

# 《Haskell函数式编程入门》

## 图书基本信息

书名：《Haskell函数式编程入门》

13位ISBN编号：9787115338013

10位ISBN编号：7115338019

出版时间：2014-3-5

出版社：人民邮电出版社

作者：张淞

页数：359

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《Haskell函数式编程入门》

## 内容概要

《haskell函数式编程入门》是一本讲解haskell这门经过精心设计和锤炼的纯函数式编程语言的书，同时也是一本通过haskell来讲解函数式编程的方法与思想的书。全书共分三个部分。第一部分介绍函数式编程在解决数学与算法问题的精简与直观的特色，让不熟悉haskell的读者对其建立初步的了解，同时通过解决一些算法问题，如斐波那契数列、八皇后问题、排序问题、24点等，引发一些对函数式编程方式的思考；第二部分介绍一些略微深入的haskell内容，包括函子、monoid、io与monad转换器等；最后一部分则涉及快速测试、惰性求值和并行编程等主题。

《haskell函数式编程入门》既适合对haskell和函数式编程感兴趣的程序员阅读，又适合作为haskell语言入门教程，供计算机科学与数学专业的学生参考。

# 《Haskell函数式编程入门》

## 作者简介

张淞 1989 年 1 月出生于黑龙江省绥芬河市，酷爱数学、物理，进入大学后开始喜欢学习与研究各类编程语言，并成为了一名 Haskell 爱好者。2012 年 7 月于英国诺丁汉大学获得计算机科学本科学位。2013 年 11 月于英国牛津大学获得计算机科学硕士学位。目前喜欢学习与研究 演算、类型系统、抽象代数、范畴论、逻辑证明、组合数学等内容。

## 书籍目录

|                             |    |
|-----------------------------|----|
| 《haskell函数式编程入门》            |    |
| 第1章 haskell简介               | 1  |
| 1.1 haskell的由来              | 1  |
| 1.2 haskell编译器的安装以及 编写环境    | 3  |
| 1.3 ghci的使用                 | 4  |
| 1.3.1 ghci中的命令              | 5  |
| 1.3.2 在ghci中调用函数            | 5  |
| 1.4 .hs和.lhs文件、注释与库函数       | 7  |
| 1.5 第一个haskell程序helloworld! | 7  |
| 本章小结                        | 8  |
| 第2章 类型系统和函数                 | 9  |
| 2.1 haskell的类型与数据           | 9  |
| 2.1.1 haskell常用数据类型         | 9  |
| 2.1.2 函数类型                  | 14 |
| 2.1.3 类型的别名                 | 17 |
| 2.1.4 类型的重要性                | 18 |
| 2.2 haskell中的类型类            | 19 |
| 2.2.1 相等类型类：eq              | 20 |
| 2.2.2 有序类型类：ord             | 20 |
| 2.2.3 枚举类型类：enum            | 21 |
| 2.2.4 有界类型类：bounded         | 21 |
| 2.2.5 数字类型类：num             | 22 |
| 2.2.6 可显示类型类：show           | 25 |
| 2.2.7 小结                    | 25 |
| 2.3 haskell中的函数             | 26 |
| 2.3.1 haskell中的值            | 26 |
| 2.3.2 函数思想入门                | 27 |
| 2.3.3 函数的基本定义格式             | 28 |
| 2.3.4 表达式                   | 30 |
| 2.3.5 参数的绑定                 | 34 |
| 2.4 haskell中的表达式            | 35 |
| 2.4.1 条件表达式                 | 35 |
| 2.4.2 情况分析表达式               | 36 |
| 2.4.3 守卫表达式                 | 37 |
| 2.4.4 模式匹配                  | 37 |
| 2.4.5 运算符与函数                | 38 |
| 2.4.6 运算符与自定义运算符            | 38 |
| 本章小结                        | 41 |
| 第3章 基于布尔值的函数                | 42 |
| 3.1 关键字module与import简介      | 42 |
| 3.2 简易布尔值的函数                | 43 |
| 3.3 与非门和或非门                 | 46 |
| 本章小结                        | 47 |
| 第4章 库函数及其应用                 | 48 |
| 4.1 预加载库函数                  | 48 |
| 4.1.1 常用函数                  | 48 |
| 4.1.2 基于列表的函数               | 50 |

