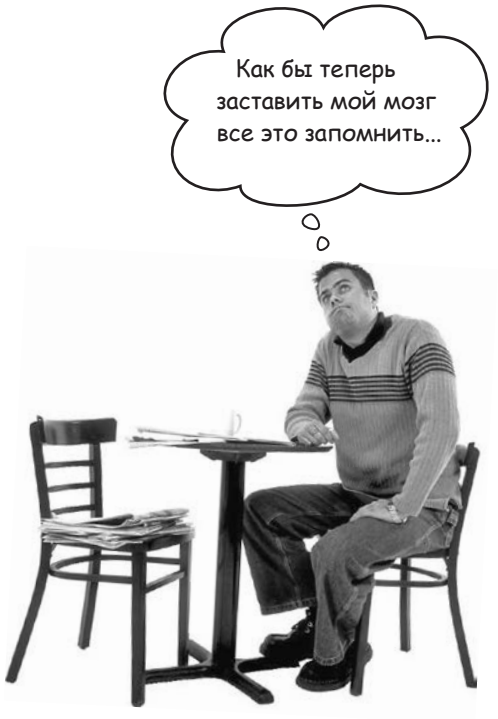
Содержание (настоящее)

Введение

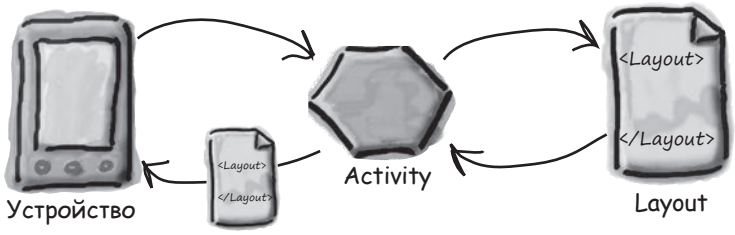
Ваш мозг думает об Android. Вы сидите за книгой и пытаетесь что-нибудь выучить, но ваш мозг считает, что вся эта писанина не нужна. Ваш мозг говорит: «Выгляни в окно! На свете есть более важные вещи, например сноуборд». Как заставить мозг думать, что ваша жизнь действительно зависит от умения разрабатывать приложения для Android?



1 Первые шаги

С головой в пучину

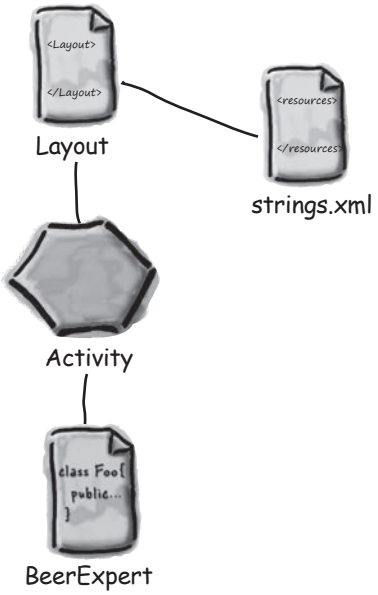
Система Android покорила мир. Все хотят иметь планшет или смартфон, а устройства на базе Android пользуются невероятной популярностью. В этой книге мы научим вас разрабатывать собственные приложения, а также покажем, как построить простое приложение и запустить его на виртуальном устройстве Android.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



2 Построение интерактивных приложений

Приложения, которые что-то делают

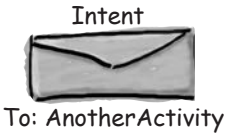
Обычно приложение должно реагировать на действия пользователя. В этой главе вы узнаете, как существенно повысить интерактивность ваших приложений. Мы покажем, как заставить приложение делать что-то в ответ на действия пользователя и как заставить активность и макет общаться друг с другом, как старые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



