



Адаптация приложения для незрячих

Вихров Виктор

разработчик Яндекс Go

1. Контент и элементы управления должны корректно озвучиваться

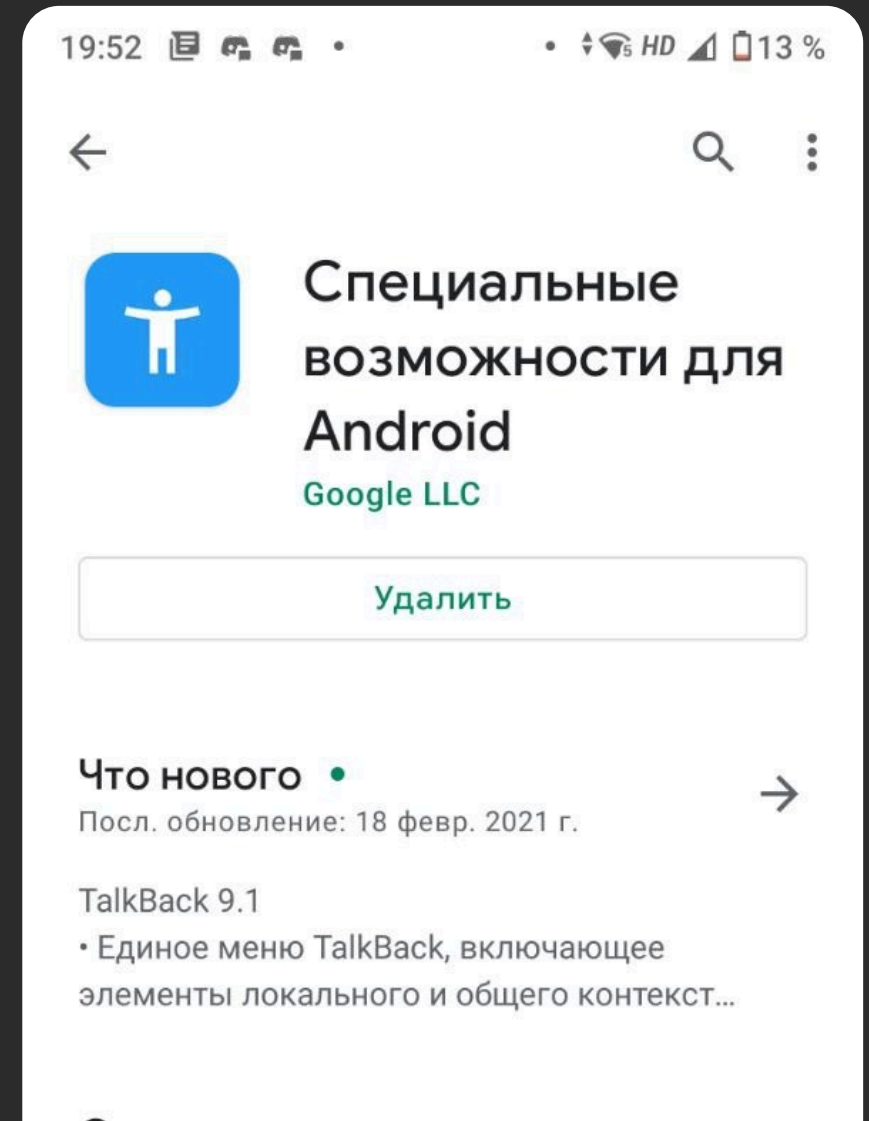
2. Навигация должна корректно работать. Обеспечить доступ к экранам, элементам управления и т.д.

Специальные возможности

TalkBack - приложения для навигации и озвучивания элементов экрана

Select to Speak (Озвучивание при нажатии) озвучивает выделенный текст

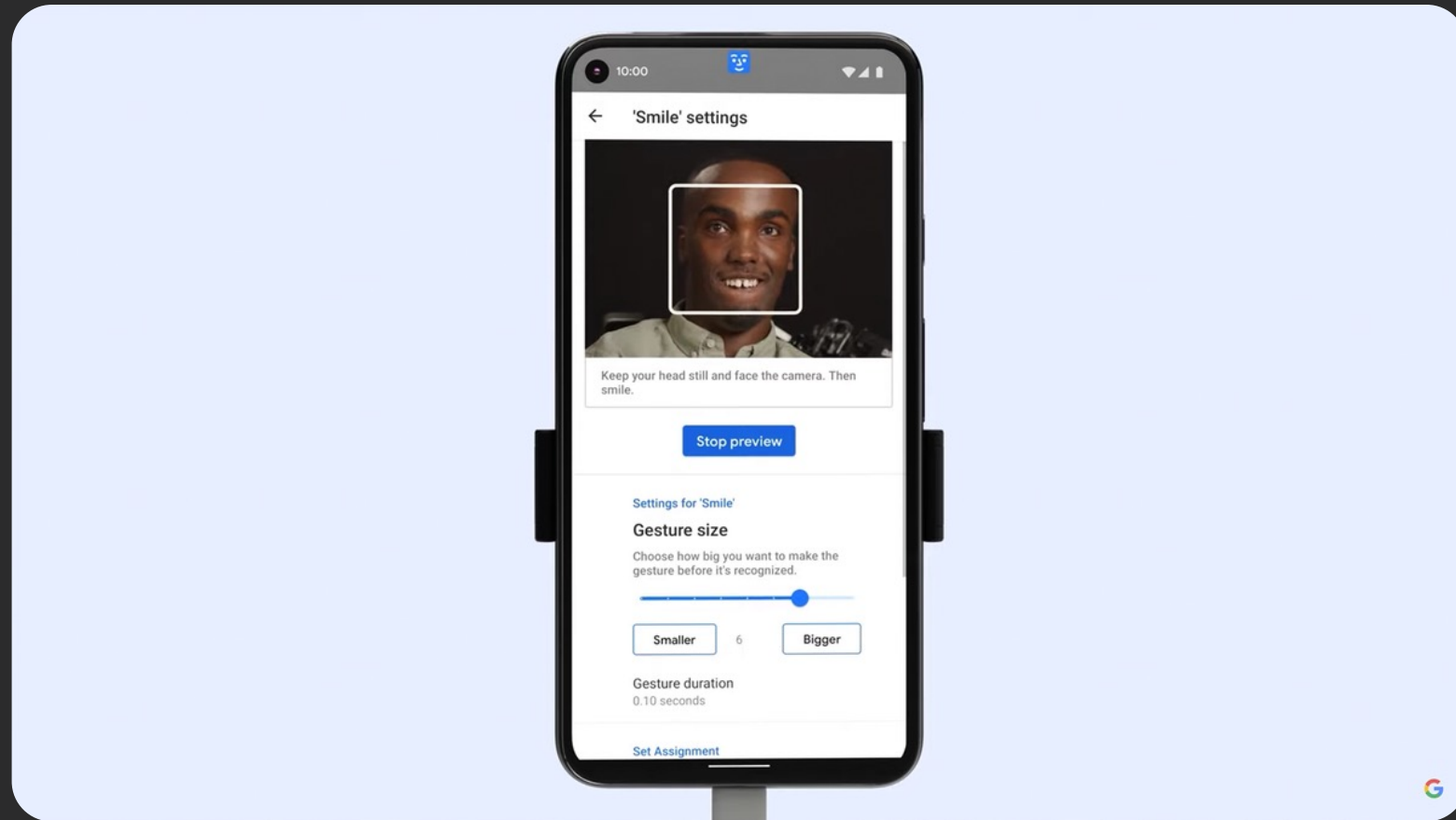
SwitchAccess приложение для управления устройством специальными переключателями



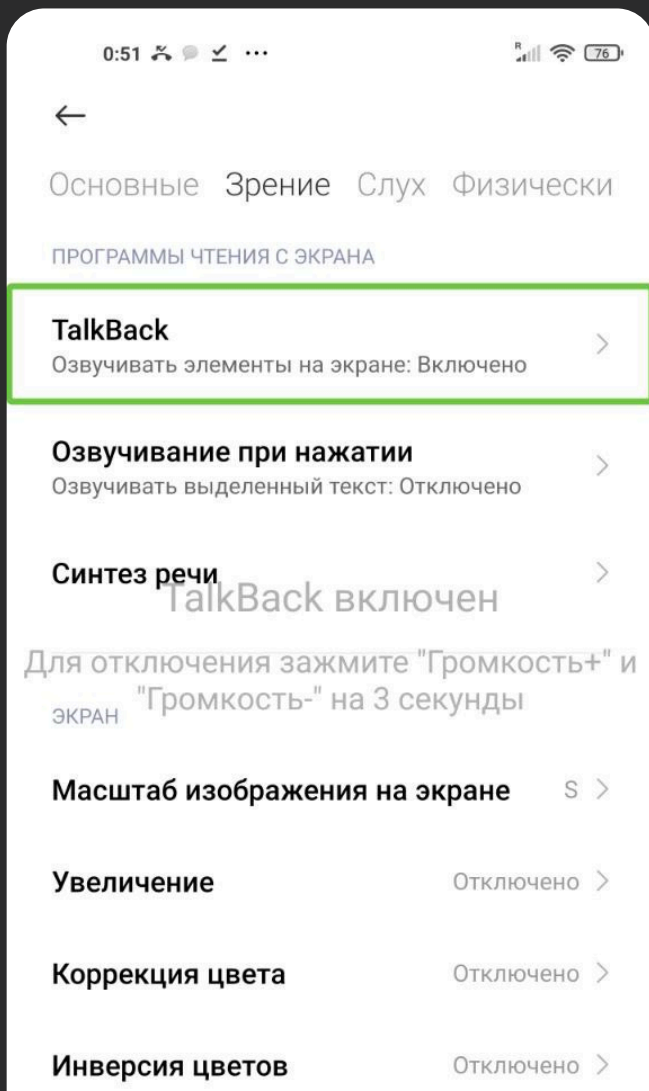
Switch access



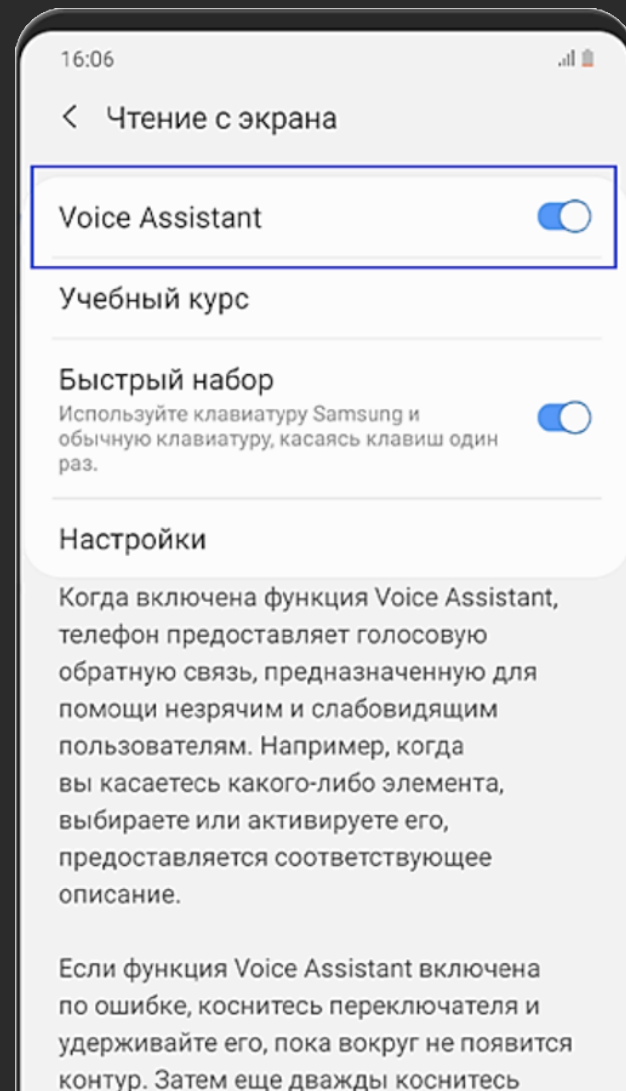
Camera Switches (Android 12)



