22 Testing for Accessibility

Тестирование доступности

[1. Введение](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#0)

В этой лаборатории вы узнаете, как тестировать и улучшать доступность ваших приложений.

Люди с нарушениями зрения, дальтонизмом, слухом, ловкостью, когнитивными нарушениями и многими другими нарушениями используют устройства Android для выполнения задач в своей повседневной жизни. Когда вы разрабатываете приложения с учетом специальных возможностей, вы улучшаете пользовательский опыт, особенно для пользователей с этими и другими потребностями в специальных возможностях.

В ходе этой лабораторной работы вы будете использовать **TalkBack** и **Switch Access**, чтобы проверить доступность приложения **Woof**.

* **TalkBack** позволяет пользователям свободно управлять своими устройствами.
* **Switch Access** позволяет пользователям перемещаться по приложению с помощью переключателей вместо сенсорного экрана.

**Что вам понадобится**

* Компьютер с установленной Android Studio.
* Устройство Android или эмулятор, у которого есть доступ к приложению Google Play Store или уже установлен Android Accessibility Suite.
* Код решения для приложения **Woof** .

## [2. Приступаем к настройке](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#1)

## Загрузите стартовый код

В Android Studio откройте папку android-basics-kotlin-compose-woof.

**URL стартового кода:**

<https://github.com/google-developer-training/basic-android-kotlin-compose-training-woof>

**Название ветки со стартовым кодом:** main

Откройте код приложения **Woof** в Android Studio.

## ****Настройте свое устройство****

Если на вашем устройстве или эмуляторе уже установлен Android Accessibility Suite, вы можете пропустить эту часть. Если вам необходимо установить Android Accessibility Suite (для доступа к TalkBack и Switch Access), следуйте инструкциям ниже.

**Примечание.** Для этой лаборатории кода рекомендуется использовать физическое устройство, если оно у вас есть. Если у вас есть физическое устройство, вам не придется выполнять следующие шаги настройки.

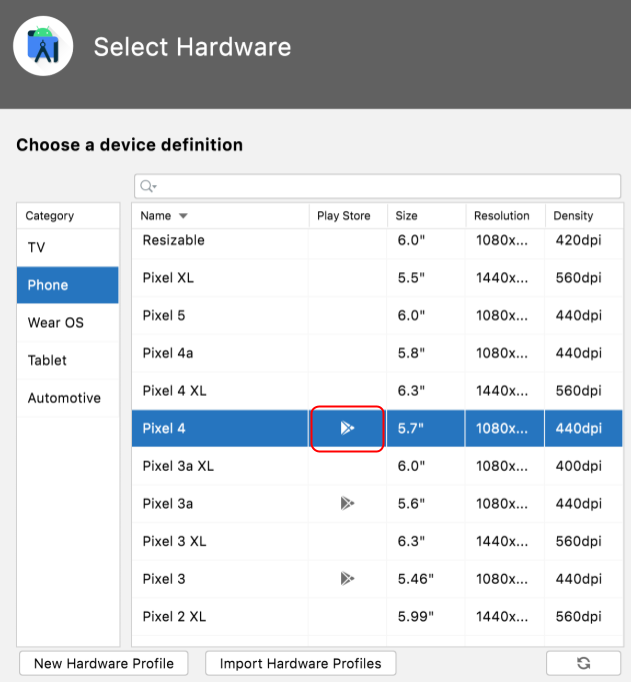
### **Настройте устройство с помощью приложения Google Play Store.**

Если вы используете физическое устройство, убедитесь, что вы:

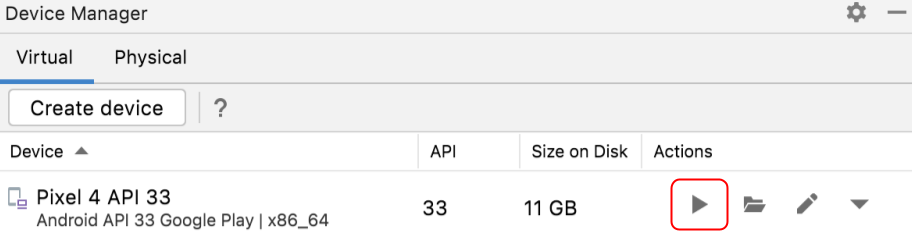
* Иметь доступ к приложению Play Store.
* Вы вошли в свою учетную запись Google.
* Имейте необходимые разрешения на своем устройстве для загрузки приложений из магазина Play Store.

