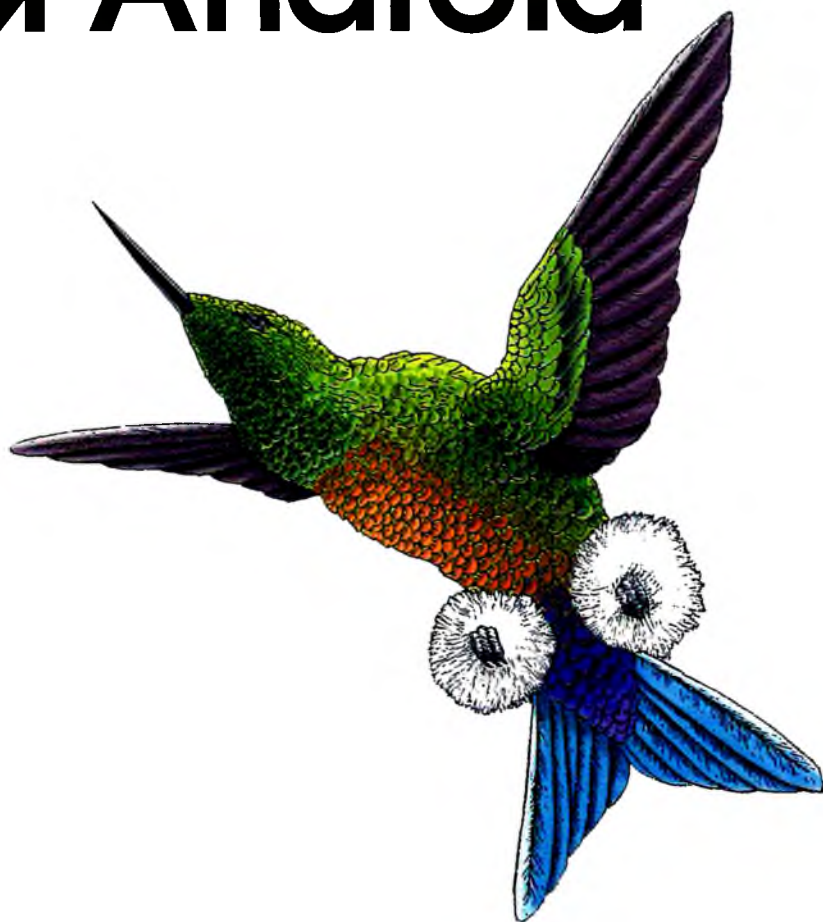


O'REILLY®

Программирование на Kotlin для Android



Пьер-Оливье Лоранс,
Аманда Хинчман-Домингес,
Дж. Блейк Мик, Майк Данн



Содержание

ОБ АВТОРАХ	11
ПРЕДИСЛОВИЕ	13
Кому адресована эта книга	14
Почему мы написали эту книгу.....	15
Как организована эта книга	15
Условные обозначения и соглашения	16
Примеры кода	17
Благодарности.....	17
ГЛАВА 1. Основы языка Kotlin.....	19
Система типов Kotlin	20
Примитивные типы.....	20
Null-безопасность	21
Тип <i>Unit</i>	23
Функциональные типы	24
Обобщенные типы	25
Переменные и функции	25
Переменные	26
Лямбда-выражения	26
Функции-расширения	27
Классы	29
Инициализация класса.....	29
Свойства.....	31
Модификатор <i>lateinit</i>	32
Свойства с отложенной инициализацией	34
Делегаты	35
Объекты-компаньоны	36
Классы данных	37
Классы перечислений	38
Запечатанные классы.....	40
Модификаторы видимости	41
Резюме	42

