

Kotlin 极简教程

陈光剑◎编著

Kotlin 快速入门教程，零基础学会 Kotlin 开发
从 Kotlin 基础知识到动手实战，包含了大量精选示例代码和应用案例



机械工业出版社
China Machine Press

Kotlin

极简教程

陈光剑◎编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Kotlin 极简教程 / 陈光剑编著. —北京: 机械工业出版社, 2017.9

ISBN 978-7-111-57993-9

I. K… II. 陈… III. JAVA 语言—程序设计—教材 IV. TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 220693 号

Kotlin 极简教程

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴 怡

责任校对: 李秋荣

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2017 年 9 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 31.75

书 号: ISBN 978-7-111-57993-9

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有 • 侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

内容简介

本书从 Kotlin 的 HelloWorld 开始，生动讲解 Kotlin 的基础知识和实战案例。全书共 16 章，分为三大部分：第一部分“Kotlin 语言基础”包括第 1~4 章，介绍如何快速开始使用 Kotlin，以及 Kotlin 中的语法、基本数据类型和类型系统；第二部分“编程进阶”包括第 5~10 章，介绍 Kotlin 标准库中的集合类、泛型、面向对象编程特性、函数式编程、协程、与 Java 的互操作；第三部分“综合实战”包括第 11~16 章，介绍如何使用 Kotlin 集成 Spring Boot、Spring MVC 等框架来开发 Web 服务端应用，如何使用 Kotlin 开发 Android 应用，如何使用 Kotlin 的 DSL 代码风格，如何操纵文件 IO、正则表达式以及多线程，如何用 Kotlin Native 与 C 语言互操作。

作者简介

陈光剑

资深开发人员，目前就职于阿里巴巴集团，参与过多种平台工具的开发，具有丰富的平台开发经验，操练过Java、Android、Scala、Groovy等语言进行领域建模、架构设计、工具开发，在工作中积累了大量经验，善于总结，乐于分享，发布过大量技术文章。

华章IT
HZBOOKS | Information Technology



Preface 序 言

首先非常感谢我的同事陈光剑在图书还没有正式发布之前能将所有章节发给我阅读，并邀请写一个序言介绍这本书。Kotlin 1.0 在 2016 年初推出后，受到非常多的程序员认可，当然你也可以理解为热捧。越来越多的项目开始使用 Kotlin 开发，而且很多技术和框架都开始支持 Kotlin，其中最知名的就是 Google 宣布 Kotlin 为 Android 开发官方语言，其他如 Spring Framework 5.0 原生支持 Kotlin，Spring Boot 2、JUnit 5 等都支持 Kotlin。众多的开发者非常看重 Kotlin 带来的便捷性和代码精简，但是 DSL、协程 (CoRoutine)、Reactive 等高级特性越来越多地被采用。在这里我就不介绍 Kotlin 众多特性了，本书会一一呈现给大家。

学习一门语言一定会涉及图书和资料，本书是不可多得的优秀图书，内容翔实，描述简洁，对编程语言有深刻的理解，融入了自己多年的开发经验。不可否认，Kotlin 官方文档写得非常不错，还有 Safari Online Books 上的 Hadi Hariri 的 Kotlin 视频教程也很好。但是对于学习一项新技术来说，快速掌握、系统了解、理解不要出现偏差，非常重要，如果有这样一本中文图书，对中国程序员会帮助非常大，大家可以花更少的时间了解和学习 Kotlin，并在实际的开发中快速使用起来。

Kotlin 非常新，而且发展也非常快，是一门非常有前景的开发语言，我个人希望 Android 和 Java 程序员都能停下来阅读这本书，系统地了解这项技术。当然没有一本书能够跟得上技术快速发展的步伐，新特性、技术实现总是快速地推陈出新，但是知识原理都是相通的。DSL 设计可能很难，但是你如果非常理解书中讲到的 Function Literals with Receiver、Invoke Operator 和 Infix notation，配合基础的 Lambda、Operator overloading 和 Function Extension，那么一切就非常简单啦。

