和信息了解不充分，选择了Thinking in C++这书。

翻译太烂，读起来费劲得很，简直还不如去读英文原版。

所以让我对Bruce这本书本身的质量产生怀疑，真的像网站上鼓吹的那么好吗？

不能迷信于网站的宣传，很多都不靠谱，而且不一定会适合自己。

- 15、学C++的人必看书！
- 16、学校书摊淘来。大学第二本 C++ 书。
- 17、首先，此书成书较早，对很多新的特性没有涉及，希望作者快出第三版:-D

我是看完《Effective C++》之后开始读的《Thinking in C++》，发现本书的文字比前者要易懂得多，当然也许是因为《Effective C++》的规模限制，作者不能展开讲解的缘故:-)

实际上两本书各有所长。《Thinking in C++》带领读者领略C++语言的各个领域，边边角角都不放过，对于初学C++的人来说，能够解答很多疑惑，比如为什么需要拷贝构造函数，对象切片到底发生了什么，C++对象模型中的虚函数表VTABLE和VPTR的概念和原理等等。此书的代码简短有效，通常一个概念用几行代码就表述清楚了，很有力量。但此书的缺陷在于有些问题浅尝辄止，没有深入。比如说，为什么拷贝构造函数参数是引用，继承层次中的虚析构函数调用过程在编译器中如何实现等（这些是面试题常问的，所以我就比较关心:-)）

而《Effective C++》很明显是面向有多年编程经验的程序员的，每个item都有特定的应用背景，然后作者用犀利的语言指出该怎么做不该怎么做。对有些问题虽然也会讲到语言细节，但更多是偏向于你该如何如何做这样的建议。《Thinking in C++》结尾的Programming Guidelines貌似有好多是

《Effective C++》里提到的。

总体上看,《Thinking in C++》更倾向于语言级别的介绍,《Effective C++》和程序的结构设计联系更多,都是适合放在手边参考的好书。

18、如果你想大幅度的提高自己的编程技术,原版是你最好的选择!

19、C++经典之作

20、习惯到网上买书了,种类挺多,买起来方便,送的还真挺快,好像我一般二三天就能收到了,挺快的。这次也一样,刷书就到了。要是学习也能这么快,嘿嘿.....天下哪有那么容易的事儿呀~~功夫不负苦心人,终于找到了一个专门教电脑的猎豹网校,想学C++,看网校上老师选的是最经典的教程,试听了一下,效果挺好。要是早点找到这类网校,不早轻松多了?

21、如果是C++新手,或者,OO新手,Thinking系列第一卷绝对是一本好书.它给你解释了怎么从C到C++的转变,接着,它慢慢的给你介绍C++的各种东西,先从数据抽象入手,再到类,再到各种特性,如访问保护,命名空间,重载操作符,继承,多态,模板(初级介绍)等等以及为什么要这样.如果你对 these 解释不够满意,那请深入学习C++ 老爸写的<C++的设计与演化>.它上面的例子都是顺水指推舟的,很合理的出现在了各个章节当中,后面的章节会继续引用前面章节讲到的例子,以告诉你一个怎样的进化过程.

而Thinking系列的第二卷,也就是这本了,我觉得它没有第一卷写得好.第二卷浅尝辄止地讲了一些高级特性以及一些常用的编程概念如并发和设计模式.当然它不可能很深入的介绍每一个高级特性,毕竟光模板都可以讲好几本书了,更何况标准库.

最后, Thinking in c++ 吧, 还有 Reading in c++, Programming into c++ 去吧!

- 22、这是第三遍了, 还是有不少收获, 这次收获主要在C++多态内部实现机制上, 以前没关注
23、机械工业拿这么烂的东西出版吗?
译者拿这么烂的翻译出来, 是不是不吓死全国人民不罢休?

