

УДК 004Dart
ББК 32.973-018.2
Б19

Бакетт, Крис.

Б19 Dart в действии / К. Бакетт ; пер. с англ. — 2-е изд., эл. — 1 файл pdf : 530 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-314-1

Dart — язык программирования для разработки веб-приложений, созданный компанией Google. Он обладает современными объектно-ориентированными средствами, как Java или C#, не отказываясь при этом от свойственных JavaScript динамичности и ориентированности на функциональное программирование. Написанные на Dart приложения транслируются в JavaScript, но могут исполняться и непосредственно в браузерах, поддерживающих Dart. В комплекте с Dart поставляются библиотеки и инструментальные средства промышленного качества. На Dart могут быть написаны как клиентская, так и серверная часть приложения, что упрощает процесс разработки.

В этой книге вы познакомитесь с языком Dart и научитесь использовать его для создания приложений, работающих в браузере, — в настольной или мобильной ОС. Это не просто учебное пособие по языку, довольно быстро автор переходит к техническим аспектам работы с Dart. На большинство вопросов, возникающих при чтении, тут же даются ответы!

Издание предназначено веб-программистам разной квалификации, в том числе малознакомым с объектно-ориентированным программированием.

УДК 004Dart
ББК 32.973-018.2

Электронное издание на основе печатного издания: Dart в действии / К. Бакетт ; пер. с англ. — Москва : ДМК Пресс, 2013. — 528 с. — ISBN 978-5-94074-918-9. — Текст : непосредственный.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

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.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

ISBN 978-5-89818-314-1

© 2013 by Manning Publications Co.
© Оформление, перевод ДМК Пресс, 2013



Содержание

Предисловие	14
Вступление	16
Благодарности	18
Об этой книге	20
Об иллюстрации на обложке	25
ЧАСТЬ I. ВВЕДЕНИЕ В DART	26
Глава 1. Здравствуй, Dart	27
1.1. Что такое Dart?	27
1.1.1. Знакомый синтаксис помогает в освоении языка	29
1.1.2. Архитектура одностраничного приложения	30
1.2. Первый взгляд на язык Dart	32
1.2.1. Строковая интерполяция	32
1.2.2. Факультативные типы в действии	34
1.2.3. Традиционная структура на основе классов	36
1.2.4. Определение подразумеваемого интерфейса	38
1.2.5. Фабричные конструкторы для предоставления реализации по умолчанию	39
1.2.6. Библиотеки и область видимости	40
1.2.7. Функции как полноценные объекты	43
1.2.8. Параллелизм с помощью изоляторов	45
1.3. Веб-программирование на языке Dart	46
1.3.1. dart:html: удобная библиотека для работы с моделью DOM браузера	47
1.3.2. Dart и HTML5	48
1.4. Инструментальная экосистема Dart	50
1.4.1. Редактор Dart	50
1.4.2. Виртуальная машина Dart	51
1.4.3. Dartium	51
1.4.4. dart2js: конвертер Dart в JavaScript	51

1.4.5. Управление пакетами с помощью pub	52
1.5. Резюме	53

Глава 2. Программа «Здравствуй, мир» на Dart..... 54

2.1. VM Dart для командных приложений	55
2.2. Программа «Здравствуй, мир» в редакторе Dart	57
2.2.1. Знакомство с инструментами, встроенными в Редактор Dart Editor	58
2.2.2. Dart-скрипты и HTML-файлы	61
2.2.3. Запуск приложения «Здравствуй, мир» в Dartium.....	62
2.2.4. Использование dart2js для конвертации в JavaScript	63
2.2.5. Генерация документации с помощью dartdoc	66
2.2.6. Отладка Dart-кода с помощью точек останова	66
2.3. Импорт библиотек для работы с пользовательским интерфейсом в браузере.....	68
2.3.1. Импорт библиотек Dart.....	69
2.3.2. Доступ к элементам DOM с помощью dart:html	70
2.3.3. Динамическое добавление новых элементов на страницу	71
2.4. Резюме	72