我这一年也在做一些 Kotlin 相关的研究，主要集中于 DSL 设计、Reactive、CoRoutine 和相关框架整合等，但是真没有能力和精力来写一本关于 Kotlin 的图书帮助到程序员，因为写本书相当不容易，写本书好书更加烧脑费时，令人佩服的是我的同事陈光剑给大家带来了好书，期待这本书给更多的人带来实惠，使更多的人能够快速学习和使用 Kotlin。

雷卷，阿里巴巴 70 后 Java 程序员

2017 年 8 月于杭州

前言 Preface

Kotlin 是 JetBrains 团队开发的一门现代的、注重工程实用性的静态类型编程语言，JetBrains 团队以开发了世界上最好用的 IDE 而著称。Kotlin 于 2010 年推出，并在 2011 年开源。Kotlin 充分借鉴并汲取了 Java、Scala、Groovy、C#、Gosu、JavaScript、Swift 等多门杰出语言的优秀特性，语法简单优雅、表现力丰富、抽象扩展方便、代码可重用性好，同时也支持面向对象和函数式编程的多范式编程。Kotlin 可以编译成 Java 字节码运行在 JVM 平台和 Android 平台，也可以编译成 JavaScript 运行在浏览器环境，而且还可以直接编译成机器码的系统级程序，直接运行在嵌入式、iOS、MacOS/Linux/Windows 等没有 JVM 环境的平台。Kotlin 源自产业界，它解决了工程实践中程序设计所面临的真实痛点，例如，类型系统可以避免指针异常的问题。

我最早是被 Kotlin 的下面这段代码所吸引：

```
package com.easy.kotlin

fun <A, B, C> compose(f: (B) -> C, g: (A) -> B): (A) -> C {
    return { x -> f(g(x)) }
}

fun isOdd(x: Int) = x % 2 != 0
fun length(s: String) = s.length
fun main(args: Array<String>) {
    val oddLength = compose(::isOdd, ::length)
    val strings = listOf("a", "ab", "abc")
    println(strings.filter(oddLength))
}
```

13 行。

这大约是在三年前，当时我在学习 Java 8 中的函数式编程以及 Lambda 表达式等新特性。那时，我也对 Scala、Groovy、Clojure、Haskell 等技术很感兴趣，在伴随着兴趣的学习过程中，我无意中看到了上面那段 Kotlin 代码，第一眼看到这么优雅的函数式编程风格，尤其是 compose 函数的定义实现，深深地吸引了我。

Swift 使用 func 关键字声明函数多个 c，怪怪的；Groovy、Scala 等语言使用 def 关键字，与函数本义联想不直接；JavaScript 使用 function 关键字又显得死板了些。而 Kotlin 中的 fun

则简单优雅地恰到好处，关键还让人自然联想到“乐趣、开心、愉快”这样的意思，使用 Kotlin 每写一个函数都是充满乐趣的。

我们不妨来看看同样的逻辑实现，如果使用 Java 8 来写会如何：

```
package com.easy.kotlin;

import java.util.ArrayList;
import java.util.List;

interface G<A, B> {
    B apply(A a);
}

interface F<B, C> {
    C apply(B b);
}

interface FG<A, B, C> {
    C apply(A a);
}

public class ComposeFunInJava<A, B, C> {
    public static void main(String[] args) {
        G<String, Integer> g = (s) -> s.length();
        F<Integer, Boolean> f = (x) -> x % 2 != 0;
        FG<String, Integer, Boolean> fg = (x) -> f.apply(g.apply(x));

        List<String> strings = new ArrayList();
        strings.add("a");
        strings.add("ab");
        strings.add("abc");
        List<String> result = new ArrayList();
        for (String s : strings) {
            if (fg.apply(s)) {
                result.add(s);
            }
        }
        System.out.println(result);
    }
}
```