3 Множественные активности и интенты

Предъявите свой интент

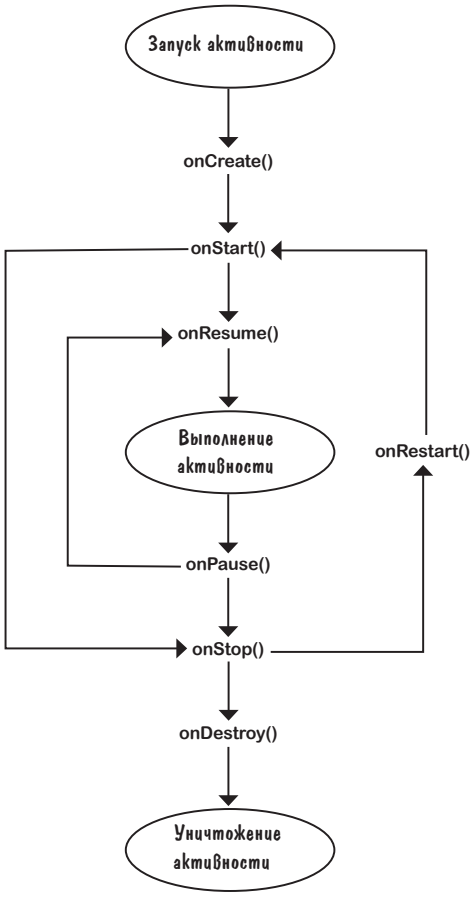
Для большинства приложений одной активности недостаточно. До настоящего момента мы рассматривали приложения с одной активностью; для простых приложений это нормально. Однако в более сложной ситуации одна активность попросту не справляется со всеми делами. Мы покажем вам, как строить приложения с несколькими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



4 Жизненный цикл активности

Из жизни активностей

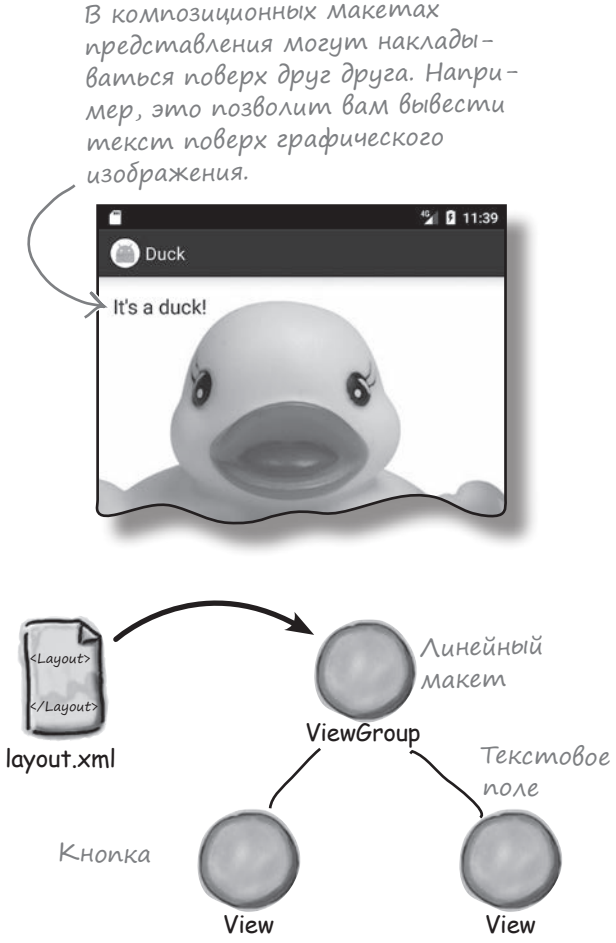
Активности образуют основу любого Android-приложения. Ранее вы видели, как создавать активности и как организовать запуск одной активности из другой с использованием интента. Но что при этом происходит, если заглянуть поглубже? В этой главе более подробно рассматривается жизненный цикл активностей. Что 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? В этой главе вы получите ответы на все эти вопросы.



