

Евгений Умрихин

Разработка Android-приложений на C# с использованием Xamarin с нуля



Вы узнаете, как:

- использовать интегрированную среду разработки (IDE);
- работать с Activity;
- выбирать и использовать управляющие элементы;
- записывать и считывать данные;
- взаимодействовать с веб-сервисами и средствами коммуникаций;
- собирать написанные на C# мобильные приложения;
- публиковать приложения в Google Play.



Евгений Умрихин

Разработка
Android-приложений
на **C#** с использованием
Xamarin с нуля

Санкт-Петербург
«БХВ-Петербург»

2021

УДК 004.43
ББК 32.973.26-018.1
У55

Умрихин Е. Д.

У55 Разработка Android-приложений на C# с использованием Xamarin с нуля. — СПб.: БХВ-Петербург, 2021. — 336 с.: ил. — (С нуля)

ISBN 978-5-9775-6671-1

Рассмотрены особенности создания Android-приложений при помощи Visual Studio Community на C#. Книга знакомит читателя со структурой проектов Xamarin.Android, с особенностями сборки и отладки приложений. Рассматриваются основные подходы к разработке Android-приложений, методы построения интерфейса, хранения данных, показана интеграция мобильных приложений с веб-службами, описаны особенности распространения и публикации приложений в магазине Google Play, разобраны основные методы монетизации мобильного контента. Представлены многочисленные примеры кода для решения различных задач, который можно использовать в своих приложениях.

Для начинающих разработчиков

УДК 004.43
ББК 32.973.26-018.1

Группа подготовки издания:

Руководитель проекта	<i>Павел Шалин</i>
Зав. редакцией	<i>Екатерина Сависте</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Карины Соловьевой</i>

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-5-9775-6671-1

© Умрихин Е. Д., 2021
© Оформление. ООО "БХВ-Петербург", ООО "БХВ", 2021

ОГЛАВЛЕНИЕ

Введение	5
Что вы узнаете из этой книги?	7
Что нужно знать?	8
Основные термины и понятия	8
Об авторе	9
 Глава 1. Освоение интегрированной среды разработки (IDE)	11
Знакомство с Visual Studio Community	11
Содержание проекта	12
Ресурсы приложения	13
Проектирование экранных форм	15
Редактор кода	16
Приложение HelloWorld	17
Локализация приложений	21
Принципы работы приложения и основы архитектуры	23
Описание исходного кода	26
Тестирование и отладка приложения	29
 Глава 2. Работа с Activity	31
Жизненный цикл приложения	31
Управление жизненным циклом	35
Атрибут <i>[Activity]</i> и основные свойства активности	38
Намерения и передача данных между Activity	41
Получение результата из Activity	46
Сохранение настроек Activity	52
Взаимодействие между Activity	54
Передача данных между Activity через статические переменные	56
Диалоговые Activity	57
 Глава 3. Выбор и использование управляющих компонентов	60
<i>Layouts</i> в Android	60
Контейнер <i>ScrollView</i>	67
Элемент <i>TextView</i>	67
Элемент <i>EditText</i>	68
Элемент <i>Button</i>	69
Элемент <i>Checkbox</i>	78
Элемент <i>RadioButton</i>	78
Элемент <i>SeekBar</i>	79
Элемент <i>ImageView</i>	86
Анимация свойств	91

Адаптеры и списки	94
Выбор даты и времени	109
Создание меню	110
Элемент <i>Toolbar</i>	116
Контекстная панель действий	119
Введение во фрагменты	124
Разработка интерфейса в стиле WhatsApp	128
Использование асинхронных методов	135
Элемент <i>SwipeRefreshLayout</i>	140
Создание экранных заставок	142
Перелистывание страниц и <i>ViewPager</i>	145
Создание собственных компонентов	148
Компоненты построения диаграмм	151
Создание интерфейса в стиле Gmail	156
Глава 4. Запись и чтение данных	165
Чтение и сохранение файлов	165
Основы работы с SQLite	174
Добавление, удаление и обновление данных в SQLite	182
Динамический поиск по базе данных SQLite	188
Глава 5. Работа с веб-сервисами	194
Введение в веб-сервисы	194
Веб-службы ASP.NET (ASMX)	195
Использование служб RESTful	207
Глава 6. Обзор основных функций	219
Телефонные звонки	219
Отправка СМС	222
Работа с контактами	226
Определение местоположения	235
Захват изображения и работа с камерой	240
Распознавание речи	248
Введение в сервисы	252
Широковещательные приёмники	257
Уведомления	261
Разработка сканера кассовых чеков	264
Оптическое распознавание символов	270
Разработка экранных виджетов	276
Вывод данных на печать	284
Компонент <i>ShortcutBadger</i>	289
Использование диспетчера оповещений	290
Отслеживание телефонных звонков	293
Глава 7. Сборка и распространение приложений	305
Создание подписанного apk	305
Публикация приложения в Play Market	311
Монетизация мобильных приложений	314
Заключение	329
Предметный указатель	332

