

<<快学Scala>>

图书基本信息

书名：<<快学Scala>>

13位ISBN编号：9787121185670

10位ISBN编号：7121185679

出版时间：2012-10

出版时间：电子工业出版社

作者：[美] C. S. 霍斯曼

页数：388

字数：453000

译者：高宇翔

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>

<<快学Scala>>

内容概要

scala是一门以java虚拟机（jvm）为目标运行环境并将面向对象和函数式编程语言的最佳特性结合在一起的编程语言。

你可以使用scala编写出更加精简的程序，同时充分利用并发的威力。

由于scala运行于jvm之上，因此它可以访问任何java类库并且与java框架进行互操作。

本书从实用角度出发，给出了一份快速的、基于代码的入门指南。

horstmann以“博客文章大小”的篇幅介绍了scala的概念，让你可以快速地掌握和应用。

实际上手的操作，清晰定义的能力层次，从初级到专家级，全程指导。

《快学scala》适合有一定的java编程经验、对scala感兴趣，并希望尽快掌握scala核心概念和用法的开发者阅读。

作者简介

Cay S.

Horstmann是《Java核心技术》卷1和卷2第8版（Sun Microsystems出版社2008年出版）的主要作者，除此之外，他还著有其他十多本面向专业程序员和计算机专业学生的书籍。