Если вы используете эмулятор, выполните следующие действия, чтобы настроить эмулятор с доступом к приложению Play Store:

1. В Android Studio откройте **Диспетчер устройств** и выберите **«Создать устройство»** .
2. Убедитесь, что вы выбрали **Pixel 4** . Обратите внимание на значок приложения Play Store в столбце **Play Store**. Значок указывает, что этот эмулятор поставляется с приложением Google Play Store.

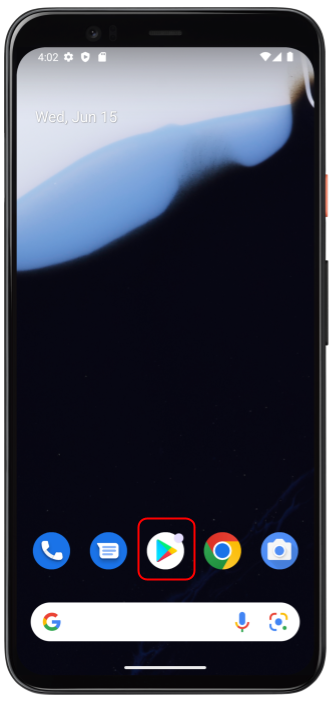


1. Нажмите **«Далее»** и приступайте к созданию эмулятора.
2. После создания эмулятора запустите его из **диспетчера устройств** , щелкнув значок стрелки.



1. Запустите приложение Play Store на эмуляторе и войдите в систему, используя действующую учетную запись Google.

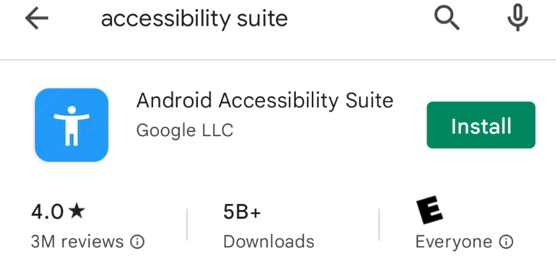
Если вы не видите панель запуска приложений Play Store в указанном ниже месте, вы можете найти ее на экране **«Все приложения»** .



### **Установите приложение Android Accessibility Suite.**

Android Accessibility Suite содержит коллекцию приложений для специальных возможностей. Он понадобится вам для использования **TalkBack** и **Switch Access** позже в этой лаборатории кода.

1. В приложении Google Play Store установите приложение [**Android Accessibility Suite**](https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback) .



## Установите приложение Woof.

Если вы создали новый профиль оборудования Pixel 4 в начале этого раздела, вам потребуется установить приложение **Woof** из загруженного вами кода решения.

[3. Используйте Woof с TalkBack](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#2)

[**TalkBack**](https://support.google.com/accessibility/android/answer/6283677) — это программа чтения с экрана Google, которая обеспечивает голосовую обратную связь, чтобы пользователи могли перемещаться по устройству, не глядя на экран. Это особенно полезно для людей с ослабленным зрением.

После включения TalkBack пользователи смогут перемещаться по своему устройству с помощью голосовой обратной связи и жестов, например смахивания и касания. Навигация с помощью TalkBack — отличный способ проверить, есть ли в вашем приложении области улучшения.

Выполните следующие действия, чтобы лучше познакомиться с TalkBack:

**Примечание.** Когда **TalkBack** включен, коснитесь, чтобы выбрать элемент, затем дважды коснитесь, чтобы активировать его. Используйте два пальца для пролистывания и навигации по системе.

1. Посмотрите следующее видео, чтобы узнать, как настроить и использовать TalkBack.

**Примечание.** Если у вас нет доступа к этому видео, ознакомьтесь с [документацией](https://support.google.com/accessibility/android/answer/6283677) .

1. Ознакомившись с **TalkBack** , примените полученные знания в приложении **Woof** !

**Примечание.** Звук TalkBack на эмуляторе может быть низкой громкости или низкого качества. Если вы столкнулись с этим, попробуйте увеличить громкость на эмуляторе.

1. Отключите функцию **TalkBack,** прежде чем перейти к следующему разделу. Выполните следующие действия, чтобы отключить **Talkback** :

На вашем устройстве или эмуляторе откройте **«Настройки»** .

Выберите **«Доступность»** , а затем **«TalkBack»** .

Отключите **параметр «Использовать TalkBack»** .

Выберите **ОК** .

Дополнительные способы отключения TalkBack вы можете узнать в [документации поддержки](https://support.google.com/accessibility/android/answer/6007100) . Имейте в виду, что некоторые из этих опций будут недоступны в эмуляторе, а другие могут быть устаревшими в определенных версиях Android.