TalkBack

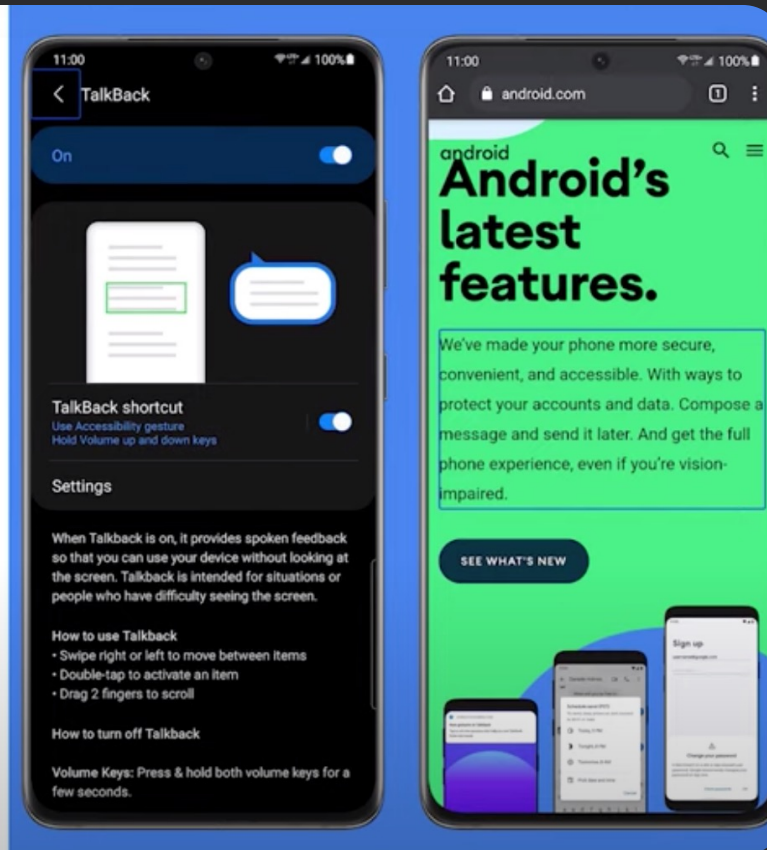


Voice Assistant

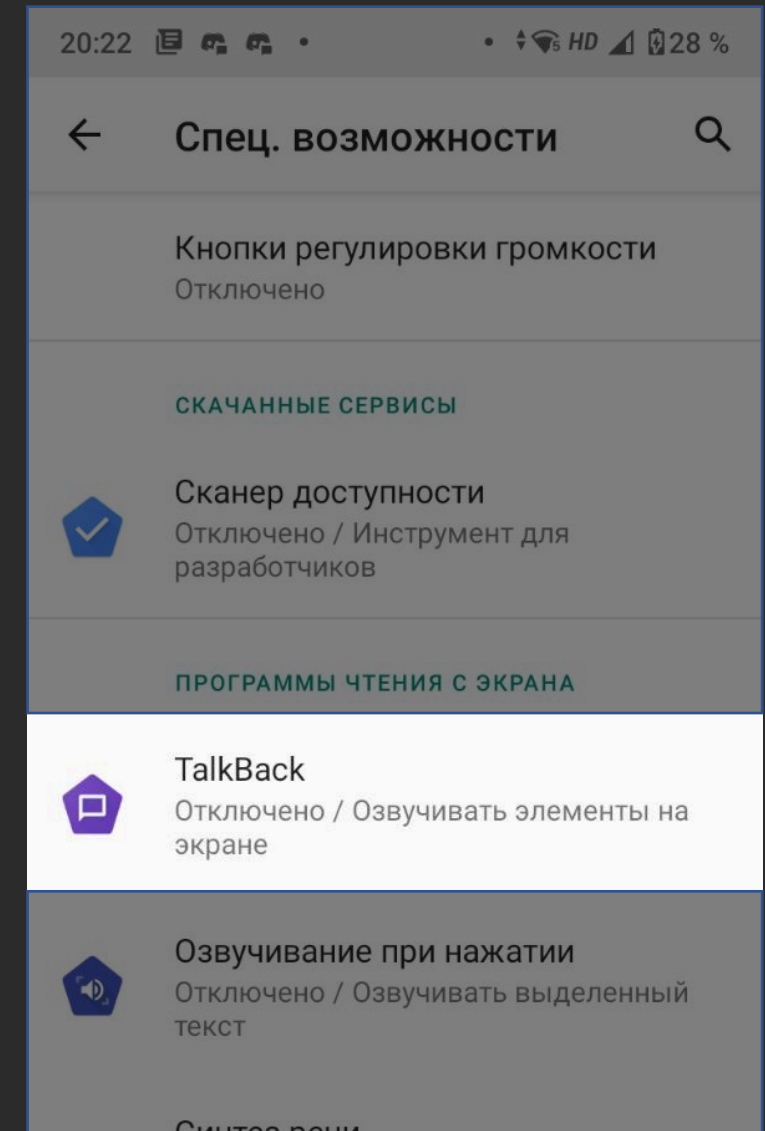
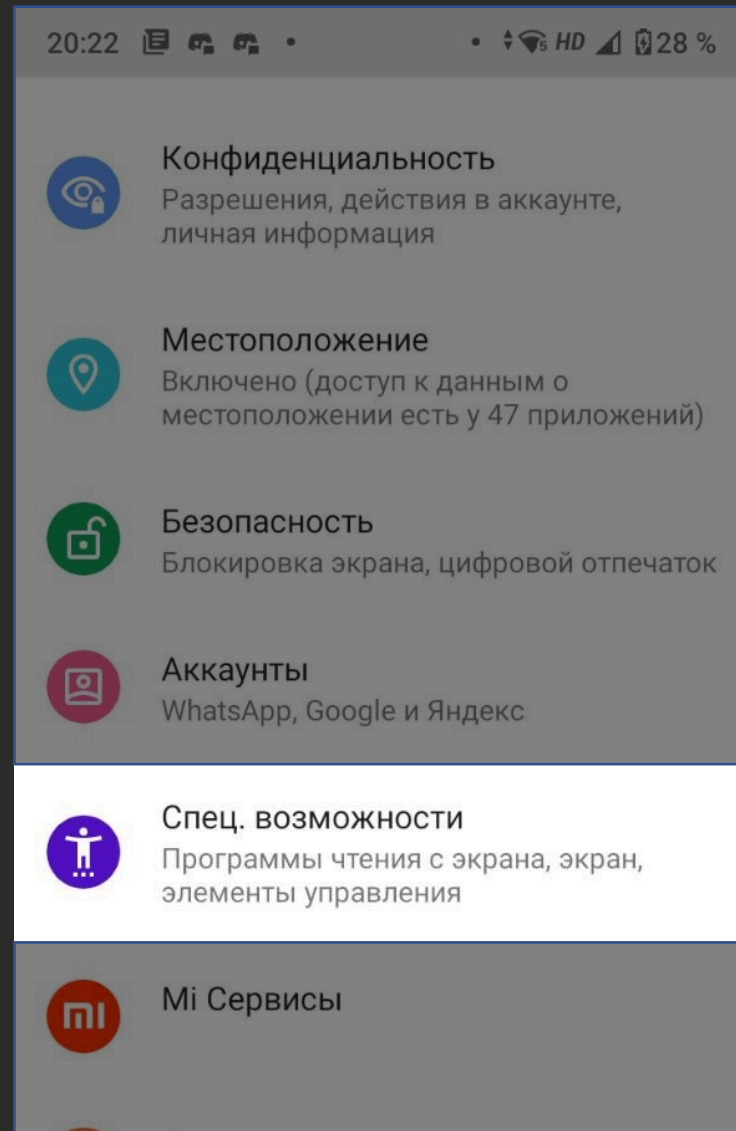


Совместная работа над Talkback

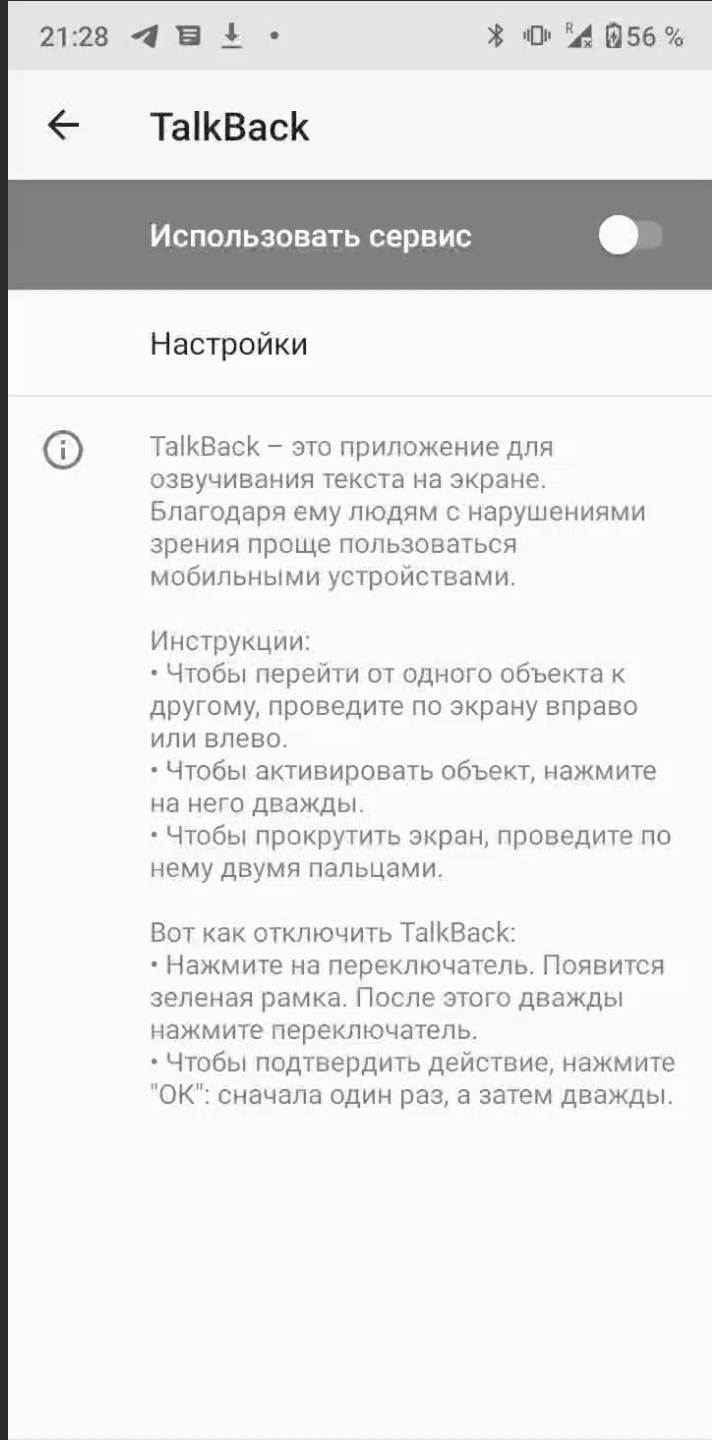
TalkBack now on
SAMSUNG
Galaxy



Как включить?



