

ОГЛАВЛЕНИЕ

Предисловие	14
Присоединяйся!	16
Как работать с книгой	17
От издательства	18
Глава 0. Установка и настройка рабочего окружения. Основы Dart	19
0.1. Установка и настройка рабочего окружения	19
0.1.1. ОС Windows	19
0.1.2. ОС macOS	22
0.1.3. ОС Ubuntu 20.04 LTS+ и Debian 11+	28
0.1.4. Установка и настройка Android Studio	32
0.2. Создание первого проекта на Dart	37
0.3. Базовые типы данных, модификаторы доступа и null-safety	41
0.3.1. Числа (int, double)	43
0.3.2. Строки (String)	44
0.3.3. Логические значения (bool)	45
0.3.4. Списки (List)	45
0.3.5. Записи (Record)	47
0.3.6. Множества (Set)	48
0.3.7. Таблицы/карты (Map)	49
0.3.8. Runes и Symbols	52
0.3.9. Модификаторы final, const и late	52
0.3.10. Null-безопасность (null-safety)	53
0.3.11. Тип данных dynamic vs Object	56
0.4. Основные операторы и pattern matching	57
0.4.1. Основные операторы Dart	57
0.4.2. Что такое Pattern Matching и Destructuring	61
0.4.3. Деструктурирование списка	62
0.4.4. Деструктурирование записи	63
0.4.5. Деструктурирование таблицы/карты	63
0.4.6. Деструктурирование экземпляра класса	65
0.5. Управление потоком выполнения кода	66
0.5.1. Условный оператор if	66
0.5.2. Оператор if-case	68

0.5.3. Тернарный оператор ?:	70
0.5.4. Оператор ??	70
0.5.5. Операторы циклов for, for-in, while и do-while	71
0.5.6. Операторы потока выполнения break, continue, return	73
0.5.7. Оператор выбора потока выполнения switch-case	74
0.5.8. Null-aware elements	76
0.6. Функции	78
0.6.1. Объявление входных аргументов функции	79
0.6.2. Необязательные аргументы функции по умолчанию	81
0.6.3. Обращение к функции с помощью переменной	82
0.6.4. Функция как входной аргумент другой функции	82
0.6.5. Type Alias	83
0.6.6. Анонимные и стрелочные функции	85
0.7. Библиотеки	86
0.7.1. Импортирование кода из файла с расширением .dart	86
0.7.2. Импортирование части функциональности	89
0.7.3. Создание и использование библиотеки	90
Проект: игра «Тетрис» v. 0	92
Разработка библиотеки ansi_cli_helper.dart	93
Разработка библиотеки blocks.dart	94
Разработка библиотеки board.dart	98
Компоновка библиотек и запуск игры	104
Задания на модификацию проекта	106
0.8. Объектно-ориентированное программирование	106
0.8.1. Конструктор класса	110
0.8.2. Статические переменные и методы класса	114
0.8.3. Методы расширения (extension methods)	115
0.8.4. Наследование и переопределение методов	116
0.8.5. Абстрактный класс и интерфейс	123
0.8.6. Модификаторы класса	127
0.8.7. Запечатанные (sealed) классы	128
0.8.8. Миксины и модификатор класса mixin	132
0.8.9. Enum (перечисления)	136
0.9. Exceptions (исключения)	138
0.9.1. Конструкция try...catch...finally	138
0.9.2. Генерация исключений и ошибок	140
0.9.3. Пользовательские исключения и ошибки	141
0.9.4. Assert (утверждение)	142
0.10. Асинхронное программирование и изоляты	143
0.10.1. Базовая концепция Event Loop-архитектуры в Dart	143
0.10.2. Очереди и цикл событий в Dart	144
0.10.3. Что такое асинхронное программирование	146

