УДК 004.438Kotlin
ББК 32.973.26-018.1
Ж53

Ж53 Жемеров Д., Исакова С.

Kotlin в действии. / пер. с англ. Киселев А. Н. – М.: ДМК Пресс, 2018. – 402 с.: ил.

ISBN 978-5-97060-497-7

Язык Kotlin предлагает выразительный синтаксис, мощную и понятную систему типов, великолепную поддержку и бесшовную совместимость с существующим кодом на Java, богатый выбор библиотек и фреймворков. Kotlin может компилироваться в байт-код Java, поэтому его можно использовать везде, где используется Java, включая Android. А благодаря эффективному компилятору и маленькой стандартной библиотеке Kotlin практически не привносит накладных расходов.

Данная книга научит вас пользоваться языком Kotlin для создания высококачественных приложений. Написанная создателями языка – разработчиками в компании JetBrains, – эта книга охватывает такие темы, как создание предметно-ориентированных языков, функциональное программирование в JVM, совместное использование Java и Kotlin и др.

Издание предназначено разработчикам, владеющим языком Java и желающим познакомиться и начать эффективно работать с Kotlin.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-61729-329-0 (англ.)
ISBN 978-5-97060-497-7 (рус.)

ɖqdpɏeɏ bƐ dədƐbɏɜə d ɕɑɑbɕɏ Ɉqdpɏqɏè

Оглавление

Предисловие ... **12**
Вступление ... **13**
Благодарности ... **14**
Об этой книге ... **15**
Об авторах ... **19**
Об изображении на обложке ... **19**
Часть 1. Введение в Kotlin ... **21**
Глава 1. Kotlin: что это и зачем ... **22**
1.1. Знакомство с Kotlin ... 22
1.2. Основные черты языка Kotlin ... 23
1.2.1. Целевые платформы: серверные приложения, Android и везде, где запускается Java ... 23
1.2.2. Статическая типизация ... 24
1.2.3. Функциональное и объектно-ориентированное программирование ... 25
1.2.4 Бесплатный язык с открытым исходным кодом ... 27
1.3. Приложения на Kotlin ... 27
1.3.1. Kotlin на сервере ... 27
1.3.2. Kotlin в Android ... 29
1.4. Философия Kotlin ... 30
1.4.1. Прагматичность ... 31
1.4.2. Лаконичность ... 31
1.4.3. Безопасность ... 32
1.4.4. Совместимость ... 33
1.5. Инструментарий Kotlin ... 34
1.5.1. Компиляция кода на Kotlin ... 35
1.5.2. Плагин для Intellij IDEA и Android Studio ... 36
1.5.3. Интерактивная оболочка ... 36
1.5.4. Плагин для Eclipse ... 36
1.5.5. Онлайн-полигон ... 36
1.5.6. Конвертер кода из Java в Kotlin ... 37
1.6. Резюме ... 37
Глава 2. Основы Kotlin ... **39**
2.1. Основные элементы: переменные и функции ... 39
2.1.1. Привет, мир! ... 40
2.1.2. Функции ... 40
2.1.3. Переменные ... 42
2.1.4. Простое форматирование строк: шаблоны ... 44
2.2. Классы и свойства ... 45
2.2.1 Свойства ... 46
2.2.2. Собственные методы доступа ... 48

2.2.3. Размещение исходного кода на Kotlin: пакеты и каталоги 49

2.3. Представление и обработка выбора: перечисления и конструкция «when» 51

2.3.1. Объявление классов перечислений 51

2.3.2. Использование оператора «when» с классами перечислений 52

2.3.3. Использование оператора «when» с произвольными объектами 54

2.3.4. Выражение «when» без аргументов 55

2.3.5. Автоматическое приведение типов: совмещение проверки и приведения типа 55

2.3.6. Рефакторинг: замена «if» на «when» 58

2.3.7. Блоки в выражениях «if» и «when» 59

2.4. Итерации: циклы «while» и «for» 60

2.4.1. Цикл «while» 60

2.4.2. Итерации по последовательности чисел: диапазоны и прогрессии 61

2.4.3. Итерации по элементам словарей 62

2.4.4. Использование «in» для проверки вхождения в диапазон или коллекцию 64

2.5. Исключения в Kotlin 65

2.5.1. «try», «catch» и «finally» 66

2.5.2. «try» как выражение 67

2.6. Резюме 68

Глава 3. Определение и вызов функций 70

3.1. Создание коллекций в Kotlin 70

3.2. Упрощение вызова функций 72

3.2.1. Именованные аргументы 73

3.2.2. Значения параметров по умолчанию 74

3.2.3. Избавление от статических вспомогательных классов: свойства и функции верхнего уровня 76