36 行，差不多是 Kotlin 的 3 倍。

我们知道，Java 是一门非常优秀的面向对象语言。但是在函数式编程方面，与其他函数语言相比，还是显得有些笨重与生涩，并且其内在体现出来的思想，依旧是面向对象的思想。

功能强大的 Scala 语言，复杂性相对较高，学习成本也高，而且 Scala 与 Java 的互操作性没有 Kotlin 好。所以，如果既想方便、流畅地使用 Java 强大与完善的生态库，又想使用更加先进的编程语言特性，无疑 Kotlin 是个非常不错的选择。

因此，我立马进入了 Kotlin 的世界！

Kotlin 之前一直是默默无闻的，直到今年（2017 年）Google IO 大会上，Google 宣布正式支持 Kotlin 为 Android 的官方开发语言，而且从 Android Studio 3.0 开始，将直接内置集成 Kotlin 而无需安装任何的插件。另外，在 Spring 5.0 M4 中也引入了对 Kotlin 专门的支持。

在学习和使用 Kotlin 的过程中，我发现我越来越喜欢 Kotlin，它是一门非常优秀、优雅、有趣、流畅实用的语言，绝对值得一试。感谢 Kotlin 团队！

本书可以说是我对 Kotlin 的使用和思考过程的粗浅总结。通过本书的写作，我加深了对 Kotlin 语言及其编程的理解，深刻体会到了学无止境的含义。写书的过程也是我系统学习与思考 Kotlin 的过程，如果本书能够对你有所帮助，将不胜欣慰。

如何阅读本书

受到 Kotlin 极简特性的感召，我希望也通过极简的表述，全面介绍 Kotlin 语言特性以及如何使用 Kotlin 进行实际项目开发。全书共 16 章，分为三大部分：第一部分“Kotlin 语言基础”包括第 1~4 章，介绍如何快速开始使用 Kotlin，以及 Kotlin 语法、基本数据类型和类型系统；第二部分“编程进阶”包括第 5~10 章，介绍 Kotlin 标准库中的集合类、泛型、面向对象编程特性、函数式编程、协程、与 Java 的互操作；第三部分“综合实战”包括第 11~16 章，介绍如何使用 Kotlin 集成 Spring Boot、Spring MVC 等框架来开发 Web 服务端应用，如何使用 Kotlin 开发 Android 应用，如何使用 Kotlin 的 DSL 代码风格，如何操纵文件 IO、正则表达式以及多线程，如何定义 Kotlin Native 与 C 语言互操作。

建议初学者最好按照顺序来阅读本书，Java 程序员可以从第 5 章开始阅读，直接研究 Kotlin 的特殊点，C 语言程序员可以先读第 16 章再考虑补充前面的基础知识。

本书章节内容如下。

第一部分 “Kotlin 语言基础”

第 1 章是 Kotlin 语言的简介，带领读者快速学习 Kotlin 的环境搭建以及常用工具的使用。该章最后还给出一个编程语言学习的小结。通过该章的学习，能够快速进入 Kotlin 的世界。

第 2 章是快速开始 Hello World，分别给出了使用命令行 REPL、可执行应用程序、Web RESTful、Android、JavaScript 等平台环境上的 HelloWorld 示例。

第 3 章介绍 Kotlin 语言的基础知识，包括 Kotlin 语言的关键字与标识符等、表达式与流程控制、运算操作符、函数及其扩展等基本内容。

第 4 章介绍 Kotlin 语言的基本类型和类型系统。包括 Kotlin 的内置基本类型：数字、字符串、布尔、数组等，以及 Kotlin 中引入的特殊的可空类型。

第二部分 “编程进阶”

第 5 章介绍 Kotlin 标准库中的集合类：List、Set、Map。通过该章的学习，我们将了解到 Kotlin 是如何扩展 Java 集合库的，这使得写代码更加简单容易。

第 6 章介绍 Kotlin 泛型的基本概念、型变以及类型边界等内容。

第 7 章介绍 Kotlin 面向对象编程的特性：类与构造函数、抽象类与接口、继承以及多重继承等基础知识，同时介绍了 Kotlin 中的注解类、枚举类、数据类、密封类、嵌套类、内部类、匿名内部类等。