0.10.4. Future API, async и await.....	147
0.10.5. Stream (поток)	154
0.10.6. Isolate (изоляты)	160
0.10.7. Async или Isolate?	165
Проект: игра «Тетрис» v. 1	166
Рефакторинг библиотеки <code>ansi_cli_helper.dart</code>	166
Рефакторинг библиотеки <code>blocks.dart</code>	169
Рефакторинг библиотеки <code>board.dart</code>	172
Разработка библиотеки <code>game.dart</code>	176
Компоновка библиотек и запуск игры	177
Задания на модификацию проекта	178
Резюме	178
Вопросы для самопроверки	179
Глава 1. Краткая история и принципы работы Flutter.....	180
1.1. Краткая история и основные нюансы.....	180
1.2. Как обстоят дела с разработкой под Web.....	185
1.3. Создание первого проекта и его запуск.....	185
1.4. Структура проекта на Flutter.....	191
1.5. Структура файла <code>pubspec.yaml</code>	193
1.6. Типы виджетов во Flutter	195
1.6.1. <code>StatelessWidget</code>	195
1.6.2. <code>StatefulWidget</code>	202
1.7. В недрах <code>BuildContext</code>	212
1.8. Передача информации по дереву элементов.....	228
1.8.1. <code>InheritedWidget</code>	230
1.8.2. <code>InheritedModel</code>	233
1.8.3. <code>InheritedNotifier</code>	236
1.8.4. <code>ChangeNotifier</code> и <code>ListenableBuilder</code>	242
1.9. Зачем нужны ключи.....	246
1.10. Жизненный цикл приложения	252
Резюме	256
Вопросы для самопроверки	257
Глава 2. Основные виджеты, их компоновка и работа с assets	258
2.1. Стили виджетов — Material vs Cupertino	258
2.2. Виджеты-«коробки»	264
2.2.1. «Коробки» для задания размеров.....	264
2.2.2. «Коробки» для отрисовки.....	268
2.2.3. Виджет для положения в пространстве	275
2.2.4. Виджет <code>Container</code>	277
2.3. Виджеты компоновки.....	278
2.3.1. Виджеты <code>Row</code> и <code>Column</code>	278
2.3.2. Виджеты <code>Flexible</code> , <code>Expanded</code> и <code>Spacer</code>	281

2.3.3. Виджет Wrap	285
2.3.4. Виджеты Stack и Positioned	286
2.3.5. Виджеты Align и Center	290
2.4. Виджеты выбора и ввода данных	293
2.4.1. Виджет TextField	293
2.4.2. Виджет Radio и его вариации	297
2.4.3. Виджет Checkbox и его вариации	302
2.4.4. Виджеты Switch и SwitchListTile	303
2.4.5. Виджет DropdownMenu	305
2.4.6. Виджет Slider	309
2.4.7. Виджет ввода времени	311
2.4.8. Виджет ввода даты	314
2.5. Виджеты кнопок	316
2.5.1. Виджет ElevatedButton	316
2.5.2. Виджеты FilledButton и OutlinedButton	319
2.5.3. Виджет TextButton	322
2.5.4. Виджет IconButton	323
2.5.5. Виджет SegmentedButton	325
2.6. Виджеты отображения данных и работа с assets	329
2.6.1. Виджеты Text и RichText	329
2.6.2. Работа со шрифтами и их импорт посредством assets	333
2.6.3. Виджет Image и его связь с assets	337
2.6.4. Виджет Icon и добавление своих значков	340
2.6.5. Assets и JSON-файлы	344
2.7. Скролируемые виджеты	347
2.7.1. Виджет SingleChildScrollView	348
2.7.2. Виджеты ListView и GridView	350
2.7.3. Виджеты PageView и Carousel	353
2.7.4. Кастомная полоса прокрутки и Slivers	357
2.8. Scaffold и его составные виджеты	362
2.8.1. Виджет AppBar	362
2.8.2. Виджеты NavigationBar и BottomAppBar	369
2.8.3. Виджет NavigationDrawer	375
2.8.4. Виджет FloatingActionButton	379
2.8.5. Виджет BottomSheet	383
2.8.6. Виджет SnackBar и способы показа сообщения пользователю	388
Проект: игра «Тетрис» v.2. Портирование на Flutter	391
Перенос и рефакторинг файлов	391
Реализация на Flutter	394
Задания на модификацию проекта	398
Резюме	398
Вопросы для самопроверки	398

