

Содержание (сводка)

	Введение	23
1	Первые шаги. <i>С головой в пучину</i>	33
2	Построение интерактивных приложений. <i>Приложения, которые что-то делают</i>	71
3	Множественные активности и интен­ты. <i>Предъявите свой ин­тен­т</i>	105
4	Жизненный цикл активности. <i>Из жизни активностей</i>	147
5	Пользовательский интерфейс. <i>Представление начинается</i>	195
6	Списковые представления и адаптеры. <i>Обо всем по порядку</i>	259
7	Фрагменты. <i>Модульная структура</i>	301
8	Вложенные фрагменты. <i>Укрощение фрагментов</i>	357
9	Панели действий. <i>В поисках короткого пути</i>	397
10	Выдвижные панели. <i>Подальше положишь...</i>	429
11	Базы данных SQLite. <i>Работа с базами данных</i>	469
12	Курсоры и асинхронные задачи. <i>Подключение к базам данных</i>	503
13	Службы. <i>К вашим услугам</i>	573
14	Материальное оформление. <i>Жизнь в материальном мире</i>	629
	Приложение I. Исполнительная среда. <i>Исполнительная среда Android</i>	681
	Приложение II. ADB. <i>Android Debug Bridge</i>	685
	Приложение III. Эмулятор. <i>Эмулятор Android</i>	691
	Приложение IV. Остатки. <i>Десять важнейших тем (которые мы не рассмотрели)</i>	694

Содержание (настоящее)

Введение

Ваш мозг и Android. Вы учитесь — готовитесь к экзамену. Или пытаетесь освоить сложную техническую тему. Ваш мозг пытается оказать вам услугу. Он старается сделать так, чтобы на эту очевидно несущественную информацию не тратились драгоценные ресурсы. Их лучше потратить на что-нибудь важное. Так как же заставить его изучить программирование для Android?

Для кого написана эта книга?	24
Мы знаем, о чем вы думаете	25
И мы знаем, о чем думает ваш мозг	25
Метапознание: наука о мышлении	27
Вот что сделали МЫ	28
Примите к сведению	30
Научные редакторы	31
Благодарности	32

1 Первые Шаги

С головой в пучину

Система Android покорила мир. Все хотят иметь планшет или смартфон, а устройства на базе Android пользуются невероятной популярностью. В этой книге мы научим вас разрабатывать собственные приложения, а также покажем, как построить простое приложение и запустить его на виртуальном устройстве Android. Попутно будут рассмотрены основные компоненты приложений Android — такие как активности и макеты. Все, что от вас потребуется — некоторые базовые знания Java...



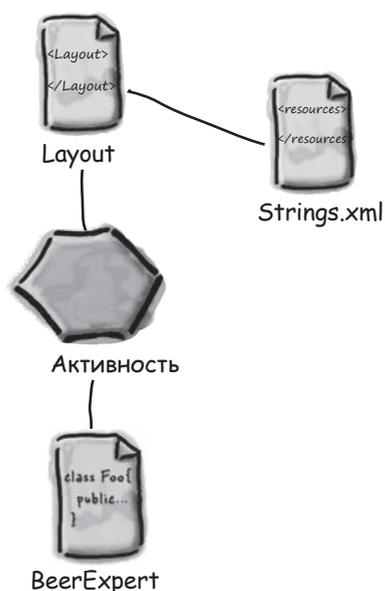
Добро пожаловать в мир Android	34
Платформа Android в разрезе	35
Вот что мы сейчас сделаем	36
Среда разработки	37
Установите Java	38
Построение простого приложения	39
Давайте построим простое приложение	40
Построение простого приложения (продолжение)	41
Активности и макеты: с высоты птичьего полета	44
Построение простого приложения (продолжение)	45
Только что вы создали свое первое Android-приложение	47
Android Studio создает всю структуру папок за вас	48
Полезные файлы в проекте	49
Редактирование кода в Android Studio	50
Запуск приложения в эмуляторе Android	55
Создание виртуального устройства Android	56
Запуск приложения в эмуляторе	59
Информация о ходе запуска отображается на консоли	60
Тест-драйв	61
Что же только что произошло?	62
Модификация приложения	63
Что содержит макет?	64
activity_main.xml состоит из двух элементов	65
Файл макета содержит ссылку на строку, а не саму строку	66
Заглянем в файл strings.xml	67
Ваш инструментарий Android	70



2 Построение интерактивных приложений

Приложения, которые что-то делают

Обычно приложение должно реагировать на действия пользователя. Из этой главы вы узнаете, как существенно повысить интерактивность ваших приложений. Мы покажем, как заставить приложение делать что-то в ответ на действия пользователя и как заставить активность и макет общаться друг с другом, как старые знакомые. Попутно вы больше узнаете о том, как на самом деле работает Android; мы расскажем о R — неприметном сокровище, которое связывает все воедино.



