

## Добро пожаловать в мир Android

Android — самая популярная мобильная платформа в мире. Согласно последним опросам, в мире свыше *трех миллиардов* активных Android-устройств, и их количество продолжает стремительно расти. Android — полнофункциональная платформа с открытым кодом на базе Linux, разрабатываемая компанией Google. Это мощная платформа разработки, включающая все необходимое для построения современных приложений. Более того, построенные приложения могут устанавливаться на множестве разных устройств — телефонах, планшетах и не только. Что же собой представляет типичное Android-приложение?

### Активности определяют, что приложение делает

Каждое приложение Android включает одну или несколько **активностей**. Активность представляет собой специальный класс, обычно написанный на Kotlin, который управляет поведением приложения и решает, как реагировать на действия пользователя. Например, если в приложении присутствует кнопка, в активность добавляется код, который указывает, что должно происходить при нажатии этой кнопки.

### Макеты определяют, как будут выглядеть экраны приложения

Типичное приложение Android состоит из одного или нескольких экранов. Внешний вид каждого экрана определяется файлом **макета** или дополнительным кодом активности. Макеты обычно определяются на языке XML, и каждый экран может включать такие компоненты, как кнопки, текст и графика.

### Также могут понадобиться другие файлы

Кроме активностей и макетов, приложениям Android часто требуются дополнительные ресурсы: графические файлы, данные приложений и т. д. В приложение можно включить любые дополнительные файлы, которые вам понадобятся.

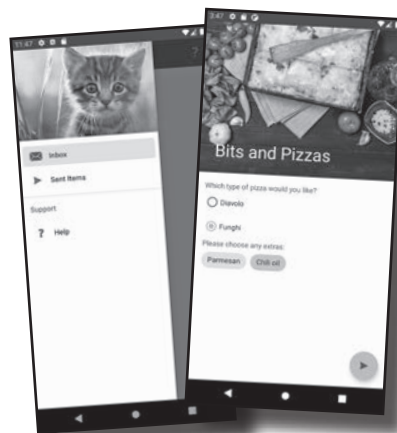
На самом деле приложение Android — всего лишь набор файлов в заранее определенных каталогах. При построении приложения все эти файлы собираются воедино, и вы получаете приложение, которое можно запустить на вашем устройстве.

Для построения приложений Android мы будем использовать Kotlin и XML. Некоторые вещи будут объясняться по ходу дела, но чтобы извлечь пользу из этой книги, читатель должен понимать Kotlin.



MainActivity

← Активности определяют, что приложение должно делать.



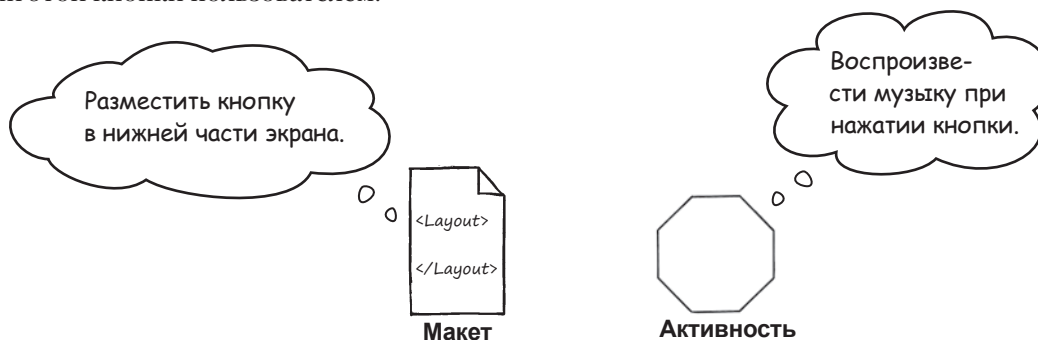
← Макеты сообщают Android, как будут выглядеть экраны вашего приложения.



← Также приложение может включать другие файлы (например, графические).

## Активности и макеты образуют костяк вашего приложения

В типичном приложении активности и макеты работают совместно для определения пользовательского интерфейса приложения. Макеты сообщают Android, как должны быть упорядочены различные элементы экрана, а активности управляют поведением приложения. Например, если в приложении присутствует кнопка, макет определяет ее позицию, а активность управляет тем, что происходит при нажатии этой кнопки пользователем.



