

## <<C++编程规范>>

### 图书基本信息

书名：<<C++编程规范>>

13位ISBN编号：9787115142054

10位ISBN编号：711514205X

出版时间：2006-1

出版时间：人民邮电出版社

作者：（美）Herb Sutter, Andrei Alexandrescu

页数：216

译者：刘基诚

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：<http://www.tushu007.com>

## <<C++编程规范>>

### 内容概要

《C++编程规范：101条规则准则与最佳实践》中，两位知名的C++专家将全球C++界20年的集体智慧和经验凝结成一套编程规范。

这些规范可以作为每一个开发团队制定实际开发规范的基础，更是每一位C++程序员应该遵循的行事准则。

书中对每一条规范都给出了精确的描述，并辅以实例说明；从类型定义到错误处理，都给出了最佳的C++实践。

即使使用C++多年的程序员也会从《C++编程规范：101条规则准则与最佳实践》中受益匪浅。

《C++编程规范：101条规则准则与最佳实践》适合于各层次C++程序员使用，也可作为高等院校C++课程的教学参考书。

## 作者简介

Herb Sutter ISO C++标准委员会主席，C++ Users Journal杂志特邀编辑和专栏作家。他目前在微软公司领导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：Exceptional C++ Style（中文版即将由人民邮电出版社出版）、Exceptional C++ Style和More Exceptional C++ Style。

## &lt;&lt;C++编程规范&gt;&gt;

## 书籍目录

组织和策略问题第0条 不要拘泥于小节(又名:了解哪些东西不应该标准化)第1条 在高警告级别干净落地进行编译第2条 使用自动构建系统第3条 使用版本控制系统第4条 在代码审查上投入设计风格第5条 一个实体应该只有个紧凑的职责第6条 正确、简单和清晰第7条 编程中应知道何时和如何考虑可伸缩性第8条 不要进行不成熟的优化第9条 不要进行不成熟的劣化第10条 尽量减少全局和共享数据第11条 隐藏信息第12条 懂得何时和如何进行并发性编程第13条 确保资源为对象所拥有。

使用显式的RAII和智能指针编程风格第14条 宁要编译时和连接时错误,也不要运行时错误第15条 积极使用const第16条 避免使用宏第17条 避免使用“魔数”第18条 尽可能局部地声明变量第19条 总是初始化变量第20条 避免函数过长,避免嵌套过深第21条 避免跨编译单元的初始化依赖第22条 尽量减少定义性依赖。

避免循环依赖第23条 头文件应该自给自足第24条 总是编写内部群include保护符,决不要编写外部#include保护符函数与操作符第25条 正确地选择通过值、(智能)指针或者引用传递参数第26条 保持重载操作符的自然语义第27条 优先使用算术操作符和赋值操作符的标准形式第28条 优先使用++和的标准形式。

优先调用前缀形式第29条 考虑重载以避免隐含类型转换第30条 避免重载&&、||或,(逗号)第31条 不要编写依赖于函数参数求值顺序的代码类的设计与继承第32条 弄清所要编写的是哪种类第33条 用小类代替巨类第34条 用组合代替继承第35条 避免从并非要设计成基类的类中继承第36条 优先提供抽象接口第37条 公用继承即可替换性。

继承,不是为了重用,而是为了被重用第38条 实施安全的改写第39条 考虑将虚拟函数声明为非公用的,将公用函数声明为非虚拟的第40条 要避免提供隐式转换第41条 将数据成员设为私有的,无行为的聚集(C语言形式的struct)除外第42条 不要公开内部数据第43条 明智地使用Pimpl第44条 优先编写非成员非友元函数第45条 总是起提供new和delete第46条 如果提供类专门的new,应该提供所有标准形式(普通、就地和不抛出)构造、析构与复制第47条 以同样的顺序定义和初始化成员变量第48条 在构造函数中用初始化代替赋值第49条 避免在构造函数和析构函数中调用虚拟函数第50条 将基类析构函数设为公用且虚拟的,或者保护且非虚拟的第51条 析构函数、释放和交换绝对不能失败第52条 致地进行复制和销毁-第53条 显式地启用或者禁止复制第54条 避免切片。

