

<<像程序员一样思考>>

图书基本信息

书名：<<像程序员一样思考>>

13位ISBN编号：9787115316585

10位ISBN编号：7115316589

出版时间：2013-6

出版时间：人民邮电出版社

作者：V. Anton Spraul

译者：徐波

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>

<<像程序员一样思考>>

内容概要

编程的真正挑战不是学习一种语言的语法，而是学习创造性地解决问题，从而构建美妙的应用。

《像程序员一样思考》分析了程序员解决问题的方法，并且教授你其他图书所忽略的一种能力，即如何像程序员一样思考。

全书分为8章。

第1章通过对几个经典的算法问题切入，概括了问题解决的基本技巧和步骤。

第2章通过实际编写C++代码来解决几个简单的问题，从而让读者进一步体会到解决问题的思路和应用。

第3到7章是书中的主体部分，分别探讨了用数组、指针和动态内存、类、递归和代码复用来解决问题的途径和实际应用。

最后，第8章从培养程序员思维的角度，进行了总结和概括，告诉读者如何才能像程序员一样思考。

<<像程序员一样思考>>

作者简介

作者：（美国）V.Anton Spraul 译者：徐波V.Anton Spraul，讲授入门级编程和计算机科学已经超过15年。

本书凝聚了他在多年的开发经历中所提炼的经验和技巧，并在面向许多遭遇瓶颈的程序员的一对一指导中收到了良好的效果。

他还是《Computer Science Made Simple》（Broadway）的作者。

徐波，浙江宁波人，熟悉C和C++、Java等编程语言。

2002年开始从事计算机技术图书翻译。

徐波技术视野广阔，翻译文笔优美。

译著有《C专家编程》、《C和指针》等。

<<像程序员一样思考>>

书籍目录

目 录	第1章 解决问题的策略	11.1 经典难题	21.1.1 狐狸、鹅和玉米	31.1.2 瓷砖滑块问题
	7.1.1.3 数独	11.1.1.4 Quarrasi锁	13.1.2 基本的问题解决技巧	16.1.2.1 总是要制订计划
	1.2.2 重新陈述问题	17.1.2.3 划分问题	18.1.2.4 从自己所知的开始	19.1.2.5 削减问题
	20.1.2.6 寻找类比	21.1.2.7 试验	21.1.2.8 避免陷入挫折感	22.1.3 习题
	23	第2章 纯粹的难题	25.2.1 本章所使用的C++简述	25.2.2 输出图案
	26.2.3 输入处理	31.2.4 追踪状态	42.2.5 结论	55
	2.6 习题	55	第3章 用数组解决问题	59.3.1 数组基础知识概述
	60.3.2 用数组解决问题	66.3.3 固定数据的数组	71.3.4 非标量数组	73.3.5 多维数组
	75.3.6 决定什么时候使用数组	78.3.7 习题	82	第4章 用指针和动态内存解决问题
	85.4.1 指针基础知识回顾	86.4.2 指针的优点	87	4.2.1 运行时确定长度的数据结构
	87.4.2.2 可改变长度的数据结构	87.4.2.3 内存共享	88.4.3 什么时候使用指针	89.4.4 内存细节
	90.4.4.1 堆栈和堆	90.4.4.2 内存的大小	93.4.4.3 生命期	94
	4.5 解决指针问题	95.4.5.1 可变长度的字符串	95.4.5.2 链表	105.4.6 结论和未来的步骤
	113	4.7 习题	114	第5章 用类解决问题
	117.5.1 类的基础知识回顾	118.5.2 使用类的目的	119.5.2.1 封装	120.5.2.2 代码的复用
	120.5.2.3 问题的细分	121.5.2.4 信息隐藏	121.5.2.5 可读性	123
	5.2.6 表达能力	123.5.3 创建一个简单的类	124.5.3.1 问题：班级花名册	124.5.3.2 基本的类框架
	125.5.3.3 支持方法	129.5.4 具有动态数据的类	132.5.5 需要避免的错误	147.5.5.1 假类
	147.5.5.2 单功能	148.5.6 习题	148	第6章 用递归解决问题
	151.6.1 递归基础知识回顾	151	6.2 头递归和尾递归	152.6.3 大递归思路
	160.6.4 常见的错误	163.6.4.1 过多的参数	164.6.4.2 全局变量	165.6.5 把递归应用于动态数据结构
	166.6.5.1 递归和链表	167.6.5.2 递归和二叉树	169.6.6 包装器函数	172.6.7 什么时候选择递归
	175.6.8 习题	179	第7章 通过代码复用解决问题	181.7.1 良好的复用和不良的复用
	182.7.2 组件基础知识回顾	183.7.3 创建组件的基础知识	186.7.3.1 探索式学习	186.7.3.2 根据需要学习
	190.7.4 选择组件类型	198.7.5 习题	204	第8章 培养程序员的思维
	207.8.1 创建自己的总体计划	207.8.1.1 扬长避短	208.8.1.2 制订总体计划	214.8.2 处理任何问题
	215.8.2.1 问题：绞型者作弊程序	216.8.2.2 寻找作弊方法	217.8.2.3 绞型者作弊所需要的操作	218.8.2.4 初始设计
	220.8.2.5 开始编写代码	221.8.2.6 对初始结果的分析	229.8.2.7 解决问题的艺术	230.8.3 学习新的编程技能
	231.8.3.1 新语言	231.8.3.2 已经熟悉的语言的新技巧	234.8.3.3 新代码库	235.8.3.4 上课
	235.8.4 结论	236.8.5 习题	237	