3.3. Добавление методов в сторонние классы: функции-расширения и свойства-расширения 78

3.3.1. Директива импорта и функции-расширения 80

3.3.2. Вызов функций-расширений из Java 80

3.3.3. Вспомогательные функции как расширения 81

3.3.4. Функции-расширения не переопределяются 82

3.3.5. Свойства-расширения 84

3.4. Работа с коллекциями: переменное число аргументов, инфиксная форма записи вызова и поддержка в библиотеке 85

3.4.1. Расширение API коллекций Java 85

3.4.2. Функции, принимающие произвольное число аргументов 86

3.4.3. Работа с парами: инфиксные вызовы и мультидекларации 87

3.5. Работа со строками и регулярными выражениями 88

3.5.1. Разбиение строк 89

3.5.2. Регулярные выражения и строки в тройных кавычках 89

3.5.3. Многострочные литералы в тройных кавычках 91

3.6. Чистим код: локальные функции и расширения 93

3.7. Резюме 96

Глава 4. Классы, объекты и интерфейсы 97
4.1. Создание иерархий классов 98
4.1.1. Интерфейсы в Kotlin 98
4.1.2. Модификаторы open, final и abstract: по умолчанию final 101
4.1.3. Модификаторы видимости: по умолчанию public 103
4.1.4. Внутренние и вложенные классы: по умолчанию вложенные 105
4.1.5. Запечатанные классы: определение жестко заданных иерархий 108
4.2. Объявление классов с нетривиальными конструкторами или свойствами 110
4.2.1. Инициализация классов: основной конструктор и блоки инициализации 110
4.2.2. Вторичные конструкторы: различные способы инициализации суперкласса 113
4.2.3. Реализация свойств, объявленных в интерфейсах 115
4.2.4. Обращение к полю из методов доступа 117
4.2.5. Изменение видимости методов доступа 118
4.3. Методы, сгенерированные компилятором: классы данных и делегирование 119
4.3.1. Универсальные методы объектов 120
4.3.2. Классы данных: автоматическая генерация универсальных методов 123
4.3.3. Делегирование в классах. Ключевое слово by 124
4.4. Ключевое слово object: совместное объявление класса и его экземпляра 127
4.4.1. Объявление объекта: простая реализация шаблона «Одиночка» 127
4.4.2. Объекты-компаньоны: место для фабричных методов и статических членов класса 130
4.4.3. Объекты-компаньоны как обычные объекты 132
4.4.4. Объекты-выражения: другой способ реализации анонимных внутренних классов 135
4.5. Резюме 136

Глава 5. Лямбда-выражения 138
5.1. Лямбда-выражения и ссылки на члены класса 138
5.1.1. Введение в лямбда-выражения: фрагменты кода как параметры функций 139
5.1.2. Лямбда-выражения и коллекции 140
5.1.3. Синтаксис лямбда-выражений 141
5.1.4. Доступ к переменным из контекста 145
5.1.5. Ссылки на члены класса 148
5.2. Функциональный API для работы с коллекциями 150
5.2.1. Основы: filter и map 150
5.2.2. Применение предикатов к коллекциям: функции «all», «any», «count» и «find» 152
5.2.3. Группировка значений в списке с функцией groupBy 154
5.2.4. Обработка элементов вложенных коллекций: функции flatMap и flatten 154
5.3. Отложенные операции над коллекциями: последовательности 156
5.3.1. Выполнение операций над последовательностями: промежуточная и завершающая операции 157
5.3.2. Создание последовательностей 160
5.4. Использование функциональных интерфейсов Java 161
5.4.1. Передача лямбда-выражения в Java-метод 162

