

## <<易读代码的艺术>>

### 图书基本信息

书名：<<易读代码的艺术>>

13位ISBN编号：9787564134471

10位ISBN编号：756413447X

出版时间：2012-6

出版时间：东南大学出版社

作者：Dustin Boswell , Trevor Foucher

页数：190

字数：250000

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>

## <<易读代码的艺术>>

### 内容概要

作为程序员，我们经常看到一些充满Bug、让我们头痛不已的丑陋代码。在过去五年中，《易读代码的艺术(影印版)》的作者波斯韦尔(Dustin Boswell)和富彻(Trevor Foucher)分析了许多这样的“糟糕代码”(这些代码有很多是出自于他们自己之手)，他们试图厘清为什么这些代码如此糟糕以及如何改进这些代码。他们得出的结论是：你必须写出让他人(这里也包括你自己)花费最少时间能够理解的代码。

## <<易读代码的艺术>>

### 作者简介

Dustin Boswell，毕业于加州理工学院，在谷歌公司从事过五年的Web信息采集基础设施和广告营销计划的研究工作。

他先后构建过多个Web站点，目前的主要研究兴趣在于大数据的处理和机器学习领域的相关技术。

Trevor Foucher，过去十年中先后在微软公司担任Windows及安全相关产品的工程师、经理和技术领导者的职务，现任职于谷歌公司，主要从事谷歌广告营销计划和搜索基础设施的研究工作。

## <<易读代码的艺术>>

### 书籍目录

#### PREFACE

##### CODE SHOULD BE EASY TO UNDERSTAND

What Makes Code "Better"?

The Fundamental Theorem of Readability

Is Smaller Always Better?

Does Time-Till-Understanding Conflict with Other Goals?

The Hard Part

#### Part One SURFACE-LEVEL IMPROVEMENTS

##### PACKING INFORMATION INTO NAMES

Choose Specific Words

Avoid Generic Names Like Imp and retual

Prefer Concrete Names over Abstract Names

Attachin9 Extra Information to a Name

How Long Should a Name Be?

Use Name Formatting to Conuey Meaning

Summary

##### NAMES THAT CAN'T BE MISCONSTRUED

Example: FilterO

Example: Clip(text, length)

Prefer rain and max for (Inclusiue) Limits

Prefer fit and last for Inelusiuie RanRes

Prefer herin and end for.Inclusiue/Exclusiue RanRes

NaminR Boolea

MatchinR Expectatio of Use

Example: EvaluatinR Multiple Name Candidates

Summary

##### AESTHETICS

Why Do Aesthetics Matter?

BearranRe Line Breaks to Be Coistent and Compact

Use Methods to Clean Up IrreRularity

Use Column AliRnment When Helpful

Pick a MeaninRful Order, and Use It Coistently

OrRanize Declaratio into Blocks

Break Code into "Parafgraphs"

Peonal Style ueus Coistency

Summary

##### KNOWING WHAT TO COMMENT

What NOT to Comment

Beordin9 Your Thouyhts

Put Youelf in the Reader's Shoes

Final Thoughts--Getting Over Writer's Block

Summary

##### MAKING COMMENTS PRECISE AND COMPACT

Keep Comments Compact

Avoid Ambiguous Pronou

## <<易读代码的艺术>>

Polish Sloppy Sentences

Describe Function Behavior Precisely

Use Input/Output Examples That Illustrate Corner Cases

State the Intent of Your Code

"Named Function Parameter" Comments

Use Information-Dee Words

Summary

### Part Two SIMPLIFYING LOOPS AND LOGIC

#### MAKING CONTROL FLOW EASY TO READ

The Order of Arguments in Conditionals

The Order of if/else Blocks

The ?: Conditional Expression (a.k.a. "Ternary Operator")

Avoid dogwhile Loops

Returning Early from a Function

The Infamous goto

Minimize Nesting

Can You Follow the Flow of Execution?

Summary

#### BREAKING DOWN GIANT EXPRESSIONS

Explaining Variables

Summary Variables

Using De Morgan's Laws

Abusing Short-Circuit Logic

Example: Wrestling with Complicated Logic

Breaking Down Giant Statements

Another Creative Way to Simplify Expressions

Summary

#### VARIABLES AND READABILITY

Eliminating Variables

Shrink the Scope of Your Variables

Prefer Write-Once Variables

A Final Example

Summary

### Part Three REORGANIZING YOUR CODE

#### EXTRACTING UNRELATED SUBPROBLEMS

Introductory Example: findClosestLocation

Pure Utility Code

Other General-Purpose Code

Create a Lot of General-Purpose Code

Project-Specific Functionality

Simplifying an Existing Interface

Reshaping an Interface to Your Needs

Taking Things Too Far

Summary

#### ONETASK AT A TIME

Tasks Can Be Small

Extracting Values from an Object

## <<易读代码的艺术>>

A Larger Example

Summary

TURNING THOUGHTS INTO CODE

Describing Logic Clearly

Knowing Your Libraries Helps

Applying This Method to Larger Problems

Summary

WRITING LESS CODE

Don't Bother Implementing That Feature--You Won't Need It

Question and Break Down Your Requirements

Keeping Your Codebase Small

Be Familiar with the Libraries Around You

Example: Using Unix Tools Instead of Coding

Summary

Part Four SELECTED TOPICS

TESTING AND READABILITY

Make Tests Easy to Read and Maintain

What's Wrong with This Test?

Making This Test More Readable

Making Error Messages Readable

Choosing Good Test Inputs

Naming Test Functions

What Was Wrong with That Test?

Test-Friendly Development

Going Too Far

Summary

DESIGNING AND IMPLEMENTING A "MINUTE/HOUR COUNTER"

The Problem

Defining the Class Interface

Attempt 1: A Naive Solution

Attempt 2: Conveyor Belt Design

Attempt 3: A Time-Bucketed Design

Comparing the Three Solutions

Summary

FURTHER READING

INDEX

## <<易读代码的艺术>>

### 媒体关注与评论

“意识到你所编写的代码如何影响他人的阅读和理解是软件开发中的一个重要环节。本书的作者通过大量启发式的例子带领读者从多个不同的方面去迎接这一挑战，在这一领域做出了伟大的工作。

” ——Michael Hunger，软件开发人员，Neo Technology

<<易读代码的艺术>>

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>