Глава 3. Создание и тестирование

Dart-приложения 74

3.1. Конструирование пользовательского интерфейса с помощью dart:html	75
3.1.1. Начальный HTML-файл	76
3.1.2. Создание элементов с помощью dart:html	77
3.1.3. Создание экземпляра Element из фрагмента HTML ...	78
3.1.4. Создание элементов по имени тега	80
3.1.5. Добавление элементов в HTML-документ.....	82
3.2. Добавление интерактивности с помощью событий браузера	86
3.2.1. Добавление предмета в список по нажатии кнопки ...	86
3.2.2. Применение гибкого синтаксиса функций в Dart для обработки событий.....	87
3.2.3. Реагирование на события браузера.....	90
3.2.4. Рефакторинг прослушивателя событий для повторного использования.....	91

3.2.5. Запрос HTML-элементов в dart:html	92
3.3. Инкапсуляция структуры и функциональности с помощью классов.....	95
3.3.1. Классы в Dart не таят неожиданностей	96
3.3.2. Конструирование класса PackItem.....	97
3.3.3. Инкапсуляция функциональности с помощью методов чтения и установки.....	99
3.4. Автономное тестирование программы.....	103
3.4.1. Создание автономных тестов	105
3.4.2. Определение ожидаемых результатов теста	106
3.4.3. Создание пользовательского сравнителя	107
3.5. Резюме.....	109

ЧАСТЬ II. ЯЗЫК DART..... 111

Глава 4. Функции и замыкания 112

4.1. Функции в Dart.....	113
4.1.1. Тип возвращаемого функцией значения и ключевое слово return	116
4.1.2. Передача функции данных с помощью параметров	119
4.2. Функции как полноценные объекты.....	126
4.2.1. Объявления локальных функций.....	128
4.2.2. Определение строгого типа функции	134
4.3. Замыкания.....	137
4.4. Резюме.....	140

Глава 5. Библиотеки и ограничение доступа..... 142

5.1. Определение и импорт библиотеки	143
5.1.1. Определение библиотеки с помощью ключевого слова library	145
5.1.2. Импорт библиотек.....	147
5.2. Скрытие функциональности путем ограничения доступа к частям библиотеки	155
5.2.1. Ограничение доступа в классах.....	157
5.2.2. Использование закрытых функций в библиотеках....	162
5.3. Организация исходного кода библиотеки	163
5.3.1. Ключевые слова part и part of.....	164
5.4. Упаковка библиотек.....	168

5.5. Скрипты – это исполняемые библиотеки	171
5.6. Резюме	173

Глава 6. Классы и интерфейсы

6.1. Определение простого класса	176
6.1.1. Программирование относительно интерфейса класса	177
6.1.2. Формализация интерфейса путем явного его определения	180
6.1.3. Реализация нескольких интерфейсов	181
6.1.4. Объявление акцессоров свойств	182
6.2. Конструирование классов и интерфейсов	184
6.2.1. Конструирование экземпляров класса	185
6.2.2. Проектирование и использование классов с несколькими конструкторами	187
6.2.3. Использование фабричных конструкторов для создания экземпляров абстрактных классов	187
6.2.4. Применение фабричных конструкторов для повторного использования объектов	190
6.2.5. Использование статических методов и свойств совместно с фабричными конструкторами	191
6.3. Создание константных классов с неизменяемыми полями	194
6.3.1. Финальные значения и свойства	195
6.3.2. Блок инициализации конструктора	195
6.3.3. Использование ключевого слова const для создания константного конструктора	196
6.4. Резюме	197

Глава 7. Расширение классов и интерфейсов

7.1. Расширение классов с помощью наследования	200
7.1.1. Наследование класса	201
7.1.2. Наследование конструкторов	203
7.1.3. Переопределение методов и свойств	205
7.1.4. Включение абстрактных классов в иерархию наследования	206
7.2. Все является объектом	210
7.2.1. Проверка отношения «является» Object	210
7.2.2. Использование отношения «является» применительно к Object	212

