

TURING

图灵程序设计丛书

한빛미디어
Hanbit Media, Inc.

128个编码好习惯

这样编码才规范

[韩] 朴晋锈 著 才璐 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

128个编码好习惯

这样编码才规范

[韩] 朴晋锈 著 才璐 译



人民邮电出版社

北京

图书在版编目(CIP)数据

这样编码才规范: 128个编码好习惯 / (韩) 朴晋锈
著; 才璐译. -- 北京: 人民邮电出版社, 2018.3
(图灵程序设计丛书)
ISBN 978-7-115-47301-1

I. ①这… II. ①朴… ②才… III. ①计算机网络—
编码程序—程序设计 IV. ①TP393

中国版本图书馆CIP数据核字(2017)第284376号

内 容 提 要

本书通过16个主题, 收录了优秀程序员应当遵守的128项基本编码准则, 涉及初期设计、间隔、缩进、标识符名称、运算符使用等多个方面, 并附有大量参考网址及文献, 帮助开发人员夯实基础, 从规范的编码操作开始, 培养良好的编码习惯, 助力未来技术成长。

◆ 著 [韩] 朴晋锈
译 才 璐
责任编辑 陈 曦
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷

◆ 开本: 880×1230 1/32
印张: 9.75

字数: 234千字 2018年3月第1版
印数: 1-4 000册 2018年3月北京第1次印刷

著作权合同登记号 图字: 01-2016-1568号

定价: 45.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字20170147号

站在巨人的肩上

Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上 Standing on Shoulders of Giants



iTuring.cn

版权声明

Coding Guidelines by Park, Jinsoo (朴晋绣)

Copyright © 2013 Hanbit Media, Inc.

All rights reserved.

Originally Korean edition published by Hanbit Media, Inc

The Simplified Chinese Language edition © 2018 by POSTS & TELECOM PRESS

The Simplified Chinese translation rights arranged with Hanbit Media, Inc through
Media Solutions.

本书中文简体字版由 Hanbit Media 授权人民邮电出版社独家出版。
未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

声明

本书提出的诸多观点仅供参考，不必完全遵守。不同情况需要适用不同规则，不同价值观或组织文化也可能导致出现与本书观点背道而驰的情形。因此，没有必要将本书视为金科玉律，各位根据自身情况选择适合的内容即可。

前言

我是拥有多年程序员开发经验的技术作家，本书是我结合自己双重身份积累的经验编写而成的。虽然各种方法论一直层出不穷，但优秀的编码习惯并没有太大变化。因此，本书更专注于始终具备使用价值的优秀的编码习惯和方法。我参考相关书籍，将实际工作中间接接触的知识和主动思考的结论进行了系统分类和详细说明。

盖房子时，如果负责基础工作的泥瓦匠手艺不精又没有竭尽全力，房屋质量就不可能过关。即使后期有技艺精湛的装潢布局使得建筑物的外观精美绝伦，根基没有打好也会成为“豆腐渣工程”。同理，决定软件系统质量的就是编码工作，这就要求负责编码的程序员具有过硬的基本功。

培养具备优秀编码习惯的一流程序员是软件开发公司和学校的职责，不是我能够干预的。但编写并出版合适的工具书，为广大程序员的成长提供帮助，是我力所能及的事情。因此，我决定在此范围内最大限度发挥自身的光和热。

本书融入了很多经验丰富的程序员的多种观点，我要向提供这些观点的程序员——可以称为“黑客”或尊称为“专家”的优秀程序员——表达衷心的感谢。正是仰仗这些人的不懈努力和智慧结晶，我们编写的代码才能越来越精美、准确、高效。因此，我希望每位程序员都读一读这本书，也希望从事与编码工作相关业务的其他人员进行参考。

各位阅读本书时不必按照顺序逐章阅读。虽然与后半部分相比，前半部分更浅显易懂，但只要是写过代码的人，就都能够理解后半部分内容。因此，不必像阅读小说一样从头开始，各位挑选感兴趣的专题研读即可。

如果感到本书内容不够深入，末尾的附录部分给出了更多参考文献，供各位进行深度学习。但我认为，只要牢记本书收录的精髓并加以运用，就足以大幅提升效率。

我一直坚持为出版的每一本书写前言，这次也不例外。虽然现在刚刚完稿，身心俱疲，但我总是安慰自己，写前言就像是前进中留下的脚印。这次完成了这本关于编码风格的书，我又向前迈了一步，留下了一个脚印。但愿今天我留下的这个脚印能为后来人指引方向。