Демонстрация

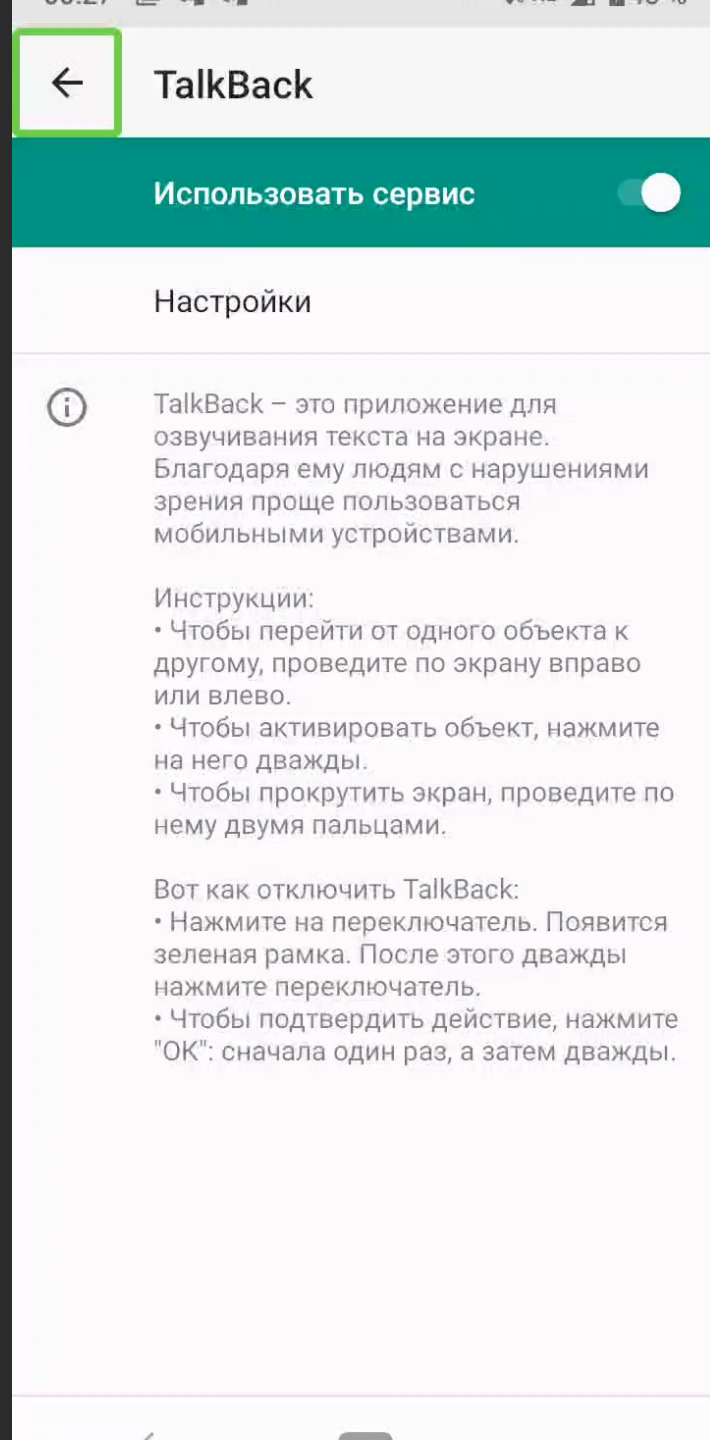


Управление

Linear navigation

1. Свайп вправо, свайп влево
2. Фокус двигается слева направо, сверху вниз

Важно! При тестирование использовать линейную навигацию

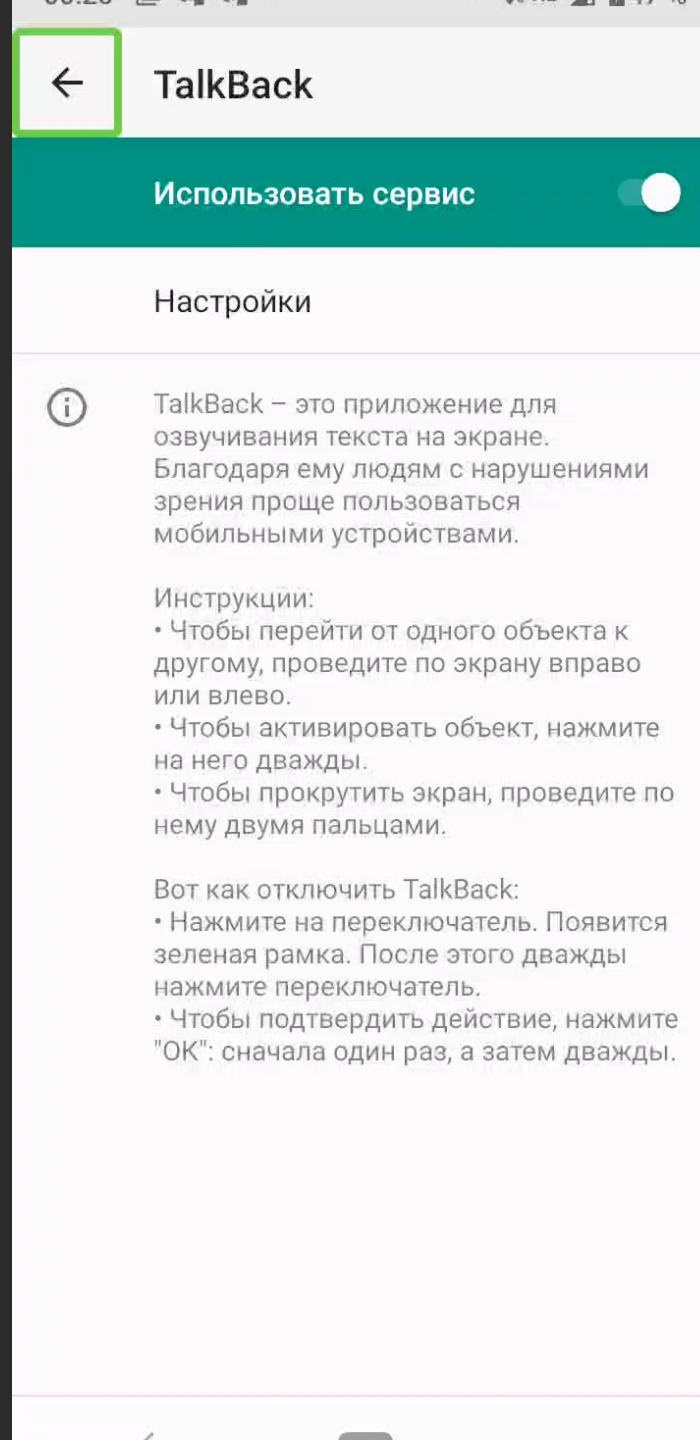


Управление

Explore by touch

1. Когда что-то пошло не так
2. Когда пользователь хорошо знает приложение

Важно! При тестирование использовать линейную навигацию



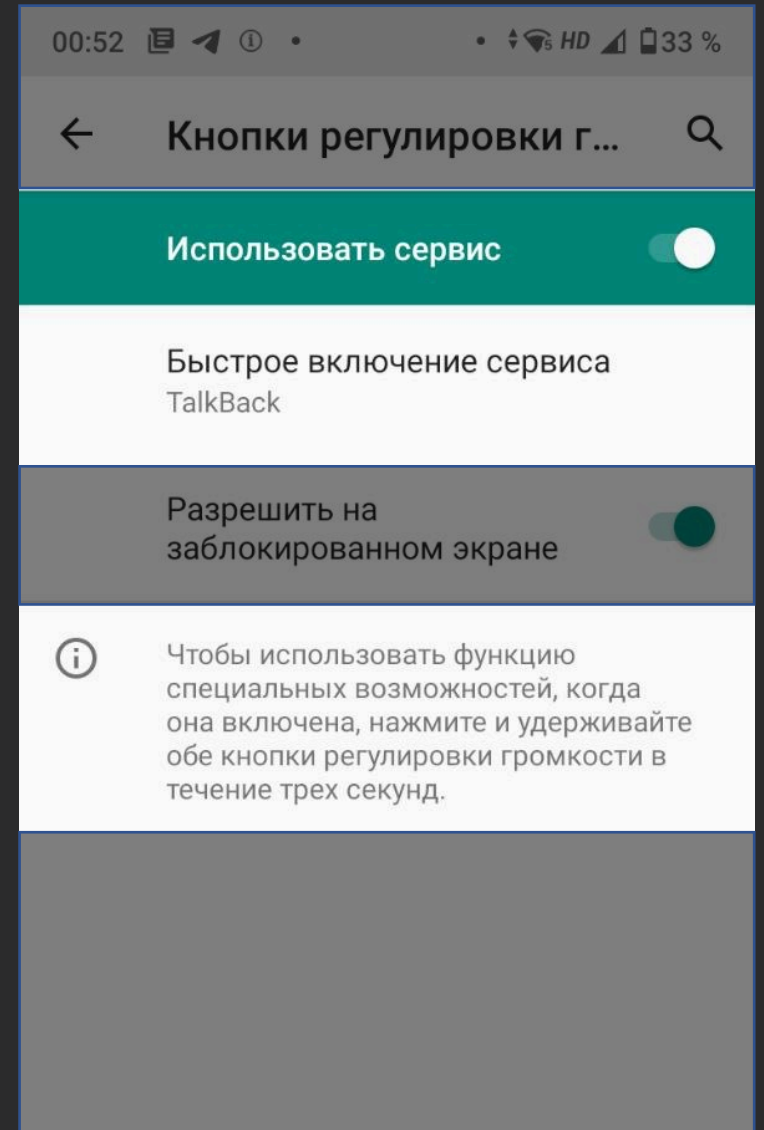
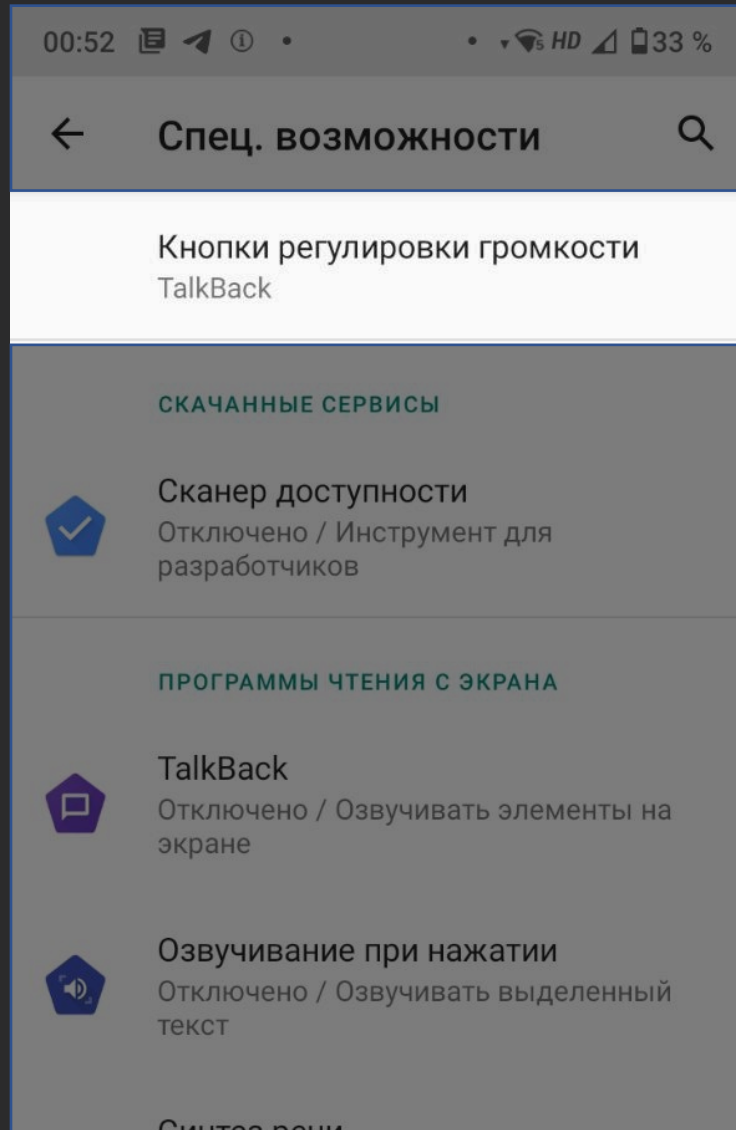
Жесты

Действие	Жест
Выбрать элемент	тап
Следующий элемент	свайп вправо
Предыдущий элемент	свайп влево
Клик по элементу	двойной тап
Лонгтап	двойной тап + удерживать
Свайп	Свайп двумя пальцами
Вернуться назад	Проведите вниз, затем влево

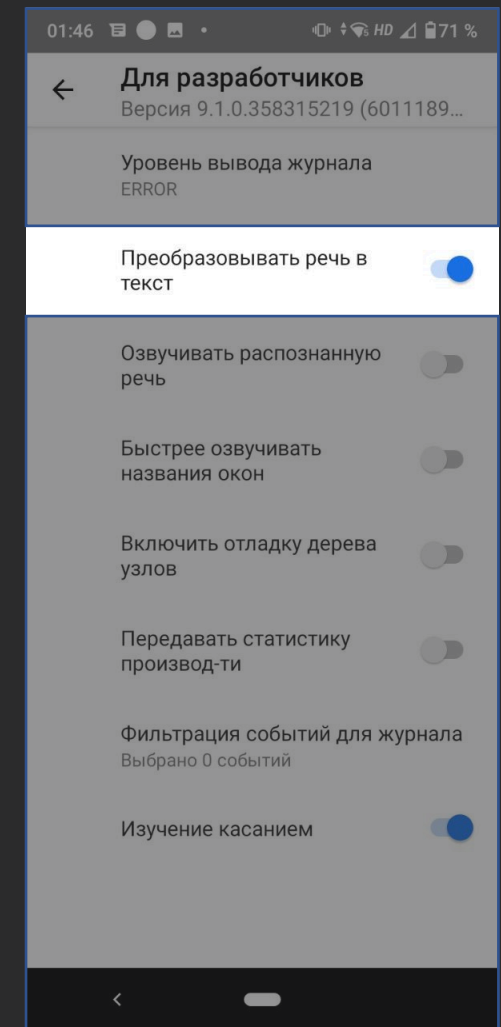
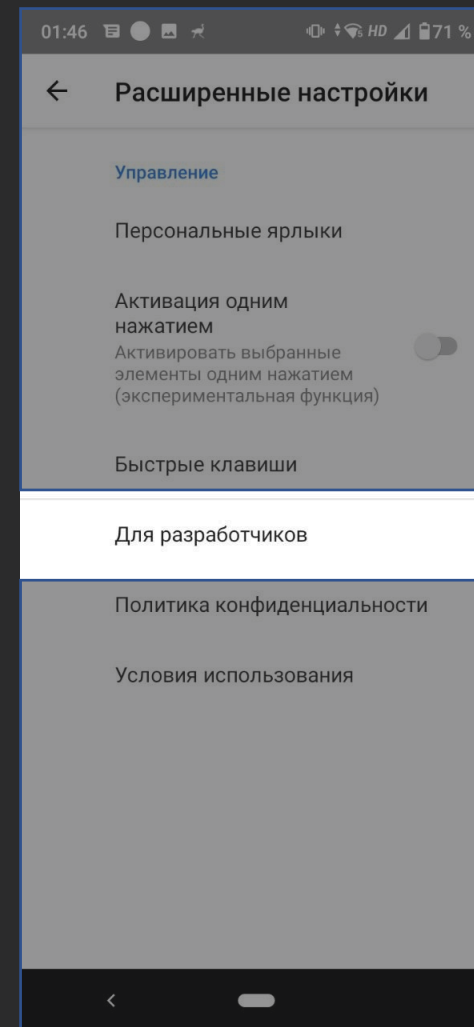
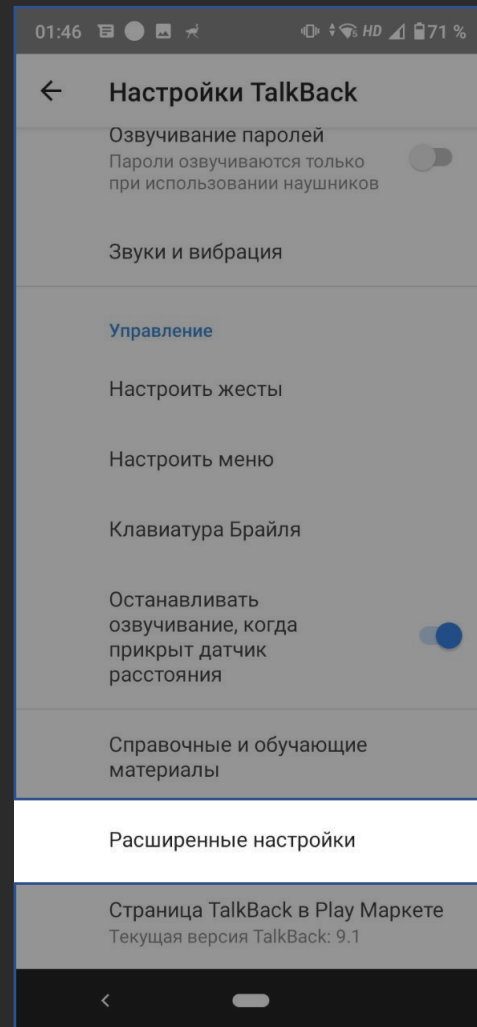
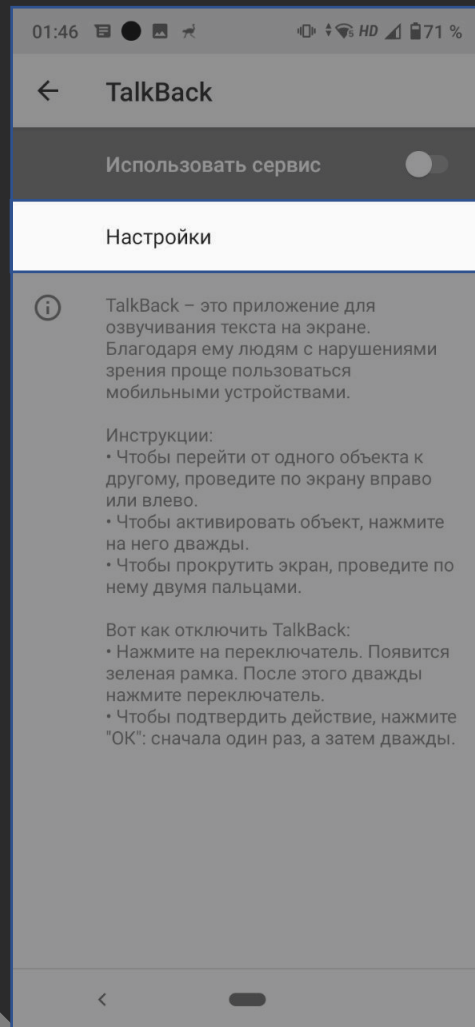


[Полный список поддерживаемых жестов по версиям](#)

