

程序设计实践

The Practice of Programming

课程简介

你认为一个**好的**程序应该是什么样的？

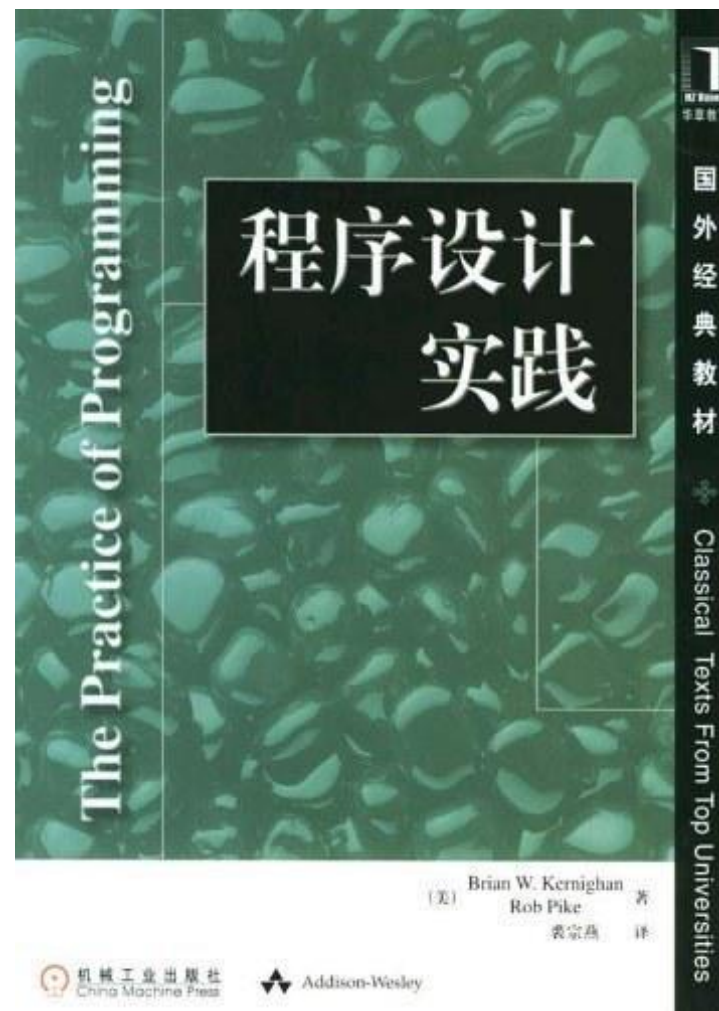
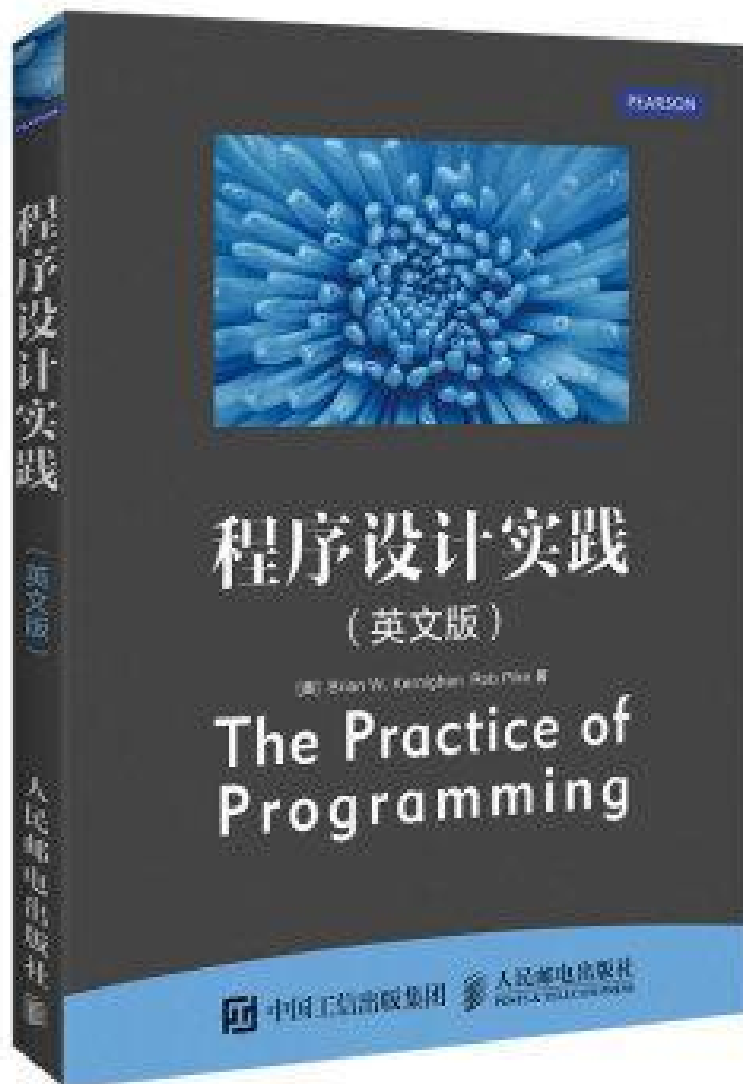