ВВЕДЕНИЕ

Мобильные устройства давно вошли в нашу жизнь. За последние годы лишь укрепился тренд в увеличении доли смартфонов и планшетов на рынках мобильных телефонов и персональных компьютеров. Бизнес все активнее переносит свои сервисы на мобильные площадки. Безусловным лидером среди таких платформ является операционная система Android. По данным различных источников, ее доля достигла 90% всего рынка мобильных операционных систем. Компания Google — разработчик этой операционной системы — предоставляет всю необходимую инфраструктуру для создания и распространения программных решений для мобильных устройств на базе Android. Вместе с тем официальной средой разработки приложений считается Android Studio, которая включает такие языки программирования, как Java, Kotlin, C++. Эти средства разработки принято считать *нативными*.

Использование нативных средств разработки — довольно правильный подход при создании качественных бизнес-приложений. Однако на практике такой способ не всегда оказывается экономически выгодным. Для бизнеса создание мобильной версии своего сервиса сопряжено со значительными расходами: приходится держать в команде несколько дорогостоящих специалистов, стоимость разработки приложений в специализированной компании может быть еще выше, притом что для каждой мобильной платформы используются свои нативные средства, а сервис должен работать минимум на двух самых популярных. В таких ситуациях использовать другие способы разработки оказывается целесообразней.

Кроме нативных существуют так называемые *гибридные технологии*. Их суть заключается в совместном использовании нативных средств разработки и веб-разработки. В гибридном приложении нативная часть — это контейнер WebView (веб-представление), расположенный на основном экране приложения и обеспечивающий взаимодействие с операционной системой. При этом весь интерфейс и весь функционал разрабатываются при помощи стека технологий, принятых в веб: HTML5 в связке с JavaScript. Чем-то это напоминает обычный браузер. Преимуществом такого подхода является то, что разработка ведется сразу для нескольких платформ за счет повторного использования кода. Это приводит к значительной экономии расходов и выходу на рынок программного продукта для нескольких платформ одновременно. К сожалению, у данного подхода есть и слабые стороны, например доступ к новым возможностям мобильной платформы происходит не сразу, а лишь после изменения фреймворка. Вносят свою лепту и низкая скорость, сложности с дизайном из-за разных подходов к нему в операционных системах, незащищенность кода.

Таких недостатков лишена одна из технологий для разработки мобильных приложений — *Xamarin* (произносится [з'эмэрин]), набирающая сейчас популярность. *Xamarin* позволяет разрабатывать приложения на языке C#, который входит в топ 10 самых востребованных языков программирования в мире и поддерживается Microsoft — одной из крупнейших ИТ-корпораций. При этом используются библиотеки классов и среда исполнения, работающие на всех платформах, таких как Android, iOS и Windows. Как и в случае применения нативных средств разработки, производится не интерпретация, а компиляция собственных приложений. Они достаточно производительны, в отличие от гибридных приложений. Это справедливо даже для игр, которые зачастую очень ресурсоемки. Имея привязки ко всем базовым пакетам SDK в мобильных платформах Android и iOS, *Xamarin* позволяет использовать библиотеки Java, Objective-C и пр. Таким образом, продукт предоставляет доступ к функциональным возможностям существующих мобильных операционных систем и имеет ряд собственных. В частности, разработка приложений осуществляется в Microsoft Visual Studio, которая имеет качественную кросс-платформенную поддержку для основных мобильных платформ и усовершенствованную систему управления проектами. В настоящее время существуют реализации этой среды разработки для двух операционных систем: Windows и macOS. Кроме того, большинство .Net-классов и технологий, таких как, например, Linq или ASP.NET, традиционно обласканных .Net-разработчиками, можно использовать в *Xamarin*-проектах.