А вот как активности и макеты работают совместно при запуске приложения на устройстве:

1 Android запускает главную активность приложения.

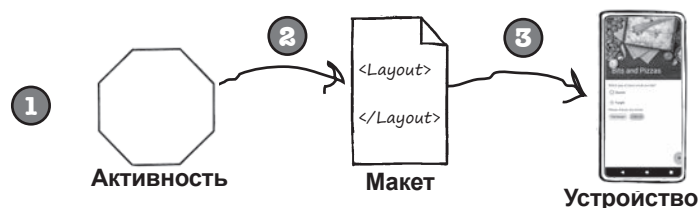
2 Активность приказывает Android использовать конкретный макет.

3 Макет отображается на устройстве.

4 Пользователь взаимодействует с макетом.

5 Активность реагирует на эти взаимодействия и обновляет изображение...

6 ...которое пользователь видит на устройстве.



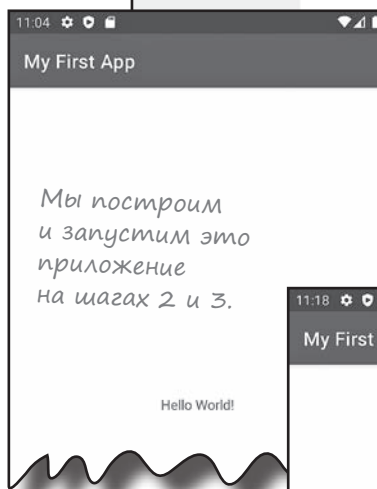
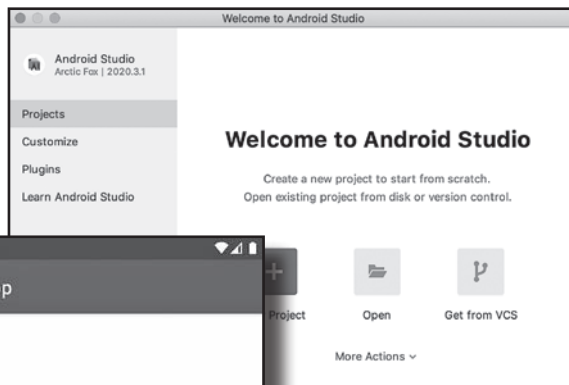
Теперь вы в общих чертах представляете, как строятся приложения Android, сделаем следующий шаг и построим простейшее приложение Android.

## Вот что мы сейчас сделаем

Давайте с ходу возьмемся за дело и построим простейшее Android-приложение. Для этого необходимо выполнить лишь несколько действий:

- 1 Подготовка среды разработки.**  
Необходимо установить среду Android Studio, включающую все необходимое для разработки Android-приложений.
- 2 Построение простейшего приложения.**  
Мы создадим в Android Studio простое приложение, которое будет выводить текст на экран.
- 3 Запуск приложения.**  
Мы запустим приложение на физическом и на виртуальном устройстве, чтобы увидеть его в действии.
- 4 Изменение приложения.**  
Наконец, мы внесем несколько изменений в созданное приложение и снова запустим его.

Для построения всех приложений в этой книге будет использоваться среда Android Studio.



### Часто задаваемые вопросы

**В:** Все Android-приложения пишутся на Kotlin?

**О:** Android-приложения также можно разрабатывать и на других языках (например, на Java), но лучше всего использовать Kotlin. Дело в том, что некоторые полезные возможности доступны только в Kotlin.

**В:** Как мне изучить Kotlin?

**О:** Наверное, мы необъективны, но на наш взгляд, Kotlin лучше всего изучать по нашей книге *Head First Kotlin*\*. В ней вы узнаете все, что необходимо знать о Kotlin, чтобы извлечь максимум пользы из книги.

**В:** Вы упоминали, что для определения внешнего вида приложения вместо фай-

лов макетов может использоваться код активности?

**О:** Да, с использованием инструментария, который называется Jetpack Compose.

Позднее в этой книге вы научитесь использовать Compose. А пока сосредоточимся на определении пользовательского интерфейса приложения в файлах макетов.

\* Гриффитс Дон, Гриффитс Дэвид. Head First Kotlin. СПб. Питер, 2020.

## Android Studio: среда разработки

Для разработки Android-приложений лучше всего использовать **Android Studio** — официальную интегрированную среду (IDE) для разработки Android-приложений.