[4. Используйте Woof с Switch Access](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#3)

Switch Access позволяет вам взаимодействовать с вашим устройством Android с помощью одного или нескольких переключателей вместо сенсорного экрана. Эта альтернатива использованию сенсорного экрана особенно полезна для пользователей с ограниченной подвижностью.

Switch Access сканирует элементы на вашем экране, выделяя каждый элемент по очереди, пока вы не сделаете выбор.

Чтобы использовать Switch Access, вам сначала понадобится один или несколько коммутаторов. Существует несколько типов переключателей, но в этой лаборатории мы будем использовать встроенные кнопки громкости на устройстве Android.

1. Посмотрите следующее видео, чтобы узнать, как настроить и использовать **[Switch Access](https://support.google.com/accessibility/android/answer/6122836" \t "_blank)** .

**Примечание.** Если у вас нет доступа к этому видео, ознакомьтесь с [документацией](https://support.google.com/accessibility/android/answer/6122836) .

1. Если вы настроили кнопки громкости в соответствии с инструкциями в видео, Switch Access позволит вам нажимать кнопку уменьшения громкости для перехода к различным элементам приложения. Как только элемент выделен, его можно выбрать с помощью кнопки увеличения громкости.

Для элементов с простым действием щелчка выбор элемента аналогичен выполнению действия касания. Для элементов, для которых доступны настраиваемые действия специальных возможностей, выбор элемента предоставляет пользователю различные действия, которые он может выполнить с этим элементом.

Когда Switch Access включен, в верхней части экрана устройства появляется вкладка **«Меню»** . При выборе вкладки открывается глобальное меню с параметрами навигации, такими как **«Назад»** и **«Домой»** , которые эквивалентны жестам на экране устройства. Некоторые параметры настраивают поведение Switch Access.

1. Ознакомившись с **Switch Access** , примените полученные знания в приложении **Woof** !
2. Отключите **Switch Access,** прежде чем перейти к следующему разделу.

[5. Улучшение доступности пользовательского интерфейса.](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#4)

Существует несколько вариантов дизайна пользовательского интерфейса, которые следует учитывать при попытке создать более доступное приложение. Помимо атрибутов и вариантов поведения, которые позволяют эффективно использовать TalkBack и Switch Access, ниже приведены некоторые оптимизации пользовательского интерфейса, которые вы можете сделать, чтобы улучшить доступность вашего приложения.

**Описание контента**

Пользователи служб специальных возможностей, таких как программы чтения с экрана (например, TalkBack), полагаются на описания контента, чтобы понять значение элементов интерфейса.

В некоторых случаях, например, когда информация внутри элемента передается графически, описания контента могут предоставлять текстовое описание значения или действия, связанного с элементом.

Если элементы пользовательского интерфейса не имеют меток содержимого, некоторым пользователям может быть сложно понять представленную им информацию или выполнить действия в интерфейсе. В Compose вы можете описывать визуальные элементы с помощью contentDescriptionатрибута. Для чисто декоративных визуальных элементов можно установить contentDescriptionзначение null. Подробнее о том, как применять описания контента, читайте в [документации](https://developer.android.com/jetpack/compose/accessibility#describe-visual) .

Сенсорный целевой размер

Любой экранный элемент, с которым кто-то может взаимодействовать, должен быть достаточно большим для надежного взаимодействия. Минимальный размер сенсорного объекта, на который можно нажать, составляет 48 dp в высоту и 48 dp в ширину. Существует ряд компонентов Material Design, для которых Compose автоматически назначает правильный минимальный целевой размер. Имейте в виду, что минимальный целевой размер сенсорного экрана относится к кликабельным компонентам размером менее 48 dp. Компоненты размером более 48 dp будут иметь цель касания, размер которой не меньше размера компонента. Следуйте этим ресурсам для получения дополнительной информации о размере сенсорной цели:

1. О минимальном целевом размере читайте в [документации Accessibility in Compose](https://developer.android.com/jetpack/compose/accessibility#minimum-target-sizes) .
2. Посмотрите раздел о размерах сенсорных целей в [видеоролике «Что нового в специальных возможностях Google»](https://youtu.be/6LsaP6oKxMY?t=166) .
3. Взгляните на код приложения **Woof** . В **MainActivity.kt** составной DogItemButtonобъект использует IconButtonсоставной объект.