Глава 3. Управление состоянием	399
3.1. Типы состояния приложения	399
3.1.1. Состояние приложения (App state)	400
3.1.2. Эфемерное состояние (Ephemeral state)	400
3.2. Инструменты Flutter для работы с состоянием приложения	402
3.2.1. setState	402
3.2.2. ChangeNotifier	403
3.2.3. ValueNotifier	406
3.2.4. InheritedWidget	407
3.3. Паттерны для управления состоянием	412
3.3.1. BLoC (Business Logic Component)	412
3.3.2. MVP (Model-View-Presenter)	416
3.3.3. MVC (Model-View-Controller)	419
3.3.4. MVVM (Model-View-ViewModel)	422
3.3.5. Сравнение MVP, MVC и MVVM	426
Проект: игра «Тетрис» v. 3. Переход на ChangeNotifier	426
Задания на модификацию проекта	431
Резюме	432
Вопросы для самопроверки	432
Глава 4. Навигация	433
4.1. Основные концепции навигации во Flutter	433
4.1.1. Разница между императивной и декларативной навигацией	433
4.1.2. Императивная навигация (Navigator 1.0)	435
4.1.3. Декларативная навигация (Navigator 2.0)	437
4.1.4. Направление навигации	438
4.2. Основные элементы навигации во Flutter	440
4.2.1. Класс Navigator	440
4.2.2. Класс Route	443
4.2.3. Класс NavigatorObserver	447
4.3. Именованные маршруты	451
4.3.1. onUnknownRoute	456
4.3.2. onGenerateRoute	456
4.4. Навигация без контекста — GlobalKey и NavigatorState	458
4.5. Инструменты декларативной навигации	460
4.5.1. RouteInformationProvider	461
4.5.2. RouteInformationParser	461
4.5.3. RouterDelegate	463
4.5.4. Настройка MyRouterDelegate	464
4.5.5. PopNavigatorRouterDelegateMixin	468
4.5.6. RouterConfig	469
4.6. Передача информации между экранами	470
4.6.1. Императивная передача с помощью RouteSettings	471
4.6.2. Декларативный подход к передаче информации	473

4.7. Вложенная навигация (nested navigation)	476
4.7.1. Императивная реализация	476
4.7.2. Декларативная реализация.....	480
Проект: игра «Тетрис» v. 4. Добавление навигации.....	485
Задания на модификацию проекта	491
Резюме	491
Вопросы для самопроверки	492
Глава 5. Работа с сетью	493
5.1. Клиент-серверная архитектура.....	493
5.1.1. API.....	494
5.1.2. HTTP-запросы	494
5.1.3. HTTP-статусы ответов сервера.....	496
5.2. Встроенный инструмент Flutter для работы с HTTP	497
5.2.1. Стандартная библиотека dart:io.....	497
5.2.2. Ограничения и проблемы dart:io.....	498
5.3. Пакет (библиотека) HTTP	498
5.3.1. Первый GET-запрос в сеть	499
5.3.2. Обработка ответа сервера	500
5.3.3. Парсинг JSON-моделей	502
5.3.4. Метод POST	506
5.3.5. Методы PUT и PATCH.....	507
5.3.6. Метод DELETE	508
5.3.7. Обработка HTTP-статусов ответов от сервера	509
5.3.8. Заголовки (headers).....	510
5.3.9. Перехватчик запросов Interceptor.....	512
5.4. Веб-сокеты.....	515
5.4.1. Начало работы с веб-сокетами	516
5.4.2. Чтение данных из веб-сокета.....	516
5.4.3. Отправка данных через сокет	517
5.4.4. Отображение данных из сокета на экране.....	518
Проект: игра «Тетрис» v. 5. Работа с сетью	523
Запуск серверной части	523
Базовая концепция приложения и структура папок проекта	523
Разработка DI-контейнера (Dependency Injection).....	525
Разработка HTTP-клиента.....	527
Отображение списка лучших результатов.....	529
Разработка экрана с главным меню.....	537
Создание нового пользователя.....	539
Модификация экрана с игрой	550
Задания на модификацию проекта	553
Резюме	554
Вопросы для самопроверки	554