Android Studio базируется на среде IntelliJ IDEA — возможно, она вам уже знакома. В нее включены редакторы кода, UI-инструменты и шаблоны, упрощающие вашу жизнь в мире Android.

Также она включает **Android SDK** (Android Software Development Kit), необходимый для разработки Android-приложений. Android SDK включает исходные файлы Android и компилятор, используемый для компиляции кода в формат Android.

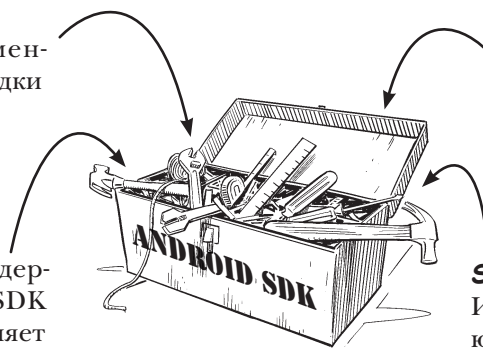
Основные компоненты Android SDK:

### SDK Tools

Полный набор инструментов для разработки и отладки Android-приложений.

### SDK Platform

Каждая версия Android содержит собственный пакет SDK Platform. Этот пакет позволяет компилировать приложения для данной версии Android и включает исходные файлы для этой версии.



### SDK Build Tools

Инструменты, необходимые для построения Android-приложений.

### SDK Platform Tools

Инструменты, взаимодействующие с платформой Android. Эти инструменты обладают обратной совместимостью, но некоторые возможности могут быть доступны только для новых версий Android.

## Необходимо установить Android Studio

Так как Android Studio включает все необходимое для разработки Android-приложений, эта среда будет использоваться для построения всех приложений, представленных в книге.

Прежде чем двигаться дальше, **необходимо установить Android Studio на вашей машине**. О том, как это сделать, более подробно рассказано на следующей странице.

Вы находитесь здесь.

первые шаги



Подготовка среды

Построение приложения

Запуск приложения

Изменение приложения

## Установка Android Studio

Чтобы извлечь максимум пользы из книги, необходимо установить среду Android Studio. Мы не приводим здесь полные инструкции по установке, потому что они быстро устаревают, но если вы будете следовать инструкциям в интернете, все будет нормально.

Сначала проверьте системные требования Android Studio:

<https://developer.android.com/studio#Requirements>

Затем загрузите Android Studio по этому адресу:

<https://developer.android.com/studio>

*Эти URL-адреса иногда изменяются. Если адреса не работают, поищите Android Studio в интернете — вы найдете соответствующие страницы.*

и выполните инструкции по установке.

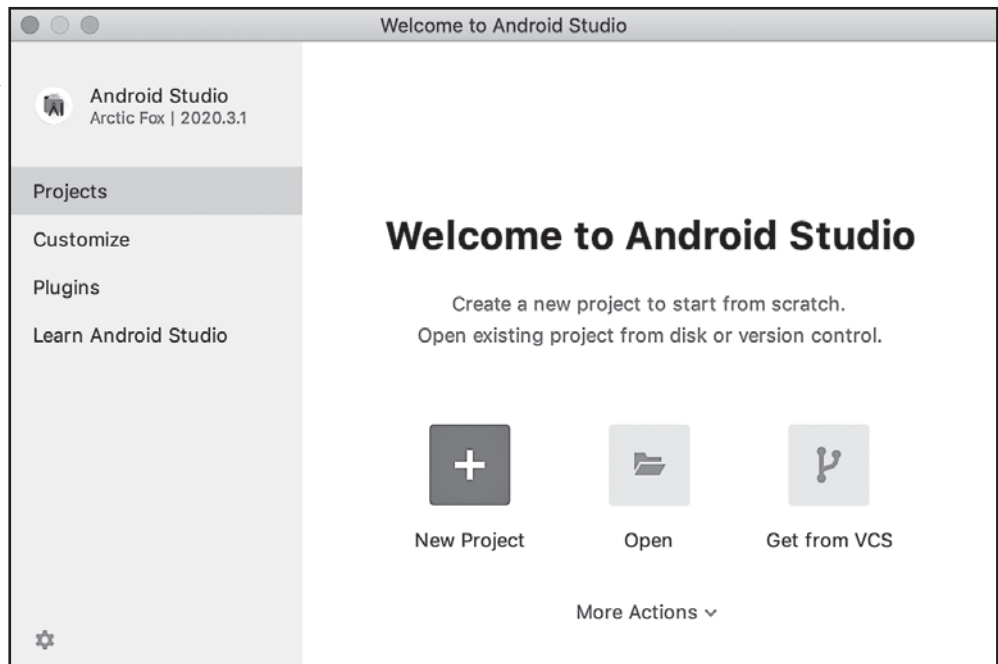
Когда установка будет завершена, откройте Android Studio и выполните инструкции по добавлению новейших инструментов SDK и библиотек поддержки.