5.4.2. SAM-конструкторы: явное преобразование лямбда-выражений в функциональные интерфейсы 164

5.5. Лямбда-выражения с получателями: функции «with» и «apply» 166

5.5.1. Функция «with» 166

5.5.2. Функция «apply» 169

5.6. Резюме 171

Глава 6. Система типов Kotlin 172

6.1. Поддержка значения null 172

6.1.1. Типы с поддержкой значения null 173

6.1.2. Зачем нужны типы 175

6.1.3. Оператор безопасного вызова: «?.» 177

6.1.4. Оператор «Элвис»: «?:» 178

6.1.5. Безопасное приведение типов: оператор «as?» 180

6.1.6. Проверка на null: утверждение «!!» 182

6.1.7. Функция let 184

6.1.8. Свойства с отложенной инициализацией 186

6.1.9. Расширение типов с поддержкой null 188

6.1.10. Типовые параметры с поддержкой null 189

6.1.11. Допустимость значения null и Java 190

6.2. Примитивные и другие базовые типы 195

6.2.1. Примитивные типы: Int, Boolean и другие 195

6.2.2. Примитивные типы с поддержкой null: Int?, Boolean? и прочие 197

6.2.3. Числовые преобразования 198

6.2.4. Корневые типы Any и Any? 200

6.2.5. Тип Unit: тип «отсутствующего» значения 201

6.2.6. Тип Nothing: функция, которая не завершается 202

6.3. Массивы и коллекции 203

6.3.1 Коллекции и допустимость значения null 203

6.3.2. Изменяемые и неизменяемые коллекции 206

6.3.3. Коллекции Kotlin и язык Java 208

6.3.4. Коллекции как платформенные типы 210

6.3.5. Массивы объектов и примитивных типов 213

6.4. Резюме 215

Часть 2. Непростой Kotlin 217

Глава 7. Перегрузка операторов и другие соглашения 218

7.1. Перегрузка арифметических операторов 219

7.1.1. Перегрузка бинарных арифметических операций 219

7.1.2. Перегрузка составных операторов присваивания 222

7.1.3. Перегрузка унарных операторов 224

7.2. Перегрузка операторов сравнения 225

7.2.1. Операторы равенства: «equals» 225

7.2.2. Операторы отношения: compareTo 227

7.3. Соглашения для коллекций и диапазонов 228

7.3.1. Обращение к элементам по индексам: «get» и «set» 228

7.3.2. Соглашение «in» 230

7.3.3. Соглашение rangeTo ... 231
7.3.4. Соглашение «iterator» для цикла «for» ... 232

7.4. Мультидекларации и функции component ... 233
7.4.1. Мультидекларации и циклы ... 235

7.5. Повторное использование логики обращения к свойству: делегирование свойств ... 236
7.5.1. Делегирование свойств: основы ... 237
7.5.2. Использование делегирования свойств: отложенная инициализация и «by lazy()» ... 238
7.5.3. Реализация делегирования свойств ... 240
7.5.4. Правила трансляции делегированных свойств ... 244
7.5.5. Сохранение значений свойств в словаре ... 245
7.5.6. Делегирование свойств в фреймворках ... 246

7.6. Резюме ... 248

Глава 8. Функции высшего порядка: лямбда-выражения как параметры и возвращаемые значения ... 249

8.1. Объявление функций высшего порядка ... 250
8.1.1. Типы функций ... 250
8.1.2. Вызов функций, переданных в аргументах ... 251
8.1.3. Использование типов функций в коде на Java ... 253
8.1.4. Значения по умолчанию и пустые значения для параметров типов функций ... 254
8.1.5. Возврат функций из функций ... 257
8.1.6. Устранение повторяющихся фрагментов с помощью лямбда-выражений ... 259

8.2. Встраиваемые функции: устранение накладных расходов лямбда-выражений ... 262
8.2.1. Как работает встраивание функций ... 262
8.2.2. Ограничения встраиваемых функций ... 264
8.2.3. Встраивание операций с коллекциями ... 265
8.2.4. Когда следует объявлять функции встраиваемыми ... 267
8.2.5. Использование встраиваемых лямбда-выражений для управления ресурсами ... 268

8.3. Порядок выполнения функций высшего порядка ... 269
8.3.1. Инструкции return в лямбда-выражениях: выход из вмещающей функции ... 270
8.3.2. Возврат из лямбда-выражений: возврат с помощью меток ... 271
8.3.3. Анонимные функции: по умолчанию возврат выполняется локально ... 273

8.4. Резюме ... 274

Глава 9. Обобщенные типы ... 276

9.1. Параметры обобщенных типов ... 277
9.1.1. Обобщенные функции и свойства ... 278
9.1.2. Объявление обобщенных классов ... 279
9.1.3. Ограничения типовых параметров ... 281
9.1.4. Ограничение поддержки null в типовом параметре ... 283