|        |                       |     |
|--------|-----------------------|-----|
| 4.1.3  | 定义历法公式                | 57  |
| 4.1.4  | 字符串处理的函数              | 58  |
| 4.2    | 字符与位函数库简介             | 60  |
| 4.2.1  | data.char             | 60  |
| 4.2.2  | data.bits             | 60  |
|        | 本章小结                  | 61  |
|        | 第5章 递归函数              | 62  |
| 5.1    | 递归函数的概念               | 62  |
| 5.2    | 简单递归函数                | 64  |
| 5.3    | 扩展递归与尾递归              | 66  |
| 5.4    | 互调递归                  | 68  |
| 5.5    | 麦卡锡的91函数              | 69  |
| 5.6    | 斐波那契数列                | 69  |
| 5.7    | 十进制数字转成罗马数字           | 73  |
| 5.8    | 二分法查找                 | 74  |
| 5.9    | 汉诺塔                   | 75  |
| 5.10   | 排序算法                  | 78  |
| 5.10.1 | 插入排序                  | 78  |
| 5.10.2 | 冒泡排序                  | 81  |
| 5.10.3 | 选择排序                  | 83  |
| 5.10.4 | 快速排序                  | 84  |
| 5.10.5 | 归并排序                  | 86  |
|        | 小结                    | 91  |
| 5.11   | 递归基本条件与程序终止           | 91  |
| 5.12   | 递归与不动点                | 92  |
| 5.13   | 无基本条件递归和惰性求值          | 94  |
|        | 本章小结                  | 96  |
|        | 第6章 列表内包              | 97  |
| 6.1    | 列表生成器                 | 97  |
| 6.2    | 素数相关趣题                | 99  |
| 6.3    | 凯撒加密                  | 101 |
| 6.3.1  | 加密                    | 102 |
| 6.3.2  | 解密                    | 102 |
| 6.4    | 排列与组合问题               | 104 |
| 6.4.1  | 排列问题                  | 104 |
| 6.4.2  | 错位排列问题                | 105 |
| 6.4.3  | 组合问题                  | 106 |
| 6.5    | 八皇后问题                 | 107 |
| 6.6    | 计算矩阵乘法                | 111 |
| 6.7    | 最短路径算法与矩阵乘法           | 112 |
|        | 本章小结                  | 116 |
|        | 第7章 高阶函数与复合函数         | 117 |
| 7.1    | 简单高阶函数                | 117 |
| 7.2    | 折叠函数foldr与foldl       | 119 |
| 7.3    | mapaccuml与mapaccumr函数 | 125 |
| 7.4    | 复合函数                  | 126 |
|        | 本章小结                  | 128 |
|        | 第8章 定义数据类型            | 129 |
| 8.1    | 数据类型的定义               | 129 |

|        |                              |     |
|--------|------------------------------|-----|
| 8.1.1  | 枚举类型                         | 129 |
| 8.1.2  | 构造类型                         | 132 |
| 8.1.3  | 参数化类型                        | 134 |
| 8.1.4  | 递归类型                         | 138 |
| 8.1.5  | 杂合定义类型                       | 140 |
| 8.2    | 类型的同构                        | 142 |
| 8.3    | 使用newtype定义类型                | 146 |
| 8.4    | 数学归纳法的有效性                    | 148 |
| 8.5    | 树                            | 150 |
| 8.6    | 卡特兰数问题                       | 151 |
| 8.7    | 霍夫曼编码                        | 152 |
| 8.8    | 解24点                         | 154 |
| 8.9    | zipper                       | 157 |
| 8.10   | 一般化的代数数据类型                   | 159 |
| 8.11   | 类型的kind                      | 162 |
| 8.11.1 | 类型的kind                      | 162 |
| 8.11.2 | 空类型的声明                       | 164 |
|        | 本章小结                         | 165 |
|        | 第9章 定义类型类                    | 166 |
| 9.1    | 定义类型类                        | 166 |
| 9.2    | haskell中常见类型类                | 169 |
| 9.2.1  | 常用类型类                        | 169 |
| 9.2.2  | functor                      | 171 |
| 9.2.3  | applicative                  | 173 |
| 9.2.4  | alternative                  | 177 |
| 9.2.5  | 简易字符识别器                      | 179 |
| 9.2.6  | read类型类                      | 182 |
| 9.2.7  | 单位半群(monoid)                 | 182 |
| 9.2.8  | foldable与monoid类型类           | 184 |
| 9.2.9  | 小结                           | 186 |
| 9.3    | 类型类中的类型依赖                    | 187 |
| 9.4    | 类型类中的关联类型                    | 192 |
| 9.5    | 定长列表                         | 193 |
| 9.6    | 运行时重载                        | 197 |
| 9.7    | existential类型                | 198 |
|        | 本章小结                         | 199 |
|        | 第10章 monad初步                 | 201 |
| 10.1   | monad简介                      | 201 |
| 10.2   | 从identity monad开始            | 204 |
| 10.3   | maybe monad                  | 206 |
| 10.4   | monad定律                      | 209 |
| 10.5   | 列表monad                      | 210 |
| 10.6   | monad相关运算符                   | 210 |
| 10.7   | monadplus                    | 211 |
| 10.8   | functor、applicative与monad的关系 | 213 |
|        | 本章小结                         | 215 |
|        | 第11章 系统编程及输入/输出              | 216 |
| 11.1   | 不纯函数与副作用                     | 216 |
| 11.2   | io monad                     | 218 |

