

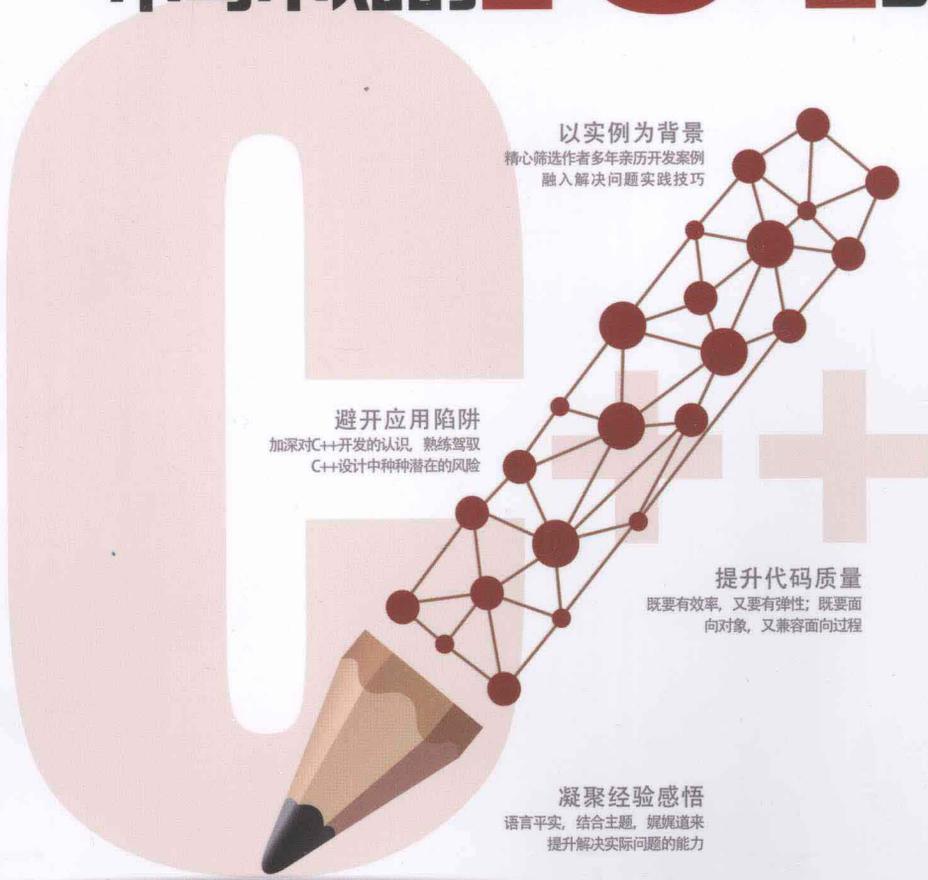
编码的法则

本书中源代码下载地址

<http://www.tdpress.com/zyzx/tsscflw/>

C++程序员不可不知的101条实用经验

刘光 编著



以实例为背景
精心筛选作者多年亲历开发案例
融入解决问题实践技巧

避开应用陷阱
加深对C++开发的认识，熟练驾驭
C++设计中种种潜在的风险

提升代码质量
既要有效率，又要有弹性；既要面
向对象，又兼容面向对象

凝聚经验感悟
语言平实，结合主题，娓娓道来
提升解决实际问题的能力

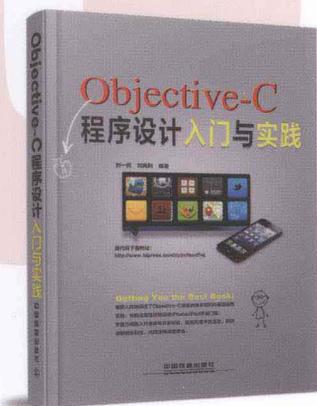
编码的法则

本书中源代码下载地址
<http://www.tdpress.com/zyzx/tsscflwj>

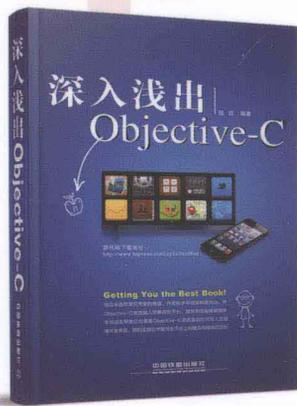
101 实用经验

C++ 程序员不可不知的 101 条

刘光 编著



书名: **Objective-C**程序设计入门与实践
定价: 49.80元



书名: 深入浅出**Objective-C**
定价: 89.00元

上架建议: 计算机/程序设计/C++



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

地址: 北京市西城区右安门西街8号
邮编: 100054
网址: <http://www.tdpress.com>

ISBN 978-7-113-17653-2



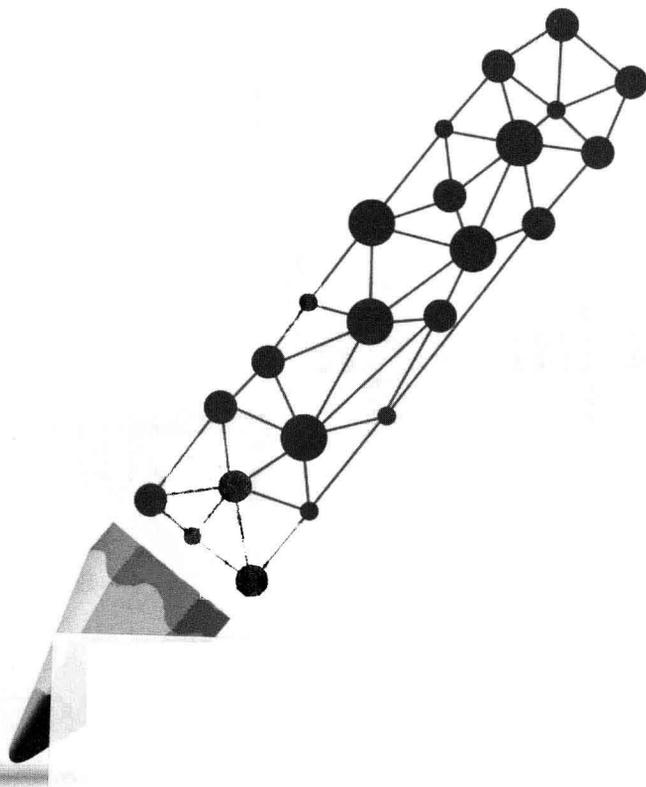
9 787113 176532 >

定价: 59.80元

编码的法则

C++程序员不可不知的101条实用经验

刘光 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书深入浅出地介绍了 C++ 语言的各种语法特性, 不仅是一本实用的参考用书, 还是笔者多年来对 C++ 语言开发的经验和感悟。内容结构独具匠心, 以实际实例为背景, 结合主题, 探讨 C++ 语言面向对象的深入细节, 有助于加深读者对 C++ 语言开发的运用。

本书涵盖了与 C++ 语言相关的大部分知识, 在提供丰富内容的同时, 进一步探讨了更高深的 C++ 特性及其应用场景, 旨在帮助有较少开发经验的 C++ 程序员加深理解, 从而避开 C++ 开发的误区和陷阱, 同时也可以帮助中高级读者进一步提高分析和解决实际问题的能力。

图书在版编目 (CIP) 数据

编码的法则 : C++ 程序员不可不知的 101 条实用经验 /