他是San

Jose

书籍目录

译者序 iii 序 xv 前言 xvii 作者简介 xix 第1章 基础 a1 1.1.1 scala解释器 1.2 声明值和变量 3 1.3 常用类型 4 1.4 算术和操作符重载 5 1.5 调用函数和方法 7 1.6 apply方法 8 1.7 scaladoc 9 练习 11 第2章 控制结构和函数 a1 13 2.1 条件表达式 14 2.2 语句终止 16 2.3 块表达式和赋值 16 2.4 输入和输出 17 2.5 循环 18 2.6 高级for循环和for推导式 19 2.7 函数 21 2.8 默认参数和带名参数 11 22 2.9 变长参数 11 22 2.10 过程 23 2.11 懒值 11 24 2.12 异常 25 练习 27 第3章 数组相关操作 a1 29 3.1 定长数组 29 3.2 变长数组：数组缓冲 30 3.3 遍历数组和数组缓冲 31 3.4 数组转换 32 3.5 常用算法 34 3.6 解读scaladoc 35 3.7 多维数组 37 3.8 与java的互操作 37 练习 38 第4章 映射和元组 a1 41 4.1 构造映射 41 4.2 获取映射中的值 42 4.3 更新映射中的值 43 4.4 迭代映射 44 4.5 已排序映射 44 4.6 与java的互操作 45 4.7 元组 45 4.8 拉链操作 46 练习 47 第5章 类 a1 51 5.1 简单类和无参方法 51 5.2 带getter和setter的属性 52 5.3 只带getter的属性 55 5.4 对象私有字段 56 5.5 bean属性 11 57 5.6 辅助构造器 59 5.7 主构造器 60 5.8 嵌套类 11 63 练习 65 第6章 对象 a1 69 6.1 单例对象 69 6.2 伴生对象 70 6.3 扩展类或特质的对象 71 6.4 apply方法 72 6.5 应用程序对象 73 6.6 枚举 74 练习 75 第7章 包和引入 a1 79 7.1 包 80 7.2 作用域规则 81 7.3 串联式包语句 83 7.4 文件顶部标记法 83 7.5 包对象 84 7.6 包可见性 85 7.7 引入 85 7.8 任何地方都可以声明引入 86 7.9 重命名和隐藏方法 87 7.10 隐式引入 87 练习 88 第8章 继承 a1 91 8.1 扩展类 91 8.2 重写方法 92 8.3 类型检查和转换 93 8.4 受保护字段和方法 94 8.5 超类的构造 94 8.6 重写字段 95 8.7 匿名子类 96 8.8 抽象类 97 8.9 抽象字段 97 8.10 构造顺序和提前定义 139 8.11 scala继承层级 100 8.12 对象相等性 11 101 练习 102 第9章 文件和正则表达式 a1 105 9.1 读取行 106 9.2 读取字符 106 9.3 读取词法单元和数字 107 9.4 从url或其他源读取 108 9.5 读取二进制文件 108 9.6 写入文本文件 108 9.7 访问目录 109 9.8 序列化 110 9.9 进程控制 a2 111 9.10 正则表达式 113 9.11 正则表达式组 114 练习 114 第10章 特质 11 117 10.1 为什么没有多重继承 117 10.2 当做接口使用的特质 119 10.3 带有具体实现的特质 120 10.4 带有特质的对象 121 10.5 叠加在一起的特质 122 10.6 在特质中重写抽象方法 124 10.7 当做富接口使用的特质 124 10.8 特质中的具体字段 125 10.9 特质中的抽象字段 126 10.10 特质构造顺序 127 10.11 初始化特质中的字段 129 10.12 扩展类的特质 131 10.13 自身类型 132 10.14 背后发生了什么 133 练习 135 第11章 操作符 11 139 11.1 标识符 139 11.2 中置操作符 140 11.3 一元操作符 141 11.4 赋值操作符 142 11.5 优先级 142 11.6 结合性 143 11.7 apply和update方法 144 11.8 提取器 12 145 11.9 带单个参数或无参数的提取器 12 146 11.10 unapplySeq方法 12 147 练习 148 第12章 高阶函数 11 151 12.1 作为值的函数 151 12.2 匿名函数 152 12.3 带函数参数的函数 153 12.4 参数(类型)推断 154 12.5 一些有用的高阶函数 155 12.6 闭包 156 12.7 sam转换 157 12.8 柯里化 158 12.9 控制抽象 159 12.10 return表达式 161 练习 162 第13章 集合 a2 165 13.1 主要的集合特质 166 13.2 可变和不可变集合 167 13.3 序列 168 13.4 列表 169 13.5 可变列表 170 13.6 集 171 13.7 用于添加或去除元素的操作符 173 13.8 常用方法 175 13.9 将函数映射到集合 177 13.10 化简、折叠和扫描 a3 178 13.11 拉链操作 181 13.12 迭代器 183 13.13 流 a3 184 13.14 懒视图 185 13.15 与java集合的互操作 186 13.16 线程安全的集合 188 13.17 并行集合 188 练习 190 第14章 模式匹配和样例类 a2 193 14.1 更好的switch 194 14.2 守卫 195 14.3 模式中的变量 195 14.4 类型模式 196 14.5 匹配数组、列表和元组 197 14.6 提取器 198 14.7 变量声明中的模式 199 14.8 for表达式中的模式 199 14.9 样例类 200 14.10 copy方法和带名参数 201 14.11 case语句中的中置表示法 201 14.12 匹配嵌套结构 202 14.13 样例类是邪恶的吗 203 14.14 密封类 204 14.15 模拟枚举 205 14.16 option类型 205 14.17 偏函数 12 207 练习 207 第15章 注解 a2 211 15.1 什么是注解 212 15.2 什么可以被注解 212 15.3 注解参数 213 15.4 注解实现 214 15.5 针对java特性的注解 216 15.5.1 java修饰符 216 15.5.2 标记接口 216 15.5.3 受检异常 217 15.5.4 变长参数 217 15.5.5 javabeans 218 15.6 用于优化的注解 219 15.6.1 尾递归 219 15.6.2 跳转表生成与内联 220 15.6.3 可省略方法 221 15.6.4 基本类型的特殊化 222 15.7 用于错误和警告的注解 223 练习 224 第16章 xml处理 a2 227 16.1 xml字面量 228 16.2 xml节点 228 16.3 元素属性 230 16.4 内嵌表达式 231 16.5 在属性中使用表达式 232 16.6 特殊节点类型 233 16.7 类xpath表达式 234 16.8 模式匹配 235 16.9 修改元素和属性 236 16.10 xml变换 237 16.11 加载和保存 238 16.12 命名空间 241 练习 242 第17章 类型参数 12 245 17.1 泛型类 246 17.2 泛型函数 246 17.3 类型变量界定 246 17.4 视图界定 248 17.5