В этой главе мы построим приложение для выбора пива	72
Что нужно сделать	73
Создание проекта	74
Мы создали активность и макет по умолчанию	75
Добавление компонентов в визуальном редакторе	76
В <code>activity_find_beer.xml</code> появилась новая кнопка	77
Подробнее о коде макета	78
Изменения в XML...	80
...отражаются в визуальном редакторе	81
Использование строковых ресурсов вместо жестко запрограммированного текста	82
Внесение изменений в макет для использования строковых ресурсов	83
Добавление значений в список	85
Передача массива строк раскрывающемуся списку	86
Тест-драйв раскрывающегося списка	86
Кнопка должна что-то делать	87
Как заставить кнопку вызвать метод	88
Как выглядит код активности	89
Добавление в активность метода <code>onClickFindBeer()</code>	90
Метод <code>onClickFindBeer()</code> должен что-то делать	91
Обновление кода активности	93
Первая версия активности	95
Что делает этот код	96
Построение вспомогательного класса Java	98
Код активности, версия 2	101
Что происходит при выполнении кода	102
Тест-драйв	103
Ваш инструментарий Android	104

Множественные активности и интенты

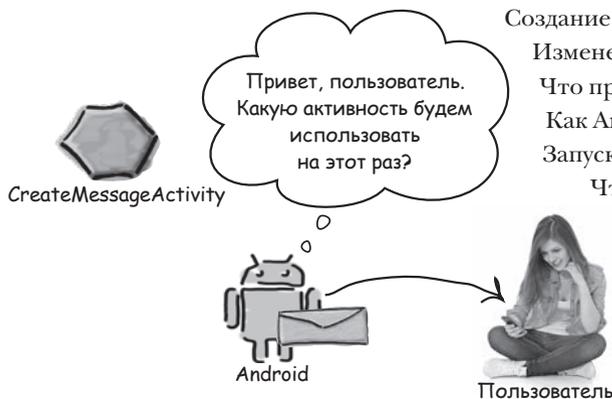
Предъявите свой интент

3

Для большинства приложений одной активности недостаточно. До настоящего момента мы рассматривали приложения с одной активностью; для простых приложений это нормально. Однако в более сложной ситуации одна активность попросту не справляется со всеми делами. Мы покажем вам, как строить приложения с несколькими активностями и как организовать взаимодействие между активностями с использованием интентов. Также вы узнаете, как использовать интенты за пределами приложения и как выполнять действия при помощи активностей других приложений на вашем устройстве. Внезапно перед вами открываются совершенно новые перспективы...



Приложение может содержать несколько активностей	106
Структура приложения	107
Создание проекта	107
Обновление макета	108
Обновление strings.xml...	109
Создание второй активности и макета	110
Знакомьтесь: файл манифеста Android	112
Интент — разновидность сообщения	114
Использование интента для запуска второй активности	115
Что происходит при запуске приложения	116
Передача текста второй активности	118
Обновление свойств надписи	119
putExtra() включает в интент дополнительную информацию	120
Обновление кода CreateMessageActivity	123
Использование информации из интента в ReceiveMessageActivity	124
Что происходит при щелчке на кнопке Send Message	125
Как работают приложения Android	127
Создание интента с указанием действия	129
Изменение интента для использования действия	130
Что происходит при выполнении кода	131
Как Android использует фильтр интентов	134
Запуск приложения на РЕАЛЬНОМ устройстве	137
Что происходит при вызове createChooser()	141
Изменение кода выбора активности	143
Тест-драйв	144
Если подходящих активностей НЕТ	145
Ваш инструментарий Android	146



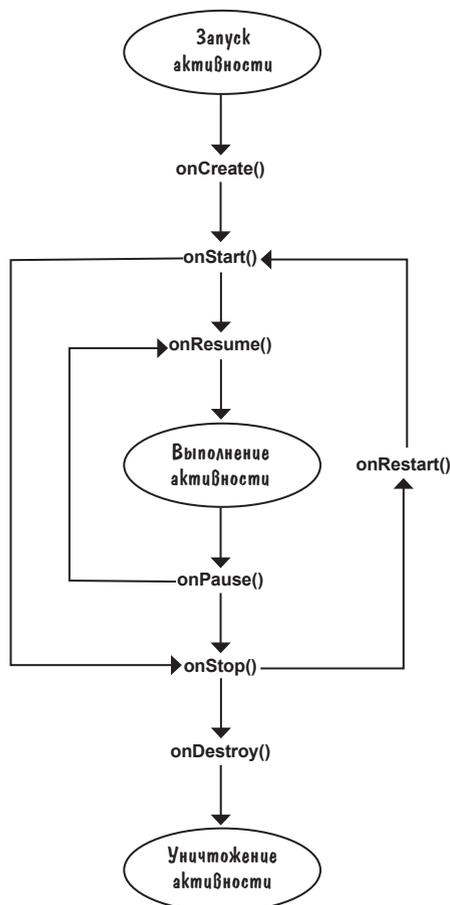
4

Жизненный цикл активности

Из жизни активностей

Активности образуют основу любого Android-приложения.

