

O'REILLY®

От Java к Kotlin

Руководство по рефакторингу



bhv®

Дункан Макгрегор,
Нэт Прайс

Оглавление

Предисловие	11
Как устроена эта книга?	12
Как были выбраны темы?	12
Уровень сложности	13
Идеальный код	14
Форматирование кода	14
Условные обозначения	15
Использование примеров кода	16
Платформа онлайн-обучения O'Reilly	17
Как с нами связаться?	17
Благодарности	18
Благодарности Дункана	18
Благодарности Нэта	19
Глава 1. Введение	21
Сущность языка программирования	21
История стилей программирования на Java (на наш взгляд)	24
Первобытный стиль	24
Стиль JavaBeans	25
Корпоративный стиль	26
Инверсия абстракции	27
Современный стиль	29
Будущее	29
Сущность Kotlin	29
Рефакторинг на Kotlin	31
Принципы рефакторинга	32
Подразумеваем хорошее покрытие тестами	33
Фиксации изменений рассчитаны на Git Bisect	33
Над чем мы работаем?	34
Приступим!	34
Глава 2. Перевод проектов с Java на Kotlin	35
Стратегия	35
Добавление поддержки Kotlin в сборку Java	37
Другие системы сборки	40
Двигаемся дальше	40
Глава 3. От классов Java к классам Kotlin	41
Исходный код	41
Простой тип значений	41

Ограничения классов данных.....	48
Двигаемся дальше.....	52
Глава 4. От необязательных типов к обнуляемым.....	54
Как представить отсутствие чего-либо?.....	54
Рефакторинг от необязательных типов к обнуляемым	56
Итерация и цикл <i>for</i>	58
Рефакторинг с помощью <i>Expand-and-Contract</i>	64
Рефакторинг в код, характерный для <i>Kotlin</i>	64
Двигаемся дальше.....	71
Глава 5. От объектов <i>JavaBeans</i> к значениям.....	72
<i>JavaBeans</i>	72
Значения	73
Почему следует выбирать значения?.....	74
Рефакторинг экземпляров <i>JavaBeans</i> в значения	74
Равенство объектов	81
Двигаемся дальше.....	82
Глава 6. От коллекций <i>Java</i> к коллекциям <i>Kotlin</i>.....	83
Коллекции <i>Java</i>	83
Не изменяйте разделяемые коллекции.....	86
Коллекции <i>Kotlin</i>	86
Рефакторинг от коллекций <i>Java</i> к коллекциям <i>Kotlin</i>	89
Подправим код на <i>Java</i>	90
Преобразование в <i>Kotlin</i>	95
Двигаемся дальше.....	99
Глава 7. От действий к вычислениям	100
Функции	100
Вычисления	101
Вычисляемое свойство или функция?	102
Действия	102
Почему это должно нас волновать?	103
Процедурное программирование	104
Почему предпочтение отдается вычислениям?	104
Рефакторинг действий в вычисления	105
Существующий код.....	105
2015 был концом времени.....	109
Улучшенный проект.....	109
Конец игры.....	115
Двигаемся дальше.....	118
Глава 8. От статических методов к функциям верхнего уровня	119
Модификатор <i>Static</i> в <i>Java</i>	119
Статическое состояние	119
Функции верхнего уровня в <i>Kotlin</i> , объекты и объекты-компаньоны.....	120
Рефакторинг от статических методов к функциям верхнего уровня	122
Удобные функции.....	123
Перемещение на верхний уровень	126

Kotlin'изация.....	127
Перемещение функций верхнего уровня	129
Двигаемся дальше.....	129
Глава 9. От многострочных функций к однострочным	130
Попытка № 1: Замена встроенным выражением	132
Попытка № 2: Ввести функцию	133
Попытка № 3: Блок <i>let</i>	138
Попытка № 4: Отступление	140
Что должен возвращать метод <i>Parse</i> ?	143
Двигаемся дальше.....	143
Глава 10. От функций к функциям-расширениям	145
Функции и методы.....	145
Функции-расширения.....	147
Расширения и типы функций	148
Свойства-расширения	149
Преобразования	149
Как именовать преобразования?.....	150
Обнуляемые параметры	151
Обнуляемые получатели	152
Обобщенные расширения	154
Функции-расширения в качестве методов	155
Рефракторинг к функциям-расширениям.....	155
Двигаемся дальше.....	171
Глава 11. От методов к свойствам	172
Поля, аксессоры и свойства.....	172
Как выбрать?.....	176
Мутирующие свойства.....	178
Рефракторинг к свойствам.....	178
Двигаемся дальше.....	184
Глава 12. От функций к операторам	185
Базовый класс: <i>Money</i>	185
Добавление определяемого пользователем оператора.....	187
Вызов нашего оператора из существующего кода Kotlin	189
Операторы для существующих классов Java	191
Соглашения для аннотаций значений.....	191
Двигаемся дальше.....	195
Глава 13. От потоков к итерируемым объектам и последовательностям	196
Потоки в Java	196
Итераторы в Kotlin	198
Последовательности в Kotlin.....	199
Замена итераций и последовательностей	199
Множественные итерации	200
Выбор между потоками, итераторами и последовательностями	203
Алгебраическая трансформация	204
Рефракторинг от потоков к итераторам и последовательностям	207
Сначала итераторы.....	210
Последовательности.....	214