ГЛАВА 2. Фреймворк коллекций Kotlin	45
Основные понятия.....	45
Совместимость с Java	46
Изменяемость	46
Перегруженные операторы	48
Создание контейнеров.....	49
Функциональное программирование	50
Сравнение функционального и процедурного программирования: простой пример	50
Функциональное программирование в Android.....	51
Функции-преобразователи	52
Булевы функции.....	52
Фильтры.....	53
Функция <i>map</i>	53
Функция <i>flatMap</i>	55
Группировка	56
Сравнение итераторов и последовательностей.....	57
Пример.....	59
Проблема	59
Реализация	60
Резюме	66
 ГЛАВА 3. Основы Android	67
Стек Android.....	67
Аппаратное обеспечение.....	67
Ядро.....	68
Системные службы	68
Среда выполнения Android	69
Приложения.....	69
Прикладное окружение Android.....	69
Намерения и фильтры намерений	70
Контекст.....	73
Компоненты приложения Android: строительные блоки	75
Компонент <i>Activity</i> и его друзья	76
Службы	80
Провайдеры контента	85
<i>BroadcastReceiver</i>	86
Архитектуры приложений Android.....	88
MVC: основы.....	88
Виджеты.....	89
Локальная модель	89
Паттерны Android.....	90
Model-View-Intent	90
Model-View-Presenter	90
Model-View-ViewModel.....	91
Резюме	92

ГЛАВА 4. Параллельное программирование в Android	94
Потокобезопасность	95
Атомарность	95
Видимость	96
Модель многопоточного выполнения Android	97
Пропуск кадров	98
Утечка памяти	101
Инструменты для управления потоками	103
<code>Looper/Handler</code>	104
Исполнители <code>Executors</code> и объекты <code>ExecutorService</code>	106
Инструменты для управления заданиями	107
<code>JobScheduler</code>	109
<code>WorkManager</code>	111
Резюме	112
 ГЛАВА 5. Потокобезопасность	 113
Пример проблемы, связанной с потокобезопасностью	113
Инварианты	115
Мьютексы	116
Потокобезопасные коллекции	116
Привязка к потоку	119
Конфликт потоков	120
Сравнение блокирующего и неблокирующего вызовов	121
Очереди работ	122
Противодавление	123
Резюме	125
 ГЛАВА 6. Организация параллелизма с использованием обратных вызовов	 127
Пример: функция обработки покупок	128
Создание приложения	130
Компонент <code>ViewModel</code>	130
Представление	131
Реализация логики	135
Обсуждение	136
Ограничения модели многопоточного выполнения	138
Резюме	139
 ГЛАВА 7. Сопрограммы	 141
Что такое сопрограмма?	141
Наша первая сопрограмма	142
Функция <code>async</code>	144
Краткий обзор структурированного параллелизма	146
Связь "родитель — потомок" в структурированном параллелизме	148
<code>CoroutineScope</code> и <code>CoroutineContext</code>	150
Функции, поддерживающие возможность приостановки	155

ГЛАВА 4. Параллельное программирование в Android.....	94
Потокобезопасность.....	95
Атомарность.....	95
Видимость.....	96
Модель многопоточного выполнения Android.....	97
Пропуск кадров.....	98
Утечка памяти.....	101
Инструменты для управления потоками.....	103
<i>Looper/Handler</i>	104
Исполнители <i>Executors</i> и объекты <i>ExecutorService</i>	106
Инструменты для управления заданиями.....	107
<i>JobScheduler</i>	109
<i>WorkManager</i>	111
Резюме.....	112
 ГЛАВА 5. Потокобезопасность	 113
Пример проблемы, связанной с потокобезопасностью.....	113
Инварианты.....	115
Мьютексы.....	116
Потокобезопасные коллекции.....	116
Привязка к потоку.....	119
Конфликт потоков.....	120
Сравнение блокирующего и неблокирующего вызовов.....	121
Очереди работ.....	122
Противодавление.....	123
Резюме.....	125
 ГЛАВА 6. Организация параллелизма с использованием обратных вызовов	 127
Пример: функция обработки покупок.....	128
Создание приложения.....	130
Компонент <i>ViewModel</i>	130
Представление.....	131
Реализация логики.....	135
Обсуждение.....	136
Ограничения модели многопоточного выполнения.....	138
Резюме.....	139
 ГЛАВА 7. Сопрограммы	 141
Что такое сопрограмма?.....	141
Наша первая сопрограмма.....	142
Функция <i>async</i>	144
Краткий обзор структурированного параллелизма.....	146
Связь "родитель — потомок" в структурированном параллелизме.....	148
<i>CoroutineScope</i> и <i>CoroutineContext</i>	150
Функции, поддерживающие возможность приостановки.....	155