在基类中考虑用克隆代替复制第55条 使用赋值的标准形式第56条 只要可行,就提供不会失败的swap(而且要正确地提供)名字空间与模块第57条 将类型及其非成员函数接口置于同名字空间中第58条 应该将类型和函数分别置于不同的名字空间中,除非有意让它们一起工作第59条 不要在头文件中或者样include之前编写名字空间using第60条 要避免在不同的模块中分配和释放内存第61条 不要在头文件中定义具有链接的实体第62条 不要允许异常跨越模块边界传播第63条 在模块的接口中使用具有良好可移植性的类型模板与泛型第64条 理智地结合静态多态性和动态多态性第65条 有意地进行显式自定义第66条 不要特化函数模板第67条 不要无意地编写不通用的代码错误处理与异常第68条 广泛地使用断言记录内部假设和不变式第69条 建立合理的错误处理策略,并严格遵守第70条 区别错误与非错误第71条 设计和编写错误安全代码第72条 优先使用异常报告错误第73条 通过值抛出,通过引用捕获第74条 正确地报告、处理和转换错误第75条 避免使用异常规范STL:容器第76条 默认时使用vector。

否则,选择其他合适的容器第77条 用vector和string代替数组第78条 使用vector(和string::c\_str)与非C++API交换数据第79条 在容器中只存储值和智能指针第80条 用push\_back代替其他扩展序列的方式第81条 多用范围操作,少用单元操作第82条 使用公认的惯用法真正地压缩容量,真正地删除元素STL:算法第83条 使用带检查的STL实现第84条 用算法调用代替手工编写的循环第85条 使用正确的STL查找算法第86条 使用正确的STL排序算法第87条 使谓词成为纯函数第88条 算法和比较器的参数应多用函数对象少用函数第89条 正确编写函数对象类型安全第90条 避免使用类型分支,多使用多态第91条 依赖类型,而非其表示方式第92条 避免使用reinterpret\_cast第93条 避免对指针使用static\_cast第94条 避免强制转换const第95条 不要使用c风格的强制转换第96条 不要对非POD

<<C++编程规范>>

进行memcpy操作或者memcmp操作第97条 不要使用联合重新解释表示方式第98条 不要使用可变长参数(...)第99条 不要使用失效对象。  
不要使用不安全函数第100条 不要多态地处理数组参考文献摘要汇总索引

## &lt;&lt;C++编程规范&gt;&gt;

## 媒体关注与评论

今年是C++ 20周年，本书的出版可以说是恰逢其时。本书是近年来最重要的C++图书，凝聚了C++界 20年的集体智慧。荣获今年Jolt生产效率大奖从某种意义上也证明了这一点。请注意它绝不是国内曾经出版的那些“规定代码应该缩进几格”之类的编码规范书所能相提并论。它首先是一本所有C++程序员都需要阅读的最佳实践指南，不仅涵盖编程，也涉及类的设计与优化等主题。书中提到的许多应该避免的问题，即使是许多C++老手也会犯。有些反映的是业界最新的经验总结，蕴涵着C++新的改进方向（毕竟，本书的作者之一是C++标准委员会的主席）。有些技术细节非常深入。同时，如果在学习C++的同时，有本书在手，就能很好地避免形成一些常见的误解，重走前人的老路。对相关各知识点的掌握也能更加深入。其次，它也是开发小组制定自己所用编程规范的很好的起点。由于作者和技术审校团队的权威背景，本书的各种规则几乎都是无可置疑的。以此为参考，能够将各开发小组的规范水平提高一个档次。最后，本书也是阅读众多C++经典文献的绝佳入口。因为内容是按知识点和条目形式组织的，每个条目最后又有相关的参考文献及其章节编号，查阅非常方便。

## <<C++编程规范>>

### 编辑推荐

《C++编程规范：101条规则准则与最佳实践》涵盖了C++程序设计的方方面面，包括设计和编码风格、函数、操作符、类的设计、继承、构造与析构、赋值、名字空间、模块、模板、泛型、异常、STL容器和算法等。

书中对每一条规范都给出了言简意赅的叙述，并辅以实例说明；书中还给出了从类型定义到错误处理等方面大量的C++最佳实践，包括许多最新总结出的和标准化的技术，即使使用C++多年的程序员也会从中受益匪浅。

#### 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:<http://www.tushu007.com>