第一图书网, tushu007.com

<<编程之魂>>

图书基本信息

书名：<<编程之魂>>

13位ISBN编号：9787121104985

10位ISBN编号：7121104989

出版时间：2010-04

出版时间：电子工业出版社

作者：Federico Biancuzzi,Shane Warden

页数：376

译者：闫怀志

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

前言

编写软件是件难事——最起码，编出来的软件如果要经得住测试、时间和不同环境的考验，确实很难。
在过去的50多年里，为了让编写软件变得更容易一些，不仅软件工程领域为此在不懈努力，编程语言也被赋以重任。
但是，真正的困难究竟是什么呢？
大多数书籍和论文在回答这个问题时，都将焦点集中在软件体系结构、需求之类的话题上。
不过，如果困难在于编写程序本身，又会怎么样呢？
换句话说，如果我们把自己当成是更具交流（语言）色彩的程序员，而不是更具工程色彩的程序员，又该如何呢？
小孩子两三岁学说话，五六岁学读书写字。
我从来没见过哪个大作家是成年以后才学习读书和写字的。

内容概要

《编程之魂：与27位编程语言创始人对话》是27位杰出的设计师与你分享他们的智慧和经验。书中以问答方式告诉为什么要创建某种编程语言、它在技术上如何开发、如何教授和学习，以及它如何顺应时代发展等。
你会发现构建成功编程语言所需的思想和步骤，它广受欢迎的原因，以及如何处理程序员常见的问题。
因此，如果你想深入学习设计成功编程语言的思想，《编程之魂：与27位编程语言创始人对话》会对你大有帮助。

作者简介

Federico Biancuzzi，是位自由职业采访者（freelance interviewer），他的采访在ONLamp、NewsForge、TheRegister、ArsTechniea等很多网站上在线出版。
Shane Warden，是位对编程语言设计和虚拟机感兴趣的自由软件开发者。
他在业余时间经营独立出版商Onyx Neon。
Press的小说分部。
他是《T'he Art 0f Agile：Development》（O ’ Reilly）的合著者。

书籍目录

推荐序 前言1 C++ Bjarne Stroustrup1.1 设计决策 1.2 使用语言 1.3 OOP和并发 1.4 关于未来 1.5 有关教学 2 Python Guido van Rossum2.1 Python方式 2.2 优秀的程序员 2.3 多种Python 2.4 权宜之计和经验 3 APL Adin Falkoff3.1 纸笔方式 3.2 基本原理 3.3 并行 3.4 遗留 4 Forth Chuck Moore4.1 Forth语言与语言设计 4.2 硬件 4.3 应用程序设计 5 BASIC Tom Kurtz5.1 BASIC背后的目标 5.2 编译器设计 5.3 语言和编程实践 5.4 语言设计 5.5 工作目标 6 AWK Al Aho6.1 算法生命周期 6.2 语言设计 6.3 Unix及其文化 6.4 文档的作用 6.5 计算机科学 6.6 培育小语言 6.7 设计一种新语言 6.8 遗留文化 6.9 变革性技术 6.10 改变世界的“ 位 ” 6.11 理论和实践 6.12 等待突破 6.13 通过实例来编程 7 Lua Luiz Henrique de Figueiredo and Roberto Ierusalimschy7.1 脚本的功能 7.2 经验 7.3 语言设计 8 Haskell Simon Peyton Jones, Paul Hudak, Philip Wadler, and John Hughes 8.1 功能性团队 8.2 函数式编程之路 8.3 Haskell语言 8.4 传播（函数式）教育 8.5 形式体系和发展 9 ML Robin Milner9.1 可靠性定理 9.2 意义理论（译注） 9.3 超越信息学 10 SQL Don Chamberlin10.1 一篇开创性的有重大影响的根本性的论文 10.2 语言 10.3 反馈和演进 10.4 XQuery和XML 11 Objective-C Brad Cox and Tom Love11.1 Objective-C工程 11.2 培育一种语言 11.3 教育和培训 11.4 项目管理和遗留软件 11.5 Objective-C和其它语言 11.6 组件、沙子和砖 11.7 作为经济现象的质量 11.8 教育 12 Java James Gosling12.1 功能或者简单性 12.2 品味的问题 12.3 并发性 12.4 设计一种语言 12.5 反馈循环 13 C# Anders Hejlsberg13.1 语言和设计 13.2 培育一种语言 13.3 C# 23813.4 计算机科学的未来 14 UML Ivar Jacobson, James Rumbaugh, and Grady Booch14.1 学习和教学 14.2 人们的角色 14.3 UML 14.4 知识 14.5 作好变革准备 14.6 使用UML 14.7 层和语言 14.8 一点可复用性 14.9 对称关系 14.10 UML 14.11 语言设计 14.12 培训开发者 14.13 创新、改进和模式 15 Perl Larry Wall15.1 革命性的语言 15.2 语言 15.3 社区 15.4 改进和革命 16 PostScript Charles Geschke and John Warnock16.1 为永恒而设计 16.2 研究和教育 16.3 长寿命接口 16.4 标准愿望 17 Eiffel Bertrand Meyer17.1 一个充满灵感的下午 17.2 可复用性和泛型 17.3 校对语言 17.4 管理成长和演进 后记 受访嘉宾 索引