7.2.3. Использование метода toString(), унаследованного от класса Object	213
7.2.4. Перехват обращений к методу noSuchMethod()	215
7.2.5. Прочая функциональность класса Object	218
7.3. Знакомство с типом dynamic.....	219
7.3.1. Использование аннотации типа dynamic.....	220
7.4. Резюме	221

Глава 8. Классы коллекций..... 223

8.1. Работа с коллекциями данных.....	224
8.1.1. Коллекции объектов	226
8.1.2. Использование конкретных реализаций интерфейса Collection.....	230
8.1.3. Создание специализированных коллекций с помощью обобщенных типов.....	233
8.1.4. Хранение списков пар ключ–значение в обобщенных словарях	237
8.2. Создание обобщенных классов	242
8.2.1. Определение обобщенного класса.....	242
8.2.2. Использование своего обобщенного класса.....	244
8.2.3. Ограничения на параметрические типы.....	245
8.3. Перегрузка операторов	246
8.3.1. Перегрузка операторов сравнения.....	247
8.3.2. Неожиданное применение перегрузки операторов	249
8.3.3. Перегрузка операторов доступа по индексу	249
8.4. Резюме	252

Глава 9. Асинхронное программирование с применением обратных вызовов и будущих значений 254

9.1. Почему веб-приложение должно быть асинхронным.....	256
9.1.1. Преобразование приложения в асинхронное	259
9.2. Использование обратных вызовов в асинхронном программировании	263
9.2.1. Добавление асинхронных обратных вызовов в Лотерею Dart.....	265
9.2.2. Ожидание завершения всех асинхронных обратных вызовов перед продолжением	266

9.2.3. Вложенные обратные вызовы как средство управления порядком асинхронного выполнения	269
9.3. Знакомство с типами Future и Completer	271
9.3.1. Передача будущих значений из одного места программы в другое	274
9.3.2. Упорядочение асинхронных вызовов путем сцепления будущих значений	275
9.3.3. Ожидание завершения всех запущенных операций получения будущих значений	276
9.3.4. Преобразование обычных значений в будущее	278
9.4. Автономное тестирование асинхронных API.....	280
9.4.1. Тестирование асинхронных функций обратного вызова	282
9.4.2. Тестирование будущих значений	283
9.5. Резюме	285

ЧАСТЬ III. КЛИЕНТСКИЕ DART-ПРИЛОЖЕНИЯ

Глава 10. Создание веб-приложения на Dart

10.1. Структура одностраничного веб-приложения	289
10.1.1. Приложение DartExpense – постановка задачи	290
10.1.2. Структура Dart-приложения.....	293
10.1.3. Поток выполнения в Dart-приложении	296
10.2. Конструирование пользовательского интерфейса с помощью dart:html	299
10.2.1. Интерфейс Element	299
10.2.2. Конструкторы элементов в действии	303
10.2.3. Организация взаимодействия с представлениями и элементами	305
10.2.4. Построение простой обобщенной сетки	309
10.3. Обработка браузерных событий с помощью dart:html ...	313
10.3.1. Управление порядком прохождения события в браузере	315
10.3.2. Наиболее распространенные типы событий	317
10.4. Резюме	319

Глава 11. Навигация и локальное хранение

данных	321
11.1. Интеграция навигации с браузером	323

11.1.1. Применение функции <code>pushState()</code> для добавления элементов в браузерную историю навигации	324
11.1.2. Подписка на событие <code>popState</code>	326
11.2. Использование куков браузера для повышения удобства работы	329
11.2.1. Сохранение данных в кукe	330
11.2.2. Чтение данных из кука	332
11.3. Локальное хранение данных с помощью Web Storage API	334
11.3.1. Преобразование объектов Dart в формат JSON	335
11.3.2. Преобразование JSON-строк в Dart-объекты	340
11.3.3. Сохранение данных в браузере	341
11.4. Резюме	346