**Если ранее вы установили более старую версию Android Studio, мы рекомендуем восстановить настройки IDE по умолчанию.** Для этого откройте меню File, выберите команду Manage IDE Settings, а затем Restore Default Settings.

*← Сбрасывает все старые настройки, которые могли храниться в Android Studio. Они могли бы помешать выполнению вашего кода.*

Когда все будет сделано, на экране появляется заставка Android Studio:

*Заставка Android Studio включает список возможных действий.*



- Подготовка среды
- Построение приложения
- Запуск приложения
- Изменение приложения

**В этой книге используется версия Android Studio 2020.3.1 (также известная как Arctic Fox). Убедитесь в том, что вы установили эту или более позднюю версию.**

Этот шаг завершен,  
мы его вычеркиваем.



*первые шаги*

Подготовка среды  
Построение приложения  
Запуск приложения  
Изменение приложения

## Построение простого приложения

Итак, среда разработки подготовлена, и можно переходить к созданию вашего первого Android-приложения. Вот как оно будет выглядеть:

Приложение, которое мы собираемся построить. Оно очень простое, но для первого Android-приложения этого вполне достаточно.



Давайте займемся построением приложения.

### Часто Задаваемые Вопросы

**В:** А мне обязательно использовать Android Studio для построения Android-приложений? Нельзя ли использовать другую IDE?

**О:** Строго говоря, все, что вам нужно, — это инструменты для написания и компиляции Kotlin-кода, а также некоторые специализированные инструменты для преобразования откомпилированного кода в форму, которая может запускаться на Android-устройствах. Тем не менее Android Studio — официальная среда разработки Android, и команда Android рекомендует именно ее.

Мы полагаем, что эта среда идеально подходит для разработки Android-приложений, поэтому мы остановили свой выбор на ней.

**В:** Возможно ли создавать Android-приложения без IDE?

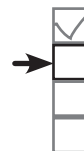
**О:** Возможно, но это заметно увеличит объем работы. Гораздо быстрее и проще работать в IDE, поэтому мы рекомендуем использовать Android Studio.