<<像程序员一样思考>>

章节摘录

版权页：插图：读者的一个想法可能是只要把所有的数据都声明为私有数据，并花费足够的时间设计成员函数列表以及它们的形参列表使其永远不需要修改，就可以确保信息隐藏。

虽然这些对于信息隐藏而言确实是必须的，但并不充分，因为信息隐藏问题可能更为微妙。

记住，类同意不更改它的任何方法的功能，而不管具体是什么情况。

在以前的章节中，我们必须确定一个函数将要处理的最小情况或者在遇到一种异常情况下该怎么办，例如在寻找一个数组的平均数时发现数组的长度为零时该怎么办。

即使是为了适应一种异常的情况而更改方法的结果都会导致接口的改变，我们应该予以避免。

这也是为什么在编程中明确考虑特殊情况如此重要的另一个原因。

许多时候，当一个程序的底层技术或应用程序编程接口（API）被更新时会出现问题，在过去有些会很可靠地返回-1的系统调用现在很可能出错，而返回一个看上去随机的、但仍然是负数的值。

避免这个问题的最好办法之一是在类或方法文档中陈述特殊情况的结果。

如果文档表示当一个特定的情况发生时将会返回-1这个错误代码，就应该重新考虑让自己的方法返回同样的值。

那么，信息隐藏是怎样影响解决问题的呢？

信息隐藏的原则告诉程序员在编写客户代码的时候就要把类的实现细节搁在一边。

或者更广泛地说，只在编写一个特定类的内部细节的时候才关注它的实现。

把实现细节暂时抛诸脑后，才能消除思维的分心，把注意力集中在解决当前问题上。

但是，把信息隐藏与问题解决相关联时，我们应该注意它的限制。

有时候，实现细节并不会对客户代码产生影响。

在前面的章节中，我们已经看到了一些基于数组的数据结构和一些基于指针的数据结构的优点和弱点。

基于数组的数据结构允许随机访问，但是无法轻易地增长或收缩。

反之，基于指针的数据结构只提供了线性访问，但是在添加或删除其中的一些元素时并不需要重新创建整个结构。

因此，以基于数组的结构为基础所创建的类的特性与那些以基于指针的结构为基础所创建的类不同。

在计算机科学中，我们常常讨论抽象数据类型的概念，这是最纯粹形式的信息隐藏：一种数据类型只是由它的操作所定义的。

在第4章，我们讨论了堆栈的概念，并描述了程序的堆栈是一块连续的内存。

但是，作为抽象数据类型，堆栈可以是任何能够添加并删除单独数据项的数据类型，并且这些数据项是按照与添加相反的顺序被删除的。

这称为后入先出的顺序（LIFO）。

没有任何硬性规定堆栈必须是一块连续的内存，用链表创建堆栈也是完全可以的。

由于连续的内存和链表具有不同的属性，因此使用某种方法实现的堆栈可能与使用其他方法实现的堆栈具有不同的属性，也许会对使用堆栈的客户产生巨大的差别。

<<像程序员一样思考>>

编辑推荐

《像程序员一样思考》编辑推荐：学会像程序员一样思考，构建创新性问题解决方

案。
《像程序员一样思考》习题极其类似Google和Facebook的面试题，对于软件工程师面试有非常大的参考价值。

《像程序员一样思考》选取的话题切中程序员的痛点，针对他们最容易陷入挣扎的领域展开讨论，引发思考。

每章后面都给出一些编程习题，使得读者能够应用该章所讨论的概念，训练和提升问题解决的能力。

《像程序员一样思考》适合初级到中级的程序员用来提升自己的问题解决能力和应用编程技能的能力，也适合计算机相关专业的学生作为参考书阅读。

<<像程序员一样思考>>

名人推荐

本书所提供的习题类似于我在接受Google和Facebook的软件工程师面试时所遇到的问题，因此对于打算通过面试寻找新工作的专业程序员，本书是极好的复习材料。

——Ariane Coffin，Wired网站的GeekMom作者在向初学者阐述难懂的概念方面具有广博的知识和丰富的经验。

本书显示了他脚踏实地、一丝不苟却又令人愉悦的写作风格。

——Adrian Woodhead，Slashdot这是我阅读后收获最大的书籍之一，因为它指导我们设计一个属于自己的系统，而不是把思维固化为只能采取一种正确的方法才能达到目的。

——Lucas Westermann，Full Circle杂志如果你能够认真研读本书，我保证它可以极大地拓展你的思维。

——David Bolton，About.com C/C++/C#不管使用什么教材向新学生讲授编程和程序逻辑，我都建议一定要把本书作为重要的参考书。

——Joe Saur，The ACM's Software Engineering Notes杂志作者所提供的建议简单、直观并且实用。

本书的阅读是一个既轻松又极有价值的过程。

——James Powell，Enterprise Systems对于所有想要培养创造性的解决问题能力的人以及已经学习了编程但觉得没有完全理解概念的人，我向他们强烈推荐本书。

——Robert Perkins，Game Vortex如果我教其他人学习编程，这肯定是我选择的教材。

——Stephen Chapman，Ask Felgall

<<像程序员一样思考>>

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>