原作者Bruce看了非吐血身亡不可。。。

不是做开发的, 就表碰开发类的书, 理解得不透就表要写雷人害人的东西

- 24、经典中的经典, 其思维方式的科学性是中国人难以达到的。
25、作为计算机专业的学生, 极力推荐这本C++的提高读物。只要是有了C++的基础, 基本语法+基本数据结构能够熟练掌握, 读这本书能让自己对C++有更深感触, 锻炼了以它为基础的编程思维! 我的入门语言是Java, Eckel的Thinking in Java也是一本不可多得的好书!
26、Thinking in Java我实在是读不懂, thinking in CPP也是一样。。。哎。。。
27、个人认为强于primer
28、全英文的我喜欢 嘎嘎
29、现在看挺便宜的
30、都传说翻译差, 看的时候就特别注意语意的流畅度, 看了前面几章算是熟悉的内容, 果然是差...

倒是前言阅读尚可, 囧。

不过在国内这样的情况估计应该也可以理解, 见多了慵懒的导师分配个任务, 无聊的学生偷懒用个翻译软件完成个小作业罢了。

如若如此, 谨表示深深的鄙视。

- 31、看这个有时候就是寄存器思维了~~~~~
32、Primer入门, 此书进阶。看完此书告一段落, 内功不够, 即将炼出内伤!
33、大一教材来着。
34、本来想扣掉一颗星给翻译, 可是谁让我没耐心看原版呢.....还是妥妥的给五星
35、好想收藏的一本书
36、研一的时候, 看完c++ primer 之后看的, 现在忘得差不多了
37、课本
38、作者对C++的理解深入骨髓, 又是顶级的教师, 再加上他对本书的精益求精, 铸就了C++书籍的一个神话。本书适合对C++和面向对象都有所了解的人, 它不但用最合适的方式描述了C++是什么, 还讲清楚了为什么。但更值得我们学习的是作者在字里行间所透露的对编程、对设计、对面向对象思想的理解以及永无止境地追求更高境界的态度。

```
while (前两章不能很好地理解) {  
    if (第一次看) {  
        看全书;  
    } else {  
        sleep(一年); // 其它线程可不能睡, 得抓紧时间夯实基础  
    }  
}
```

- 39、E文原书读的头大,
还有, 中文第一版的翻译,,,, 实在不让人很满意
并且Bruce Eckel的书总带有那么一点点“禅意”
总体来说并不适合新手
买一本放在书橱里收藏还是不错滴
40、C++ 学习
41、好书, 不要看这本的中文版
42、2007.06 | COL | 不错, 不过没有读完

43、Volume 1

44、无论内容、封面、纸张，堪称一流

45、对于计算机编程人员，就算截止到今天，能够很好地使用面向对象思想的人，也绝对不多，一是因为这个思想属于设计层面；另一个因为大家工作的层次都是面对函数，面对系统的机会很少。

本书确实能让那些每天精通于函数的人们豁然开朗，发现原来，面向对象真的是可以利用的一种很好的方法呀。

希望你读完本书，能够开始逐渐走出面向过程的语言，进入虚函数的实际使用，进入对象的世界。

46、之所以给了推荐是面向那些对C有足够掌握或者已经接触C++的人。如果C程序不够玩转而且从没接触过CS的面向对象语言的专有名词，建议不要阅读此书。

47、匆匆看了一遍，这本书内容很丰富，也有翻译的原因，读起来不太顺畅，看来还是尽量看英文版的

Thinking in C++，它不是逐一去讲C++的教程，所谓Thinking主要在于对C++的理解，章节的分配有点专题的味道，比如继承与组合等

所以一开始学C++，看这本书可能不太合适

高级的话题如异常，模板，设计模式等将在第二卷中有详细说明

总体来说，翻译的不好是一大遗憾，但最主要还得看自己是否看得下去

48、好书，可惜读了没几页。。。。。。。。。。惭愧

49、可以说，是我第一次认真读的编程英文原著。非常经典，写的东西很容易看懂。

50、大部分还是蛮不错的，亮点在OOP上（这和作者对Smalltalk、Java的热衷相关吧）。不过大部分都已经过时了，不要说C++98 / C++03的标准没有更新，对template技术的梳理也是比较落后，所以不是很合适在C++14的时代当作标准读物了。

51、I read a lot of book about programming, this is definately one of those few I would recommand others to read!