Функции, поддерживающие возможность приостановки, "под капотом"	156
Использование сопрограмм и функций, поддерживающих возможность приостановки: практический пример	160
Не ошибитесь с модификатором <i>suspend</i>	163
Резюме	164

ГЛАВА 8. Структурированный параллелизм и сопрограммы

166

Функции, поддерживающие возможность приостановки	166
Настройка места действия	167
Традиционный подход с использованием <i>java.util.concurrent.ExecutorService</i>	168
Вспомним, что такое <i>HandlerThread</i>	172
Использование приостанавливаемых функций и сопрограмм	175
Сравнение приостанавливаемых и традиционной многопоточности: итоги	179
Отмена	179
Жизненный цикл сопрограмм	180
Отмена сопрограммы	182
Отмена задачи, делегированной сторонней библиотеке	184
Сопрограммы, которые действуют согласованно, чтобы их можно было отменить	188
Функцию <i>delay()</i> можно отменить	190
Обработка отмены	191
Причины отмены	192
Супервизия	195
Функция <i>supervisorScope</i>	197
Параллельная декомпозиция	197
Автоматическая отмена	199
Обработка исключений	199
Необработанные и открытые исключения	199
Открытые исключения	201
Необработанные исключения	204
Резюме	207
Размышления напоследок	208

ГЛАВА 9. Каналы

209

Обзор каналов	209
Рандеву-канал	211
Неограниченный канал	215
Объединенный канал	216
Буферизованный канал	217
Функция <i>produce</i>	218
Взаимодействующие последовательные процессы	219
Модель и архитектура	219
Первая реализация	220
Выражение <i>select</i>	225
Собираем все воедино	227

Мультиплексор и демуплексор	228
Проверка производительности	229
Противодавление	231
Сходства с моделью акторов	232
Последовательное выполнение внутри процесса	232
Размышления напоследок	233
Взаимоблокировки в CSP	233
Каналы и взаимоблокировки	236
Ограничения каналов	236
"Горячие" каналы	238
Резюме	239
ГЛАВА 10. Потoki	241
Введение в потоки	241
Более реалистичный пример	242
Операторы	244
Терминальные операторы	245
Примеры использования холодного потока	246
Вариант 1: интерфейс с API на базе функции обратного вызова	246
Вариант 2: параллельное преобразование потока значений	251
Вариант 3: создание собственного оператора	253
Обработка ошибок	257
Блок <i>try/catch</i>	257
Разделение ответственности важно	260
Нарушение прозрачности исключения	260
Оператор <i>catch</i>	261
Материализация исключений	264
Горячие потоки и <i>SharedFlow</i>	267
Создаем <i>SharedFlow</i>	268
Регистрируем подписчика	268
Отправляем значения в <i>SharedFlow</i>	269
Использование <i>SharedFlow</i> для потоковой передачи данных	269
Использование <i>SharedFlow</i> в качестве шины событий	275
<i>StateFlow</i> : специализированная версия <i>SharedFlow</i>	276
Пример использования <i>StateFlow</i>	277
Резюме	279
ГЛАВА 11. Вопросы производительности и инструменты профилирования Android	280
Android Profiler	282
Network Profiler	285
CPU Profiler	291
Energy Profiler	301
Memory Profiler	303
Обнаружение утечек памяти с помощью LeakCanary	308
Резюме	312

ГЛАВА 12. Снижение потребления ресурсов за счет оптимизации производительности.....	314
Достижение плоской иерархии представлений с помощью <i>ConstraintLayout</i>	315
Сокращение количества операций рисования с помощью экземпляров класса <i>Drawable</i>	319
Минимизация данных в сетевых вызовах.....	324
Организация пула и кэширование объектов <i>Bitmap</i>	324
Избавляемся от ненужной работы	326
Использование статических функций	329
Минификация и обфускация с <i>R8</i> и <i>ProGuard</i>	329
Резюме	331
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	333