刘光编著. — 北京: 中国铁道出版社, 2014. 1
ISBN 978-7-113-17653-2

I. ①编… II. ①刘… III. ①C 语言—程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 269680 号

书 名: 编码的法则: C++ 程序员不可不知的 101 条实用经验
作 者: 刘 光 编著

策 划: 荆 波
责任编辑: 张 丹
责任印制: 赵星辰

读者热线电话: 010-63560056
特邀编辑: 赵树刚
封面设计: 多宝格·付巍

出版发行: 中国铁道出版社 (北京市西城区右安门西街 8 号 邮政编码: 100054)
印 刷: 三河市华业印装厂
版 次: 2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷
开 本: 787mm×1092mm 1/16 印张: 28.25 字数: 627 千
书 号: ISBN 978-7-113-17653-2
定 价: 59.80 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社读者服务部联系调换。电话: (010) 51873174

打击盗版举报电话: (010) 51873659

前 言

C++是当今主流程序设计语言的一种，由美国 AT&T 贝尔实验室的 Bjarne Stroustrup 博士于 20 世纪 80 年代发明和实现的，目的是解决 C 语言开发过程中遇到的棘手问题。C++语言起初被称之为“C with Classes”、“a better C”，作为 C 语言的增强版而出现，直到 1998 年才正式更名为 C++。

C++是一种颇受争议的语言。这主要体现在其复杂性和流行性两个方面。Linux 之父就称其为“糟糕程序的垃圾语言”，但其优良的兼容性却使其颇受欢迎。

首先 C++是一种复杂、难以掌握的语言，C++的复杂难以掌握不仅表现在其复杂的语法，更体现在其提供的 4 种编程思维模型上面，包括：**procedural-based**，**object-based**，**object-oriented**，**generic paradigm**。这些是 C++的设计目的决定的。C++语义的目标是：既要有效率，又要有弹性；既要面向对象，又要兼容面向过程。学习起来自然不可能很简单。

再者 C++是一门流行的语言。在 Tiobe 编程语言热度排行榜中 C++稳居前三。这主要原因是 C++具有面向对象的逻辑表达优势，还具有 C 语义在时间和空间上的效率。这是每个程序员都向往的。