Ранее вы видели, как создавать активности и как организовать запуск одной активности из другой с использованием интента. Но что при этом происходит, если заглянуть поглубже? В этой главе более подробно рассматривается жизненный цикл активностей. Что происходит при создании или уничтожении активностей? Какие методы вызываются, когда активность становится видимой и появляется на переднем плане, и какие методы вызываются, когда активность теряет фокус и скрывается? И как выполняются операции сохранения и восстановления состояния активности?

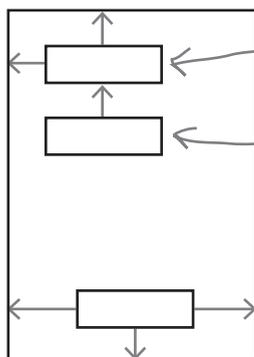


Как на самом деле работают активности?	148
Приложение Stopwatch	150
Разметка макета приложения Stopwatch	151
Как работает код активности	153
Добавление кода кнопок	154
Метод runTimer()	155
Объекты Handler позволяют планировать выполнение кода	156
Полный код runTimer()	157
Полный код StopwatchActivity	158
Что происходит при запуске приложения	160
Поворот экрана изменяет конфигурацию устройства	164
От рождения до смерти: состояния активности	165
Жизненный цикл активности: от создания до уничтожения	166
Активность наследует свои методы жизненного цикла	167
Что делать при изменении конфигурации?	168
Что происходит при запуске приложения	171
Жизненный цикл активности: видимость	175
Необходимо реализовать еще два метода жизненного цикла	176
Обновленный код StopwatchActivity	179
Что происходит при запуске приложения	180
Жизненный цикл активности: видимость	183
Прекращение отсчета времени при приостановке активности	186
Что происходит при запуске приложения	187
Тест-драйв	188
Полный код активности	189
Краткое руководство по методам жизненного цикла	193
Ваш инструментарий Android	194

5 Пользовательский интерфейс

Представление начинается

Давайте честно признаем: создавать хорошие макеты нужно уметь. Если вы строите приложения, которые должны использоваться людьми, необходимо позаботиться о том, чтобы эти макеты выглядели в точности так, как вам нужно. До настоящего момента мы едва затронули тему создания макетов; пришло время разобраться поглубже. Мы познакомим вас с другими типами макетов, которые могут использоваться в программах, после чего будет представлен обзор основных компонентов графического интерфейса и способов их использования. К концу главы вы увидите, что несмотря на внешние различия, у всех макетов и компонентов графического интерфейса больше общего, чем кажется на первый взгляд.



Представления могут размещаться относительно родительского макета...

...или относительно других представлений.

Три ключевых макета: относительный, линейный и табличный	197
Позиционирование представлений относительно родительского макета	200
Позиционирование представлений относительно других представлений	202
Атрибуты для позиционирования представлений относительно других представлений	203
Относительный макет: итоги	205
В линейных макетах представления выводятся в одну строку или столбец	206
Начало настройки линейного макета	209
Назначение веса одному представлению	211
Назначение весов нескольким представлениям	212
Атрибут android:gravity: список значений	214
Другие допустимые значения атрибута android:layout-gravity	216
Полная разметка линейного макета	217
Линейный макет: итоги	218
Добавление представлений в табличный макет	222
Создание табличного макета	223
Строка 0: добавление представлений в конкретные строки и столбцы	225
Строка 1: представление занимает несколько столбцов	226
Строка 2: представление занимает несколько столбцов	227
Полный код табличного макета	228
Табличный макет: итоги	229
Ваш инструментарий Android	257

6 Списковые представления и адаптеры

Обо всем по порядку

Хотите знать, как лучше организовать Android-приложение?

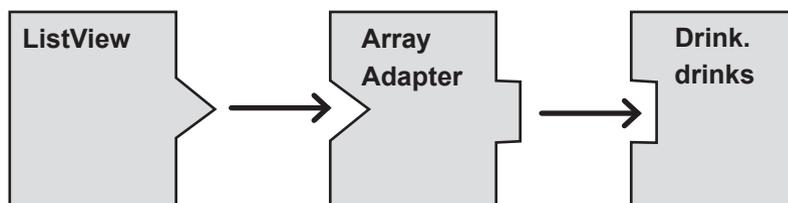
Мы рассмотрели основные структурные элементы, используемые при построении приложений; теперь пора привести знания в порядок. В этой главе мы покажем, как взять разрозненные идеи и превратить их в классное приложение. Мы покажем, как списки данных могут стать основой структуры вашего приложения и что связывание списков позволяет создавать мощные и удобные приложения. Попутно вы в общих чертах узнаете, как при помощи слушателей событий и адаптеров сделать ваше приложение более динамичным.