章节摘录

2 Python Guido van Rossum2.1 Python方式 开发编程语言和开发“ 普通的 ”软件项目有什么区别？
Guido van Rossum：除了与大多数软件项目打交道以外，你最重要的用户还是程序员自己。
这给语言项目提供了高级别的“ 元 ”内容。
在软件项目的依存关系树中，编程语言差不多处于最底层：其他的所有东西都依赖于一种或多种语言。
这也使得很难修改一种语言：不兼容的修改会影响很多“ 依存物 ”，以至于这样做通常并不可行。
换句话说，一旦发布之后，所有的错误都会固定不变。
C++很可能就是这种情况最极端的例子，它必須满足兼容性的需求，事实上，这种需求也许会要求20年前编写的代码仍然有效。
您如何调试一种语言呢？
Guido：你不用调试它。
在语言设计领域中，敏捷开发方法学并没有什么意义：在该语言稳定之前，很少有人会用它，而且你在语言定义中找不着bug，直到你已经有很多的用户为止，而此时你要修改已经为时太晚。
当然，实现中可能会有很多内容需要调试，比如所有老程序，不过语言设计自身大多要求提前仔细设计，因为bug的成本极高。
对于应该何时将某个特性扩展到库中，或者它何时需要核心语言的支持．您是如何做出决定的？
Guido：从历史来看，我对此已经有一个很好的答案。
我很早就注意到，每个人都想把自己喜欢的特性添加到语言中，而且大多数人都对语言设计相对没有经验。
每个人都在提议“ 把这个加到语言中 ”，“ 加一条实现某种功能的语句 ”在很多情况下，答案是“ 好的，通过编写这样的两、三行代码，你就可以实现那种功能或者极为类似的功能，而且它也没那么难 ”。
你可以使用词典，也可以将列表、元组数组和正则表达式组合起来，或者是编写一些元类——等等这一切事情。
我对这个问题的最初答案甚至可能是来自Linus，他好像也有类似的理念。

后记

只有一个词能形容我从这次采访中获得的感受——狂热。
每一位受访嘉宾都会给出你所期望的回报——深层次知识、历史性发现及实践洞察力——不过正是他们对于语言设计、实现与发展的狂热方才显示出了巨大的感染力。
例如，Anders Hejlsberg和James Gosling再次唤起了我对c#和Java的兴趣。
Chuck Moore和Adin Falkoff说服我去研究Forth和APL，而这两种语言在我出生前就已经发明出来了。
AIAho通过描述他的编译器类来诱惑我。
我们采访的每个人都给我提供了很多想法，我真希望有时间来研究它们！
承蒙各位帮助，我对此感激不尽，不仅仅是因为你们给予了我和Federico时间来采访，还因为你们开辟了许多丰富多彩的创新领域。
我从这次经历中获得的最佳经验是： 永远不要低估设计或实现简单性的价值。
人们可以一直增加复杂性。
而大师会力图消除复杂性。
充满热情来努力满足你的求知欲。
很多最佳的发明创新和发现都是在正确的时间正确的位置追求正确的答案时完成的。
了解一个领域的过去和现在。
每一位受访嘉宾都是和其他聪明的、努力工作的人们一起工作的。
我们的领域取决于这种信息共享。
语言可能会持续不断地修改，不过这些宗师们面临的问题仍然会困扰我们——而他们的答案仍然适用。
诸如如何维护软件？
如何找到一个问题的最佳解决方案？
如何令用户惊奇并赞赏？
如何在要处理不可避免的修改要求而又不能中断必须继续工作的情况下获得解决方案？
这次采访对这些问题提供了很好的答案。
我希望本书在你自己寻找灵感时会对你有所帮助。
——Shane Warden

编辑推荐

《编程之魂:与27位编程语言创始人对话》：Adin D.Falkoff：APL；Thomas E.Kurtz：BASlC；Charles H.Moore：FORTH；Robin Milner：ML.；Donald D.Chamberlin：SQL；Alfred Aho.Peter Weinberger DBrian Kernighan：AWK；Charles Geschke和lJohn Warnock：PostScript；Biarne Stroustrup：C++；Bertrand Meyer：Eiffel；Brad Cox and T0m Love：Objective · C；Larrv Wall：Perl；Simon Peyton Jones，Paul Hudak，Philip Wadlerl：]John Hughes：Haskell；Guido van Rossum：python；Luiz Henrique de Figueiredot和Roberto lerusalimschy：Lua；James Goslinq：Java；Grady Booch.Ivar Jacobson31：1James Rumbaugh：UML；Anders Hejlsberg：Delphi的发明者和C#的主要开发者。
《编程之魂》采访了数位极具影响力的编程语言创建者。
在这本独一无二的采访集中，您会了解具体设计决策的过程，包括这些前辈必须作出的折中平衡，以及他们的经历对于今天编程的影响。
受访嘉宾包括：如果您对那些具有远见卓识并为计算机行业的发展殚精竭虑的人感兴趣，您会发现《编程之魂》有着无穷的魅力。

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