

《C++沉思录》

图书基本信息

书名：《C++沉思录》

13位ISBN编号：9787115106223

10位ISBN编号：7115106223

出版时间：2002-11-01

出版社：人民邮电出版社

作者：Andrew Koenig, Barbara Moo

页数：386 页

译者：黄晓春, 孟岩(审校)

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.tushu000.com

《C++沉思录》

内容概要

《C++ 沉思录》集中反映了C++的关键思想和编程技术，不仅告诉你如何编程，还告诉你为什么要这样编程。本书曾出现在众多的C++专家推荐书目中。这将是C++程序员的必读之作。因为：它包含了丰富的C++思想和技术，从详细的代码实例总结出程序设计的原则和方法。不仅教你如何遵循规则，还教你如何思考C++编程。既包括面向对象编程也包括泛型编程。探究STL这一近年来C++最重要的新成果的内在思想。

《C++沉思录》

作者简介

Andrew Koenig 和 Barbara E . Moo 是夫妻。

Andrew Koenig 是AT&T公司Shannon实验室大规模编程研究部门中的成员，同时也是C++标准委员会的项目编辑。他的编程经验超过30年，其中有15年在使用C++，他已经出版了超过150篇和C++有关的论文，并且在世界范围内就这个主题进行过多次演讲。

Barbara E . Moo 是独立咨询顾问，在软件领域中拥有超过20年的工作经验的独立咨询顾问。在AT&T工作的近15年中，她参与了第一个使用C++编写的商业产品的开发，领导了公司中第一个C++编译器项目，管理了AT&T中广受赞誉的World Net Internet Service Business的开发。

书籍目录

第0章 序幕

第一篇 动机

第1章 为什么我用C++

第2章 为什么用C++工作

第3章 生活在现实世界中

第二篇 类和继承

第4章 类设计者的核查表

第5章 代理类

第6章 句柄：第一部分

第7章 句柄：第二部分

第8章 一个面向对象程序范例

第9章 一个课堂练习的分析（上）

第10章 一个课堂练习的分析（下）

第11章 什么时候不应当使用虚函数

第三篇 模板

第12章 设计容器类

第13章 访问容器中的元素

第14章 迭代器

第15章 序列

第16章 作为接口的模板

第17章 模板和泛型算法

第18章 泛型迭代器

第19章 使用泛型迭代器

第20章 迭代器配接器

第21章 函数对象

第22章 函数配接器

第四篇 库

第23章 日常使用的库

第24章 一个库接口设计实例

第25章 库设计就是语言设计

第26章 语言设计就是库设计

第五篇 技术

第27章 自己跟踪自己的类

第28章 在簇中分配对象

第29章 应用器、操纵器和函数对

第30章 将应用程序库从输入输出中分离出来

第六篇 总结

第31章 通过复杂性获取简单性

第32章 说了HELLO WORLD后再做什么

精彩短评

- 1、收获较少，许多东西已经知道了，和前面读过的书知识重叠
- 2、确实很沉思
- 3、不错，OO
- 4、有些沉思已经过时，但是不影响其历史地位以及价值。
- 5、我要证明自己也是学过C++滴！
- 6、还不错，适合有一定c++基础的同学。由浅入深娓娓道来，读着很轻松，但又值得深思。技术到了极致就会变成哲学，设计的哲学。
- 7、好久没有看语言类的书籍了。这本书整体感觉还行，但没有期望的那么高~
- 8、两个大牛
- 9、说白了就是抽象和加层，至于怎么抽象怎么加层，值得沉思~
- 10、和关注度不匹配的一部经典！
- 11、没读懂
- 12、讲故事的书，人家说很好，但是我的感觉很一般
- 13、改变了对c++看法
- 14、能触发人思考的文字才是好文字
- 15、读书笔记总结的书籍，原创风格。
- 16、知识较旧，讲的知识比较基础，多是启发性的，适合初级程序员。例如像模板这一篇，围绕容器类的设计讲了很多内容，你说有用也有用，但是作为用STL用了这么久的程序员，都有一种似曾相识的感觉。总体感觉不是太有用。
- 17、大一C++期末复习的时候浏览过。期末编程考试是开卷的，题目刚好在书中出现，且当时我也带了这本书，结果托大没看书。。
- 18、说实话，这本书算是真的引导我去理解c++的第一本书
- 19、还行...
- 20、额，翻了，没看懂，C++程度暂时没有那么高.....暂时封藏了.....
- 21、校馆
- 22、很多设计模式的思想，句柄类讲的不错
- 23、经典就是经典！
- 24、印象最深的就是引用计数了。
- 25、Cpp大师如是说
- 26、不是介绍C++语言而是阐述C++程序设计的书。作者对C++的把握毋庸置疑。但书已经有点老了。由于是由文章整理的，所以系统性也不够强。可以不做必读书。
- 27、其实里边很多思想在STL里边有所体现，还是吐槽一下翻译，如果没有了解过C++一些东西，书中有些地方看起来不知所云。总体来说还是一本不错的书
- 28、翻译的不是很好，未读完
- 29、没看完。
- 30、一本老书，但强烈推荐
- 31、unfinish
- 32、写的很不错，虽然名气不大
- 33、没什么新鲜玩意。
- 34、入门书
- 35、虽然这本书所阐述的内容还是C++98，但是里面所展示的C++的应用精髓是每一个C++程序员都应该认真学习和掌握的。C++绝对不是一门语言那么简单，这本书给我们展示一个在实际生活之中的强有力的C++，教会读者如何规避可能的不良习惯。通过逐步引导的方式给我们阐明每一种最佳实践背后的道理。我觉得每一个程序员都需要读这本书。
- 36、书比较老了，有些东西现在已经属于常识了。
- 37、一本好玩的书
- 38、虽然老吧,但是其实阅读起来非常的愉快的,很多思想其实不过时.