Вывести начальный экран со списком всех команд.

Вывести подробную информацию по каждому блюду.

Вывести подробную информацию по каждому напитку.

Каждое приложение начинается с идей	260
Проведите классификацию идей: верхний уровень, категории, детализация/редактирование	261
Навигация по активностям	262
Навигация с использованием списковых представлений	263
Построим приложение Starbuzz	264
Активность детализации с информацией о напитке	265
Структура приложения Starbuzz	266
Макет верхнего уровня содержит изображение и список	270
Полный код макета верхнего уровня	272
Обработка щелчков компонентом ListView	273
Полный код TopLevelActivity	275
Создание списковой активности	281
Связывание списковых представлений с массивами при помощи адаптера массива	283
Добавление адаптера массива в DrinkCategoryActivity	284
Что происходит при выполнении кода	285
Как мы обрабатывали щелчки в TopLevelActivity	288
Полный код DrinkCategoryActivity	290
Активность детализации выводит данные из одной записи	291
Обновление представлений	293
Код DrinkActivity	295
Тест-драйв	298
Ваш инструментарий Android	300



Фрагменты

1

Модульная структура

Вы уже умеете создавать приложения, которые работают одинаково независимо от устройства, на котором они запускаются.

Но что, если ваше приложение должно выглядеть и вести себя по-разному в зависимости от того, где оно запущено — на телефоне или планшете? В этой главе мы покажем, как в приложении выбрать наиболее подходящий макет по размерам экрана устройства. Также вы познакомитесь с фрагментами — механизмом создания модульных программных компонентов, которые могут повторно использоваться разными активностями.

Структура приложения Workout	305
Класс Workout	307
Добавление фрагмента в проект	308
Добавление фрагмента в макет активности	311
Передача идентификатора фрагменту	312
Присваивание идентификатора	313
Жизненный цикл фрагмента	315
Ваш фрагмент наследует методы жизненного цикла	316
Создание фрагмента со списком	320
Создание спискового фрагмента	322
Использование ArrayAdapter для заполнения ListView	323
Обновленный код WorkoutListFragment	324
Включение фрагмента WorkoutListFragment в макет MainActivity	325
Связывание списка с детализацией	327
Логическое отделение фрагмента от активности	328
Добавление интерфейса к списковому фрагменту	329
Реализация интерфейса активностью	330
Фрагменты должны поддерживать кнопку возврата	331
Транзакции фрагментов	333
Обновленный код MainActivity	334
Тест-драйв	335
Поворот устройства нарушает работу приложения	336
Код WorkoutDetailFragment	337
Телефон и планшет	338
Структура приложения для планшета и телефона	339
Выбор имен папок	341
Макет MainActivity для телефона	347
Полный код DetailActivity	351
Использование различий в макетах	352
Обновленный код MainActivity	353
Тест-драйв	354
Ваш инструментарий Android	355

Значит, фрагмент будет содержать только списковое представление. Интересно... Для активностей, содержащих только список, мы использовали активность ListActivity. Нет ли чего-нибудь похожего для фрагментов?



Вложенные фрагменты

Укращение фрагментов

Вы уже видели, что использование фрагментов в активностях способствует повторному использованию кода и делает приложения более гибкими. В этой главе мы покажем, как вложить один фрагмент внутрь другого. Вы научитесь пользоваться диспетчером дочерних фрагментов для укращения строптивных транзакций фрагментов. А попутно вы узнаете, почему так важно знать различия между активностями и фрагментами.

Создание вложенных фрагментов	358
Жизненные циклы фрагментов и активностей похожи...	359
Код StopwatchFragment	364
Макет StopwatchFragment	367
Добавление фрагмента в WorkoutDetailFragment	369
Добавление фрейма в месте нахождения фрагмента	370
Отображение фрагмента в коде Java	371
getManager() создает транзакции на уровне активности	372
Вложенным фрагментам — вложенные транзакции	373
Отображение фрагмента в методе onCreateView() родителя	374
Полный код WorkoutDetailFragment	375
Почему при нажатии кнопки происходит сбой?	377
Обратимся к разметке макета StopwatchFragment	378
Атрибут onClick вызывает методы активности, а не фрагмента	379
Удаление атрибутов onClick из макета фрагмента	380
Реализация OnClickListener фрагментом	381
Связывание OnClickListener с кнопками	383
Код StopwatchFragment	384
При повороте устройства активность создается заново	387
Код WorkoutDetailFragment	390
Тест-драйв	391
Ваш инструментарий Android	396

Посмотрите-ка — транзакции фрагментов. Я должна немедленно применить их.