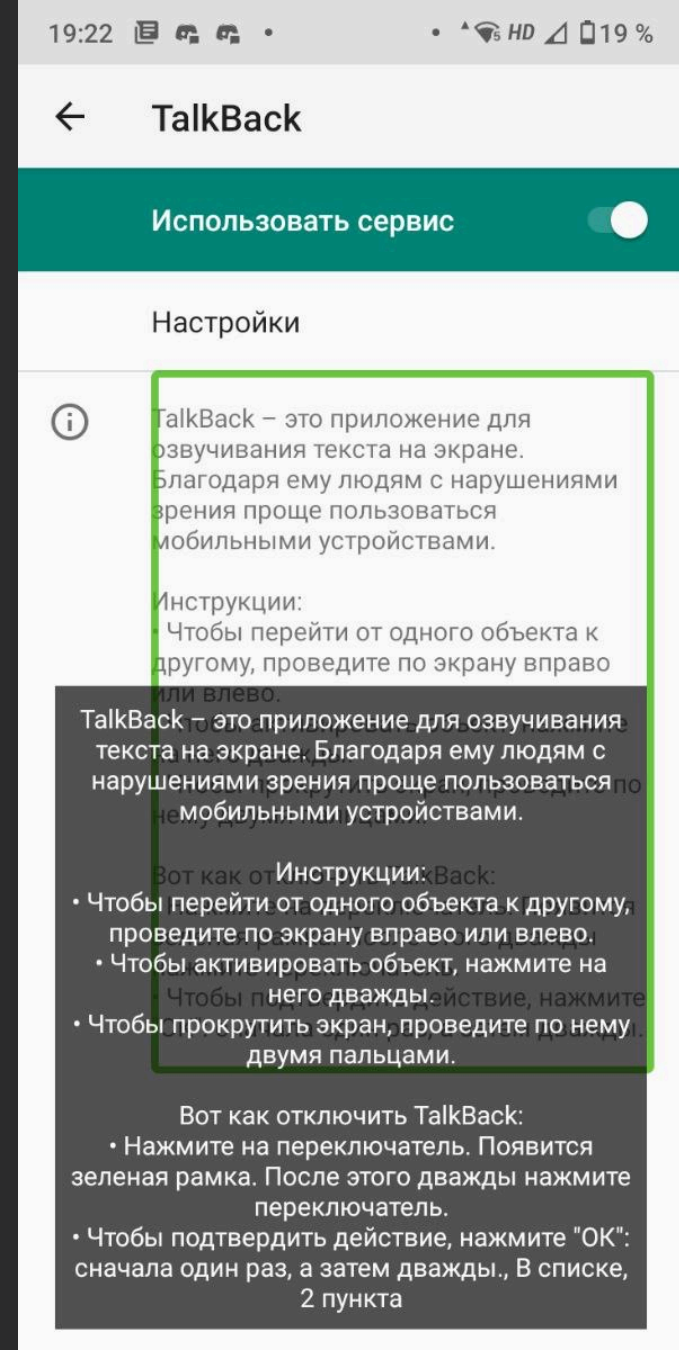
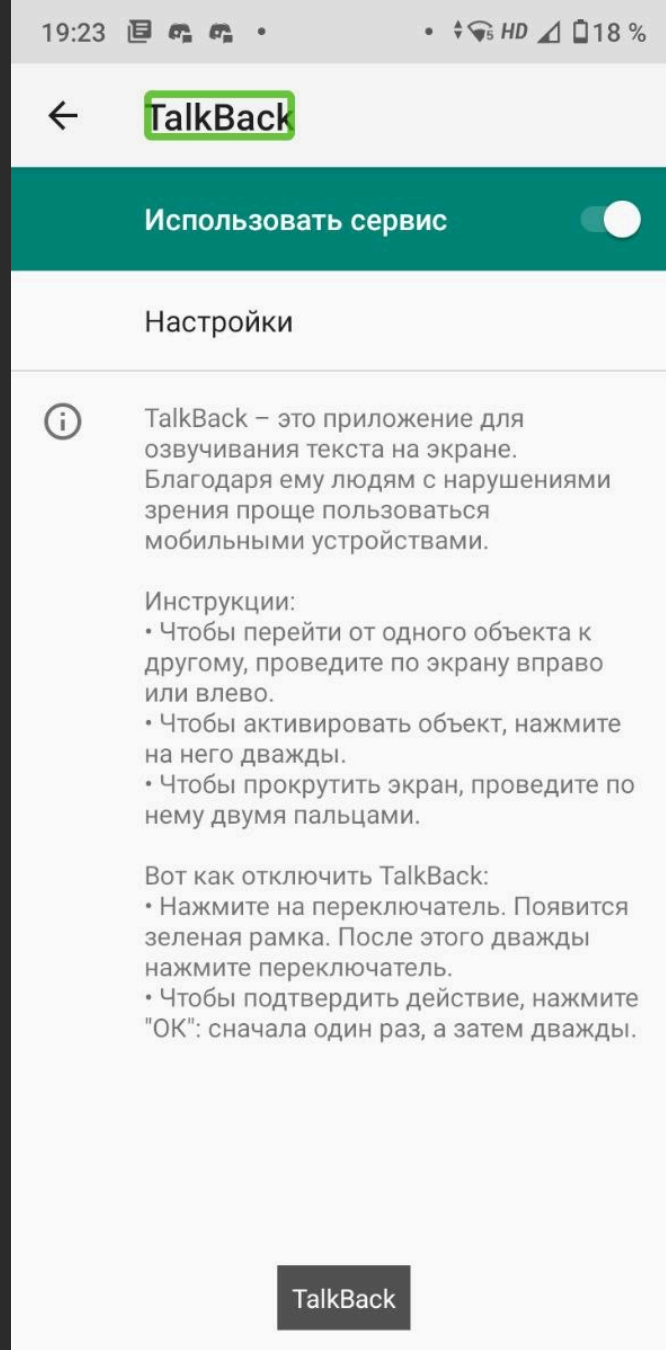
Быстрое включение



Включение субтитров



Субтитры



Ввод данных


Голосовой ввод

02:14

5 4G 61 %



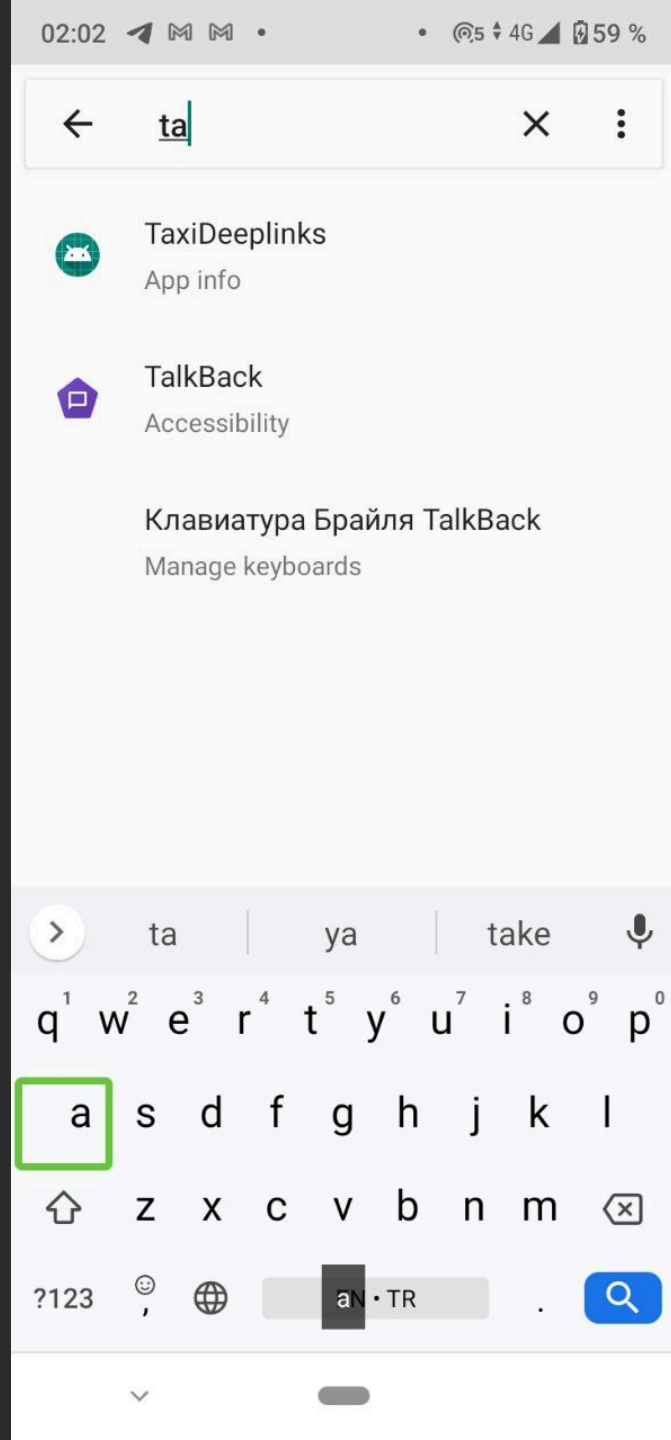
Выберите кнопку
голосового поиска или
отдайте голосовую
команду