52、本书涵盖了C++编程的所有理论知识，很好，值得一看！

53、当时热衷于C#，读此书只是对比着学习面向对象

54、现在学JAVA了，呵呵...还想要买他的thinkinginjava，但是书块头太大了，怕看不完，呵呵...

55、没有一页一页地看，第一卷大概翻了一遍，主要讲一些设计思想和实现方式。开卷有益。

56、Bruce Eckel写过很多经典的计算机书籍，尤其是在计算机语言方面，包括C++和Java，尤其是C++造诣很高，这本书是大学面向对象课程的教材，即使上了研究生，再读读这本书还是很有收获！

57、书籍说明

最经典的C++书籍之一

适合在入门之后，在对整个C++的思维方式进行梳理

大牛的经典作品，一定要读

当做进阶的C++书籍来读，一定会有收获的

阅读建议

第三本C++书籍！！

58、网友都说中文版翻译的不好，于是买了e版。。没有想象中的难，开始把几个术语弄懂，书中基本没有什么不认识的词，非常好懂。很多内容越看越想看。

59、从这个事例上还是可以说明学英语是很重要的，要是能深得原著精髓该多好。。

60、书是好书，但是翻译的语文是体育老师教的吧

61、粗览一遍，上学期码的代码简直弱爆了呢。。。

62、我内牛满面的英语

63、详细的将书看了一篇，加深了对C++的认识，但书的章节组织较凌乱，本人认为不适合初学者

64、虽然C++领域的经典书籍犹如过江之鲫、车载斗量，但其中的可供初学者入门的书籍却并不多。可能C++阵营里的牛人太多了，都不屑于写入门级新手教程。虽然Bjarne Stroustrup大叔说学习C++不需要学习C语言，但实践表明有C语言基础还是很有帮助的。另外大叔自己写的那本《The C++ Programming Language》初学者不要看！

好了，如果你已有C语言使用经验，想要继续学习C++，那么有两本书值得推荐：《Thinking in C++》和《C++ Primer》，前者浅显易懂，后者广博渊深，都是学习C++良师益友。

《Thinking in C++》分为两本：volume I（卷1）和volume II（卷2）。卷1的作者是著名的Bruce Eckel，卷2是他和人合著的。我感觉是卷2相比卷1语言上要晦涩不少，不如看《C++ Primer》和《The Standard Template Library》。

《Thinking in C++》卷1和《Thinking in Java》最大的特点就是结合面向对象的设计思想来讲解C++、Java程序设计，在阐述OOP方面做得非常好！其他C++书籍无出其右。

推荐直接看英文版。

65、很多公司面试经常问的一个题“C++类的构造函数可以是虚函数吗？为什么？”，如果你还不知道答案，那么你需要认真研读这本书了，而且不是跳跃式的阅读，因为这个问题在最后一部分，我相信前面部分也有很多你需要了解的内容。一流的作者讲为什么要这么实现，有什么设计上的考虑，二流的作者讲他是怎么实现的，也就是原理分析，三流的作者让你背书，总之重点背下来了考试没问题，四流的作者翻译手册，没有自己的理解就敢上市销售，最末一等，用词典翻译，连计算机体系内的kill都能翻译成杀死。本书我归为一流，书名是c++编程思想，但是不限于c++

开发人员阅读。因为了解编译器如何工作的，了解内存中的一些状态，对于触类旁通的分析类似PHP编译器同样有帮助，五星推荐给大家。

66、这本书带领我走入了C++和面向对象的大门，Bruce Eckel独特的思路可以让读者理清面向对象的脉络，抓住面向对象的核心，同时又能学习到C++语言本身最重要的那些知识。