Глава 12. Взаимодействие с другими системами и языками

12.1. Взаимодействие с JavaScript	349
12.1.1. Отправка данных из Dart в JavaScript	352
12.1.2. Получение в JavaScript данных, посланных из Dart	355
12.1.3. Отправка данных из JavaScript в Dart	358
12.2. Взаимодействие с внешними серверами	361
12.2.1. Правило одного домена	363
12.2.2. Использование JSONP для запроса данных у внешнего сервера	364
12.3. Построение допускающих установку браузерных приложений, не требующих сервера	367
12.3.1. Применение технологии AppCache для запуска приложений в автономном режиме	368
12.3.2. Создание пакета приложения, допускающего установку в Chrome	373
12.4. Резюме	377

ЧАСТЬ IV. DART НА СТОРОНЕ СЕРВЕРА

Глава 13. Работа с файлами и протоколом HTTP на сервере

13.1. Запуск серверного Dart-скрипта	381
--	-----

13.1.1. Доступ к аргументам командной строки	384
13.1.2. Доступ к файлам и папкам с помощью dart:io.....	386
13.2. Обслуживание HTTP-запросов от браузера.....	393
13.2.1. Класс <code>HttpServer</code>	395
13.2.2. Передача статических файлов по HTTP.....	397
13.3. REST API для клиентов	399
13.3.1. Отправка содержимого каталога в формате JSON	402
13.3.2. Отправка содержимого файла в формате JSON	403
13.3.3. Добавление пользовательского интерфейса на стороне клиента	404
13.4. Резюме	409

Глава 14. Отправка, синхронизация

и сохранение данных	410
14.1. Передача приложения <code>DartExpense</code> с сервера	411
14.2. Использование веб-сокетов для организации двусторонней связи	411
14.2.1. Соединение через веб-сокет на стороне клиента	414
14.2.2. Обработка подключения через веб-сокет на сервере.....	416
14.2.3. Использование веб-сокетов для межбраузерной синхронизации	419
14.3. Сохранение данных в базе <code>CouchDB</code> с помощью класса <code>HttpClient</code>	426
14.3.1. Краткое введение в <code>CouchDB</code>	427
14.3.2. Совместное использование модельного класса <code>Expense</code> в коде клиента и сервера.....	430
14.3.3. Добавление поддержки сохранения данных на сервере.....	431
14.4. Резюме	437

Глава 15. Организация параллелизма

с помощью изоляторов	438
15.1. Использование изоляторов как единиц работы	439
15.1.1. Создание изолятора	440
15.1.2. Односторонняя связь с изолятором	442
15.1.3. Двусторонняя связь с изолятором	446

15.2. Динамическая загрузка кода.....	452
15.2.1. Создание изолятора для загружаемого файла.....	454
15.2.2. Определение динамически загружаемого исходного файла	455
15.3. Запуск нескольких исполнителей.....	457
15.4. Резюме	463

Приложение А. Справочное руководство

по языку	464
A.1. Объявления переменных	464
A.1.1. Объявление переменных с ключевым словом var или именем типа.....	465
A.1.2. Объявление финальных (доступных только для чтения) переменных	466
A.1.3. Синтаксис литералов	466
A.1.4. Обобщенные списки и словари	474
A.2. Функции	475
A.2.1. Длинная синтаксическая форма.....	477
A.2.2. Короткая синтаксическая форма.....	477
A.2.3. Параметры функции.....	478
A.2.4. Функции как полноценные объекты	479
A.3. Управление потоком выполнения и итерирование	482
A.3.1. Ветвление потока выполнения	482
A.3.2. Циклы и итерирование	487

Приложение В. Определение классов

и библиотек	491
V.1. Классы и интерфейсы.....	491
V.1.1. Определение классов	491
V.1.2. Наследование классов	502
V.1.3. Абстрактные классы.....	505
V.1.4. Неявные интерфейсы.....	506
V.1.5. Статические свойства и методы.....	508
V.2. Библиотеки и ограничение доступа	509
V.2.1. Определение библиотек	509
V.2.2. Ограничение доступа к элементам библиотеки.....	510

Предметный указатель.....	512
----------------------------------	------------