第 8 章介绍 Kotlin 函数式编程的相关内容，其中重点介绍了 Kotlin 中的高阶函数、Lambda 表达式、闭包等核心语法，并给出相应的实例说明。还探讨了关于 Lambda 演算、Y 组合子与递归等函数式编程思想的内容。

第 9 章介绍 Kotlin 中的协程；包括协程的创建、执行、取消等操作。在该章的后半部分，主要探讨挂起函数的组合执行、协程上下文与调度器、通道与管道等相关内容。最后，对协程与线程进行了简单比较，简要介绍了 Kotlin 的协程 API 库。

第 10 章介绍 Kotlin 与 Java 的互操作。

第三部分 “综合实战”

第 11 章介绍如何使用 Kotlin 集成 Spring Boot、SpringMVC 等框架来开发 Web 服务端应用，给出了一个完整的开发实例。最后，简单介绍了 Spring 5.0 中对 Kotlin 的支持特性。

第 12 章介绍使用 Kotlin 集成 Gradle 开发的相关内容。

第 13 章介绍如何使用 Kotlin 开发 Android 应用的具体方法，其中用到了 Anko、ButterKnife、Realm 等相关框架。

第 14 章介绍 Kotlin 中 DSL 的相关内容。我们将会看到 Kotlin 的扩展函数和高阶函数（Lambda 表达式）特性，为定义 Kotlin DSL 提供了极大的支持。使用 DSL 的代码风格，可以让程序更加直观易懂、简洁优雅。

第 15 章介绍 Kotlin 文件 IO 操作、正则表达式以及多线程相关的内容。

第 16 章简单介绍了 Kotlin Native，并给出了 Kotlin 与 C 语言互操作的完整实例。

谁适合阅读本书

本书适合于所有程序员，不管你是前端开发者、Android/iOS 开发者，还是 Java 开发者、C 语言开发者，学习 Kotlin 都会非常容易，本书将帮助读者快速进入 Kotlin 世界。

如果你目前还不是程序员，但想进入编程世界，那么可以尝试从 Kotlin 开始学习。虽然本书中的部分内容需要一定的 Java 编程基础，但是 Kotlin 本身的极简特性能激发你对编程的兴趣。

代码下载

几乎在每章末尾处，我都附上了该章示例工程源代码地址。这些源码都在 <https://github.com/EasyKotlin>。读者可以根据需要，自由下载学习。

致谢

在本书的写作出版过程中，得到了很多人的帮助和陪伴。

首先要感谢的是我的妻子和两个可爱的孩子。正是有了你们的陪伴，我的生活才更加有意义。我要感谢我的父母，虽然他们可能不知道我写的东西是什么，但是我在写书过程中，脑海里会经常浮现出他们的笑脸，正是由于他们的辛勤养育，我才能长成今天的我。

我要衷心地感谢本书的策划编辑吴怡女士。在本书的写作修改过程中，她耐心细致地对稿件进行了详尽审阅和批注，还提出了很多宝贵的修改建议，并不断给予我鼓励，才使我快速完成了这本书。还感谢审稿编辑陈佳媛、曲熠对本书提出的宝贵意见，同时感谢本书出版过程中所有付出辛劳的华章公司工作人员。

在此，我还要特别感谢我们公司的技术大牛雷卷（陈立兵），非常感谢他能够抽出宝贵时间审阅本书，并给出了内容的勘误，极大地提升了本书的质量，最后，还为本书写了序。真的非常感谢！