## Как построить приложение

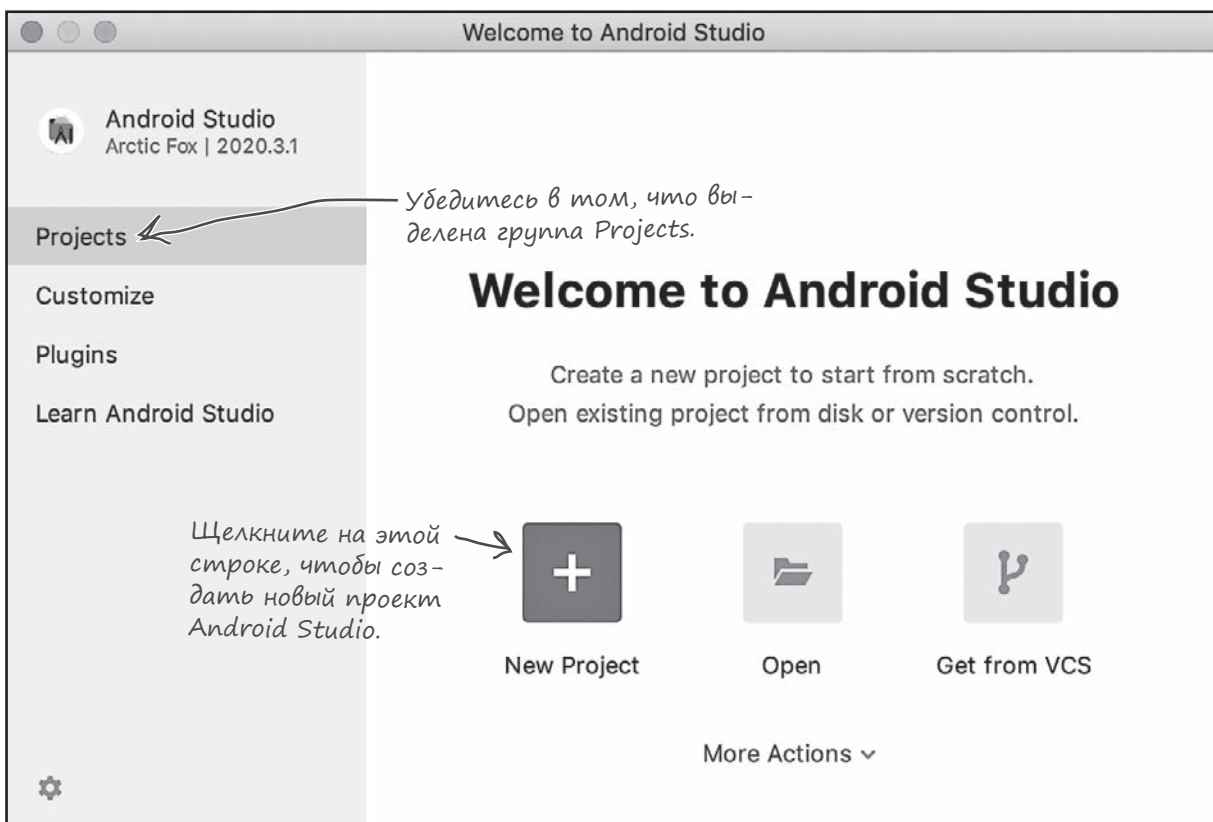
Каждый раз, когда вы создаете новое приложение, для него необходимо создать новый проект. Убедитесь в том, что среда Android Studio открыта, и повторяйте за нами.

### 1. Создание нового проекта.

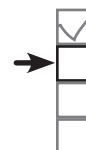
На заставке Android Studio перечислены некоторые возможные операции. Мы хотим создать новый проект; убедитесь в том, что выбрана категория Projects, и щелкните на варианте «New Project».



Подготовка среды  
Построение приложения  
Запуск приложения  
Изменение приложения