Извлечение части конвейера	218
Итоговая уборка	220
Двигаемся дальше.....	221
Глава 14. От накопления объектов к преобразованиям.....	222
Вычисление с использованием параметров накопления.....	222
Рефакторинг функций над немутуирующими данными.....	227
Давайте проделаем это снова	233
Обогащение обнаруженной нами абстракции	237
По поводу имен.....	238
Двигаемся дальше.....	239
Глава 15. От инкапсулированных коллекций к псевдонимам типов.....	241
Защитные копии	242
Создание коллекций доменов.....	243
Коллекции с другими свойствами.....	244
Рефакторинг инкапсулированных коллекций.....	245
Преобразование операций в распирения.....	246
Замена псевдонима типа	248
Рефакторинг коллекций с другими свойствами	254
Двигаемся дальше.....	257
Глава 16. От интерфейсов к функциям.....	259
Объектно-ориентированная инкапсуляция	260
Функциональная инкапсуляция	263
Функциональные типы в Java.....	266
Сочетание и совмещение	269
Выразительные типы функций.....	271
Сравнение подходов.....	272
Соединение	275
Объектно-ориентированный или функциональный?	276
Наследие Java.....	279
Прослеживаемость.....	280
Рефакторинг от интерфейсов к функциям.....	280
Введение функций.....	284
Двигаемся дальше.....	288
Глава 17. От мокинга к маппингу.....	289
Повреждение теста, вызванное моком.....	290
Замена мокинга маппингом	292
Но действительно ли мы хотим уйти от моков?	297
Двигаемся дальше.....	301
Глава 18. От открытых классов к запечатанным.....	302
Полиморфизм или запечатанные классы?.....	306
Преобразование интерфейса в запечатанный класс	307
Двигаемся дальше.....	315
Глава 19. От проверяемых исключений к типам результатов.....	316
Перезагрузка интернет-соединения	316
Обработка ошибок до появления методов, основанных на исключениях	318
Обработка ошибок с исключениями.....	319

Java и проверяемые исключения	320
Как должен быть оформлен сбой <i>parseInt</i> ?	321
Kotlin и исключения	322
За пределами исключений: функциональная обработка ошибок	322
Обработка ошибок в Kotlin	326
Рефакторинг от исключений к обработке ошибок	327
HTTP	328
Наша стратегия преобразования	329
Начнем с самого низа	330
Именованье	331
Контракт	335
Отступление	339
Еще исправления	343
Слой	349
Двигаемся дальше	351
Глава 20. От выполнения ввода/вывода к передаче данных	353
Прослушивание тестов	353
Ввод/вывод данных	356
Рефакторинг для удобства чтения	361
Эффективное написание	361
Еще о стимулах к работе	361
Эффективное считывание	365
Двигаемся дальше	370
Глава 21. От исключений к значениям	371
Определение того, что может пойти не так	371
Представление ошибок	379
А как насчет ввода/вывода?	387
Двигаемся дальше	388
Глава 22. От классов к функциям	389
Приемочный тест	389
Модульное тестирование	394
Заголовки	399
Различные разделители полей	408
Последовательности	415
Считывание из файла	421
Сравнение с Commons CSV	426
Двигаемся дальше	432
Глава 23. Продолжение путешествия	433
О сущностях	433
Функциональное мышление	434
Простая конструкция	436
Функциональное программирование и текстуальные рассуждения	437
Рефакторинг	438
Рефакторинг и функциональное мышление	439
Предметный указатель	441
Об авторах	443
Об изображении на обложке	444