上下文界定 249 17.6 manifest上下文界定249 17.7 多重界定 250 17.8 类型约束I3 250 17.9 型变252
17.10 协变和逆变点 253 17.11 对象不能泛型 255 17.12 类型通配符 256 练习 257 第18章 高级类型I2 259
18.1 单例类型 259 18.2 类型投影 261 18.3 路径262 18.4 类型别名 263 18.5 结构类型 264 18.6 复
合类型 265 18.7 中置类型 266 18.8 存在类型 267 18.9 scala类型系统 268 18.10 自身类型 269 18.11 依赖
注入 271 18.12 抽象类型I3 272 18.13 家族多态 I3274 18.14 高等类型I3 278 练习 281 第19章 解析a3 285 19.1
文法286 19.2 组合解析器操作287 19.3 解析器结果变换289 19.4 丢弃词法单元 291 19.5 生成解析
树 291 19.6 避免左递归 292 19.7 更多的组合子 294 19.8 避免回溯 296 19.9 记忆式解析器 297 19.10
解析器说到底是什么 297 19.11 正则解析器 299 19.12 基于词法单元的解析器299 19.13 错误处理 302 练习
302 第20章 actor a3 305 20.1 创建和启动actor 306 20.2 发送消息 307 20.3 接收消息 308 20.4 向其
其他actor发送消息309 20.5 消息通道 310 20.6 同步消息和future 311 20.7 共享线程 313 20.8 actor的生
命周期 316 20.9 将多个actor链接在一起 317 20.10 actor的设计318 练习 320 第21章 隐式转换和隐式参
数I3 323 21.1 隐式转换 324 21.2 利用隐式转换丰富现有类库的功能 324 21.3 引入隐式转换 325 21.4
隐式转换规则 326 21.5 隐式参数 328 21.6 利用隐式参数进行隐式转换 329 21.7 上下文界定 330 21.8
类型证明 331 21.9 @implicitnotfound注解333 21.10 canbuildfrom解读 333 练习 336 第22章 定界延续I3
339 22.1 捕获并执行延续340 22.2 “运算当中挖个洞” 341 22.3 reset和shift的控制流转 342 22.4
reset表达式的值 343 22.5 reset和shift表达式的类型 344 22.6 cps注解 345 22.7 将递归访问转化为迭
代347 22.8 撤销控制反转 349 22.9 cps变换 353 22.10 转换嵌套的控制上下文356 练习 358 词汇表 360 索
引 367

章节摘录

版权页：插图：你也可以在命令行中使用这些名称：`scalac -Xelide-below INFO myprog.scala` 如果不指定`-Xelide-below`标志，那些被注解的值低于1000的方法会被省略，剩下SEVERE的方法和断言，但会去掉所有警告。

说明：ALL和OFF级别可能会让人感到困惑。

注解`@elide (ALL)`表示方法总是被省略，而`@elide (OFF)`表示方法永不被省略。

但`-Xelide-below OFF`的意思是要省略所有方法，而`-Xelide-below ALL`的意思是什么都不要省略。这就是后来又增加了MAXIMUM和MINIMUM的原因。

Predef模块定义了一个可被忽略的`assert`方法。

例如，我们可以写：`def makeMap (keys: Seq (String) , values: Seq (String)) = { assert (keys.length == values.length , ""lengths don't match ") }` 如果我们用不匹配的两个参数来调用该方法，则`assert`方法将抛出`AssertionError`，报错消息为“`assertion failed: lengths don't match`”。

如果要禁用断言，可以用`-Xelide-below 2001`或`-Xelide-below MAXIMUM`。

注意在缺省情况下断言不会被禁用。

相比Java断言，这是个受欢迎的改进。

注意：对被省略的方法调用，编译器会帮我们替换成Unit对象。

如果你用到了被省略方法的返回值，则一个`ClassCastException`会被抛出。

最好只对那些没有返回值的方法使用`@elidable`注解。

<<快学Scala>>

编辑推荐

《快学Scala》适合有一定的Java编程经验、对Scala感兴趣，并希望尽快掌握Scala核心概念和用法的开发者阅读。

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>