Xamarin представлен двумя коммерческими продуктами: *Xamarin.Android* (ранее назывался MonoDroid) и *Xamarin.iOS* (ранее назывался MonoTouch), которые созданы на базе Mono — проекта по созданию полноценного воплощения системы .NET Framework на базе свободного программного обеспечения и опубликованных стандартах .NET ECMA. Mono впервые был выпущен уже почти два десятилетия назад, является практически ровесником .NET Framework и имеет поддержку значительного числа операционных систем: Windows, Linux, BSD (FreeBSD, OpenBSD, NetBSD), Solaris, macOS (Mac OS X) и др. Кроме того, существует еще технология кросс-платформенной разработки *Xamarin.Forms*, позволяющая использовать одну логику для всех платформ. В этом случае можно графический интерфейс определить один раз, на C# реализовать требуемую логику и получить работающие приложения для Android, iOS и Windows. Эта платформа функционально состоит из субплатформ *Xamarin.Android* и *Xamarin.iOS*. Благодаря такой организации через них запросы приложений направляются к прикладным интерфейсам на устройствах под управлением ОС Android или iOS.

Появление *Xamarin* позволило приобщиться многочисленной армии C#-программистов к мобильной разработке, что в некотором смысле можно считать революционным событием. Прежде всего, это открывает некоторые перспективы для Mono стать более популярным и востребованным проектом. Наряду с этим специалисты C# теперь не являются простыми наблюдателями. *Xamarin* дал им шанс стать полноценными мобильными разработчиками. Несмотря на то что эта возможность появилась уже давно, до сих пор, кроме официального руководства, которое в ряде случаев является довольно сложным для понимания, фактически нет ни одного

учебного пособия на русском языке, при помощи которого можно было бы получить общие представления об этой платформе разработки в доступной форме на практических примерах. Вместе с тем имеются многочисленные руководства на английском языке, что свидетельствует о невероятной популярности технологии.

В данной книге делается попытка изменить сложившуюся ситуацию. В ней будут рассмотрены основы создания мобильных приложений с использованием Xamarin.Android, которая позволяет разрабатывать нативные приложения для наиболее популярной мобильной операционной системы.

Что вы узнаете из этой книги?

Глава 1. Освоение интегрированной среды разработки (IDE). В этой главе мы познакомимся с основными инструментами разработки приложений для Android при помощи Visual Studio, рассмотрим структуру проектов таких приложений и напишем самую первую программу.

Глава 2. Работа с Activity. Данная глава посвящена основному и важнейшему экранному представлению Android-приложения — компоненту, с которым пользователи могут взаимодействовать при выполнении каких-либо действий, например когда требуется набрать номер телефона, сделать фото, отправить письмо или просмотреть карту.

Глава 3. Выбор и использование управляющих компонентов. Здесь мы познакомимся с основными компонентами, которые используются во всех Android-приложениях для построения визуального интерфейса, и научимся разрабатывать свои.

Глава 4. Запись и чтение данных. В этой главе рассказано, как сохранить данные на устройстве и прочитать их. Также вы познакомитесь с популярной мобильной системой управления данными SQLite.

Глава 5. Работа с веб-сервисам. Здесь пойдет речь об основах построения многоуровневых приложений, использующих веб-службы для обработки данных. Мы создадим простую веб-службу на ASP.NET, которая будет запрашивать информацию в созданной нами базе данных, модифицировать в ней данные и предоставлять их мобильному приложению.

Глава 6. Обзор основных функций. В этой главе продемонстрированы основные приемы работы с телефонными звонками, СМС-сообщениями, контактами, уведомлениями, камерой; описано, как определять местоположение, осуществлять распознавание речи. Рассмотрена организация функций приложения в виде сервиса, и показано, как запустить его через широкоэвентельный приёмник. Также представлены практические примеры использования сканера, распознавания образов, печати и ряда других важных функций в разрабатываемых приложениях.

Глава 7. Сборка и распространение приложений. Здесь рассмотрен порядок создания арк-пакетов для распространения программ через магазин Google Play, а также некоторые способы монетизации приложений. Приведен пример создания платного контента и биллинга.