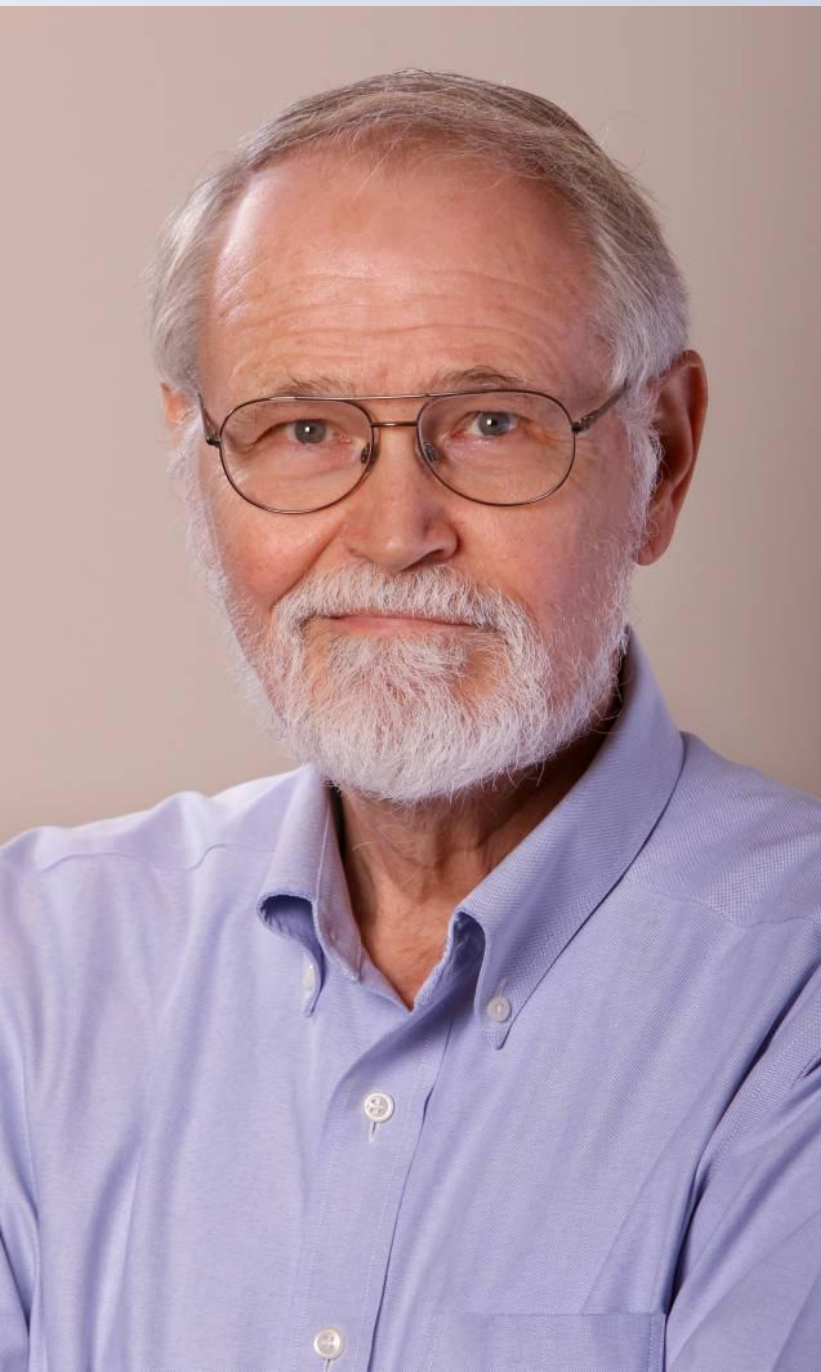
课程目标

《程序设计实践》课程主要介绍一些程序设计的实践方法，面向的是已经先修了程序设计语言，算法，数据结构等基础课程之后的学生。

《程序设计实践》课程从软件开发的完整过程的角度审视程序设计中需要注意的方方面面的问题，了解如何写出不但运行正确，而且可读性好，设计优雅，接口清晰，运行健壮，跨平台，高性能的程序。