 Найти песню

Кнопка голосового поиска, нажмите и
произнесите запрос

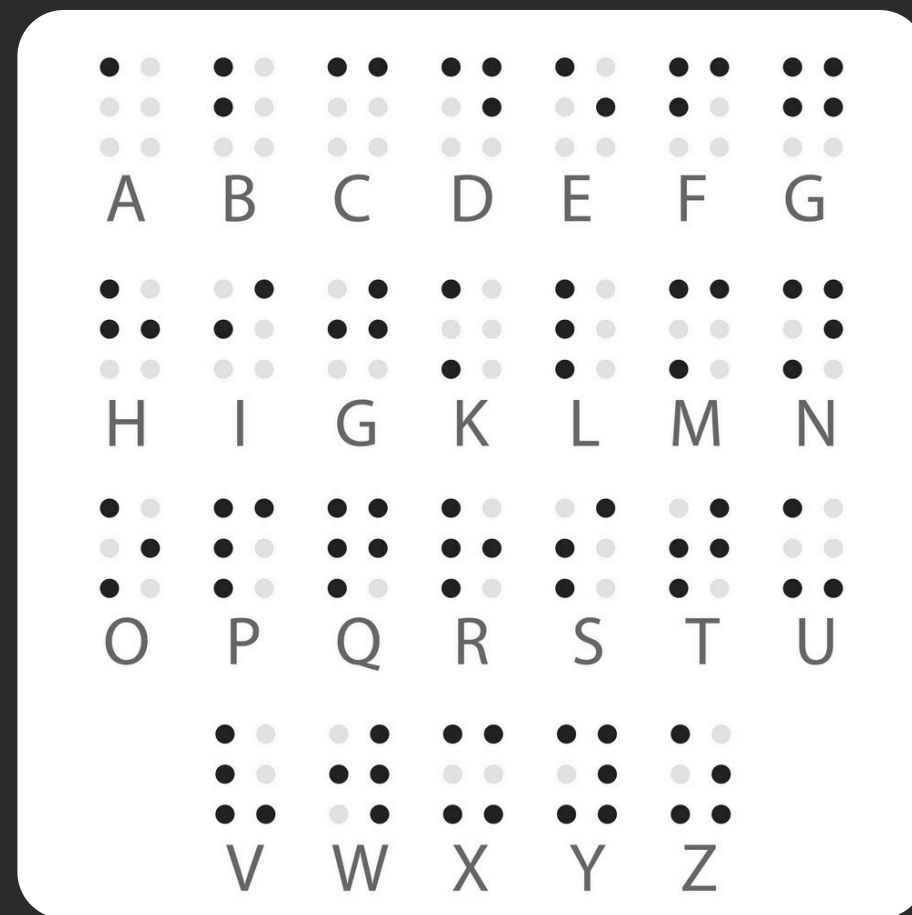
Ввод данных

Клавиатура



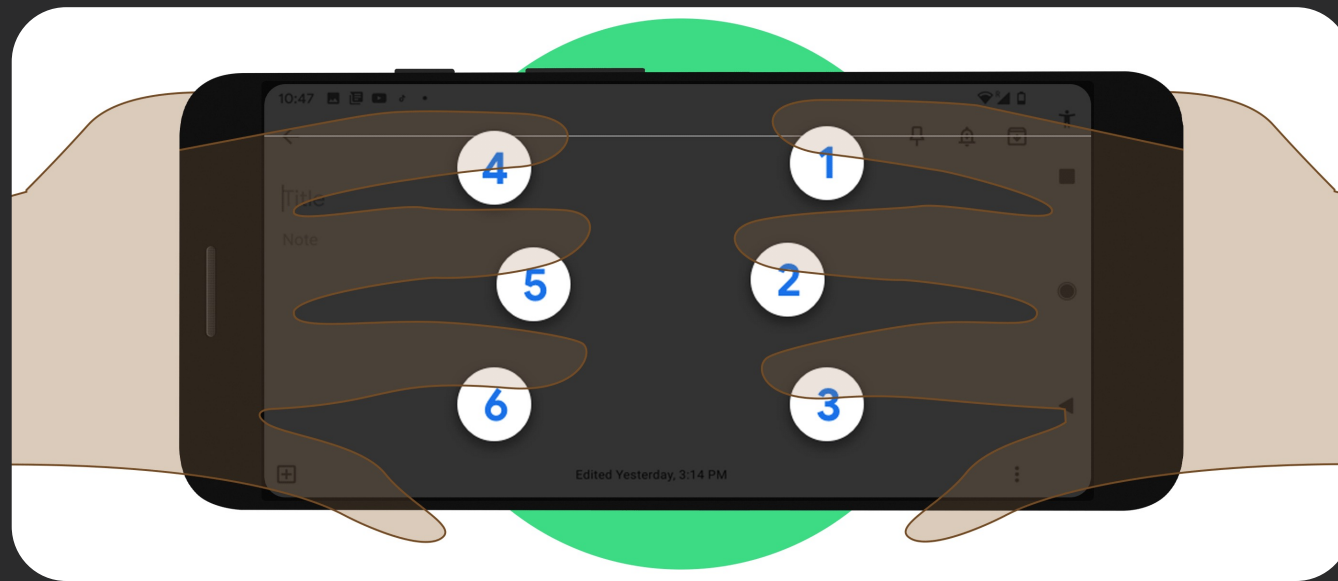
Ввод данных

Шрифт Брайля



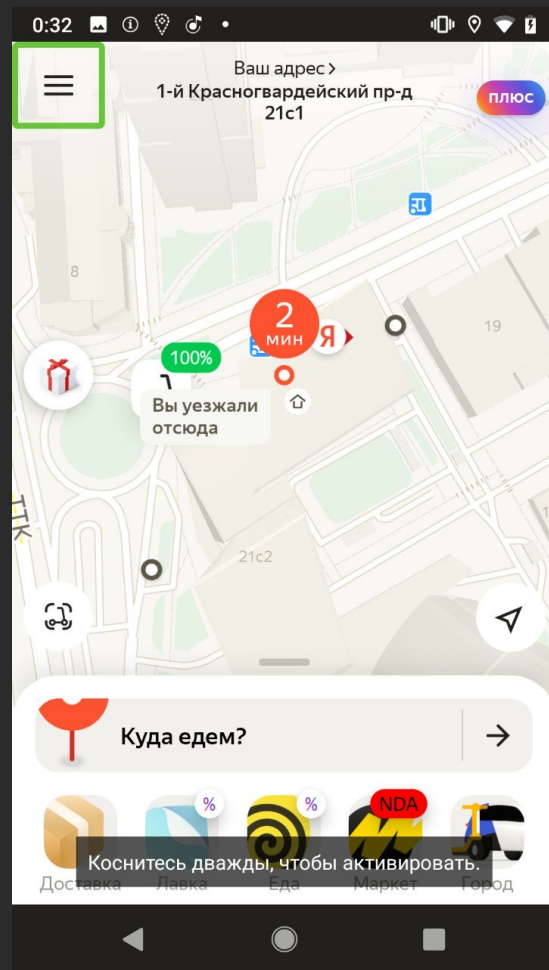
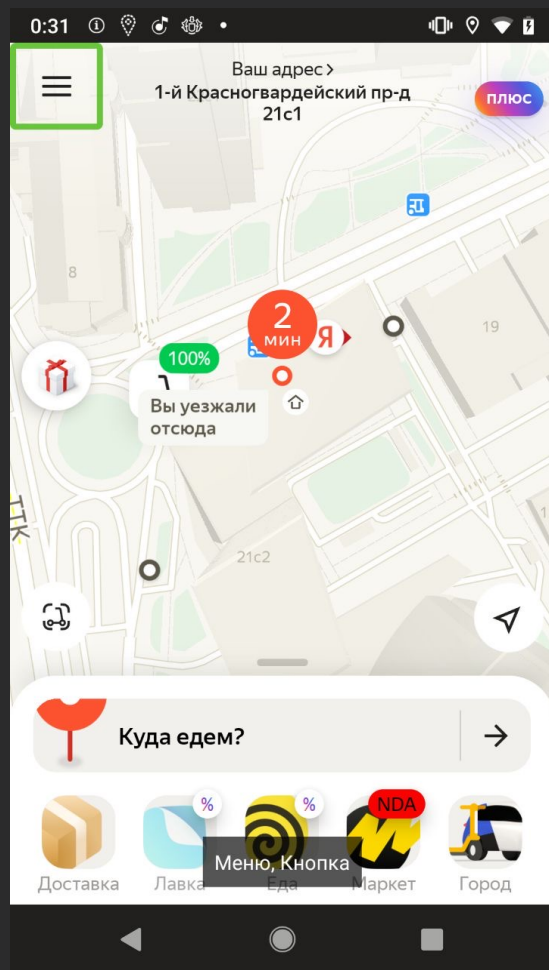
Ввод данных

Клавиатура Брайля

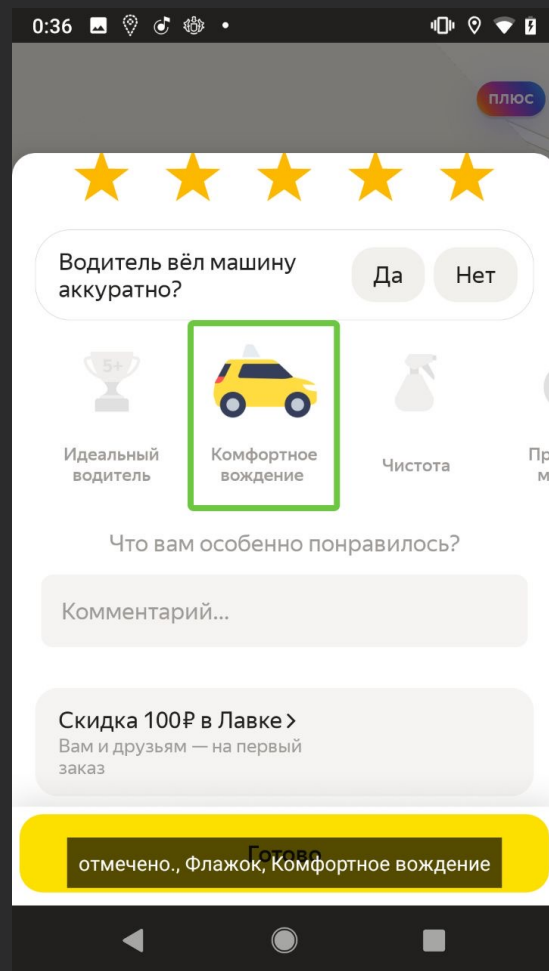
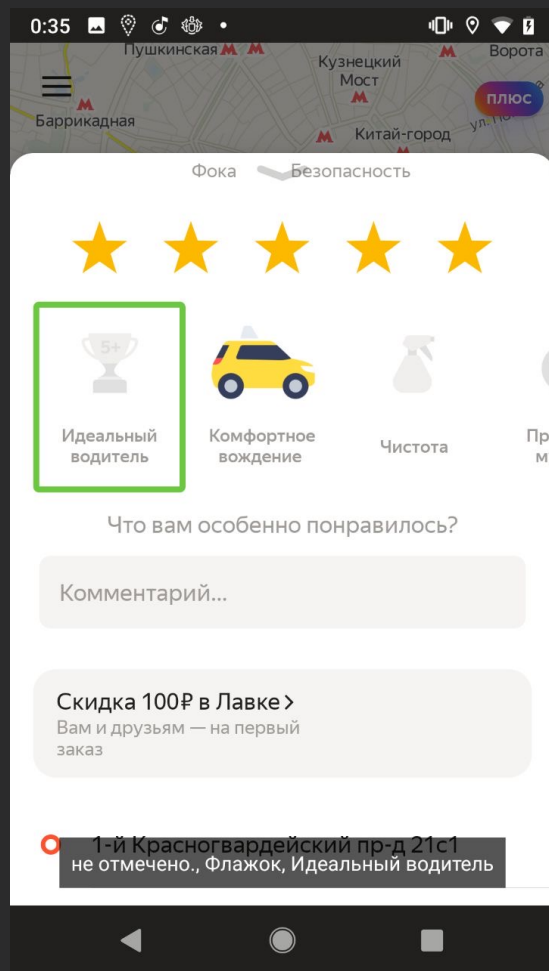



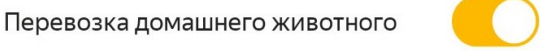

Общие подходы к доступности

Озвучивание элементов

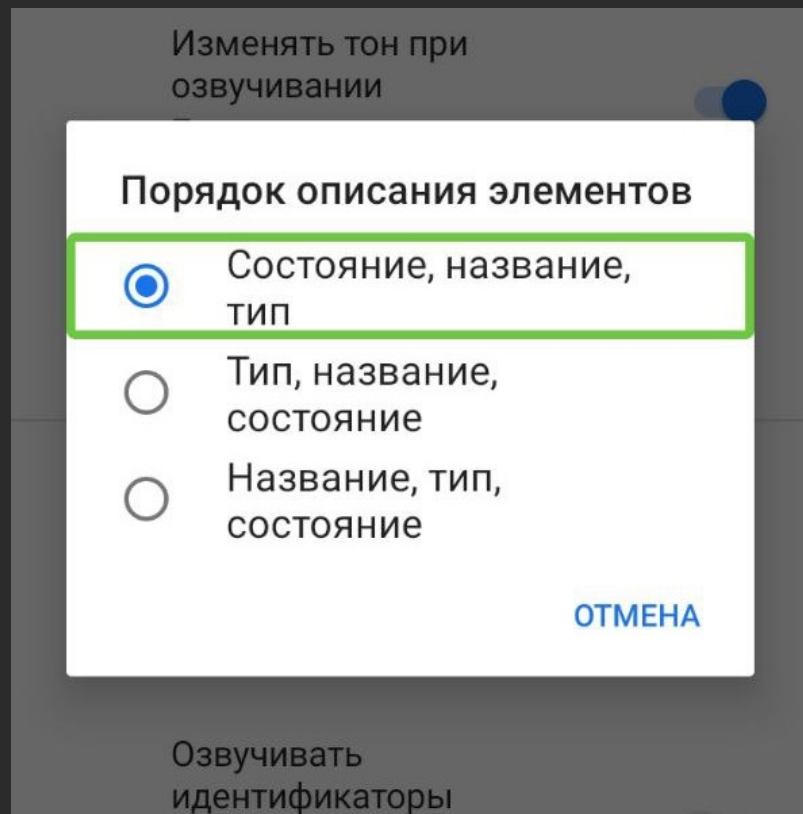


Озвучивание элементов

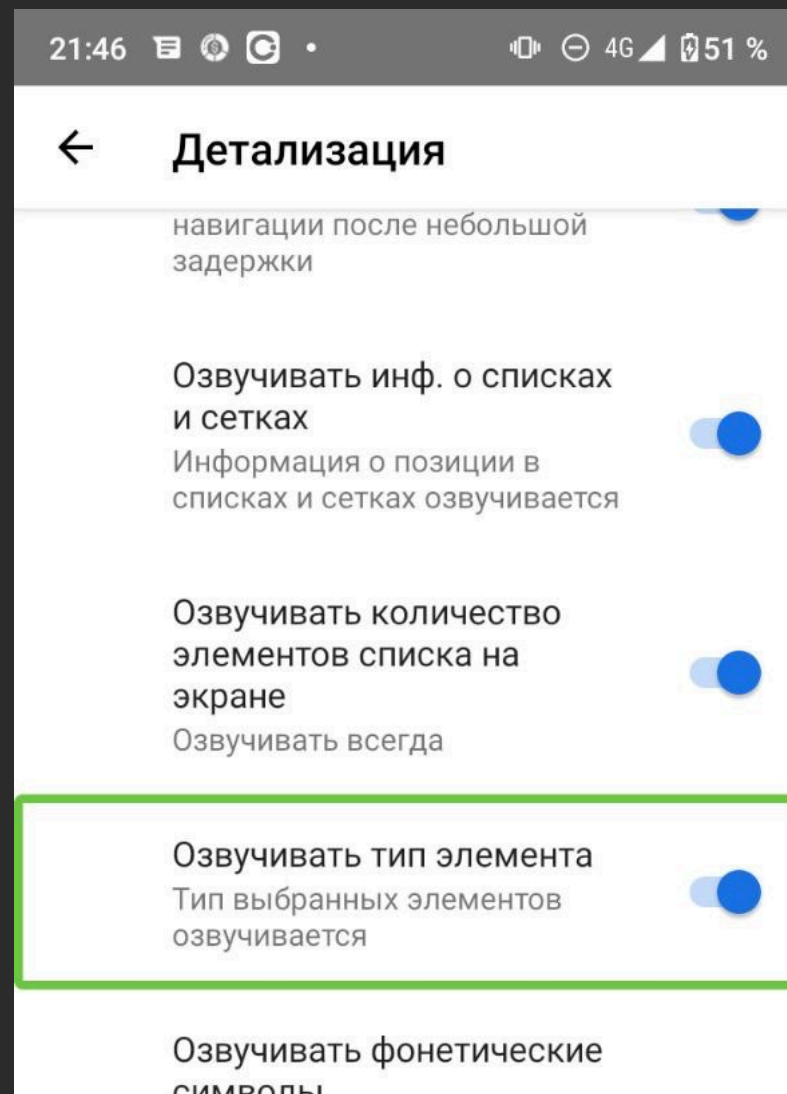
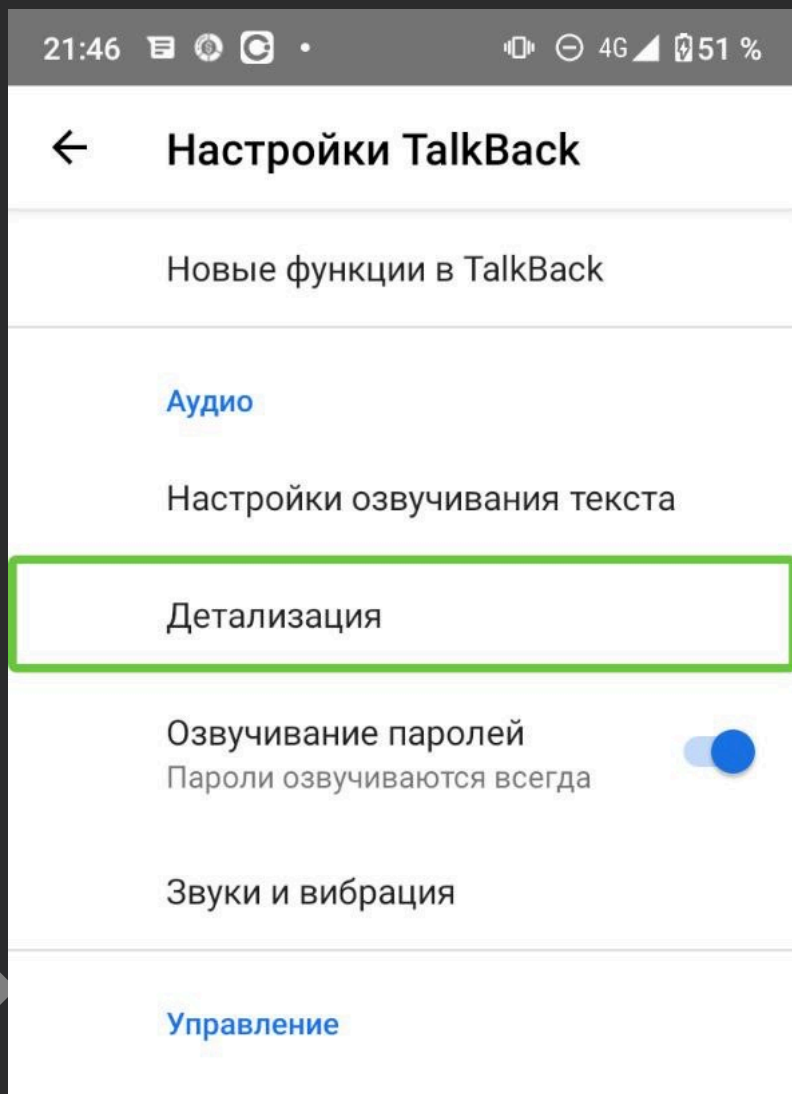