朴晋鏊

2013年9月

第 1 章 基础知识概述

- 1.1 编码风格 2
- 1.2 编码风格教育缺失 3
- 1.3 打磨编码风格的时机 7
- 1.4 必须学习编码风格的原因 9
- 1.5 编码风格比数学或英语更重要 11
- 1.6 所有编程语言都需要编码风格 12
- 1.7 选择用 C 语言阐述编程风格的原因 14
- 1.8 编码风格有益于编译执行方式和混合执行方式 17
- 1.9 基于组件的软件开发方式与编码风格 20
- 1.10 码农的力量不容小觑 23
- 1.11 将编码惯例文档化以统一应用 24

第 2 章 程序设计相关编码准则

- 2.1 遵循最新标准 28
- 2.2 合理限制开发人员的规模 29
- 2.3 维护旧程序比开发新程序更常见 31
- 2.4 不要认为修改程序很容易 32
- 2.5 慎重采用新技术 34
- 2.6 不要采用 RAF 策略 36

第 3 章 间隔相关编码准则

- 3.1 一行只写一条语句 42
- 3.2 区分声明语句和执行语句 44
- 3.3 区分段落 46
- 3.4 区分各种控制语句 50
- 3.5 区分各函数 54
- 3.6 运算符前后需留出空格 61
- 3.7 不要在一元运算符与操作数之间插入空格 63
- 3.8 分号前不要插入空格 63
- 3.9 不要滥用 Tab 键 64
- 3.10 逗号后必须插入一个空格 65
- 3.11 逗号后不要插入太多空格 65
- 3.12 变量初始化时的列对齐 66
- 3.13 一行只声明一个变量 70

第 4 章 缩进相关编码准则

- 4.1 大括号的位置 74
- 4.2 统一大括号的位置 77
- 4.3 内部代码块需要缩进 78
- 4.4 输出部分需要缩进 81
- 4.5 不要毫无意义地缩进 84
- 4.6 保持缩进程度的一致性 86
- 4.7 选择合适的缩进程度 87
- 4.8 不要编写凸出形式的代码 88

第 5 章 注释相关编码准则

- 5.1 多种注释形态 92
- 5.2 区分单行注释和注释框 94
- 5.3 添加“变量字典编写专用注释” 98
- 5.4 向程序插入伪代码 100
- 5.5 通过注释标注程序目标 102
- 5.6 程序起始部分必须添加头注释 104
- 5.7 在等于运算符旁添加注释 110
- 5.8 在大括号闭合处添加注释 112
- 5.9 在函数内部添加详细介绍函数的注释 115
- 5.10 注释标记原则 116

第 6 章 标识符名称定义相关编码准则 I

- 6.1 系统化定义变量名 118
- 6.2 用匈牙利表示法命名变量 119
- 6.3 用变量名前缀表示变量数据类型 120
- 6.4 用变量名前缀表示变量存储类型 122
- 6.5 用函数名前缀表示函数功能 125
- 6.6 编写个人专属前缀 127

第 7 章 标识符名称定义相关编码准则 II

- 7.1 用有意义的名称命名 130
- 7.2 不要使用相似的变量名 131

- 7.3 在不影响含义的前提下尽可能简短命名 133
- 7.4 用下划线和大小写区分较长变量名 135
- 7.5 变量名不要以下划线开始 136
- 7.6 不要过度使用下划线 137
- 7.7 合理使用大小写命名标识符 139
- 7.8 不要滥用大小写区分 I 141
- 7.9 不要滥用大小写区分 II 142
- 7.10 不能用相同名称同时命名类和变量 143
- 7.11 用大写字母表示变量名中需要强调的部分 144

第 8 章 运算符相关编码准则

- 8.1 恰当应用条件运算符有助于提高可读性 146
- 8.2 不要凭借运算符优先级排列算式 147
- 8.3 指针运算符应该紧接变量名 148
- 8.4 慎选移位运算，多用算术运算 150
- 8.5 不要追求极端效率 151