学生在《程序设计实践》课程作业中能够亲身实践课程讲授的内容，从而理解这些方法的意义，让这些方法能够对以后的毕业设计，研究生阶段的科研工作，或者毕业后的工作岗位上从事的编程工作有所帮助。

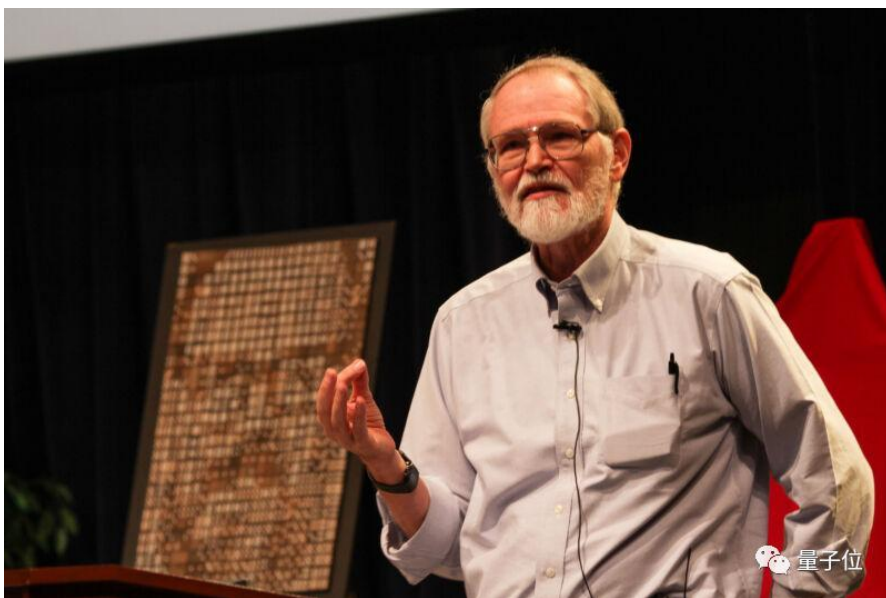
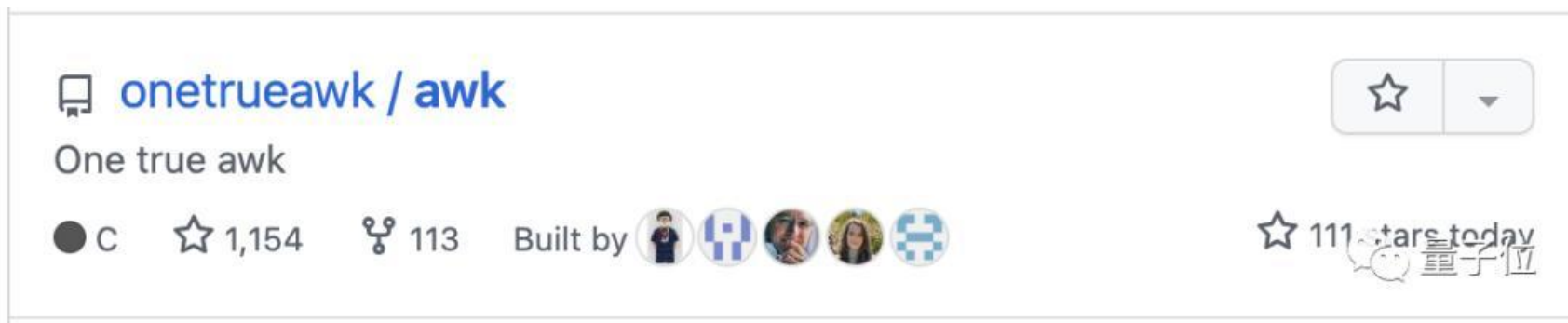




