

## 目录

第一章	Groovy 之旅.....	5
1.1.	groovy 背景.....	6
1.1.1.	groovy 是什么? .....	7
1.1.2.	与 java 共事: 无缝集成.....	7
1.1.3.	强劲代码: 一个特性丰富的语言.....	9
1.1.4.	后台公司支持的社区驱动.....	12
1.2.	groovy 能为你做什么.....	13
1.2.1.	java 内行人士的 groovy.....	13
1.2.2.	对于脚本编写人员的 groovy .....	14
1.2.3.	为敏捷开发人员 .....	14
1.3.	运行 groovy.....	15
1.3.1.	使用 groovysh 编写 “HelloWorld” .....	16
1.3.2.	使用 groovyConsole .....	19
1.3.3.	使用 groovy.....	20
1.4.	编译和运行 groovy .....	21
1.4.1.	使用 groovyc 编译 groovy.....	22
1.4.2.	使用 java 运行编译好的 groovy 脚本.....	22
1.4.3.	使用 Ant 编译和运行 .....	23
1.5.	Groovy 的 IDE 和编辑器支持.....	24
1.5.1.	集成 IntelliJ IDEA.....	25
1.5.2.	Eclipse 插件 .....	25
1.5.3.	其他文本编辑器的支持.....	26
1.6.	摘要.....	26
第一部分	groovy 语言.....	28
第二章	前奏: groovy 基础.....	30
2.1	一般代码外观 .....	30
2.1.1	groovy 代码的注释 .....	30
2.1.2	比较 groovy 和 java 的语法.....	31
2.1.3	简洁优雅的代码.....	32
2.2	探索语言的断言功能.....	33
2.3	Groovy 预览.....	35
2.3.1	声明类 .....	36
2.3.2	使用脚本.....	36
2.3.3	GroovyBeans .....	37
2.3.4	处理文本.....	38
2.3.5	数字也是对象.....	40
2.3.6	使用 lists/maps/ranges.....	41
2.3.7	代码块: 闭包.....	45
2.3.8	groovy 结构控制.....	47
2.4	在 java 环境中运行 groovy.....	49
2.4.1	我的类就是你的类.....	49

2.4.2 GDK:groovy 类库 .....	50
2.4.3 groovy 的生命周期 .....	52
2.5 概要 .....	55
第三章 groovy 数据类型 .....	56
3.1 无处不在的对象 .....	56
3.1.1 java 类型——专有类型和引用类型 .....	56
3.1.2 Groovy 的答案：一切都是对象 .....	57
3.1.3 自动装箱和拆箱 .....	59
3.1.4 没有中间层的拆箱 .....	60
3.2 可选类型的概念 .....	61
3.2.1 指定类型 .....	61
3.2.2 静态类型 VS 动态类型 .....	62
3.3 重载操作符 .....	62
3.3.1 可重写的操作符一览 .....	63
3.3.2 重写操作符实战 .....	64
3.3.3 确保正确的工作 .....	66
3.4 使用字符串 .....	68
3.4.1 字符串的样式 .....	68
3.4.2 使用 Gstring 进行工作 .....	71
3.4.3 从 java 到 groovy .....	72
3.5 使用正则表达式 .....	74
3.5.1 在字符串中使用模式 .....	76
3.5.2 应用模式 .....	78
3.5.3 模式实战 .....	80
3.5.4 模式和性能 .....	82
3.5.5 模式分类 .....	83
3.6 使用数字 .....	84
3.6.1 造型和数字运算符 .....	84
3.6.2 GDK 为数字提供的方法 .....	86
3.7 概要 .....	87
第四章 集合类型 .....	89
4.1 使用 ranges .....	89
4.1.1 规定 ranges .....	90
4.1.2 range 是对象 .....	92
4.1.3 range 实战 .....	93
4.2 使用 list .....	95
4.2.1 声明 list .....	95
4.2.2 使用 list 操作符 .....	96
4.2.3 使用列表方法 .....	100
4.2.4 list 实战 .....	104
4.3 使用 map .....	106
4.3.1 声明 map .....	106
4.3.2 使用 map 操作符 .....	108
4.3.3 map 实战 .....	112
4.4 groovy 集合中需要注意的地方 .....	114

4.4.1 了解并发修改.....	114
4.4.2 识别副本和修改在语义上的不同.....	115
4.5 摘要.....	115
第五章 闭包.....	117
5.1 出身名门的闭包.....	117
5.2 闭包例子.....	118
5.2.1 使用迭代 (iterator) .....	119
5.2.2 处理资源.....	120
5.3 声明闭包.....	123
5.3.1 简单的声明方式.....	123
5.3.2 使用赋值的方式声明闭包.....	124
5.3.3 引用一个方法作为闭包.....	125
5.3.4 比较.....	127
5.4 应用闭包.....	128
5.4.1 调用闭包.....	128
5.4.2 更多的闭包方法.....	131
5.5 理解范围.....	134
5.5.1 简单的变量范围.....	135
5.5.2 闭包范围.....	135
5.5.3 工作中的范围使用：典型的累加测试.....	138
5.6 从闭包返回结果.....	140
5.7 设计模式的支持.....	141
5.7.1 Visitor 模式.....	142
5.7.2 Builder 模式.....	142
5.7.3 其他相关模式.....	143
5.8 结束语.....	143
第六章 groovy 的控制结构.....	145
6.1 groovy 真相.....	145
6.1.1 评估 Boolean 测试.....	145
6.1.2 将 Boolean 测试分配给变量.....	147
6.2 条件执行结构.....	148
6.2.1 普通的 if 语句.....	148
6.2.2 三元条件操作符.....	149
6.2.3 switch 语句.....	150
6.2.4 使用断言进行安全检查.....	153
6.3 循环.....	157
6.3.1 while 循环.....	157
6.3.2 for 循环.....	158
6.4 退出代码块和方法.....	160
6.4.1 正常终止：return/break/continue.....	160
6.4.2 异常：throw/try-catch-finally.....	161
6.5 总结.....	162
第七章 groovy 风格的动态面向对象.....	163
7.1 定义类和脚本.....	163
7.1.1 定义属性和本地变量.....	164

7.1.2 方法和参数 .....	167
7.1.3 安全的引用符号 (?). .....	171
7.1.4 构造器（构造方法） .....	173
7.2 组织类和脚本 .....	175
7.2.1 文件到类的关系 .....	175
7.2.2 在包中组织类 .....	177
7.2.3 类路径更长远的考虑 .....	181
7.3 高级 OO 特性 .....	183
7.3.1 使用继承 .....	183
7.3.2 使用接口 .....	183
7.3.3 Multimethods .....	184
7.4 使用 GroovyBean 工作 .....	186
7.4.1 声明 Bean .....	187
7.4.2 使用 bean 工作 .....	189
7.4.3 为任何对象使用 bean 方法 .....	192
7.4.4 属性、访问方法、隐射和扩展 (Fields/accessors/maps/Expando) .....	193
7.5 使用强劲的特性 .....	195
7.5.1 使用 GPath 来查询对象 .....	195
7.5.2 注入展开操作符 .....	200
7.5.3 使用 use 关键字进行混入 .....	201
7.6 在 groovy 进行元程序编程 .....	204
7.6.1 理解元类 (MetaClass) 的概念 .....	205
7.6.2 方法调用和拦截 .....	207
7.6.3 方法拦截实战 .....	208
7.7 总结 .....	212