5 Представления и группы

Представление начинается

Вы уже видели, как происходит размещение компонентов графического интерфейса на экране в линейных макетах. Тем не менее это была лишь вершина айсберга. В этой главе мы **заглянем поглубже**, и вы узнаете, как *на самом деле* работают линейные макеты. Вы познакомитесь с компонентом **FrameLayout** — простым компонентом, предназначенным для размещения представлений. Также в этой главе 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***



6 Макеты с ограничениями

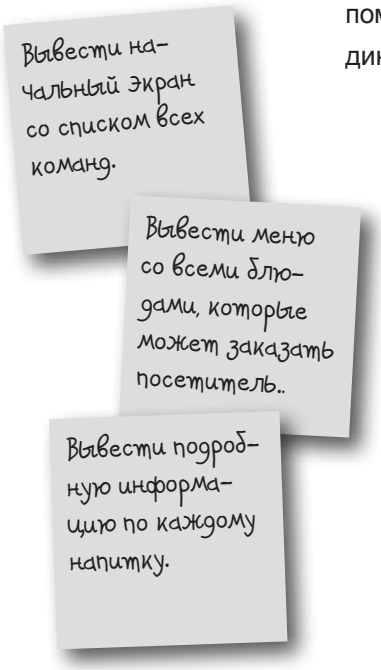
Расставить по местам

Давайте честно признаем: создать хороший макет не так просто. Это нужно уметь. Если вы строите приложения, которыми будут пользоваться люди, нужно позаботиться о том, чтобы они выглядели именно так, как было задумано. До сих пор мы показывали, как пользоваться линейными и композиционными макетами... Но что, если ваш макет имеет более сложную структуру? Для таких случаев в Android появилась новая возможность — макеты с ограничениями, разновидность макетов, которая обычно строится в визуальном режиме по схеме. Вы узнаете, как при помощи ограничений задавать позицию и размеры представлений независимо от размера экрана и ориентации. В завершение главы мы покажем, как сэкономить время, поручив Android Studio вычислить и добавить ограничения за вас.

7 Списковые представления и адаптеры

Обо всем по порядку

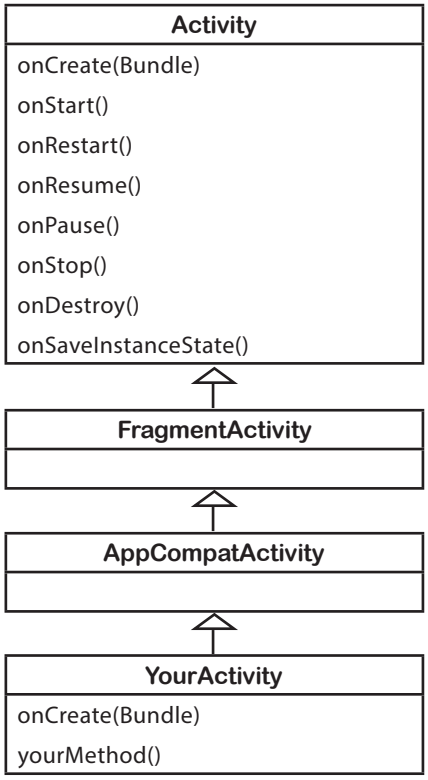
Хотите знать, как лучше организовать Android-приложение? Мы рассмотрели основные структурные элементы, используемые при построении приложений; теперь пора привести знания в порядок. В этой главе мы покажем, как взять разрозненные идеи и превратить их в классное приложение. Мы покажем, как списки данных могут стать основой структуры вашего приложения и как 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.

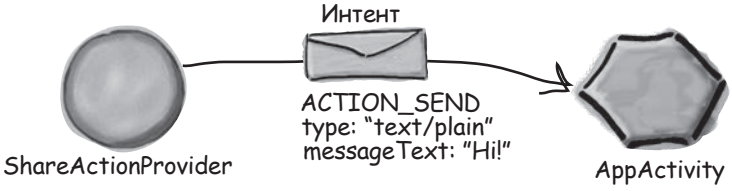


8 Библиотеки поддержки и панели приложений

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 приложений. Вы узнаете, как запускать другие активности из элементов действий на панели приложения, как передавать данные другим приложениям при помощи провайдера передачи информации и как перемещаться в иерархии приложения с использованием кнопки Вверх на панели приложения. Заодно вы познакомитесь с библиотеками поддержки Android, с которыми ваши приложения будут выглядеть современно даже в старых версиях Android.