Brian W. Kernighan

- C语言，UNIX初始参与者
- awk文本分析工具作者之一

80岁还嗖嗖改代码，他是Unix命名人，发明“Hello World”，他说解决问题全靠拖



2022年8月22日的一则新闻，布莱恩·柯林汉，AWK的GitHub2022库中，他提交了新的pull request，给AWK添加了一直没法做到的Unicode支持。



Rob Pike

- UTF-8编码发明者之一
- go语言之父



裘宗燕

- 教材中文版译者
- 北大数学院教授

知乎

首发于
极光日报



Hevin

编程话题下的优秀答主

+ 关注他

37 人赞同了该文章

The Practice of Programming，中文译为《程序设计实践》，英文版最早出版于 1999 年。

书中内容涵盖了排错、测试、性能、可移植性、设计、界面、风格和记法等方面，讨论了程序设计中实际的、又是非常深刻和具有广泛意义的思想、技术和方法。

这本书的作者 Brian W.Kernighan和 Rob Pike 在贝尔实验室的计算机科学研究中心工作。Brian Kernighan 是 Addison-Wesley 的“专业计算丛书”顾问编辑，也是《C程序设计语言》的合著者之一。Rob Pike 则是 Plan 9 和 Inferno 操作系统的主要结构设计与实现者。

作者工作这么多年，在不同的城市兜兜转转，就这本 The Practice of Programming 一直带着。

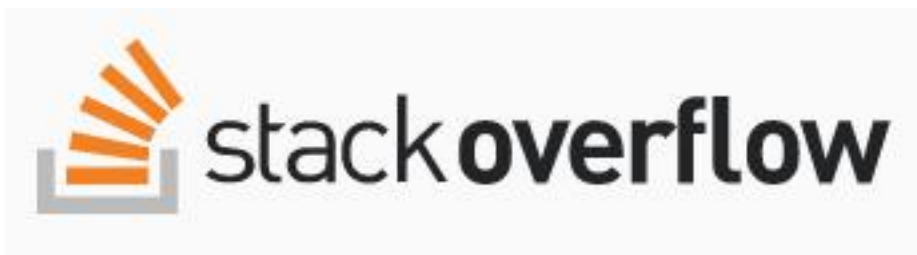
但是如今距离这本书最早出版已经 18 年了，又是什么让作者在现在还想要重读呢？作者就在文中回顾了自己觉得书中经久不衰的一些内容，比如测试和 debug 的一些方法与思想。

如果你还没有看过这本书，又觉得没有时间或嫌太老没必要看的话，不妨就看看这篇文章吧。：)

原文：[The Practice of Programming: 18 Years Later](#)

发布于 2017-01-24

学习工具



最著名的软件开发技术问答网站

- 同学们能遇到的绝大部分BUG的解决方法都能找到
- 使用方法 <https://stackoverflow.com/tour>



CSDN：专业开发者社区

- 软件开发的大部分知识解读
- 使用方法 <https://www.csdn.net>

学习工具



开源软件社区

- 学习优秀的软件源码，识别和分析差的代码
- 码云gitee: <https://gitee.com/explore>
- GitHub中文社区: <https://www.githubs.cn/>

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

从命名，表达式，语句，段落，宏定义，常量，注释等几个方面，讨论什么样的风格会让程序的可读性易维护性更好，更加不容易出错。

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

如何根据项目需求选择数据结构，
算法？

让程序设计有一个良好的开端。

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

接口分为程序与程序接口以及人与程序接口两个方面。接口为什么重要？如何设计良好的接口？

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

程序的哪些地方最容易出错？

排错的工具，方法？

排错的基本流程是什么？

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

如何设计测试驱动和测试桩？

如何设计自动测试脚本？

如何进行压力测试？

如何进行性能测试？

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

影响性能的主要原因是什么？

如何寻找性能瓶颈？

如何优化提高性能？

性能分析的工具，方法。

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

当一个程序面对不同的服务器厂商，不同的操作系统，不同的编译器，不同语言的客户，应该注意哪些问题？

课程内容

风格

设计与实现

接口

排错

测试

性能

可移植性

记法

记法为什么很重要？我们应该
如何看待已有的记法？必要的
时候是否应该设计自己的记法？
记法解释执行的原理？

课程安排

共**16**周，**16**次课

10周，课堂讲授

6周，程序设计实验以及大作业验收

学习方法

- 课堂讲授

任课老师课上讲解相关知识，占20学时（10次课）

- 课下自学

学生以课上知识为基础，通过老师推荐的参考资料和学习工具，查阅资料和动手实践，从而加深理解知识点和扩展知识面

- 实验

学生利用上机时间完成实验作业，占12学时

- QQ群

QQ群(？)