我还要感谢在我的工作学习中认识的所有朋友和同事们，能够认识你们并跟你们一起学习、共事，是我的荣幸！

请联系我

虽然在本书写作与修改的过程中，我竭尽全力追求简单正确、清晰流畅地表达内容，但是限于自身水平和有限的时间，仍会有错误与疏漏之处，还望各位读者不吝指正。

对本书的任何问题、意见或者建议，欢迎通过邮件 universsky@163.com 与我交流。

快乐生活，快乐学习，快乐分享，快乐实践出真知。祝大家阅读愉快！

陈光剑

2017年8月于杭州

Contents 目 录

序 言 前 言

第一部分 Kotlin 语言基础

第 1 章 Kotlin 简介 2

1.1 Kotlin 简史 2

1.1.1 Kotlin 概述 3

1.1.2 Kotlin 元年: 2016 3

1.2 快速学习工具 4

1.2.1 云端 IDE 4

1.2.2 本地命令行环境搭建 5

1.2.3 使用 IntelliJ IDEA 9

1.2.4 使用 Eclipse 13

1.2.5 使用 Gradle 构建 Kotlin 项目 17

1.2.6 编程语言学习小结 18

1.3 本章小结 20

第 2 章 快速开始 HelloWorld 21

2.1 命令行版 HelloWorld 21

2.2 应用程序版 HelloWorld 21

2.3 Web RESTful HelloWorld 24

2.4 Android 版的 HelloWorld 34

2.5 JavaScript 版 HelloWorld 39

2.6 本章小结 49

第 3 章 Kotlin 语言基础 50

3.1 包 50

3.2 声明变量和值 56

3.3 变量类型推断 57

3.3.1 省去变量类型 57

3.3.2 使用 is 运算符进行类型检测 59

3.4 字符串与其模板表达式 59

3.5 流程控制语句 60

3.5.1 if 表达式 60

3.5.2 when 表达式 62

3.5.3 for 循环 64

3.5.4 while 循环 64

3.5.5 break 和 continue 65

3.5.6 return 66

3.5.7 标签 69

3.5.8 throw 表达式 71

3.6 代码注释 72

3.7 语法与标识符 73

3.7.1 修饰符 73

3.7.2 关键字 (保留字) 75

3.7.3	操作符和操作符的重载	81
3.7.4	操作符优先级	82
3.7.5	一元操作符	83
3.7.6	二元操作符	85
3.8	扩展函数和扩展属性	90
3.8.1	扩展函数	90
3.8.2	扩展属性	93
3.9	空指针安全	93
3.10	标准库 API 简介	96
3.11	本章小结	97

第4章 基本数据类型与类型系统 98

4.1	什么是类型	98
4.2	编译时类型与运行时类型	100
4.2.1	弱类型与强类型	100
4.2.2	静态类型与动态类型	101
4.2.3	显式类型与隐式类型	101
4.3	根类型 Any	101
4.4	基本类型	102
4.4.1	数字类型	104
4.4.2	字符类型与转义符	109
4.4.3	布尔类型	111
4.4.4	字符串类型	112
4.4.5	数组类型	117
4.5	可空类型	120
4.5.1	Kotlin 中的 null	120
4.5.2	可空类型 String? 与安全调用 ?	122
4.5.3	可空性的实现原理	124
4.5.4	可空类型层次体系	126
4.6	kotlin.Unit 类型	127
4.7	kotlin.Nothing 类型	128
4.8	类型检测与类型转换	131

4.8.1	is 与 !is 运算符	131
4.8.2	as 运算符	133
4.9	本章小结	134

第二部分 编程进阶

第5章 集合类 136

5.1	集合类的基本概念	136
5.1.1	集合类是一种数据结构	137
5.1.2	集合类是 ADT	137
5.1.3	连续存储和离散存储	138
5.2	Kotlin 集合类简介	138
5.3	List	139
5.3.1	创建不可变 List	140
5.3.2	创建可变集合 MutableList	142
5.3.3	遍历 List 元素	143
5.3.4	List 元素操作函数	146
5.3.5	List 集合类的基本运算函数	151
5.3.6	过滤操作函数	159
5.3.7	映射操作函数	166
5.3.8	分组操作函数	169
5.3.9	排序操作符	170
5.3.10	生产操作符	171
5.4	Set	174
5.4.1	空集	174
5.4.2	创建 Set	176
5.4.3	使用 Java 中的 Set 类	177
5.4.4	Set 元素的加减操作 plus minus	179
5.5	Map	179
5.5.1	创建 Map	180
5.5.2	访问 Map 的元素	184