9.2. Обобщенные типы во время выполнения: стирание и овеществление параметров типов ... 284
9.2.1. Обобщенные типы во время выполнения: проверка и приведение типов ... 284
9.2.2. Объявление функций с овеществляемыми типовыми параметрами ... 287
9.2.3. Замена ссылок на классы овеществляемыми типовыми параметрами ... 290
9.2.4. Ограничения овеществляемых типовых параметров ... 291
9.3. Вариантность: обобщенные типы и подтипы ... 292
9.3.1. Зачем нужна вариантность: передача аргумента в функцию ... 292
9.3.2. Классы, типы и подтипы ... 293
9.3.3. Ковариантность: направление отношения тип–подтип сохраняется ... 296
9.3.4. Контравариантность: направление отношения тип–подтип изменяется на противоположное ... 300
9.3.5. Определение вариантности в месте использования: определение вариантности для вхождений типов ... 303
9.3.6. Проекция со звездочкой: использование * вместо типового аргумента ... 306
9.4. Резюме ... 311

Глава 10. Аннотации и механизм рефлексии ... 313

10.1. Объявление и применение аннотаций ... 314
10.1.1. Применение аннотаций ... 314
10.1.2. Целевые элементы аннотаций ... 315
10.1.3. Использование аннотаций для настройки сериализации JSON ... 318
10.1.4. Объявление аннотаций ... 320
10.1.5. Метааннотации: управление обработкой аннотаций ... 321
10.1.6. Классы как параметры аннотаций ... 322
10.1.7. Обобщенные классы в параметрах аннотаций ... 323
10.2. Рефлексия: интроспекция объектов Kotlin во время выполнения ... 325
10.2.1. Механизм рефлексии в Kotlin: KClass, KCallable, KFunction и KProperty ... 326
10.2.2. Сериализация объектов с использованием механизма рефлексии ... 330
10.2.3. Настройка сериализации с помощью аннотаций ... 332
10.2.4. Парсинг формата JSON и десериализация объектов ... 336
10.2.5. Заключительный этап десериализации: callBy() и создание объектов с использованием рефлексии ... 340
10.3. Резюме ... 345

Глава 11. Конструирование DSL ... 346

11.1. От API к DSL ... 346
11.1.1. Понятие предметно-ориентированного языка ... 348
11.1.2. Внутренние предметно-ориентированные языки ... 349
11.1.3. Структура предметно-ориентированных языков ... 351
11.1.4. Создание разметки HTML с помощью внутреннего DSL ... 352
11.2. Создание структурированных API: лямбда-выражения с получателями в DSL ... 354
11.2.1. Лямбда-выражения с получателями и типы функций-расширений ... 354
11.2.2. Использование лямбда-выражений с получателями в построителях разметки HTML ... 358
11.2.3. Построители на Kotlin: поддержка абстракций и многократного использования ... 363

11.3. Гибкое вложение блоков с использованием соглашения «invoke» ... 366
11.3.1. Соглашение «invoke»: объекты, вызываемые как функции ... 367
11.3.2. Соглашение «invoke» и типы функций ... 367
11.3.3. Соглашение «invoke» в предметно-ориентированных языках: объявление зависимостей в Gradle ... 369
11.4. Предметно-ориентированные языки Kotlin на практике ... 371
11.4.1. Цепочки инфиксных вызовов: «should» в фреймворках тестирования ... 371
11.4.2. Определение расширений для простых типов: обработка дат ... 374
11.4.3. Члены-расширения: внутренний DSL для SQL ... 375
11.4.4. Anko: динамическое создание пользовательских интерфейсов в Android ... 378
11.5. Резюме ... 380

Приложение A. Сборка проектов на Kotlin ... 382

A.1. Сборка кода на Kotlin с помощью Gradle ... 382
A.1.1. Сборка Kotlin-приложений для Android с помощью Gradle ... 383
A.1.2. Сборка проектов с обработкой аннотаций ... 384
A.2. Сборка проектов на Kotlin с помощью Maven ... 384
A.3. Сборка кода на Kotlin с помощью Ant ... 385

Приложение B. Документирование кода на Kotlin ... 387

B.1. Документирующие комментарии в Kotlin ... 387
B.2. Создание документации с описанием API ... 389

Приложение C. Экосистема Kotlin ... 390

C.1. Тестирование ... 390
C.2. Внедрение зависимостей ... 391
C.3. Сериализация JSON ... 391
C.4. Клиенты HTTP ... 391
C.5. Веб-приложения ... 391
C.6. Доступ к базам данных ... 392
C.7. Утилиты и структуры данных ... 392
C.8. Настольные приложения ... 393

Предметный указатель ... 394