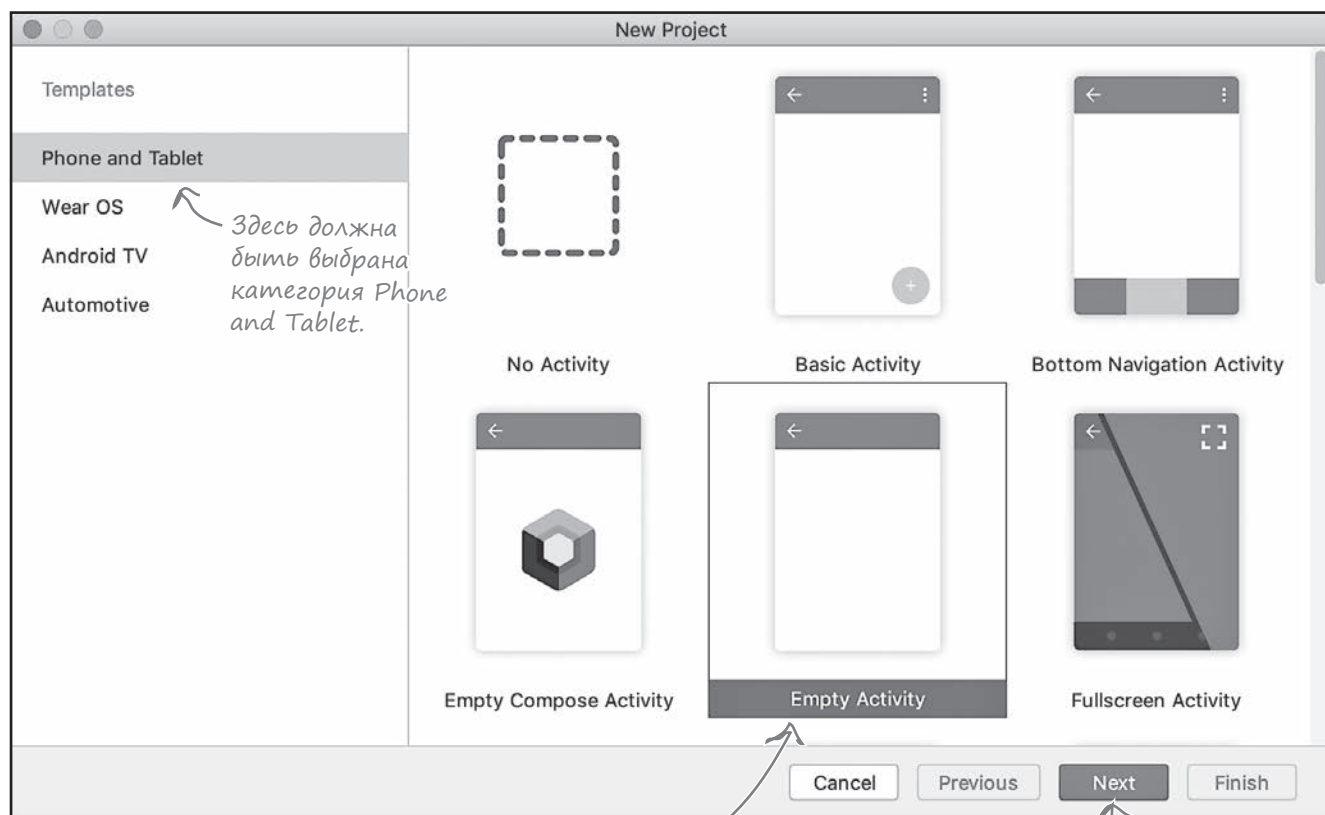




Подготовка среды  
 Построение приложения  
 Запуск приложения  
 Изменение приложения

## 2. Выбор шаблона приложения.

Далее необходимо задать тип создаваемого проекта Android Studio. Мы создадим приложение с пустой активностью, которое будет работать на телефоне или на планшете, поэтому проследите за тем, чтобы была выбрана категория Phone and Tablet, и выберите пустую активность (Empty Activity). Через несколько страниц вы узнаете, что дает вам шаблон Empty Activity, а пока перейдите к следующему шагу нажатием кнопки Next.



*Здесь должна быть выбрана категория Phone and Tablet.*

*Существуют и другие типы активностей, которые вы можете выбрать, но для данного упражнения следует выбрать пустую активность (Empty Activity).*

*Когда все будет сделано, нажмите кнопку Next.*

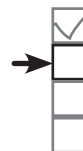


### 3. Настройка проекта.

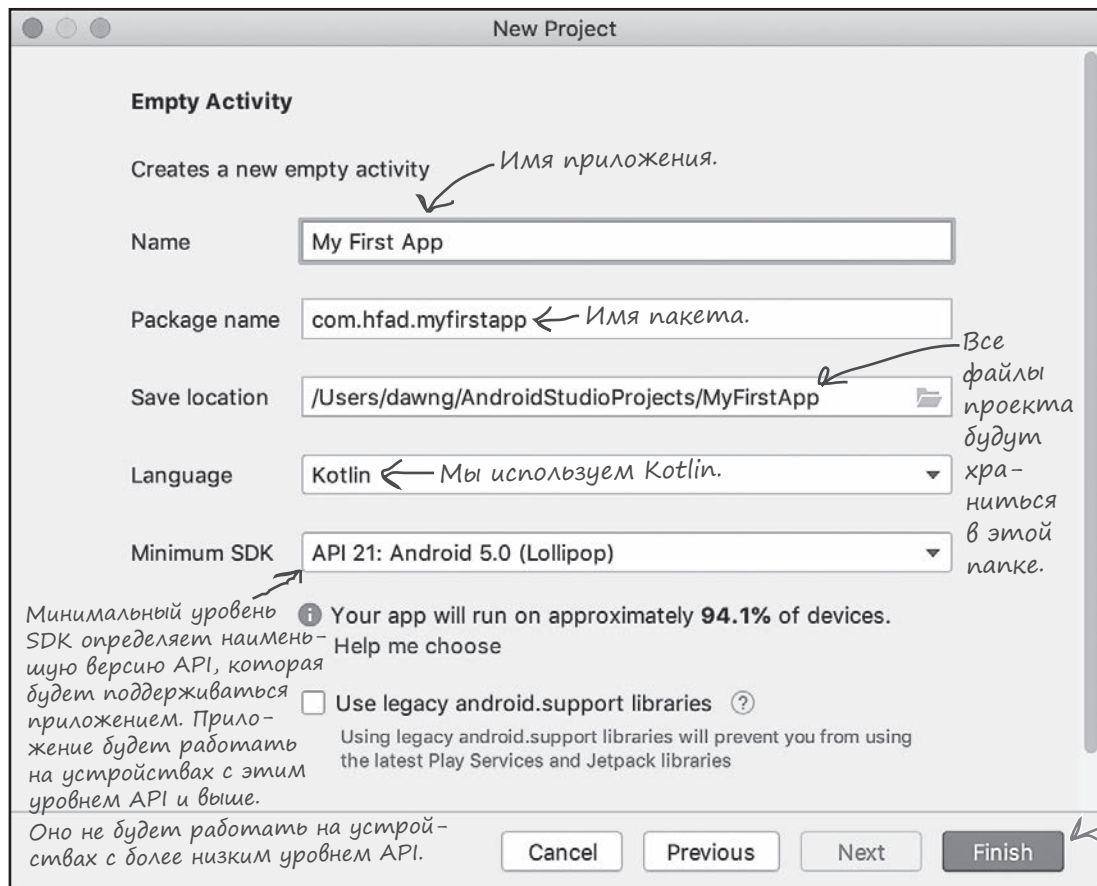
Теперь необходимо настроить конфигурацию приложения: указать его имя, имя пакета и где должны храниться его файлы. Введите имя «My First App» и имя пакета «com.hfad.myfirstapp», подтвердите место сохранения по умолчанию.