《C++沉思录》

- 39、书是好书，翻译的不太好，试图读了好几次，第三次才读完，确实不错，属于实战派的
- 40、也许我应该早一点看这本书
- 41、书是本好书，可惜这个翻译实在是.....
- 42、很容易懂的OOP书籍，风格很大众
- 43、其实这本才是真正的"C++编程思想"，而且作者介绍方式独特，别去纠缠翻译的功力了，对于此种书，作者贯穿书中的思想才是最重要的。
- 44、夫妻程序员就是好啊
- 45、C++泛型的思路，学习的过程。
- 46、内容是作者几十年的技术文章的汇总，相当部分都已经比较老了，和当前的C++最佳实践出入较大，不推荐再看了
- 47、生活是复杂的原因是：它既有现实的一面，又有充满幻想的一面。
- 48、一年了，终于看完了，收获很大的
- 49、去年看的，看的不太懂。还想再读
- 50、受益匪浅

《C++沉思录》

精彩书评

- 1、当然和成书的时间有关，friend关键字和handle手法贯穿全书。其实是有点不适合现代cpp的。一些思想看看也行。
- 2、之前是看到有人推荐，后来去书店里面买到了。目前看到第5章，应该说才是真正开始看了。看到第四章的类设计者的核查表时的那14条，再联想到之前看的C++ Primer里面的知识，总是能激发我的思考。真是不错

章节试读

1、《C++沉思录》的笔记-第266页

熟悉boost库

2、《C++沉思录》的笔记-第40页

将拷贝构造函数和赋值操作符声明私有

3、《C++沉思录》的笔记-第279页

缩短变量作用域

4、《C++沉思录》的笔记-代理类

代理类在实践中还是很常用的。主要能处理控制内存分配和把不用类型的对象放入同一个容器的情况。个人觉得很明显的两个好处是：

- 1、不用到处new和delete还担心操作不配对了
- 2、兼容和规避了继承层次中纯虚基类的"伪对象"的表示问题

5、《C++沉思录》的笔记-第296页

设计一个类是有技术的

6、《C++沉思录》的笔记-第143页

简化内存管理5.4节

7、《C++沉思录》的笔记-27

跟踪类

8、《C++沉思录》的笔记-第92页

先写类的使用再写类的定义再写类的实现

9、《C++沉思录》的笔记-第365页

c++里复制一个对象不等于按位复制组成这个对象的字节，用拷贝构造函数和赋值操作符为什么？

10、《C++沉思录》的笔记-第120页

设计方案：简单

2：省内存，灵活，复杂

如何选择，根据问题和其背景

11、《C++沉思录》的笔记-第103页

面向对象是否真的灵活暂且不谈。
但是第10章关于用面向对象设计字符图像的话题，消耗很大的智商，写出来的程序很难理解。
不知这样做到底为了什么？难道为了灵活而不惜丧失程序的简单易读。

我还是倾向于KISS原则。

这越来越让我觉得C++只不过是一个智商玩具。
面向对象类与类之间的相互关联错综复杂，晦涩难读。

第10章代码详见博客园<http://www.cnblogs.com/unixfy/p/3463846.html>

12、《C++沉思录》的笔记-第374页

先掌握构造类的技术，再用继承
能用模板很好解决，为什么用oo
用类表示概念

13、《C++沉思录》的笔记-第108页

框内信息由此类负责，框外由彼类负责

14、《C++沉思录》的笔记-类设计者的核查表

现在定义类、设计里面的接口时好像较随意，需要用一個什么样的方法就开一个什么样的方法，
参数类型、返回类型等也有时候定得比较草率，这样灰常不好.....养成良好的习惯和规范是很有必要的。