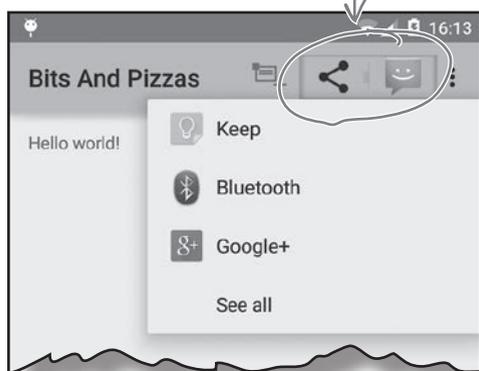
9

Панели Действий

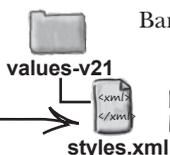
В поисках короткого пути

Все мы предпочитаем короткие пути к цели. В этой главе вы узнаете, как ускорить выполнение команд ваших приложений при помощи панелей действий. Вы узнаете, как запускать другие активности из элементов действий на панели действий, как передавать данные другим приложениям при помощи провайдера передачи информации и как перемещаться в иерархии приложения с использованием кнопки Вверх на панели действий. Попутно вы узнаете, как обеспечить единый стиль внешнего вида и поведения приложений с использованием тем оформления, и познакомитесь с пакетом библиотеки поддержки Android.

Так выглядит действие передачи информации на панели действий. Если щелкнуть на нем, на экране появляется список приложений, которые могут использоваться для передачи информации.



API 21? Идеальное совпадение...



Name: AppTheme
Parent: Theme.Material.Light

Хорошее приложение имеет четкую структуру	398
Типы навигации	399
Начнем с панели действий	400
Библиотеки поддержки Android	401
Библиотеки поддержки можно включить в ваш проект	402
Использование современных тем в приложении themes	403
Применение темы в AndroidManifest.xml	404
Определение стилей в файлах стилевых ресурсов	405
Назначение темы по умолчанию в styles.xml	406
Что происходит при запуске приложения	407
Добавление элементов действий на панель действий	408
Файл ресурсов меню	409
Атрибут меню showAsAction	410
Добавление нового элемента действия	411
Создание OrderActivity	414
Запуск OrderActivity элементом действия Create Order	415
Полный код MainActivity.java	416
Передача информации с панели действий	418
Добавление провайдера в файл menu_main.xml	419
Информация задается при помощи интента	420
Полный код MainActivity.java	421
Код MainActivity.java (продолжение)	422
Навигация с кнопкой Вверх	423
Назначение родителя активности	424
Добавление кнопки Вверх	425
Тест-драйв	426
Ваш инструментарий Android	427

10

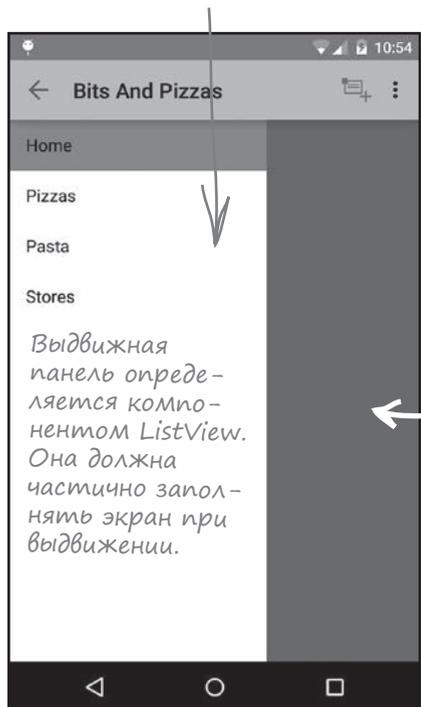
Выдвижные панели

Подальше положишь...

С приложением намного приятнее работать, когда в нем хорошо организована навигация. В этой главе мы представим

выдвижные панели (drawers) — чтобы открыть их, следует провести пальцем по экрану или прикоснуться к значку на панели действий. Вы увидите, как использовать выдвижные панели для вывода списка ссылок, ведущих к основным разделам приложения. Также вы увидите, как переключение фрагментов упрощает переход к этим разделам и ускоряет их отображение.

Информация выводится во фрейме, при этом она заполняет экран. В настоящий момент фрейм частично скрыт выдвижной панелью.