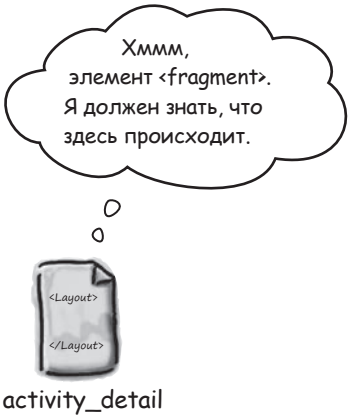




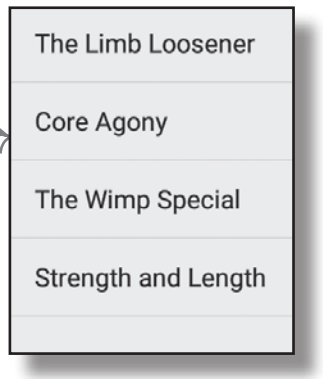
9 Фрагменты

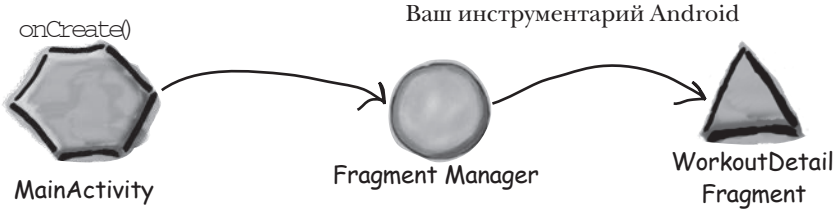
Модульная структура

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 таком случае вам понадобятся фрагменты — модульные программные компоненты, которые могут повторно использоваться разными активностями. Мы покажем, как создавать простые фрагменты и списковые фрагменты, как добавлять их в активности и как организовать взаимодействие между фрагментами и активностями.



Списковый фрагмент содержит собственное списковое представление, так что вам не придется создавать его самостоятельно: достаточно предоставить списковому фрагменту данные.

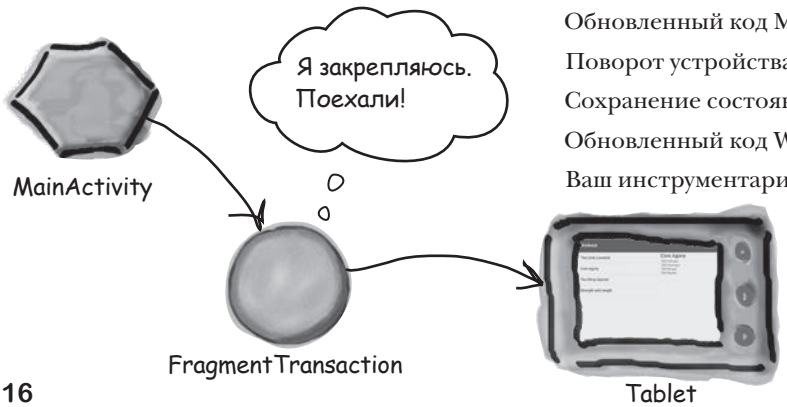




10 Фрагменты для больших интерфейсов

Разные размеры, разные интерфейсы

До сих пор наши приложения запускались только на устройствах с малыми экранами. Но что, если пользователи используют планшетные устройства? В этой главе вы увидите, как создать гибкий пользовательский интерфейс, чтобы ваше приложение выглядело и работало по-разному в зависимости от типа устройства, на котором оно запущено. В этой главе вы научитесь управлять поведением приложения с использованием кнопки Назад при помощи стека возврата и транзакций фрагментов. Наконец, вы узнаете, как сохранять и восстанавливать состояние фрагментов.