5.5.3 Map 操作符函数	186
5.6 本章小结	192

第6章 泛型 193

6.1 泛型简介	193
6.1.1 泛型与多态	193
6.1.2 为何引进泛型	194
6.1.3 Java 与 Kotlin 中的泛型	195
6.2 型变	197
6.2.1 Java 的类型通配符	197
6.2.2 泛型和数组的型变	201
6.3 Kotlin 中的泛型	205
6.3.1 out T 与 in T	205
6.3.2 声明处型变	206
6.3.3 类型投影	206
6.4 泛型类	207
6.5 泛型函数	208
6.6 本章小结	208

第7章 面向对象编程 209

7.1 面向对象编程思想	209
7.1.1 一切皆是映射	209
7.1.2 二进制 01 与易经阴阳	211
7.1.3 从面向过程到面向对象	212
7.2 类与构造函数	215
7.2.1 声明类	216
7.2.2 构造函数	216
7.2.3 类的属性	218
7.2.4 函数(方法)	219
7.3 抽象类	219
7.3.1 抽象类的定义	219
7.3.2 抽象函数	220
7.3.3 抽象属性和抽象类	221

7.4 接口	223
7.4.1 接口定义	223
7.4.2 实现接口	224
7.4.3 覆盖冲突	225
7.4.4 接口中的属性	225
7.5 抽象类和接口的差异	226
7.6 继承	227
7.6.1 open 类	227
7.6.2 多重继承	228
7.7 枚举类	230
7.7.1 枚举类基本用法	231
7.7.2 初始化枚举值	231
7.7.3 使用枚举常量	232
7.8 注解类	233
7.8.1 声明注解	233
7.8.2 使用注解	234
7.9 单例模式与伴生对象	234
7.9.1 单例模式	234
7.9.2 object 对象	235
7.9.3 嵌套 object 对象	236
7.9.4 匿名 object	238
7.9.5 伴生对象	239
7.10 密封类	242
7.11 数据类	243
7.11.1 构造函数中的 val/var	243
7.11.2 领域实体类	245
7.11.3 数据类的概念	245
7.11.4 数据类的限制	246
7.11.5 数据类的解构	248
7.11.6 标准数据类 Pair 和 Triple	248
7.12 嵌套类	249
7.12.1 嵌套类: 类中的类	249
7.12.2 内部类	250

7.12.3 匿名内部类	250	9.2 协程的基本操作	297
7.13 委托	252	9.2.1 搭建项目	297
7.13.1 代理模式	252	9.2.2 简单协程示例	298
7.13.2 类的委托	253	9.2.3 使用 launch 函数启动协程	299
7.13.3 委托属性	255	9.2.4 CommonPool 对象	299
7.14 本章小结	260	9.2.5 挂起函数	301
第 8 章 函数式编程	261	9.2.6 runBlocking 函数	302
8.1 函数式编程概述	261	9.2.7 等待一个协程任务执行完毕	303
8.1.1 函数式编程与面向对象编程	262	9.2.8 取消协程	306
8.1.2 函数式编程基本特性	265	9.2.9 协程取消失效	307
8.1.3 函数式编程核心思想	266	9.2.10 计算代码协程的取消方式	308
8.2 在 Kotlin 中使用函数式编程	284	9.2.11 在 finally 中的协程代码	310
8.2.1 Kotlin 中的函数	284	9.2.12 协程中不被取消的代码块	312
8.2.2 扩展函数	286	9.2.13 设置协程超时时间	313
8.2.3 中缀函数	286	9.3 挂起函数	314
8.2.4 函数参数	287	9.3.1 按默认顺序执行	314
8.2.5 函数返回类型	289	9.3.2 使用异步并发执行	315
8.2.6 单表达式函数	289	9.4 协程上下文与调度器	316
8.2.7 函数作用域	289	9.4.1 调度和线程	316
8.2.8 泛型函数	290	9.4.2 父子协程	318
8.2.9 高阶函数	290	9.5 通道与管道	319
8.2.10 匿名函数	291	9.5.1 通道 vs 阻塞队列	319
8.2.11 Lambda 表达式	291	9.5.2 关闭通道和迭代遍历元素	320
8.2.12 it: 单个参数的隐式名称	292	9.5.3 生产者 - 消费者模式	321
8.2.13 闭包	292	9.5.4 使用管道生产无限序列	322
8.2.14 带接收者的函数字面值	293	9.5.5 管道与无穷质数序列	323
8.2.15 具体化的类型参数	293	9.5.6 通道缓冲区	324
8.2.16 尾递归 tailrec	294	9.5.7 构建无穷惰性序列	324
8.3 本章小结	295	9.6 协程与线程	325
第 9 章 轻量级线程: 协程	296	9.6.1 轻量级的协程	325
9.1 协程的基本概念	296	9.6.2 协程 vs 守护线程	326
		9.6.3 区别	327
		9.6.4 协程的好处	328