难学易用，可以说是 C++语言最大的特点。同样也是其独具魅力的地方。C++复杂的特性导致 C++语言比较难以驾驭。烦琐的面向对象特性，使得初学者很容易陷入误区，而一知半解的掌握又使得 C++设计工程布满更多的陷阱。为了使您更好地了解 C++语言特性，也为了更好地避免种种潜在的风险而不得不对其有更深入的认识，这也是本文写作的初衷。

在 C++语言发展的最初阶段，人们想知道 C++是个什么东西。现在 C++语言已发展成熟，如何正确高效地使用它，已经成为我们亟待解决的问题。对于那些刚加入 C++开发行列的程序员来说。别人的前车之鉴，或许能对自己有很大的帮助，可以帮助自己快速、准确地掌握 C++语言的特征。

章节安排

全书分为四篇，共 13 章，由浅入深地讲解 C++ 语言的使用注意事项和优秀实践。每篇内容可总结如下：第一篇讲述 C++ 基本的语言特征，第二篇讲述类和数据抽象，第三篇讲述 C++ 面向对象的机制，第四篇讲述 C++ 的一些高级特征。

第一篇：包括 1~8 章，内容涵盖 C++ 基本语言特征。涉及概念包括 C++ 的关键词、数组、指针、表达式、函数、字符串、预处理指令等最基本的语言特征。

第 1 章：以 C 语言为参照，对 C++ 的语言特性进行了一些提纲挈领的介绍。重点介绍 C++ 和 C 语言的区别和联系，面向对象的概念等。

第 2 章：类型是所有程序的基础，类型告诉数据代表什么意思以及针对此数据可执行的操作。C++ 定义的基本类型有：字符型、整型、浮点型等，除此之外还可自定义类型。本章介绍 C++ 中全局变量和局部变量各自的应用特点，列举了结构体和联合体的适用场景，并对类型转换进行了一些深度探讨。

第 3 章：介绍关键字的用法介绍和注意事项，包括 `const`, `volatile`, `mutable`, `static`, `sizeof`, `new` 和 `delete`，以及关键字使用中的陷阱。

第 4 章：数组和指针是 C++ 最为重要的两个概念，数组保存某种类型的一组对象，数组一旦创建，就不允许添加新元素了，而指针可以遍历数组中的元素。本章所有的论题均涉及两者常见的一些应用陷阱。

第 5 章：本章的主题是表达式和语句。C++ 提供了丰富的运算符，并定义针对内置类型操作符含义，除此之外还支持操作符重载。此章节的议题分类讨论一些容易陷入误区的表达式和语句。

第 6 章：本章的主题是函数，函数是 C++ 语言重要的组成部分，其和 C++ 中的类型、数组等概念具有同等的地位，对函数的讨论，包括参数传递、重载、覆盖事项、以及内联函数等。

第 7 章：介绍字符和字符串的使用注意事项。每个可执行程序都会涉及字符和字符串。因为程序要接受用户输入，产生对应输出（显示）。用户输入的数据计算机如何解释，以及计算后的输出显示。这些都涉及字符的编码。

第 8 章：预处理是 C++ 程序编译的第一个步骤，预处理在源代码编译之前进行一些文本性质的操作。具体包括注释删除，插入被 `#include` 包含的文件内容、定义和替

换有`#define` 指令定义的符号，以及确定代码部分内容是否应根据一些条件编译指令进行编译。本章议题主要涉及预处理指令，包括 `define`、`assert` 和`##`等。本章的议题将向你展示 C++编译器预处理所进行的不为人知的事情，以及其存在的陷阱。

第二篇：包括 9、10 两章。内容涵盖 C++的类和数据抽象封装。这些是 C++区分面向过程语言最根本的特征之一。

第 9 章：本章议题涉及类的设计和实现，包括类的构造、析构、数据（成员函数）访问权限、类的继承等，这些都是类的类型基础。本章的议题将向你展示类的类型设计和实现的陷阱和注意事项。

第 10 章：本章主要讨论操作符的重载，操作符重载过程的一些注意事项。本章将详细为你讲述重载操作符的一些内容。运用本章提供的信息，我想你会从以前的困扰中走出，像专家一样将“操作符重载”玩弄于股掌之间。

第三篇：包括 11、12 两章。内容涵盖 C++面向对象编程机制。它们是 C++语言面向对象的灵魂所在。

第 11 章：本章议题围绕继承和多态展开。将为你阐述继承和多态的实现机制，以及两者的关系。通过本章议题的阅读，可帮助你对继承和多态有更为深刻的认识。

第 12 章：本章讨论的议题为模板和范型。通过本章的阅读可帮你加深对“面向对象所依赖的多态是运行时的多态，而泛型模板所依赖的多态是编译时的多态或参数式多态”的认识。