Элемент			 Комфортное вождение
Описание	Меню	Перевозка домашнего животного	Комфортное вождение
Тип	Кнопка	Переключатель	Флажок
Состояние	Прим. Может быть отключена (isEnabled == false)	Включен	Отмечен
Подсказка	Коснитесь дважды чтобы активировать	Коснитесь дважды, чтобы включить или выключить	Коснитесь дважды, чтобы включить или выключить

Порядок можно менять



Детализацию можно настраивать



1. Описание

2. Тип

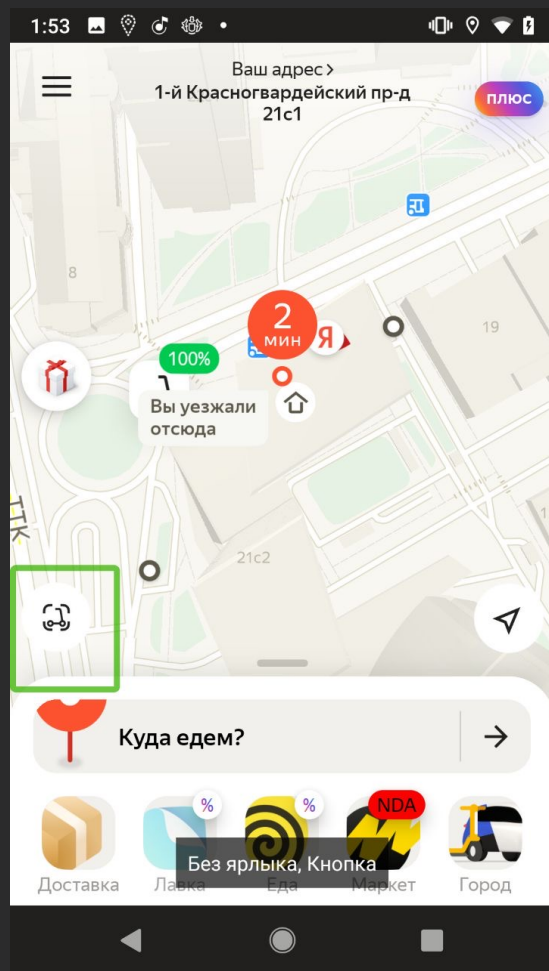
3. Состояние

4. Подсказка

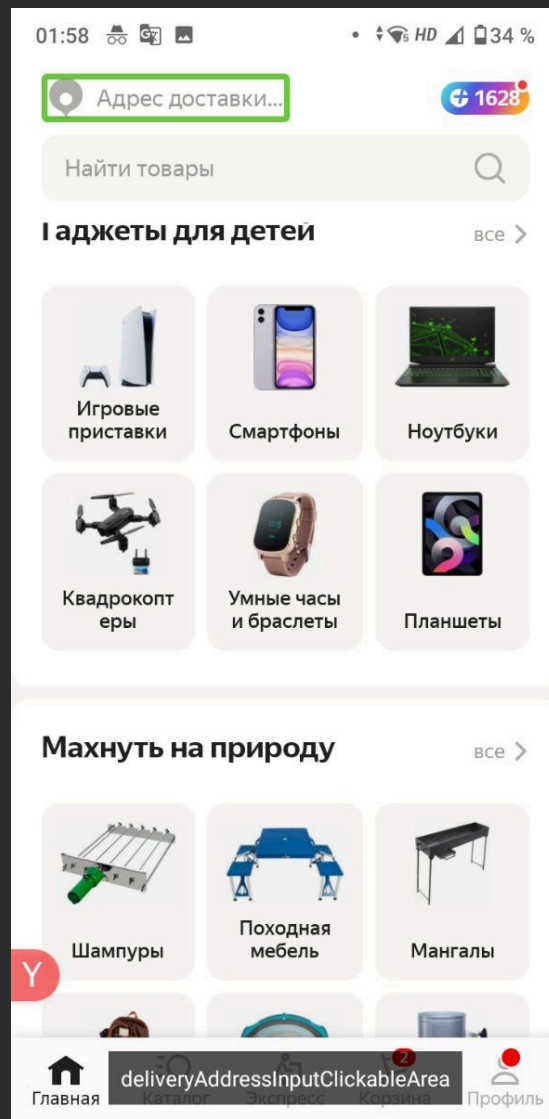
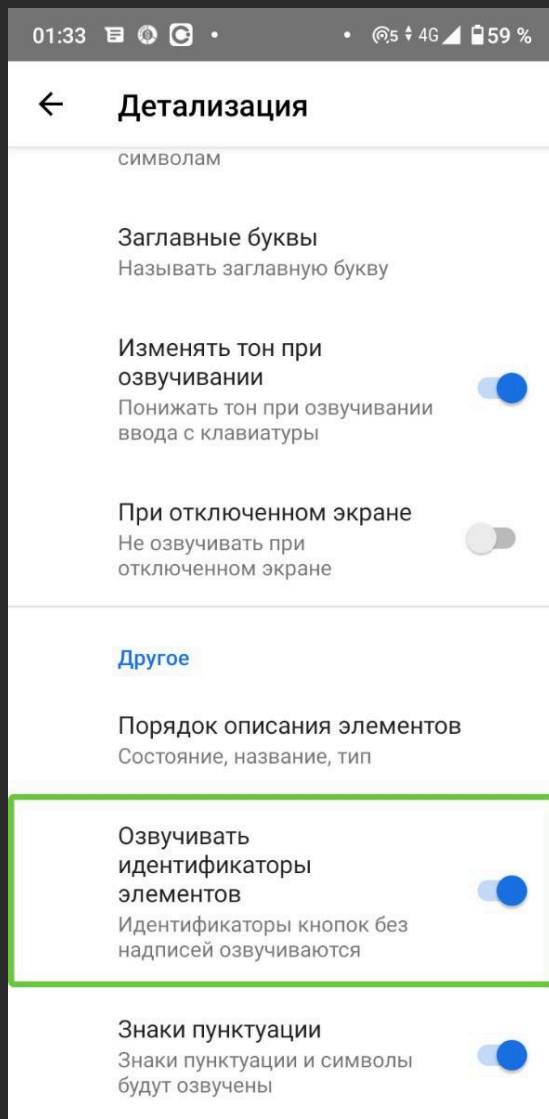
Откуда берется описание?

- `View` элементы умеют предоставлять свой текстовый контент в качестве описания (например, атрибут `text` в `TextView`, `Button`)
- У `View` можно указать `contentDescription`
- Если контент и `contentDescription` отсутствует, элемент не сможет быть озвучен

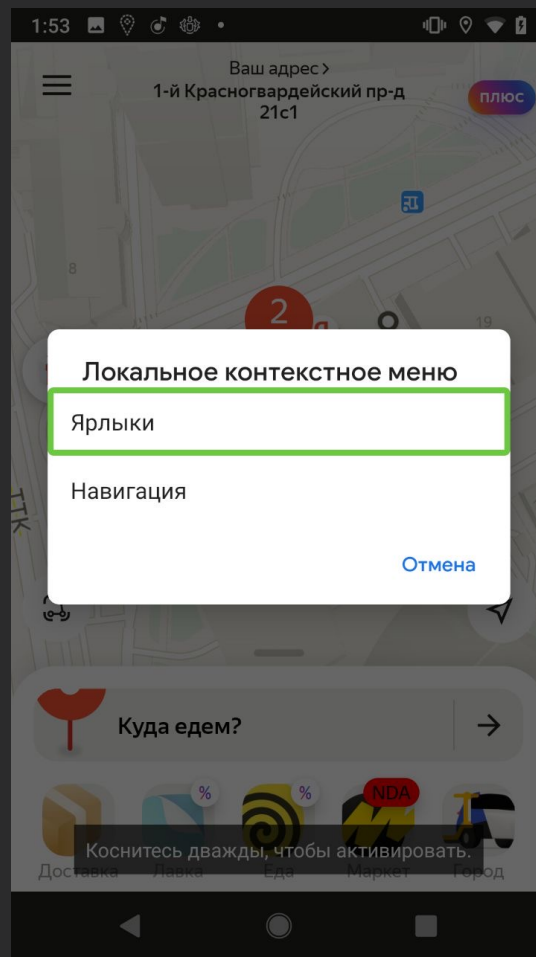
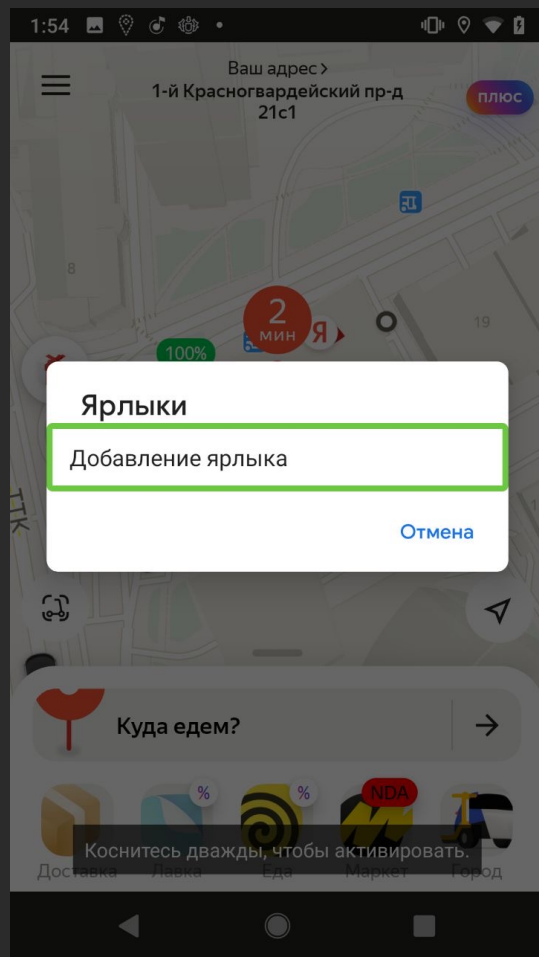
Нет текста и contentDescription



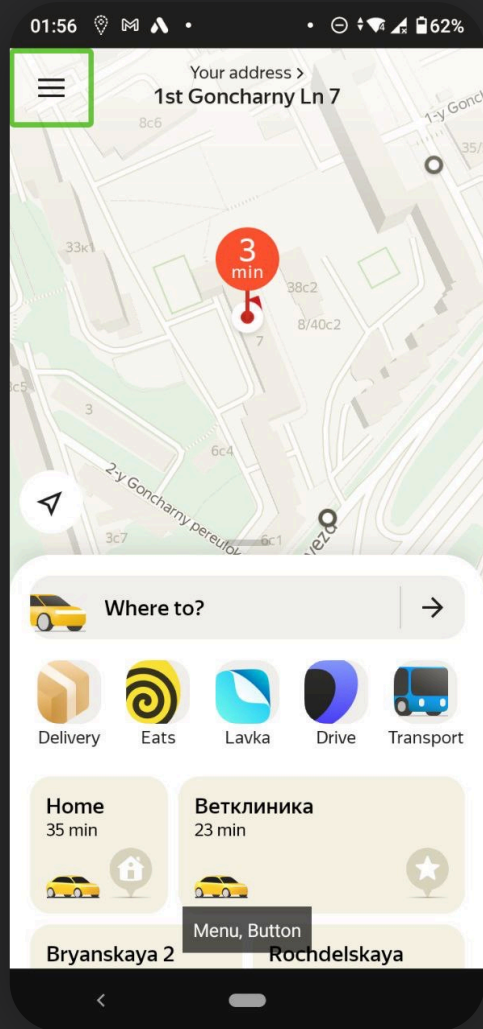
Озвучивание идентификаторов



Пользователи могут называть такие элементы



Атрибут contentDescription



```
menuButton.setContentDescription(  
    getString(R.string.menu_button_content_description));
```

<ImageView

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:contentDescription="@string/my_location_description"
```

```
>
```


Текстовые поля. Compound Drawable.

11458 

125 

Не нужно добавлять тип элемента или состояние в описание

~~Кнопка меню~~

~~Кнопка заказать отключена~~

Меню

Заказать

Важная информация помещается в начало

~~Баллов плюса: 777~~

~~активных заказов: 1~~

777 баллов плюса

1 активный заказ

Не нужно делать слишком подробное описание

~~Нажмите эту кнопку чтобы поделиться новостью~~

~~Нажмите чтобы поделиться новостью~~

Поделиться

Описание должно быть уникальным

~~editAddress.contentDescription = getString(R.string.common_edit)~~

~~editOrder.contentDescription = getString(R.string.common_edit)~~

~~editProfile.contentDescription = getString(R.string.common_edit)~~

editAddress.contentDescription = getString(R.string.edit_address)

editOrder.contentDescription = getString(R.string.edit_order)

editProfile.contentDescription = getString(R.string.edit_profile)

1. Описание

2. Тип

3. Состояние

4. Подсказка

Subclasses should only override this if they are implementing something that should be seen as a completely new class

// класс View

```
public CharSequence getAccessibilityClassName() {  
    return View.class.getName();  
}
```

// класс Button

```
@Override  
public CharSequence getAccessibilityClassName() {  
    return Button.class.getName();  
}
```

className проверяется на соответствие базовым View классам

```
// Inheritance: View->TextView->Button
```

```
if (ClassLoadingCache.checkInstanceOf(className, android.widget.Button.class)) {  
    return ROLE_BUTTON;  
}
```