Возвращаемся к приложению Pizza	430
Подробнее о выдвижных панелях	431
Структура приложения Pizza	432
Создание фрагмента TopFragment	433
Создание фрагмента PizzaFragment	434
Создание фрагмента PastaFragment	435
Создание фрагмента StoresFragment	436
Добавление DrawerLayout	437
Полная разметка activity_main.xml	438
Инициализация списка на выдвижной панели	439
OnClickListener и обработка щелчков на вариантах спискового представления	440
Метод selectItem() — что было сделано	443
Изменение заголовка на панели действий	444
Заккрытие выдвижной панели	445
Обновленный код MainActivity.java	446
Открытие и закрытие выдвижной панели	448
Использование ActionBarDrawerToggle	449
Изменение элементов панели действий во время выполнения	450
Обновленный код MainActivity.java	451
Управление открытием и закрытием выдвижной панели	452
Синхронизация состояния ActionBarDrawerToggle	453
Обновленный код MainActivity.java	454
Обработка изменений конфигурации	457
Реакция на изменения в стеке возврата	458
Назначение меток фрагментам	459
Поиск фрагмента по метке	460
Полный код MainActivity.java	461
Тест-драйв	467
Ваш инструментарий Android	468

11

Базы данных SQLite

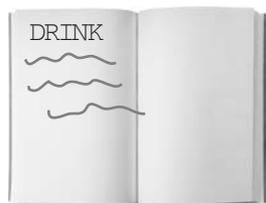
Работа с базами данных

Какая бы информация ни использовалась в приложении — рекордные счета или тексты сообщений в социальной сети, — эту информацию необходимо где-то хранить.

В Android для долгосрочного хранения данных обычно используется база данных SQLite. В этой главе вы узнаете, как создать базу данных, добавить в нее таблицы и заполнить данными, — все это делается при помощи удобных вспомогательных объектов SQLite. Затем будет показано, как выполнить безопасное обновление структуры базы данных и как вернуться к предыдущей версии в случае необходимости.



onCreate()



SQLite database

Name: "starbuzz"
Version: 1

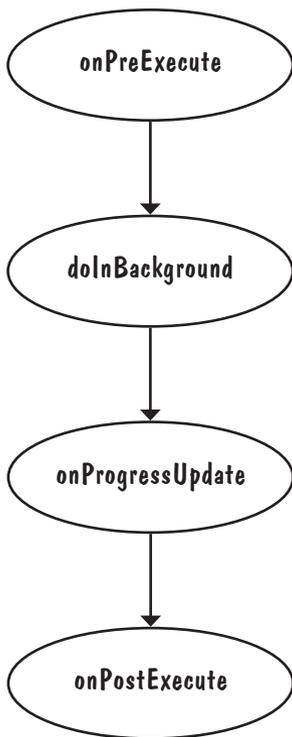
Возвращение в Starbuzz	470
Android хранит информацию в базах данных SQLite	471
Android включает классы SQLite	472
Текущая структура приложения Starbuzz	473
Переход на работу с базой данных	474
Помощник SQLite управляет базой данных	475
Помощник SQLite	475
Создание помощника SQLite	476
Внутри базы данных SQLite	478
Таблицы создаются командами SQL	479
Вставка данных методом insert()	480
Обновление записей методом update()	481
Сложные условия	482
Код StarbuzzDatabaseHelper	483
Что делает код помощника SQLite	484
А если структура базы данных изменится?	487
Номера версий баз данных SQLite	488
Обновление базы данных: сводка	489
История продолжается...	490
Как помощник SQLite принимает решения	491
Метод onUpgrade()	492
Метод onDowngrade()	493
Обновление базы данных	494
Обновление существующей базы данных	497
Переименование таблиц	498
Полный код помощника SQLite	499
Что происходит при выполнении кода	501
Ваш инструментарий Android	502