Глава 6. Локальное хранение данных	555
6.1. SharedPreferences	555
6.1.1. Начало работы.....	556
6.1.2. Запись данных.....	557
6.1.3. Запись пользовательских данных	558
6.1.4. Чтение данных	559
6.1.5. Чтение кастомных данных.....	560
6.1.6. Удаление данных и очистка	562
6.1.7. Пример интеграции с интерфейсом	563
6.1.8. Как давать имена ключам.....	566
6.1.9. Что принято и что не принято хранить в SharedPreferences.....	566
6.1.10. Новые API — SharedPreferencesAsync и WithCache	568
6.2. Secure Storage	569
6.2.1. Начало работы.....	569
6.2.2. Запись и чтение данных.....	570
6.2.3. Чтение и запись типизированных данных	571
6.2.4. Параметры кастомизации — AndroidOptions.....	572
6.2.5. Параметры кастомизации — IOSOptions	573
6.2.6. Параметры кастомизации — другие платформы	575
6.3. SQLite.....	575
6.3.1. Реляционные базы данных и язык SQL.....	576
6.3.2. Начало работы.....	577
6.3.3. Создание первой таблицы.....	578
6.3.4. Запись данных.....	580
6.3.5. Чтение данных	582
6.3.6. Обновление данных	585
6.3.7. Удаление данных	585
6.3.8. Миграции.....	586
Проект: игра «Тетрис» v. 6. Сохранение, кэширование данных и игра без Интернета	587
Разработка сервиса локального хранилища	587
Рефакторинг контейнера зависимостей	589
Рефакторинг запуска приложения	591
Рефакторинг HTTP-клиента	593
Рефакторинг функционала UserRepository.....	594
Задания на модификацию проекта.....	598
Резюме.....	599
Вопросы для самопроверки	599
Глава 7. Тестирование приложений	600
7.1. Теория тестирования.....	600
7.1.1. Какими бывают тесты.....	601
7.1.2. Подходы к тестированию	601

7.1.3. Паттерн AAA.....	602
7.1.4. Тестирование во Flutter.....	603
7.2. Unit-тесты.....	604
7.2.1. Основные функции для написания unit-тестов.....	605
7.2.2. Тестирование калькулятора.....	606
7.3. Виджет-тесты	614
7.3.1. WidgetTester	615
7.3.2. testWidgets().....	616
7.3.3. Finder-классы	617
7.3.4. Matcher-классы.....	619
7.3.5. Тестируем калькулятор	620
7.4. Асинхронные виджет-тесты.....	621
7.5. Интеграционные тесты	624
Проект: игра «Тетрис» v. 7. Тестирование.....	626
Задания на модификацию проекта	632
Резюме	633
Вопросы для самопроверки	633
Глава 8. Локализация приложения	634
8.1. Интернационализация vs локализация	635
8.2. Подключение и настройка библиотеки flutter_localizations.....	635
8.3. Интерполяция строк (placeholder)	640
8.4. Плюрализация строк (plural).....	643
8.5. Еще один механизм плюрализации (select)	645
8.6. Форматирование даты и времени	648
8.7. Форматирование чисел и данных о валюте	651
8.8. Локализация для iOS (продуктов Apple).....	654
8.9. Экранирование фигурных скобок.....	654
Проект: игра «Тетрис» v. 8. Локализация игры.....	655
Задания на модификацию проекта	660
Резюме	661
Вопросы для самопроверки	661
Глава 9. Сборка приложения.....	662
9.1. Режимы сборок	662
9.1.1. Debug — отладочный режим	662
9.1.2. Profile — режим для анализа производительности	666
9.1.3. Release — режим для публикации	667
9.1.4. Основные различия release, profile и debug.....	667
9.2. Подпись сборки под Android	668
9.2.1. Android APK	668
9.2.2. Android ABB	668
9.2.3. APK vs AAB.....	669
9.2.4. Процесс подписи release-сборки приложения	669

9.3. Подпись сборки под iOS/MacOS.....	675
9.3.1. Регистрация уникального идентификатора Bundle ID	675
9.3.2. Создание приложения в App Store Connect	675
9.3.3. Настройка проекта приложения в XCode перед релиз-сборкой.....	676
9.3.4. Создание релизного архива.....	677
9.4. Подпись сборки под ОС «Аврора»	678
9.4.1. Генерация файла-запроса на сертификат	679
9.4.2. Запрос сертификата	680
9.4.3. Сборка и подпись приложения	680
Резюме.....	682
Вопросы для самопроверки	682
Заключение	683
Лабораторные практикумы к книге.....	685
Лабораторный практикум по Dart	685
Лабораторный практикум по Flutter	686