Также необходимо сообщить Android Studio, какой язык программирования вы будете использовать, и задать минимальную версию SDK. Этот параметр определяет наименьшую версию Android, которая будет поддерживаться приложением: уровни SDK более подробно рассматриваются на следующей странице.

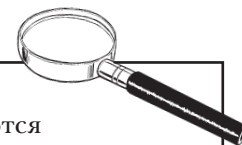
Выберите язык Kotlin и минимальную версию API 21, чтобы приложение работало на большинстве устройств. Когда вы щелкнете на кнопке Finish, Android Studio создаст проект. Через пару страниц вы узнаете, что при этом происходит.



Подготовка среды  
**Построение приложения**  
 Запуск приложения  
 Изменение приложения



## Версии Android под увеличительным стеклом



Вероятно, вам не раз доводилось слышать, как применительно к Android упоминаются разные «вкусные» названия: Nougat (нуга), Oreo и Pie (пирог). Что это за кондитерская?

Каждой версии Android присваивается номер и кодовое имя. Номер версии определяет конкретную версию Android (например, 9.0), тогда как кодовое имя представляет собой чуть более общее «дружественное» имя, которое может объединять сразу несколько версий Android (например, Pie). Под «уровнем API» понимается версия API, используемых приложением. Например, Android версии 9.0 соответствует уровень API 28.

После Pie кодовые имена Android перестали быть «вкусными». Например, Android версии 10.0 называется просто «Android 10».

Версия	Кодовое имя	Уровень API
1.0		1
1.1	Petit Four	2
1.5	Cupcake	3
1.6	Donut	4
2.0–2.1	Eclair	5–7
2.2–2.2.3	Froyo	8
2.3–2.3.7	Gingerbread	9–10
3.0–3.2.6	Honeycomb	11–13
4.0–4.0.4	Ice Cream Sandwich	14–15
4.1–4.3.1	Jelly Bean	16–18
4.4–4.4.4	KitKat	19–20
5.0–5.1.1	Lollipop	21–22
6.0–6.0.1	Marshmallow	23
7.0–7.1.2	Nougat	24–25
8.0–8.1	Oreo	26–27
9.0	Pie	28
10.0	10	29
11.0	11	30
12.0	12	31

*Сейчас эти версии уже никем не используются.*

*На большинстве устройств используется один из этих уровней API.*

При разработке Android-приложений необходимо учитывать, с какими версиями Android должно быть совместимо ваше приложение. Если вы укажете, что приложение совместимо только с самой последней версией SDK, может оказаться, что оно не запускается на очень многих устройствах. Чтобы получить информацию о процентном распределении версий по устройствам, выберите вариант «Help me choose» при создании нового проекта.