让我印象最为深刻的就是第一章就高屋建瓴地俯瞰了面向对象的核心思想。从抽象的演进讲到对象以及对象的接口，再慢慢揭开面向对象的核心：封装（信息隐藏），继承，多态。结合一些容易理解的示例（以Shape和Circle，Square的关系来讲解的，至今它仍是我最喜欢引用的关于继承和多态的例子），同时避开语言的细节，Bruce Eckel把面向对象的思想展现在读者面前。最后，Bruce还提到了异常处理，分析设计以及极限编程，并以C++为什么成功这一主题作为第一章的结束

第二章延续了第一章的思路，以C++的视角讲解了如何创建和使用对象。如果第一章是面向对象理论的鸟瞰，那么第二章就让读者以实践来认识面向对象。

之后开始了正式的C++讲解过程：C in C++讲述了C++中的那个C子集，并提到了和纯C的不同之处。然后，在读者有了基础后，Bruce开始逐渐展开C++语言的核心：数据抽象，信息隐藏，构造和析构，函数重载等等，同时不忘将第一章里的思想融入其中，提醒读者以对象的方式思考问题。

C++中比较困难的主题：如拷贝构造函数，虚函数及其实现内幕，模板等等都有专门的章节讲解

总之，Thinking in C++将C++的核心讲解得很透彻，如果认真读了，绝对能成为C++领域的“入门级高手”。

本书还有第二卷，讨论了一些更加高深的C++主题：如异常处理，泛型编程等等，读完第一卷可以尝试一下。

67、C++学习经典之一

68、英文版很给力

69、内容很详实，但是翻译的可是真的差啊，简直就是直接用机器翻译的，真的够差劲！关于内容没得说，该讲的都有讲解，而且具有一定的深度，非常好！但就是翻译影响了阅读体验啊！一句话要读好几遍才能理解翻译者的意思！唉！

70、公正的说，这本书的中文翻译已经非常好了，当然，你也可以选择原著。

71、不好意思，错点了“有关键情节”……

72、这本书的基础语法讲的比较好，对于c++标准库讲的就比较差了。

73、和Effective C++，一起看的，里面对C++的一些特性的讲解还不错，以前一些不注意的东东看了之后有一些体会，可能还需要再看即便，网上找到一个china_pub版本的感觉挺好的

74、用时再读

75、还好。比较经典

76、送货速度很给力，包装完好，下来就是好好品味书籍了。

77、写得很不错，一章一章，C++中的典型问题基本上都涉及到了。讲得非常清楚，入门好书。

78、面向对象程序设计的教材。

开始觉得不怎么样。

陈奇老师的课讲的还是很赞的。

回头再看看。

79、如果不考虑翻译得狗屁不通的话，这本书还是很经典的。当然虽然翻译得很差，但毕竟作者原文写得很好，也算抵消了一些翻译的硬伤。。。从内容上说，基本无可挑剔，唯一的遗憾是没有大的例子，毕竟要学OO，光看些玩具模型是不够的。

80、c++的太多了，经典的书籍也不少，这个也算是经典之一吧，当时上课时用的教材，一开是上课没注意听，后来突击考试时仔细看了一遍，有点后悔自己没有好好读一下此书，原因是上课之前我已经接触过另外一本同样不错的c++教材，认为自己学的还不错，但后来才发现我只是学了语法，而没有掌握内在的机制，这本书对c++的机制阐述的很好，不过前几章我感觉有点罗嗦，但这也是国外教材的一大特色吧，总是怕读者读不懂。总之好书，值得常备。

81、C++必读书

82、据说不错

83、算是完整的读过一遍，作者对C++的理解透彻，并且是难得的教学高手，至少我总是看书的时候有一个疑问，随后不久就被他解答了，就像牵着手带路那么体贴，可惜中文版实在看不下去，只看了两第三章。

84、太罗嗦了

85、经典好书

86、@fly2fire

毕竟是长辈们... 说太直接也不太妥。

87、这本书买到现在我也就看了那么几次，所以不好发表太多议论。从整体看，我最不满意他的排版，封面。还有印刷质量。书中介绍的东西也不多，不够详尽。

88、C++的经典之作

89、经典吧...毕竟是经典

90、建议读影印版，这个中文版翻译的有些差强人意。

91、整体的印刷质量不好

92、很不错的C++书籍

93、Bruckel所著的编程思想。大师级别，毋庸置疑的强。

94、这才是真正的深入浅出！

95、慵懒...