11 Динамические фрагменты

Вложение фрагментов

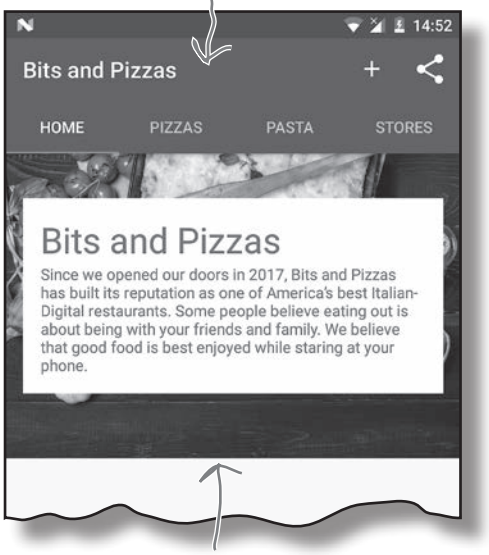
До сих пор мы занимались созданием и использованием статических фрагментов. Но что, если вы хотите придать своим фрагментам немного динамики? У динамических фрагментов много общего с динамическими активностями, но есть и важные различия, которые необходимо учитывать. В этой главе вы научитесь преобразовывать динамические активности в рабочие динамические фрагменты. Вы узнаете, как использовать транзакции фрагментов для хранения состояния фрагмента. Наконец, мы покажем, как вложить один фрагмент в другой и как диспетчер дочерних фрагментов помогает решать проблемы с некорректным поведением стека возврата.

12 Design support library

Виджеты и жесты

Хотите разрабатывать приложения с полнофункциональным, современным интерфейсом? С выходом библиотеки Android Design Support Library разработчикам стало намного проще создавать приложения с современным пользовательским интерфейсом. В этой главе мы представим некоторые из ее ключевых аспектов. Вы научитесь создавать вкладки, чтобы пользователям было удобнее работать с системой навигации ваших приложений. Вы узнаете, как определить анимацию панелей инструментов, чтобы их можно было сворачивать или разворачивать по желанию пользователя. Вы научитесь добавлять плавающие кнопки действий для стандартных пользовательских действий. Наконец, мы познакомим вас с уведомлениями Snackbar — короткими содержательными сообщениями, с которыми может взаимодействовать пользователь.

Панель инструментов должна прокручиваться при прокрутке контента в TopFragment.

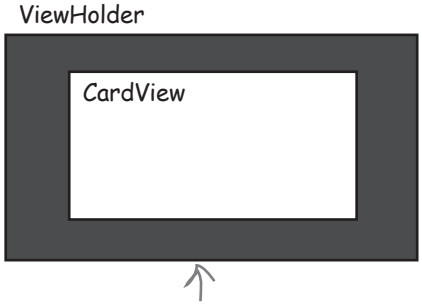


Этот контент добавляется в TopFragment.

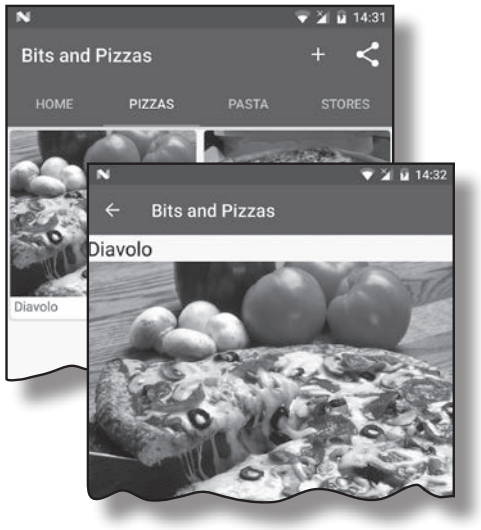
13 RecyclerView и карточки

Переработка отходов

Вы уже видели, что скромное списковое представление играет ключевую роль во многих приложениях. Но по сравнению с компонентами материального оформления, которые были представлены выше, он слишком примитивен. В этой главе будет представлен компонент RecyclerView — более мощный списковый компонент, который обладает большей гибкостью и совмещается с принципами материального оформления. Вы научитесь создавать адаптеры, приспособленные к вашим данным, и сможете полностью изменить внешний вид списка всего двумя строками кода. Мы также покажем, как при помощи карточек имитировать в приложениях эффект трехмерного материального оформления.