第 9 章 编写清晰代码所需编码准则

- 9.1 不要投机取巧，应致力于编写清晰易懂的程序 154
- 9.2 切忌混淆 while 语句中关系运算符和赋值运算符的优先级 156
- 9.3 不要进行隐式“非零测试” 158
- 9.4 不要在条件表达式中使用赋值语句 159
- 9.5 避免产生副作用 161
- 9.6 函数原型中也要标注参数的数据类型 164

- 9.7 形式参数也需要命名 166
- 9.8 必须标注返回值的数据类型 168
- 9.9 留意结果值 169
- 9.10 在 for 语句等条件表达式中谨慎运算 171
- 9.11 大量使用冗余括号 172
- 9.12 如果 else 语句使用大括号, 那么 if 语句也应该使用 175
- 9.13 函数末尾务必编写 return 语句 176

第 10 章 编写可移植代码所需编码准则

- 10.1 文件名不超过 14 个字符 178
- 10.2 不要在文件名中使用特殊字符 180
- 10.3 利用条件编译提高可移植性 181
- 10.4 了解编译器的限制 183
- 10.5 需考虑数据类型大小可能变化 185
- 10.6 不要指定绝对路径 186
- 10.7 可移植性和高效性二选一 187
- 10.8 用数组代替指针以提高可移植性 188
- 10.9 选择可移植性更好的编程语言 189
- 10.10 不要插入低级语言编写的代码 190

第 11 章 编写精确代码所需编码准则

- 11.1 计算机并不如想象得那么精确 192
- 11.2 需要进行精确计算时避开浮点数运算 193
- 11.3 double 型比 float 型更适合精确计算 194

- 11.4 确认整数型大小 197
- 11.5 必须明确计算单位 198
- 11.6 特别留意除法运算 200
- 11.7 尽量避免数据类型转换 203
- 11.8 精通编程语言的语法 205
- 11.9 留意可能出现的非线性计算结果 206

第 12 章 提升性能所需编码准则

- 12.1 重视性能, 限制输出 210
- 12.2 用简单形式改写运算表达式 211
- 12.3 需要高效处理大文件时应使用二进制文件 212
- 12.4 了解并使用压缩 / 未压缩结构体优缺点 213
- 12.5 根据运行环境选择编程语言 216
- 12.6 根据情况选择手段 218
- 12.7 选择更优秀的数据结构 219

第 13 章 编写易于理解的代码所需编码准则

- 13.1 不要使用 goto 语句 222
- 13.2 不要替换 C 语言组成要素 224
- 13.3 缩短过长数据类型名称 226
- 13.4 使用 if 语句而非三元运算符 229
- 13.5 数组维数应限制在三维之内 230
- 13.6 考虑驱动函数 main 函数的作用 231
- 13.7 将常量替换为符号常量或 const 形态常量 233

- 13.8 考虑变量声明部分的顺序 234
- 13.9 尽可能不使用全局变量 236
- 13.10 遵循 KISS 原则 237

第 14 章 用户接口处理相关编码准则

- 14.1 确保保存输入值的变量足够大 240
- 14.2 转换说明符和参数个数应保持一致 241
- 14.3 使用 fgets() 和 sscanf() 函数而非 scanf() 函数 243
- 14.4 使用 fflush() 函数清空标准输入 / 输出设备缓冲 245

第 15 章 编写零漏洞代码所需编码准则

- 15.1 数组下标应从 0 开始 252
- 15.2 置换字符串时必须使用括号 254
- 15.3 文件必须有开就有关 255
- 15.4 不要无视编译器的警告错误 259
- 15.5 掌握并在编码时防止运行时错误 260
- 15.6 用静态变量声明大数组 265
- 15.7 预留足够大的存储空间 267
- 15.8 注意信息交换引发的涌现效果 268

第 16 章 提升生产效率所需编码准则

- 16.1 在对立关系中事先确定侧重于哪一方 272
- 16.2 慎重采用最新工具 273
- 16.3 记住所有标准库 274

- 16.4 最大程度划分模块 274
- 16.5 明确区分术语 276
- 16.6 明确区分结构体、枚举体、共用体 277
- 16.7 明确区分概念 278
- 16.8 明确区分对象、类、实例 279

附录

- 参考网站及搜索方法 281
- 主要参考文献 288
- 后记 I: 从“图书出版”角度解读软件开发 289
- 后记 II: 从码农到程序员 297