和懒惰又贪婪是不一样的

96、太多看不懂，简直是天书。

97、无可争议的好书！感觉甚至比CPP Primer更顺畅，很多东西解释的更详细。CPP Primer只告诉你应该怎么做，但是这本书会告诉你为什么要这么做，争取再读一遍

98、读完了《c++编程思想》的1st edition，当然是中文版的，虽然有些地方翻译的很拗口，但确实知道作者在表达些什么意思~

99、看的第三本c++的书，自然的描述、简单的单词、轻松的氛围，看完这本书，自己已经比较全面的了解C++语法、功能点，知道了不少C++还能做的事情。大师Bruce Eckel,佳作 Thinking in c++

。

温馨提示：一定要看英文版。

100、合订本也一样的烂。无奈读中文还是舒畅一些。拿了本中文版和英文电子版，中文看不明白了就看看英文版。

101、又去了解了一下C++

精彩书评

- 1、书籍说明最经典的C++书籍之一适合在入门之后，在对整个C++的思维方式进行梳理大牛的经典作品，一定要读当做进阶的C++书籍来读，一定会有收获的阅读建议第三本C++书籍！！
- 2、作者对C++的理解深入骨髓，又是顶级的教师，再加上他对本书的精益求精，铸就了C++书籍的一个神话。本书适合对C++和面向对象都有所了解的人，它不但用最合适的方式描述了C++是什么，还讲清楚了为什么。但更值得我们学习的是作者在字里行间所透露的对编程、对设计、对面向对象思想的理解以及永无止境地追求更高境界的态度。`while (前两章不能很好地理解) {if (第一次看) {看完全书;} else {sleep(一年); // 其它线程可不能睡，得抓紧时间夯实基础}}`
- 3、习惯到网上买书了，种类挺多，买起来方便，送的还真挺快，好像我一般二三天就能收到了，挺快的。这次也一样，刷书就到了。要是学习也能这么快，嘿嘿.....天下哪有那么容易的事儿呀~~功夫不负苦心人，终于找到了一个专门教电脑的猎豹网校，想学C++，看网校上老师选的是最经典的教程，试听了一下，效果挺好。要是早点找到这类网校，不早轻松多了？
- 4、Bruce Eckel写过很多经典的计算机书籍，尤其是在计算机语言方面，包括C++和Java，尤其是C++造诣很高，这本书是大学面向对象课程的教材，即使上了研究生，再读读这本书还是很有收获！
- 5、看的第三本c++的书，自然的描述、简单的单词、轻松的氛围，看完这本书，自己已经比较全面的了解C++语法、功能点，知道了不少C++还能做的事情。大师Bruce Eckel,佳作 Thinking in c++。
温馨提示：一定要看英文版。

章节试读

1、《C++编程思想》的笔记-第157页

Note that the c heap manager is fairly primitive. It gives you chunks of memory and takes them back when you free() them. There's no facility for heap compaction, which compresses the heap to provide bigger free chunks.

2、《C++编程思想》的笔记-zdfg

c++的经典书籍
数目对我来说之一

3、《C++编程思想》的笔记-

传说中C++的经典，期待中。

4、《C++编程思想》的笔记-14: Inheritance & Composition

But, and this is an important point, as soon as you decide to write your own copy-constructor and assignment operator, the compiler assumes that you know what you ' re doing and does not automatically call the base-class versions, as it does in the synthesized functions. If you want the base class versions called (and you typically do) then you must explicitly call them yourself.既然是用户通常做的事情，为什么不设为默认呢？是考虑到可以在拷贝构造函数中用列表调用构造函数吗？这更不常见吧。还是实现难度问题？好吧正解恐怕是“不带参数”原则。这有些过分了吧，似乎有违“最合常理”（“最少惊讶”似乎是个不太好的翻译）原则。也许是出于安全性的考虑吧。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com