```
// Inheritance: View->ImageView
```

```
if (ClassLoadingCache.checkInstanceOf(className, android.widget.ImageView.class)) {  
    return node.isClickable() ? ROLE_IMAGE_BUTTON : ROLE_IMAGE;  
}
```


ROLE константы определяют строковый ресурс для озвучивания

```
public static String roleToSummaryString(@Role.RoleName int role, Context context) {  
    switch (role) {  
        case Role.ROLE_BUTTON:  
            return context.getString(R.string.value_button);  
        case Role.ROLE_CHECK_BOX:  
            return context.getString(R.string.value_checkbox);  
        case Role.ROLE_IMAGE:  
            return context.getString(R.string.value_image);  
        case Role.ROLE_IMAGE_BUTTON:  
            return context.getString(R.string.value_button);  
    }  
}
```

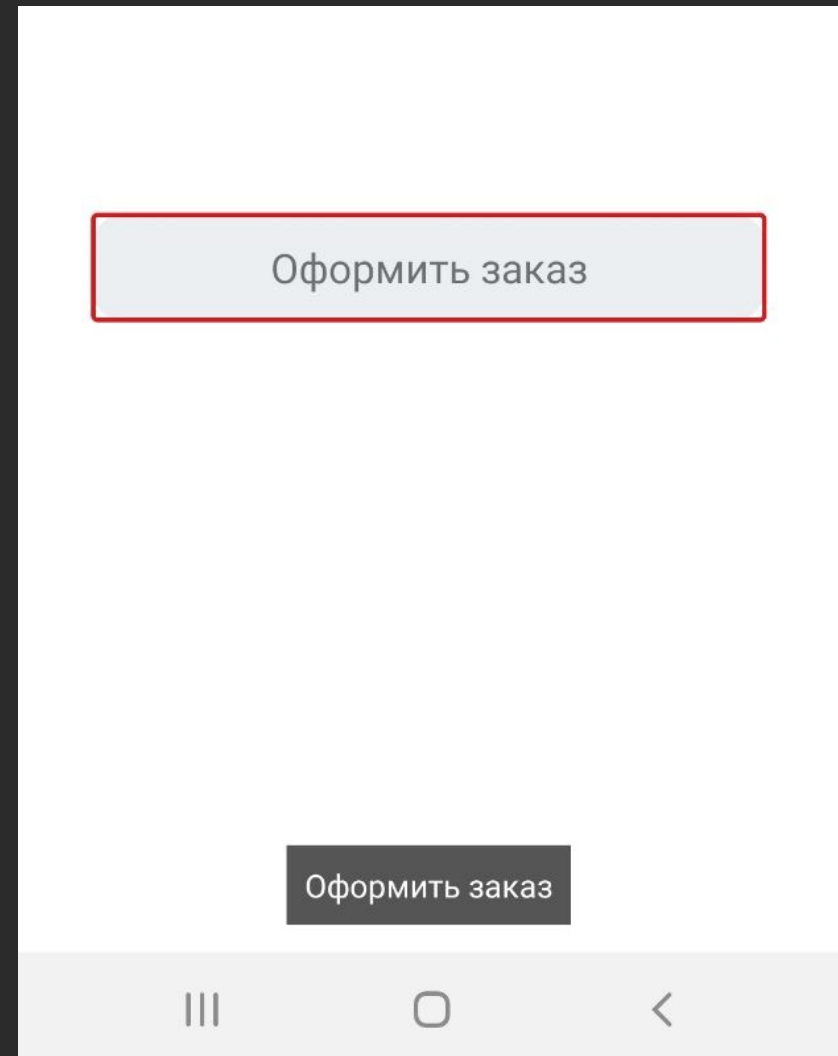
Неизвестный тип не озвучивается

```
return ROLE_NONE;
```

TextView используется вместо Button

<TextView

```
android:layout_width="match_parent"  
android:layout_height="48dp"  
android:layout_gravity="center_vertical"  
android:layout_marginHorizontal="20dp"  
android:background="@drawable/round_rect_shape"  
android:gravity="center"  
android:text="Оформить заказ"  
android:textSize="18sp" />
```



Составной элемент

```
<androidx.constraintlayout.widget.ConstraintLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content">
```

```
<TextView
```

```
    android:id="@+id/title"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/invite_friend_title"/>
```

```
<TextView
```

```
    android:id="@+id/subtitle"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/invite_friend_subtitle"/>
```

```
<ImageView
```

```
    android:id="@+id/image"  
    android:layout_width="60dp"  
    android:layout_height="60dp"  
    android:src="@drawable/discount" />
```

Получите скидку!

Пригласите друга и получите скидку 10%



Задать className

```
class PromoButton(context: Context): ConstraintLayout(context) {  
  
    //...  
  
    override fun getAccessibilityClassName(): CharSequence {  
        return Button::class.java.name  
    }  
}
```

roleDescription

```
class AdBanner(context: Context): View(context) {  
  
    override fun onDraw(canvas: Canvas?) {  
        super.onDraw(canvas)  
        //  
    }  
  
    override fun onInitializeAccessibilityNodeInfo(info: AccessibilityNodeInfo) {  
        super.onInitializeAccessibilityNodeInfo(info)  
        val node = AccessibilityNodeInfoCompat.wrap(info)  
        node.roleDescription = "Banner"  
    }  
}
```

1. Описание

2. Тип

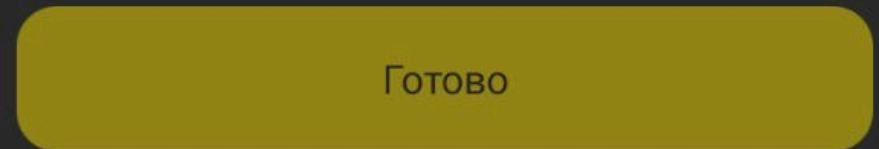
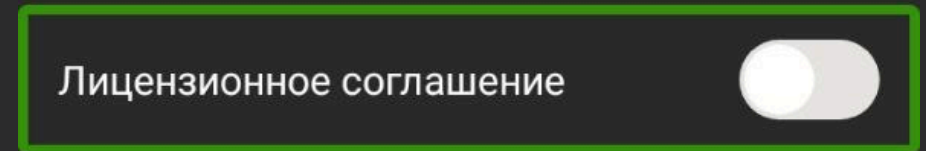
3. Состояние

4. Подсказка

Состояние элемента

Состояние берется из атрибутов View элемента

```
item.setChecked(mChecked);
```



Переключатель, отключено, Лицензионное соглашение

Кнопка отключена