- 11.3 输入/输出处理 222
  - 11.3.1 control.monad中的函数 222
  - 11.3.2 系统环境变量与命令行参数 224
  - 11.3.3 数据的读写 225
  - 11.3.4 格式化输出printf函数 228
  - 11.3.5 printf函数的简易实现 229
- 11.4 星际译王词典 233
  - 11.4.1 二分法查找 234
  - 11.4.2 散列表的使用 237
- 11.5 简易异常处理 239
- 11.6 haskell中的时间 244
- 本章小结 245
- 第12章 记录器monad、读取器monad、状态monad 246
  - 12.1 记录器monad 246
    - 12.1.1 monadwriter 248
    - 12.1.2 记录归并排序过程 249
  - 12.2 读取器monad 250
    - 12.2.1 monadreader 251
    - 12.2.2 变量环境的引用 252
  - 12.3 状态monad 253
    - 12.3.1 状态monad标签器 254
    - 12.3.2 用状态monad实现栈结构 255
    - 12.3.3 状态monad、funapp单位半群和读取器monad的关系 257
    - 12.3.4 monadstate 258
    - 12.3.5 基于栈的计算器 258
  - 12.4 随机数的生成 270
  - 本章小结 271
- 第13章 monad转换器 273
  - 13.1 从identityt monad转换器开始 273
  - 13.2 monad转换器组合与复合monad的区别 276
  - 13.3 monad转换器的组合顺序 278
  - 13.4 lift与liftio 281
  - 13.5 简易monad编译器 282
  - 13.6 语法分析器monad组合子 286
    - 13.6.1 简易语法分析器的实现 287
    - 13.6.2 parsec库简介 291
    - 13.6.3 上下文无关文法 296
    - 13.6.4 基于语法分析器的计算器 300
  - 本章小结 304
- 第14章 quickcheck简介 305
  - 14.1 测试函数属性 305
  - 14.2 测试数据生成器 308
  - 本章小结 310
- 第15章 惰性求值简介 311
  - 15.1 演算简介 311
  - 15.2 bottom 313
  - 15.3 表达式形态和thunk 314
    - 15.3.1 whnf、hnf与nf 314
    - 15.3.2 thunk与严格求值 315

|        |               |     |
|--------|---------------|-----|
| 15.4   | 求值策略          | 319 |
| 15.4.1 | 引值调用          | 319 |
| 15.4.2 | 按名调用          | 320 |
| 15.4.3 | 常序求值          | 320 |
| 15.5   | 惰性求值          | 321 |
| 15.6   | 严格模式匹配与惰性模式匹配 | 322 |
| 第16章   | 并行与并发编程       | 324 |
| 16.1   | 确定性的并行计算      | 325 |
| 16.2   | 轻量级线程         | 333 |
| 16.2.1 | 调度的不确定性       | 333 |
| 16.2.2 | 基本线程通信        | 334 |
| 16.2.3 | 信道            | 337 |
| 16.2.4 | 简易聊天服务器       | 337 |
| 16.3   | 软件事务内存        | 341 |
| 16.3.1 | 软件事务内存简介      | 341 |
| 16.3.2 | 软件事务内存的使用     | 343 |
| 16.3.3 | 哲学家就餐问题       | 347 |
| 16.3.4 | 圣诞老人问题        | 350 |
| 16.4   | 异步并发库简介       | 355 |
|        | 本章小结          | 357 |
|        | 参考文献          | 358 |
|        | 后记            | 359 |



# 《Haskell函数式编程入门》

## 精彩短评

- 1、排版错误多如牛毛，但还算基础扎实的入门书。
- 2、好难看…… 不明不白的……
- 3、主要讲Haskell的用法，而没有讲清楚fp本身的概念，可能作者已经假定读者都有相关的基础。很多例子都是从数据结构上照搬过来的，没有必要。不过89年的学生能写出这样一本书，我已经是在跪着读了。
- 4、书不错，错误有点多，不过还好有勘误表[https://github.com/HaskellZhangSong/Introduction\\_to\\_Haskell](https://github.com/HaskellZhangSong/Introduction_to_Haskell)
- 5、讲的不是很透
- 6、入门不如《趣学指南》，实用不如《Real Word Haskell》，定位很尴尬，而且有些地方讲得很模糊。本书如某楼所说，基本上都在聊 Haskell 的语法，挂着函数式入门的书名，很少聊到函数式的东西。说实话，我觉得《JavaScript 函数式编程指南》这书讲函数式都讲得比这书好。
- 7、作者好年轻...书能够写成这样说明对Haskell和函数型语言达到了一定层次的理解。但愿我在这个年龄也能。
- 8、兄弟的书，顶一下！
- 9、太多错误，讲解一般。但中国人写的还是值得关注。
- 10、很适合入门，期待作者下一个版本更多内容。印刷有很多细节错误，排版也一般，对出版社印象分减1。。。

# 《Haskell函数式编程入门》

## 精彩书评

1、初学从前开始看有许多地方无法理解。在不介绍语法的情况下就放代码还不给解释。几乎每2页都会出现1次的“印刷错误”。很多新概念没有解释清楚，反复看几遍也不一定能明白。书的前面几章给出很多代码但是没有给出足够的解释，作为初学者表示根本不知所云。有数不清的印刷错误，初学时无法分辨出来会造成很大困扰，除非看之前先照作者放到github上的所谓“勘误表”把整本书改一遍。不知如此多错误的书是怎么出版的Lol买了非常后悔，不如网上的免费电子书！初学推荐免费的Learn You a Haskell for Great Good！

# 《Haskell函数式编程入门》

## 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:[www.tushu111.com](http://www.tushu111.com)