考核方式

本课程的考核采取平时成绩+大作业的方式。

成绩组成：

✓ 平时成绩

- 通过小实验作业考核
- 占成绩30%

✓ 大作业

- 代码审查、功能验证、技术提问、文档检查
- 占成绩70%

✓ 作业提交

- 作业提交至“教学云平台”
<https://uccloud.bupt.edu.cn/>

大作业

程序题目： 基于领域特定语言的客服机器人设计与实现

描述： 领域特定语言（Domain Specific Language, DSL）可以提供一种相对简单的文法，用于特定领域的业务流程定制。本作业要求定义一个领域特定脚本语言，这个语言能够描述在线客服机器人（机器人客服是目前提升客服效率的重要技术，在银行、通信和商务等领域的复杂信息系统中有着广泛的应用）的自动应答逻辑，并设计实现一个解释器解释执行这个脚本，可以根据用户的不同输入，根据脚本的逻辑设计给出相应的应答。

基本要求：

- 脚本语言的语法可以自由定义，只要语义上满足描述客服机器人自动应答逻辑的要求。
- 程序输入输出形式不限，可以简化为纯命令行界面。
- 应该给出几种不同的脚本范例，对不同脚本范例解释器执行之后会有不同的行为表现。

程序题目：基于领域特定脚本语言的客服机器人的设计与实现

作业描述：

领域特定语言（**Domain Specific Language, DSL**）可以提供一种相对简单的文法，用于特定领域的业务流程定制。本作业要求定义一个领域特定脚本语言，这个语言能够描述在线客服机器人（机器人客服是目前提升客服效率的重要技术，在银行、通信和商务等领域的复杂信息系统中有着广泛的应用）的自动应答逻辑，并设计实现一个解释器解释执行这个脚本，可以根据用户的不同输入，根据脚本的逻辑设计给出相应的应答。

基本要求：

脚本语言的语法可以自由定义，只要语义上满足描述客服机器人自动应答逻辑的要求。程序输入输出形式不限，可以简化为纯命令行界面。

应该给出几种不同的脚本范例，对不同脚本范例解释器执行之后会有不同的行为表现。

评分标准：

本作业考察学生规范编写代码、合理设计程序、解决工程问题等方面的综合能力。满分**100分**，具体如下：

风格：满分**15分**，其中代码注释**6分**，命名**6分**，其它**3分**。

设计和实现：满分**30分**，其中数据结构**7分**，模块划分**7分**，功能**8分**，文档**8分**。

接口：满分**15分**，其中程序间接口**8分**，人机接口**7分**。

测试：满分**30分**，测试桩**15分**，自动测试脚本**15分**

记法：满分**10分**，文档中对此脚本语言的语法的准确描述。

注意：抄袭或有意被抄袭均为**0分**

大作业

本作业考察学生规范编写代码、合理设计程序、解决工程问题等方面的综合能力。满分100分，具体评分标准如下：

风格：

满分15分，其中代码注释6分，命名6分，其它3分。

设计和实现：

满分30分，其中数据结构7分，模块划分7分，功能8分，文档8分。

接口：

满分15分，其中程序间接口8分，人机接口7分。

测试：

满分30分，测试桩15分，自动测试脚本15分

记法：

满分10分，文档中对此脚本语言的语法的准确描述。

注意：抄袭或有意被抄袭均为0分

大作业

● 大作业早验收加分奖励办法：

- 实验阶段第1周验收：奖励5分
- 实验阶段第2周验收：奖励4分
- 实验阶段第3周验收：奖励3分
- 实验阶段第4周验收：奖励2分
- 实验阶段第5周验收：奖励1分
- 验收分数和奖励分数相加为实际大作业分数，不超过100分