第四篇：包括 13 章，主要讨论 C++编程中一些高级特征。在某些适当的场景下是有用的，但并不是所有的场景下都必须。没有掌握它们，虽然你可以写出运行良好的程序，但是，如果你掌握并使用了它们，你所编写的代码会在扩展性、可维护性方面，更加具有优势。

第 13 章：本章讨论的议题主要围绕 C++的部分高级特性。涉及虚函数、命名空间、运行效率、代码可读性等诸多方面。掌握它们是你迈向 C++高级程序员的必经之路。

读者对象

本书适合有一定的 C/C++ 语言基础，并想深入了解 C++ 面向对象程序设计方法的初学者；有兴趣了解 C++，并应用于实际工作中的 C++ 程序开发人员。

致谢

感谢远在故乡的父母，你们的关心和爱护，是我面对困难和失落时重新奋起的动力源泉。让我不敢懈怠，在任何时候都不轻言放弃。感谢张新华在本书成文时，对我的默默陪伴和支持，愿你生活幸福。感谢吴玲这段时间的关心与呵护，使我从生活的阴霾中走出。感谢林曙光对本书的校审工作。

谨以此书献给爱我的朋友和家人，以及那些热爱 C++ 的同道中人。

刘 光
2013 年 12 月

目 录

第 1 篇 C++基本语言特征

第 1 章 快速入门	1
实用经验 1 C++程序由哪些部分构成	2
实用经验 2 既有面向过程，为何还要面向对象	6
实用经验 3 掌握在 C++中如何使用 C	10
实用经验 4 C++比 C 加了什么	15
第 2 章 变量和类型	21
实用经验 5 计算机是如何存储变量的	22
实用经验 6 确保每个对象在使用前已被初始化	28
实用经验 7 局部变量和全局变量的差别	31
实用经验 8 掌握变量定义的位置与时机	37
实用经验 9 引用难道只是别人的替身	39
实用经验 10 枚举和一组预处理的#define 有何不同	42
实用经验 11 为何 struct x1{struct x1 stX};无法通过编译	46
实用经验 12 实现可变数组 struct{int namelen; char namestr[1];}	50
实用经验 13 typedef 使用的陷阱	54
实用经验 14 优化结构体中元素的布局	58
实用经验 15 既有结构，为何引入联合	64
实用经验 16 提防隐式转换带来的麻烦	67
实用经验 17 深刻理解 void 和 void*	73
实用经验 18 如何判定变量是否相等	77
第 3 章 关键字	84
实用经验 19 尽可能多地使用 const	85
实用经验 20 volatile 和 mutable 用在何处	89
实用经验 21 尽量用 new/delete 替代 malloc/free	95
实用经验 22 使用 new/delete 时要采用相同的形式	101
实用经验 23 sizeof 和对象的大小	104

实用经验 24	谨慎使用 static	112
第 4 章	数组和指针	118
实用经验 25	理解指针的本质	119
实用经验 26	论数组和指针的等价性	124
实用经验 27	再论数组和指针的差异性	127
实用经验 28	充满疑惑的数组指针和指针数组	130
实用经验 29	禁止以多态的形式处理数组	133
实用经验 30	防微杜渐, 拒绝野指针	136
实用经验 31	臭名昭著的空指针到底是什么	139
实用经验 32	多维数组和指针	143
实用经验 33	引用和指针的差异	148
第 5 章	表达式和语句	154
实用经验 34	运算符引发的混乱	155
实用经验 35	尽量使用 C++ 转换操作符	159
实用经验 36	表达式求值顺序不要想当然	163
实用经验 37	switch-case 的陷阱	167
实用经验 38	a b 和 a && b 的陷阱	172
实用经验 39	“悬挂” else 引起的问题	176
实用经验 40	标示语句结束的分号的思考	179
实用经验 41	goto 真的一无是处吗	182
实用经验 42	条件操作符和逗号操作符	187
实用经验 43	同魔鬼数字说再见	189
实用经验 44	关于循环语句的讨论	191
第 6 章	函数	196
实用经验 45	禁止函数返回局部变量的引用	197
实用经验 46	函数传值、传指针及传引用的效率分析	201
实用经验 47	内联函数会像宏一样替换吗	207
实用经验 48	函数重载需考虑什么	210
实用经验 49	不要让 main 返回 void	216
实用经验 50	尽量拒绝使用变参函数	219
实用经验 51	如何降低函数的圈复杂度	225
第 7 章	字符和字符串	230
实用经验 52	关于字符编码的讨论	231

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《编码的法则 C++程序员不可不知的101条实用经验》刘光 编.pdf

请登录 <https://shgis.com/post/3168.html> 下载完整文档。

手机端请扫码查看：