12

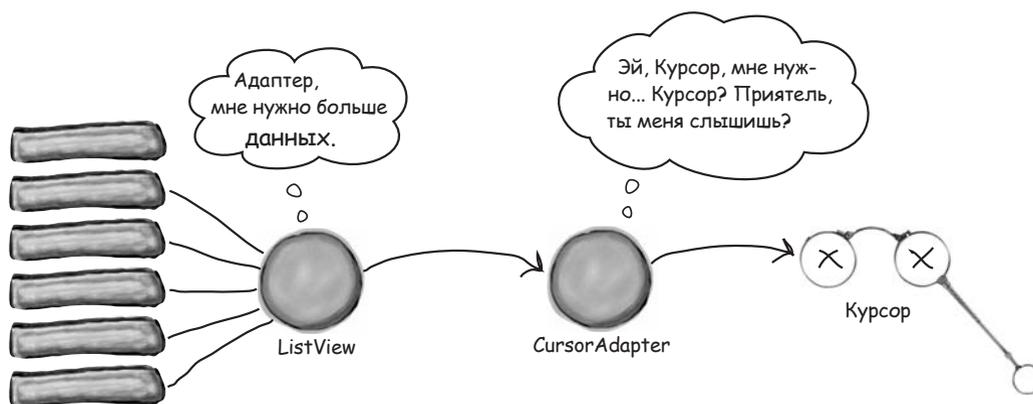
Курсоры и асинхронные задачи

Подключение к базам данных

Как же подключиться из приложения к базе данных SQLite? В предыдущей главе было показано, как создать базу данных SQLite с использованием помощника SQLite. Пора сделать следующий шаг — узнать, как работать с базой данных из активностей. В этой главе вы узнаете, как использовать курсоры для получения информации из базы данных, как перемещаться по набору данных с использованием курсора и как получить данные из курсора. Затем вы узнаете, как использовать адаптеры курсоров для их связывания со списковыми представлениями. В завершение мы покажем, как написание эффективного многопоточного кода с объектами AsyncTask ускоряет работу приложений.



Текущий код DrinkActivity	506
Определение таблицы и столбцов	510
Упорядочение данных в запросах	512
Функции SQL в запросах	513
Код DrinkActivity	522
Текущий код DrinkCategoryActivity	525
Код DrinkActivity	545
Обновленный код TopLevelActivity.java	556
Метод onPreExecute()	563
Метод doInBackground()	564
Метод onProgressUpdate()	565
Метод onPostExecute()	566
Класс AsyncTask	567
Код DrinkActivity.java	569
Ваш инструментарий Android	572



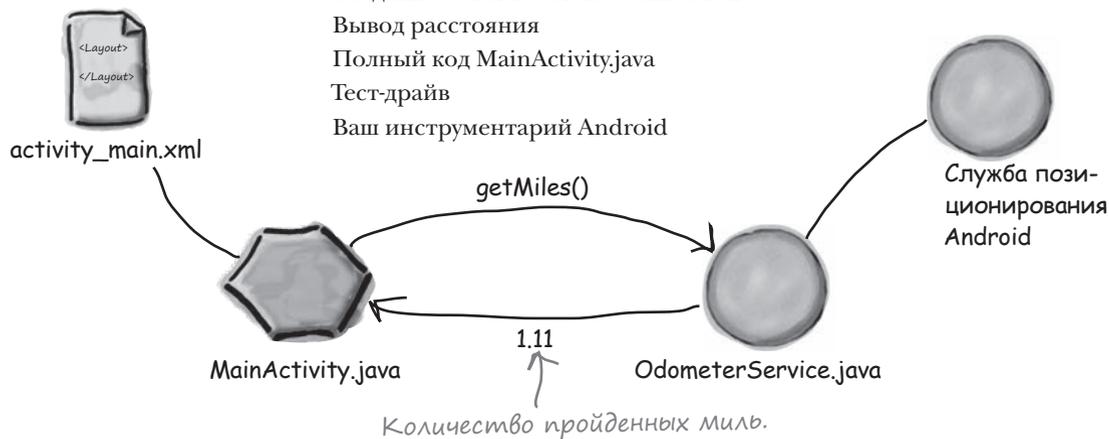
13

СЛУЖБЫ

К вашим услугам

Существуют операции, которые должны выполняться постоянно. Например, если вы запустили воспроизведение музыкального файла в приложении-проигрывателе, вероятно, музыка не должна останавливаться при переключении на другое приложение. В этой главе вы узнаете, как использовать службы для подобных ситуаций, а заодно научитесь пользоваться некоторыми встроенными службами Android. Служба уведомлений поможет вам держать пользователей в курсе дел, а при помощи службы позиционирования пользователь сможет узнать, где он находится.

Запускаемая служба	575
Класс IntentService — взгляд издалека	577
Объявление службы в AndroidManifest.xml	580
Службы запускаются вызовом startService()	582
Полный код DelayedMessageService.java	586
Использование службы уведомлений	589
Запуск активности из оповещения	591
Отправка уведомлений с использованием службы уведомлений	593
Полный код DelayedMessageService.java	594
Создание проекта Odometer	603
Определение Binder	605
Четыре ключевых метода класса Service	607
Получение информации о местонахождении	608
Добавление LocationListener в код службы	609
Регистрация LocationListener	610
Передача пройденного расстояния активности	611
Полный код OdometerService.java	612
Обновление AndroidManifest.xml	614
Обновление макета MainActivity	618
Создание объекта ServiceConnection	619
Вывод расстояния	621
Полный код MainActivity.java	622
Тест-драйв	626
Ваш инструментарий Android	627