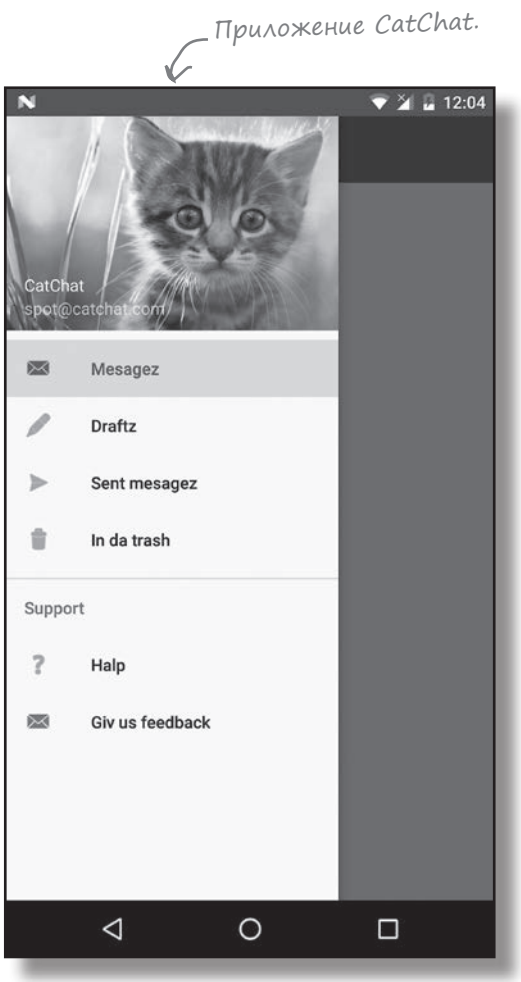
Каждый объект ViewHolder содержит CardView. Макет CardView был создан ранее в этой главе.



14 Выдвижные панели

Подальше положишь...

Как вы уже могли убедиться, вкладки сильно упрощают навигацию в приложениях. Но если вкладок *очень много* или вы захотите *разбить их на группы* — используйте навигационные выдвижные панели. В этой главе вы научитесь создавать навигационные панели, которые вызываются *из-за края активности одним прикосновением*. Вы узнаете, как назначить навигационной панели заголовок и как создать структурированное меню для перехода ко всем основным точкам приложения. Наконец, мы покажем, как создать слушателя, чтобы навигационная панель *реагировала на жесты.*



15 Базы данных SQLite

Работа с базами данных

Какая бы информация ни использовалась в приложении — рекордные счета или тексты сообщений в социальных сетях — 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.

16 Курсоры

Получение данных

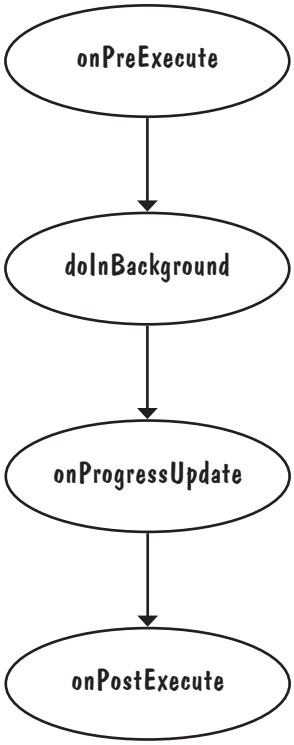
Как же подключиться из приложения к базе данных SQLite?