@Composable  
private fun DogItemButton(  
   expanded: Boolean,  
   onClick: () -> Unit,  
   modifier: Modifier = Modifier  
) {  
   IconButton(onClick = onClick) {  
       Icon(  
           imageVector = if (expanded) Icons.Filled.ExpandLess else Icons.Filled.ExpandMore,  
           tint = MaterialTheme.colors.secondary,  
           contentDescription = stringResource(R.string.expand\_button\_content\_description),  
       )  
   }  
}

1. Это компонент IconButton Material Design. В [документации кIconButton](https://developer.android.com/reference/kotlin/androidx/compose/material/package-summary#IconButton(kotlin.Function0,androidx.compose.ui.Modifier,kotlin.Boolean,androidx.compose.foundation.interaction.MutableInteractionSource,kotlin.Function0)) составному элементу указано, что минимальный размер сенсорного объекта составляет 48 dp x 48 dp.
2. Код ниже является исходным кодом для IconButton. Обратите внимание, что модификатор устанавливает значение minimumTouchTargetSize().

@Composable  
fun IconButton(  
   onClick: () -> Unit,  
   modifier: Modifier = Modifier,  
   enabled: Boolean = true,  
   interactionSource: MutableInteractionSource = remember { MutableInteractionSource() },  
   content: @Composable () -> Unit  
) {  
   Box(  
       modifier = modifier  
           .minimumTouchTargetSize()  
           .clickable(  
               onClick = onClick,  
               enabled = enabled,  
               role = Role.Button,  
               interactionSource = interactionSource,  
               indication = rememberRipple(bounded = false, radius = RippleRadius)  
           ),  
       contentAlignment = Alignment.Center  
   ) {  
       val contentAlpha = if (enabled) LocalContentAlpha.current else ContentAlpha.disabled  
       CompositionLocalProvider(LocalContentAlpha provides contentAlpha, content = content)  
   }  
}

1. **Примечание** . Код, который вы видите выше, нельзя найти непосредственно в коде приложения **Woof** . Это исходный код IconButton из библиотеки Compose Material. В качестве дополнительного шага, если вы хотите найти этот код для себя, вы можете щелкнуть правой кнопкой мыши вызов IconButtonв MainActivityи выбрать **«Перейти» > «Декларация» или «Использование»** .
2. **Цветовой контраст**
3. Цвета, которые вы выбираете для интерфейса вашего приложения, влияют на то, насколько легко пользователи смогут его прочитать и понять. Достаточный цветовой контраст облегчает чтение и понимание текста и изображений.
4. Помимо преимуществ для пользователей с различными нарушениями зрения, достаточный цветовой контраст помогает всем пользователям при просмотре интерфейса на устройствах в экстремальных условиях освещения, например, при воздействии прямых солнечных лучей или на дисплее с низкой яркостью.
5. Подробнее о том, как оптимизировать цветовой контраст, можно прочитать в [документации по специальным возможностям Android](https://support.google.com/accessibility/android/answer/7158390) . По этой ссылке вы найдете информацию о коэффициентах контрастности, которая поможет вам принять решение о том, какие цвета использовать. Кроме того, вы можете использовать [этот инструмент](https://webaim.org/resources/contrastchecker/) для проверки цветов фона и переднего плана на предмет достаточного коэффициента цветовой контрастности. Рекомендуемое соотношение мелкого текста составляет 4,5:1, а рекомендуемое соотношение крупного текста — 3,0:1.
6. Для приложения **Woof** наш дизайнер подобрал для нас цвета и позаботился о том, чтобы они имели достаточный цветовой контраст. Создавая собственное приложение, не забудьте проверить цветовой контраст. Инструмент [«Цвет»](https://material.io/resources/color/#!/?view.left=0&view.right=0) для Material Design имеет вкладку «Специальные возможности», на которой вы можете видеть соответствующие цвета текста поверх основных и дополнительных цветов.

[6. Заключение](https://developer.android.com/codelabs/basic-android-kotlin-compose-test-accessibility?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-3%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-test-accessibility#5)

Важно помнить о доступности при создании новых приложений и добавлении новых функций в существующие приложения. Интегрируя функции и сервисы специальных возможностей, вы можете повысить удобство использования вашего приложения, особенно для пользователей с ограниченными возможностями.

Узнать больше

* [Доступность на Android](https://www.youtube.com/playlist?list=PLWz5rJ2EKKc8OENfLdh3zM5T6IRdlVYKj)
* [Доступный дизайн](https://m3.material.io/foundations/accessible-design/overview)
* [Доступность в Jetpack Compose](https://developer.android.com/codelabs/jetpack-compose-accessibility#0)
* [Сделайте свое приложение для Android более доступным](https://developer.android.com/courses/pathways/make-your-android-app-accessible)