书中核查表大概内容：1.你的类需要一个构造函数吗？

2.你的数据成员是私有的吗？

3.你的类需要一个无参的构造函数吗？

4.是不是每个构造函数初始化所有的数据成员？

5.类需要析构函数吗？

6.类需要一个虚析构函数吗？

7.你的类需要复制构造函数吗？

8.你的类需要一个赋值操作符吗？

9.你的赋值操作符能正确地将对象赋给对象本身吗？

10.你的类需要定义关系操作符吗？

11.删除数组时你记住了用delete[]吗？

12.记得在复制构造函数和赋值操作符的参数类型中加上const了吗？

13.如果函数有引用参数，它们应该是const引用吗？

14.记得适当地声明成员函数为const的了吗？

ps：最近做东西深刻感受到的一点是前期花一些时间来设计良好的继承关系和机制是非常非常非常重要的！

顺便确认和备忘一下一个知识点：class A

class B : public A

A *pA = new B;

delete pA;当A的destructor不为virtual时只调用~A()。

15、《C++沉思录》的笔记-第372页

程序库优于语言细节
易理解类的概念和学会如何构造类技术
复用库代码，而不是凡是自己动手

16、《C++沉思录》的笔记-第86页

数据抽象和多态为可变化的扩展服务的

17、《C++沉思录》的笔记-第327页

清楚理解问题来简化设计，直接实现
一个同时释放资源的类

18、《C++沉思录》的笔记-第249页

望而生畏写法

19、《C++沉思录》的笔记-16

模块化接口的设计

20、《C++沉思录》的笔记-第31页

为问题挑工具，不是为工具挑问题

21、《C++沉思录》的笔记-第131页

list的实现lisp

22、《C++沉思录》的笔记-第109页

c++使用
实现

23、《C++沉思录》的笔记-27

内存管理跟踪类

24、《C++沉思录》的笔记-第5页

类为了适应扩展
提供了放置信息的地方，信息隐藏

25、《C++沉思录》的笔记-第29页

开始读

26、《C++沉思录》的笔记-第1页

好书

27、《C++沉思录》的笔记-第246页

函数不能返回函数，能返回函数对象

28、《C++沉思录》的笔记-9

这章不错

29、《C++沉思录》的笔记-第367页

例子不错

30、《C++沉思录》的笔记-第165页

stl的类函数为什么这么设计

31、《C++沉思录》的笔记-第52页

用代理对象不是对象本身解决内存分配和同继承体系中的不同对象放入同一容器

32、《C++沉思录》的笔记-第148页

当vector变大时，原来的指针可能失效，应用下标

33、《C++沉思录》的笔记-第286页

构造函数中对new失败如何处理是最好呢

34、《C++沉思录》的笔记-泛型之前

Andrew Koenig

- 1.trace 开关/与流向的设计问题
- 2.动态字符串malloc回收问题,容易导致过分关心内存泄露
- 3.细节隐藏
- 4.字符串大小越界的问题
- 5.不能专注于业务上，特别处理错误，统一编码规范等问题特别多
- 6.抽象分解子问题，并互相独立
- 7.抽象和内存管理 -- 析构函数对资源管理的运用
- 8.系统的整体性 -- 图书馆的查询系统替换/对接问题
- 9.接上8的，编程语言的交互性
- 10.协作系统无法执行垃圾回收机制的语言
- 11.动态绑定
- 12.Handle类

- 13.Private Public 成员的设计安全问题
 - 14.析构函数的陷阱
 - 15.类复制,例string& operation=(const string&s) 先判断对象先释放旧值
 - 16.不必要的函数,"只为用到的东西付出"
 - 17.抽象类,容器类,代理类--不同类相互关联
 - 18.指针内存管理
 - 19.Handle h(p) //这应该是什么含义 -- operation->() 避免对象复制,统一销毁方式
 - 20.引用计数句柄,绑定对象
 - 21.句柄绑定缺陷 -> 必须定义具有类型T成员的新类
 - 22.句柄类,与智能指针.....
 - 23.OOP 3要素:数据抽象、继承、动态绑定 (virtual func 为静态绑定)
 - 24.局步化内存分配,灵活性,效率问题
 - 25.运算树的例子很好的解释了继承的好处
 - 26.Picture 类的迭代开发模式
 - 27.不喜欢操作符重载
 - 28.一维数组解决二维数组问题
 - 29.虚函数的设计与开销问题
 - 30.指针引用调用成员函数,与虚函数调用花时间翻倍
 - 31.模板与STL
 - 32.容器内容的复制
 - 33.operator [] 的难题
 - 34.后面几章直接读STL template 的书比较好
-
- 101.墨菲法则 -- 一件事只要有可能变糟,就一定会变糟

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.tushu000.com