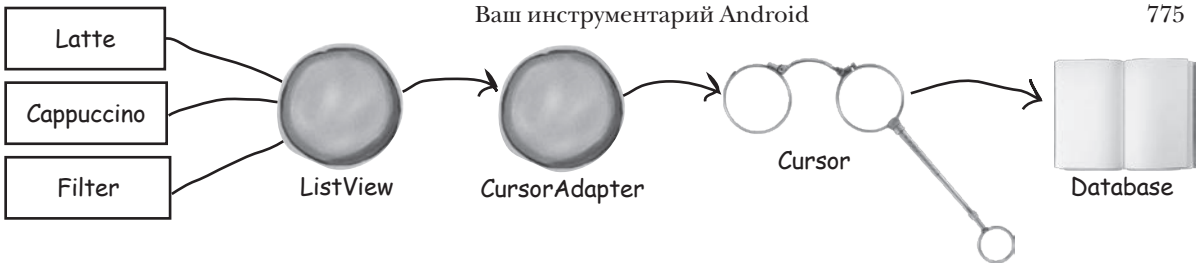
В предыдущей главе было показано, как создать базу данных SQLite с использованием помощника SQLite. Пора сделать следующий шаг — узнать, как работать с базой данных из активностей. Эта глава посвящена чтению данных из базы. Вы узнаете, как использовать курсоры для получения информации из базы данных, как перемещаться по набору данных с использованием курсора и как получить данные из курсора. Затем мы покажем, как использовать адаптеры курсоров для их связывания со списковыми представлениями.

17 Курсоры и асинхронные задачи

Выполнение в фоновом режиме

В большинстве приложений данные должны обновляться. Вы научились создавать приложения, читающие данные из баз данных SQLite. Но что, если данные приложения должны обновляться? В этой главе вы узнаете, как научить приложение реагировать на ввод данных пользователем и обновлять значения в базе данных. Также вы узнаете, как обновлять содержимое экрана после модификации данных. В завершение мы покажем, как написание эффективного многопоточного кода с объектами AsyncTask ускоряет работу приложений.

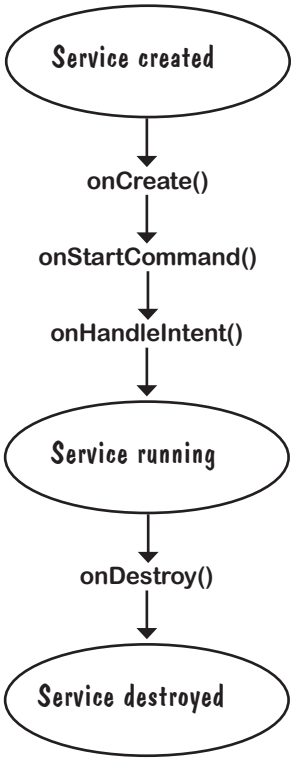




18 Службы

К вашим услугам

Существуют операции, которые должны выполняться постоянно, какое бы приложение ни обладало фокусом. 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 — компоненты, выполняющие операции в фоновом режиме, научитесь создавать службы при помощи класса IntentService. Также мы разберемся в том, как жизненный цикл служб связан с жизненным циклом активности. Заодно вы научитесь регистрировать сообщения и держать пользователей в курсе дел с использованием встроенной службы уведомлений Android.



19 Связанные службы и разрешения

Связаны вместе

Запускаемые службы отлично подходят для фоновых операций — а если вам нужна служба с большей интерактивностью? В этой главе вы научитесь создавать связанные службы — разновидность служб, с которыми могут взаимодействовать ваши активности. Вы узнаете, как выполнить привязку к службе и как отменить ее после завершения работы для экономии ресурсов. Служба позиционирования Android поможет вам получать информацию местонахождения от GPS-приемника вашего устройства. Наконец, вы научитесь пользоваться моделью разрешений Android, включая обработку запросов разрешений во время выполнения.