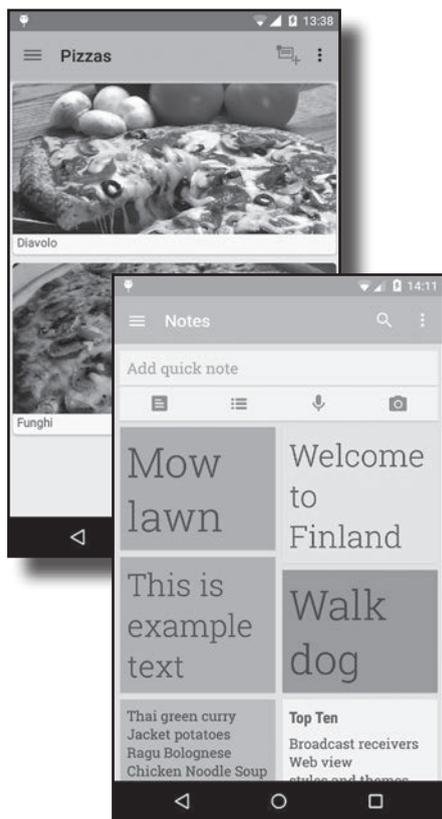


14

Материальное оформление

Жизнь в материальном мире

В API уровня 21 компания Google представила концепцию материального оформления. В этой главе вы узнаете, что такое материальное оформление и как реализовать его принципы в ваших приложениях. Мы начнем с карточек, которые могут повторно использоваться в приложениях для обеспечения целостности оформления. Затем будет рассмотрен компонент RecyclerView — хороший друг спискового представления. Попутно вы узнаете, как создавать адаптеры и как полностью изменить внешний вид RecyclerView всего двумя строками кода.



Знакомство с материальным оформлением	630
Структура приложения Pizza	632
Создание представлений CardView	635
Полная разметка card_captioned_image.xml	636
Создание базового адаптера	638
Определение класса ViewHolder	639
Создание ViewHolder	640
Добавление данных в карточки	642
Полный код CaptionedImagesAdapter.java	643
Создание RecyclerView	644
Добавление RecyclerView в макет	645
Код PizzaMaterialFragment.java	646
Размещение представлений в RecyclerView	647
Полный код PizzaMaterialFragment.java	649
Использование нового фрагмента PizzaMaterialFragment в MainActivity	650
Что происходит при выполнении кода	651
Создание PizzaDetailActivity	659
Что должен делать код PizzaDetailActivity.java	660
Код PizzaDetailActivity.java	661
Выбор вариантов в RecyclerView	663
Прослушивание событий представлений в адаптере	664
Повторное использование адаптеров	665
Добавление интерфейса в адаптер	666
Реализация слушателя в PizzaMaterialFragment.java	668
Информацию — на передний план	671
Полная разметка fragment_top.xml	676
Полный код TopFragment.java	677
Ваш инструментарий Android	679



Исполнительная среда Android

Приложения Android должны работать на устройствах с маломощными процессорами и ограниченной памятью. Java-приложения могут расходовать много памяти. Кроме того, если бы приложения выполнялись на виртуальной машине Java (JVM, Java Virtual Machine), на маломощных устройствах запуск приложения мог бы занимать много времени. Чтобы избежать этих проблем, система Android вместо JVM использует для запуска своих приложений другую виртуальную машину, которая называется ART (Android runtime). В этом приложении вы узнаете, как ART удается обеспечить нормальное выполнение Java-приложений на компактном маломощном устройстве.



Android Debug Bridge

В этой книге для всех задач Android-программирования использовались средства среды разработки. Однако в некоторых случаях инструменты командной строки попросту более эффективны — например, если Android Studio упорно отказывается видеть ваше устройство Android, а вы *знаете*, что оно есть. В этой главе мы познакомимся с **Android Debug Bridge (сокращенно adb)** — утилитой командной строки, предназначенной для взаимодействия с эмулятором или устройствами Android.



Эмулятор Android

У вас никогда не возникало ощущение, что эмулятора приходится ждать целую вечность? Безусловно, эмулятор Android полезен. С его помощью можно увидеть, как приложение будет работать на других устройствах — помимо физических устройств, имеющихся у вас. Но иногда он кажется слегка... заторможенным. В этом приложении мы объясним, почему эмулятор кажется медленным. Более того, мы приведем несколько полезных советов для ускорения его работы.



Десять важнейших тем (которые мы не рассмотрели)

Но и это еще не все. Осталось еще несколько тем, о которых, как нам кажется, вам следует знать. Делать вид, что их не существует, было бы неправильно — как, впрочем, и выпускать книгу, которую поднимет разве что культурист. Прежде чем откладывать книгу, ознакомьтесь с этими **лакомыми кусочками**, которые мы оставили напоследок.