9.6.5 Kotlin 协程的内部机制	328
9.7 本章小结	329

第 10 章 Kotlin 与 Java 互操作

10.1 在 Kotlin 中调用 Java	330
10.1.1 调用 Okhttp 库示例	331
10.1.2 使用 Java 的集合类	332
10.1.3 调用 Java 中的 Getter 和 Setter	333
10.1.4 调用 Java 中返回 void 的 方法	334
10.1.5 调用来自 Java 中的 null	335
10.1.6 使用 Java 的泛型	336
10.1.7 数组的互操作	337
10.1.8 可变参数的互操作	337
10.1.9 检查 Java 中的异常	337
10.1.10 调用 Java 的 Object 方法	338
10.1.11 访问 Java 静态成员	340
10.1.12 使用 Java 的反射	340
10.1.13 SAM 转换	341
10.2 在 Java 中调用 Kotlin	342
10.2.1 访问 Kotlin 属性	342
10.2.2 调用 Kotlin 的包级函数	344
10.2.3 调用 Kotlin 中的实例字段	347
10.2.4 调用 Kotlin 中的静态字段	348
10.2.5 调用 Kotlin 中的静态方法	349
10.2.6 Kotlin 与 Java 的可见性	350
10.2.7 生成默认参数函数的重载	352
10.2.8 显式声明 Kotlin 中的异常	353
10.2.9 关键字冲突的互操作	355
10.2.10 Nothing 类型	355
10.3 类型映射	356
10.4 Kotlin 与 Java 的简单实例对比	357

10.5 本章小结	364
-----------------	-----

第三部分 综合实战

第 11 章 使用 Kotlin 集成 Spring

Boot 开发 Web 服务端	366
11.1 Spring Boot 简介	366
11.2 系统架构技术栈	367
11.3 准备工作	367
11.3.1 创建工程	367
11.3.2 Gradle 配置文件说明	369
11.4 数据库层开发	370
11.4.1 Endpoint 监控接口	373
11.4.2 实体类	375
11.4.3 数据访问层代码	377
11.5 后端逻辑开发	379
11.5.1 控制器逻辑层	379
11.5.2 启动初始化 CommandLine- Runner	380
11.5.3 应用启动类	380
11.5.4 Model 数据绑定	382
11.5.5 模板引擎视图页面	383
11.5.6 运行测试	383
11.6 前端后端集成	384
11.6.1 引入前端组件	384
11.6.2 实现写文章模块	386
11.6.3 加上导航栏	388
11.6.4 抽取公共模板文件	389
11.6.5 写文章的控制器层接口	390
11.6.6 前端 Ajax 请求	390
11.6.7 文章详情页	391
11.6.8 添加 Markdown 支持	393
11.6.9 文章列表分页搜索	395