Рассмотрение каждой темы сопровождается многочисленными несложными примерами, дающими необходимые представления об используемой технологии. С целью увеличения эффективности изучения материала их имеет смысл воспроизводить в реальной среде разработки. В книге в ряде случаев применяются некоторые упрощения. Так, например, в основном тексте, с целью сокращения объема описательной части, сигнатуры методов могут быть представлены в сокращенном виде с использованием пустых скобок () либо упоминаются лишь названия методов, а входные параметры при этом опущены. Для получения полных сведений по применению тех или иных методов, в том числе для того, чтобы отследить их аргументы, следует обращаться к листингам с примерами использования этих процедур.

Все примеры проектов, приведенных в данной книге, были реализованы и протестированы в средах разработки Microsoft Visual Studio 2017/2019. Вместе с тем возможны отличия в работе представленного кода и в описанных методиках для более ранних версий Visual Studio.

Надеюсь, книга окажется полезной всем, кто интересуется мобильной разработкой и хотя бы немного знаком с C#, но пока еще не решается приступить к написанию своего первого Android-приложения.

Что нужно знать?

Предполагается, что читатели знакомы с основами работы в среде Windows, поскольку мы будем использовать для разработки мобильных приложений Visual Studio для Windows.

Вы также должны уметь работать в среде Android, иметь представления, как устанавливать, запускать и удалять приложения, работать с магазином приложений, производить настройки устройства и сопряжение с ПК.

Необходимо иметь некоторые базовые знания о языке разметки XML, языке управления реляционными базами данных SQL, об основных синтаксических конструкциях языка C# и архитектуре .Net-платформы в целом.

Основные термины и понятия

Определим несколько терминов, чтобы избежать их неправильного толкования.

Термин «гаджет» в широком смысле означает небольшое устройство, приспособление, предназначенное для облегчения и усовершенствования различных видов деятельности человека. Гаджеты широко распространены в самых разных сферах: в спорте — фитнес-трекеры, смарт-браслеты, спортивные устройства, в том числе и "умная" одежда; в медицине — электронные пластыри, трикодеры, гидрокоптеры, экзоскелеты. В программном обеспечении зачастую гаджет — небольшое приложение, предоставляющее дополнительную информацию, например прогноз погоды или курс валют. В данной книге под этим термином следует понимать мобильное устройство, работающее под управлением операционной системы Android. Этот термин следует относить именно к аппаратным средствам.

Виджет — это небольшое вспомогательное приложение, которое выполняет определенную функцию. Иногда так называют элементы управления для построения макета интерфейса. В целом в книге под этим термином следует понимать именно примитив графического интерфейса пользователя, имеющий стандартный внешний вид и выполняющий стандартные действия, элемент управления, компонент интерфейса или контрол. Тем не менее следует обращать внимание на контекст употребления термина. В одном из разделов будет упоминание о плавающем виджете для голосового ввода, который представляет собой микропрограмму. Кроме того, в одной из тем будут рассматриваться так называемые экранные виджеты, которые тоже являются микроприложениями, а не графическими примитивами.

Термин **Activity** еще не устоялся в русском языке у разработчиков. Некоторые используют слово "активность", другие — "деятельность" или "действие". Будем понимать под этим термином экранное представление, экран или окно, компонент приложения, с которым пользователи могут взаимодействовать для выполнения каких-либо действий.

В Android-приложениях многие операции работают через **намерения**. Под этим термином будем понимать программный механизм, который позволяет пользователям координировать функции различных видов деятельности для достижения цели задачи. Реализуется через объект обмена сообщениями `Intent`, с помощью которого можно запросить выполнение действия.

Жизненный цикл — это стадии процесса, охватывающие различные состояния системы, начиная с момента возникновения необходимости в такой системе и заканчивая ее полным уничтожением. Этот термин применяется в различных областях знаний, в том числе существует понятие жизненного цикла программного обеспечения. Однако в данной книге этот термин будет употребляться в контексте компонентов приложения, т. е. это совокупность стадий существования компонента, которые он проходит за период своего существования. Этот термин часто употребляется и по отношению к активности в целом.

Лейауты — группа компонентов, используемых для создания макета графического интерфейса пользователя, которые являются контейнерами для других элементов управления. Как правило, в тегах таких элементов употребляется лексема `Layout`.

Об авторе

Умрихин Евгений Дмитриевич — программист-разработчик с 15-летним опытом работы в крупной региональной страховой компании. Имеет опыт разработки и внедрения распределенных IT-решений с использованием веб-сервисов и мобильных платформ. Обладатель ряда авторских свидетельств об официальной регистрации программ для ЭВМ и ученой степени кандидата технических наук.