```
doneButton.setAlpha(0.5f)  
doneButton.setOnClickListener(null)
```

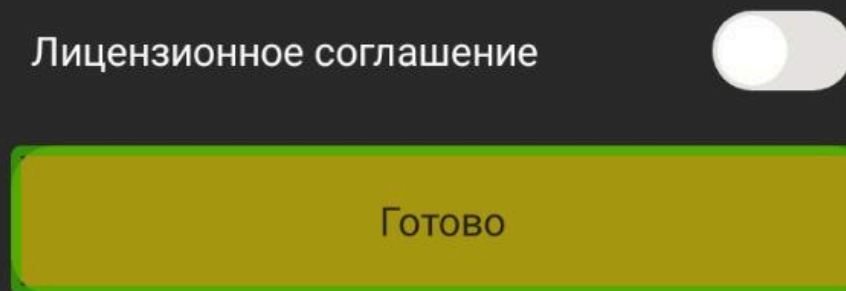
Лицензионное соглашение



Готово

Нужно использовать атрибуты View элемента

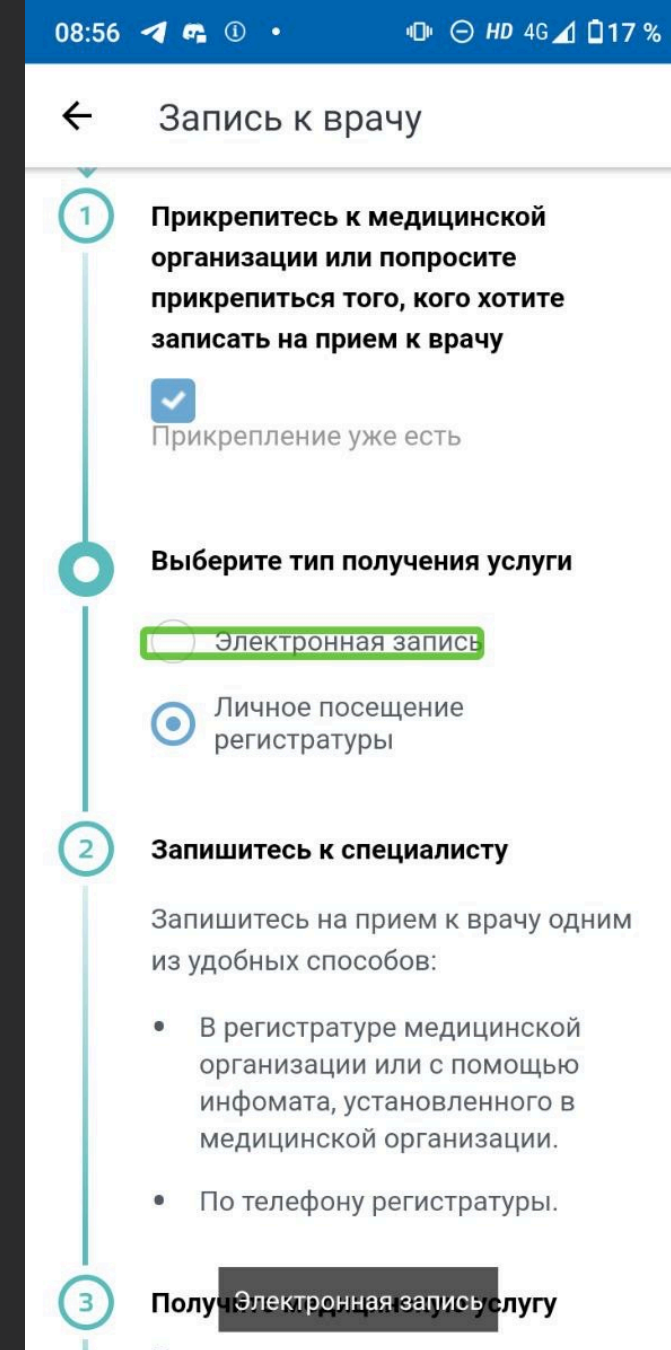
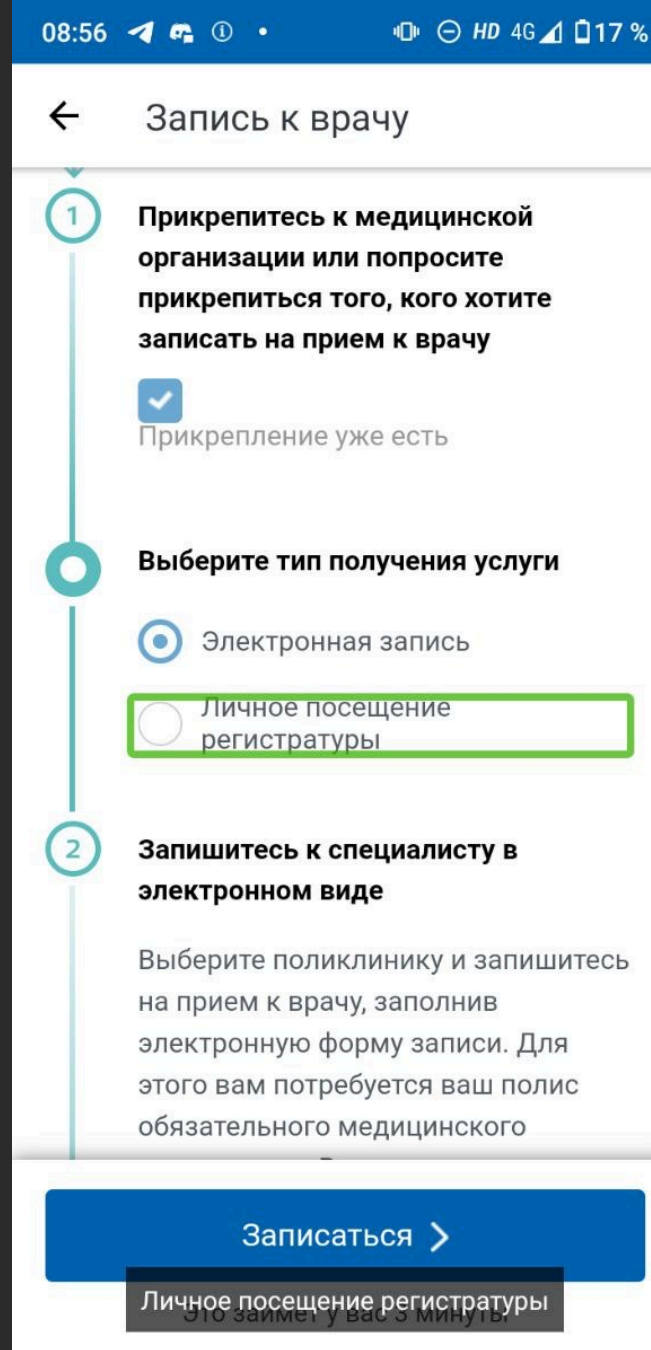
accept.*isEnabled* = isChecked



Нет типа и состояния

Непонятно
какой элемент выбран

Непонятно, что отметили
элемент выбранным



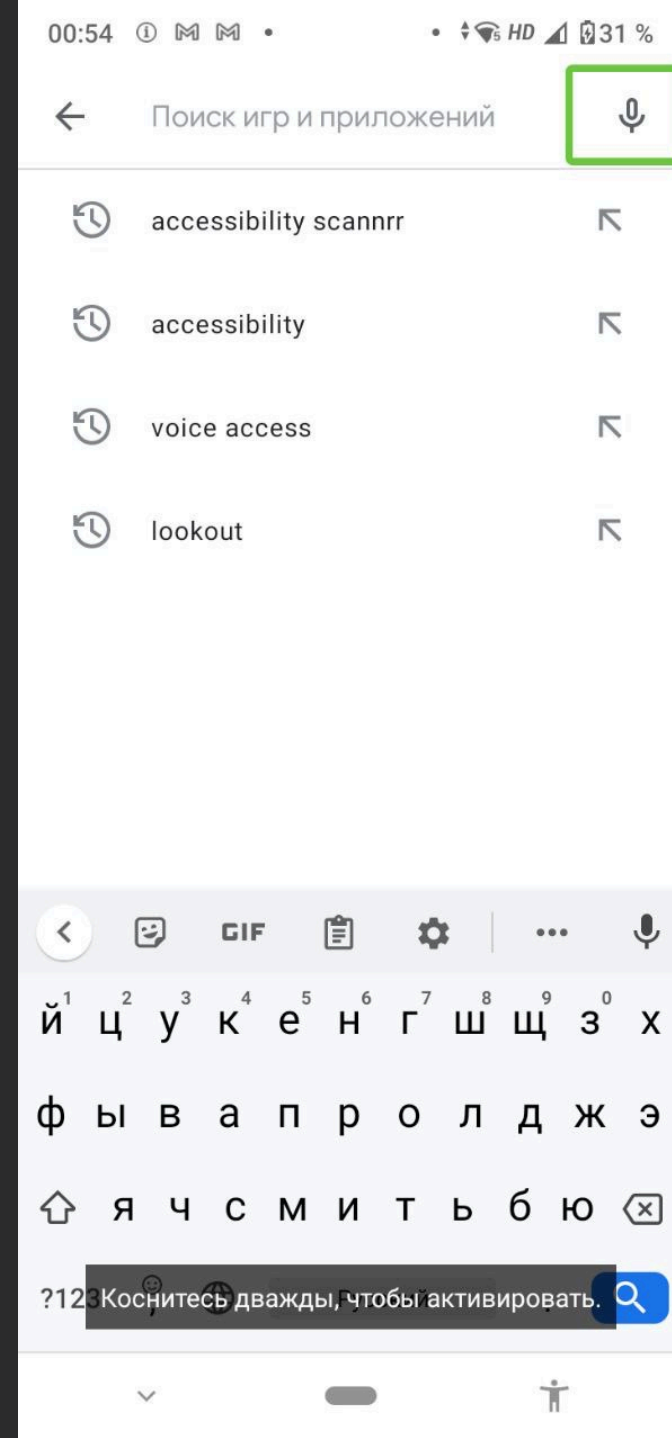
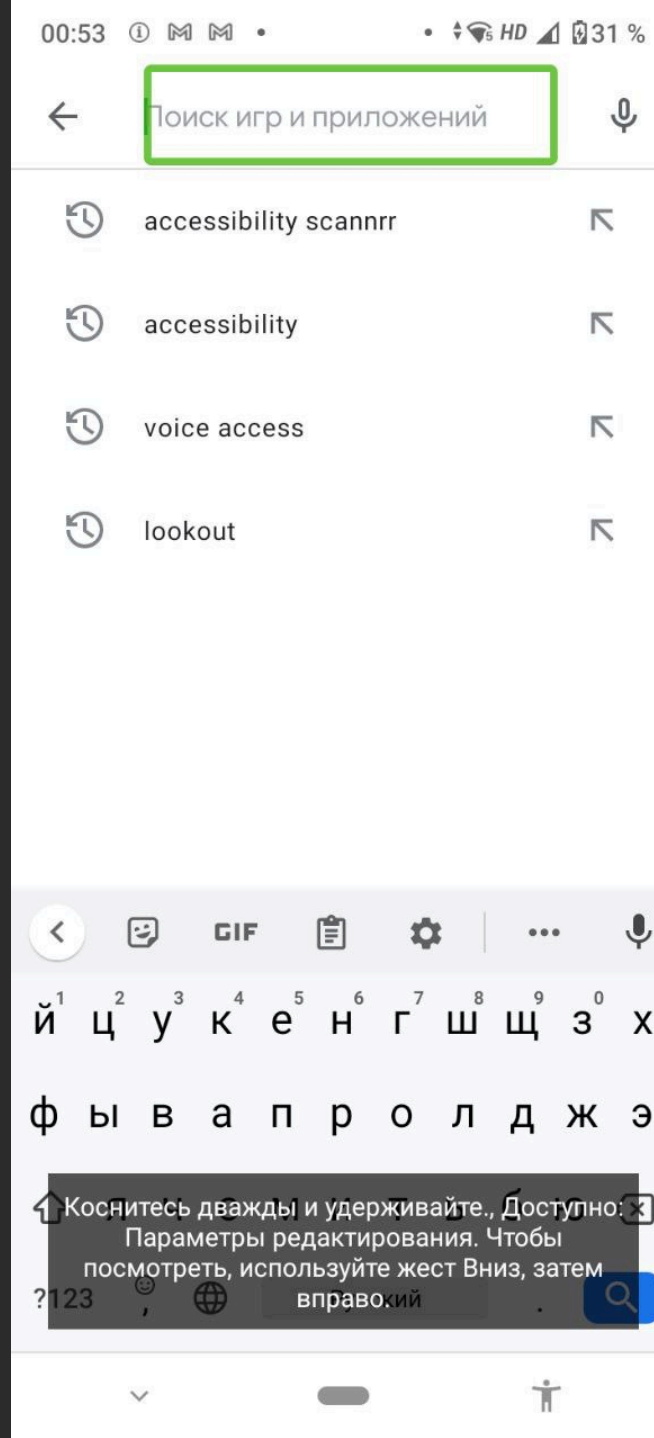
1. Описание

2. Тип

3. Состояние

4. Подсказка

Подсказка



Можно изменять текст подсказки

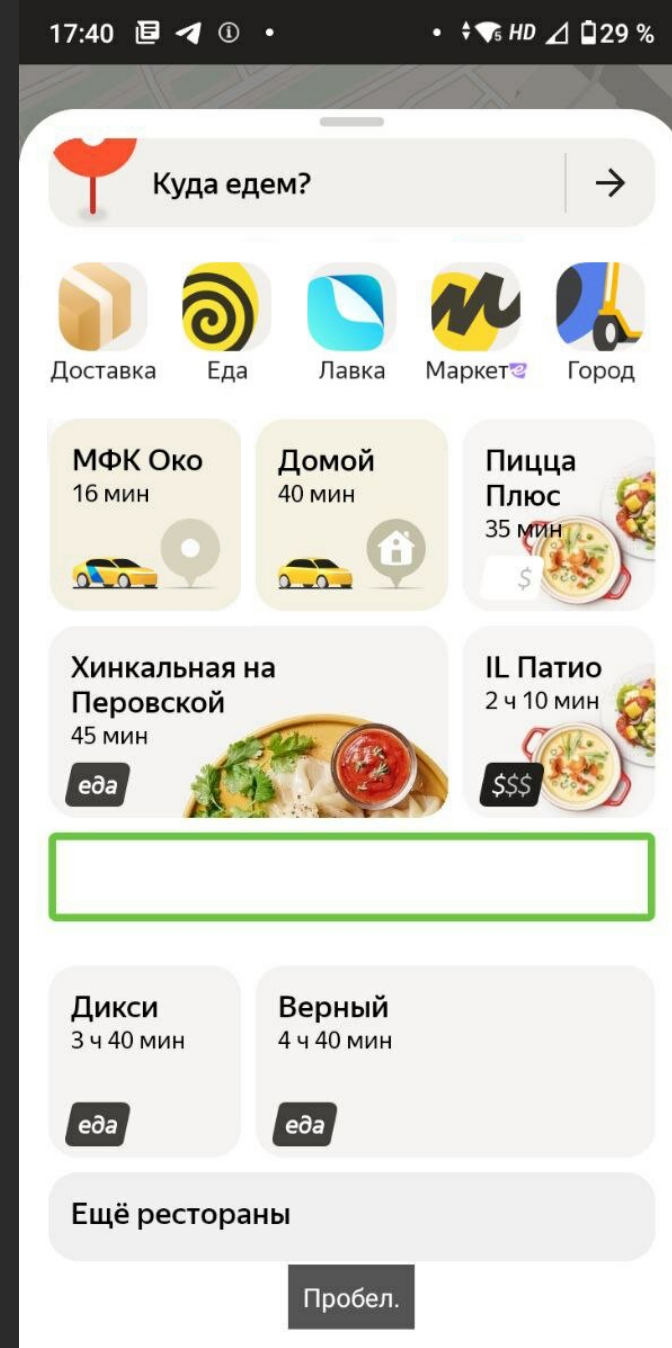
```
override fun onInitializeAccessibilityNodeInfo(info: AccessibilityNodeInfo) {  
    super.onInitializeAccessibilityNodeInfo(info)  
    info.addAction(AccessibilityNodeInfo.AccessibilityAction(  
        AccessibilityNodeInfo.ACTION_CLICK,  
        "Применить изменения"  
    ))  
}
```

Коснитесь дважды, чтобы Применить изменения.

Что можно скрывать?

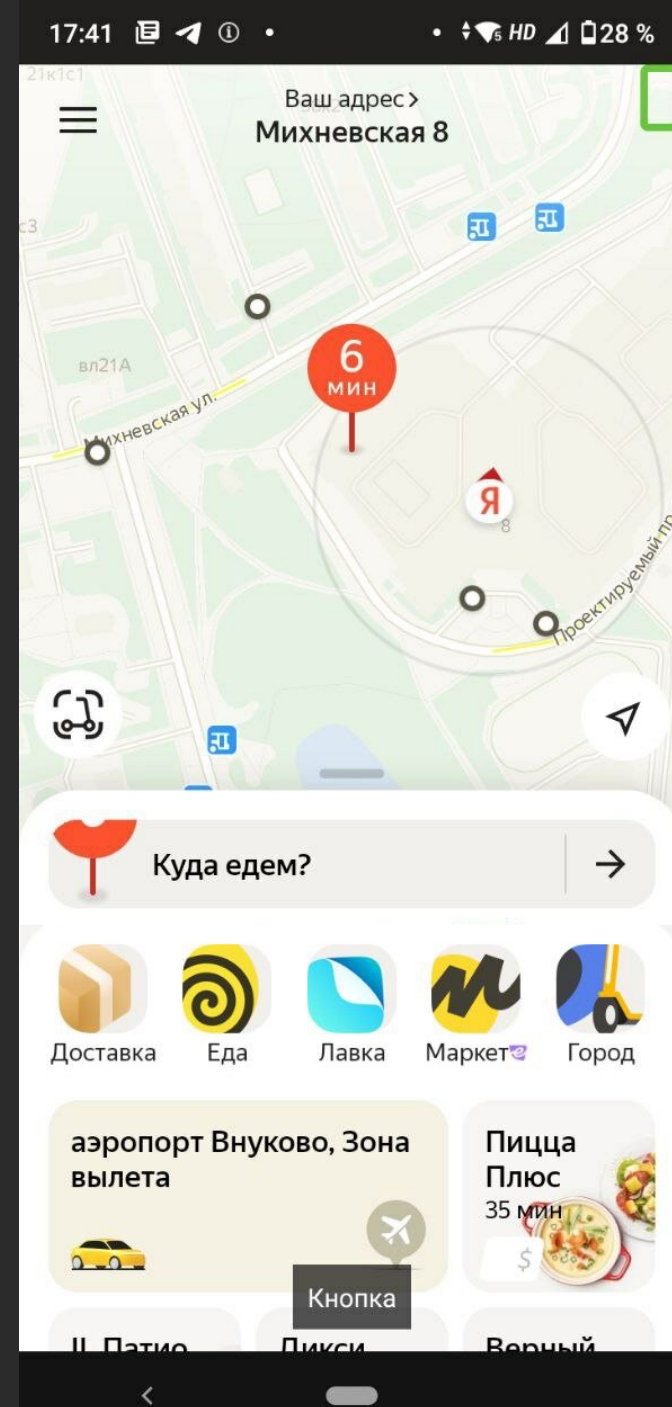
Декоративные элементы

Разделители и другие декоративные элементы



Пустой элемент

Элементы, которые не видны зрячему пользователю или неактивны в данный момент



Атрибут importantForAccessibility

Атрибут

```
<attr name="importantForAccessibility" format="integer">
```

Значения

```
auto | yes | no | noHideDescendants
```

Как скрыть декоративный элемент

```
<View
```

```
    android:id="@+id/divider"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="1dp"
```

```
    android:layout_marginHorizontal="8dp"
```

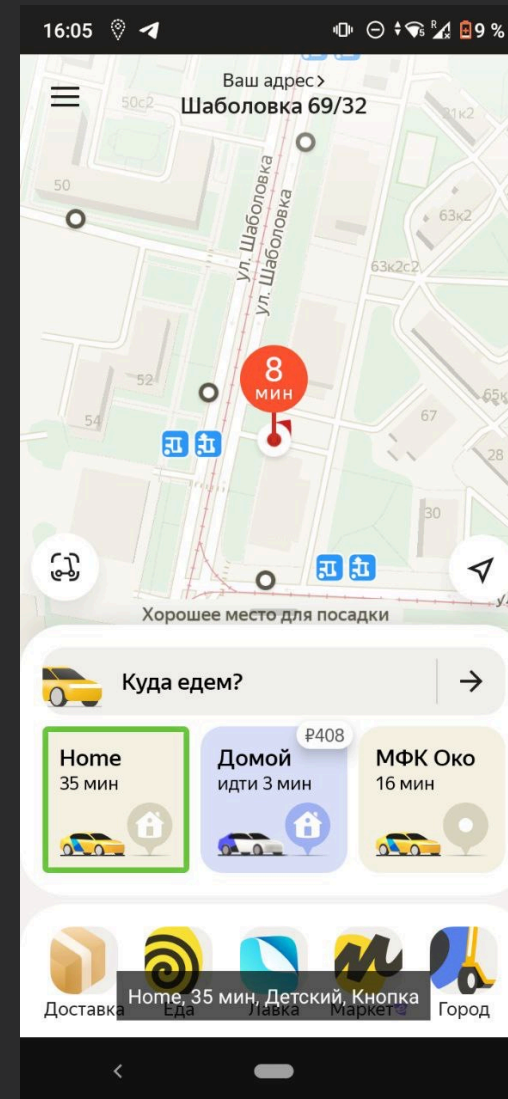
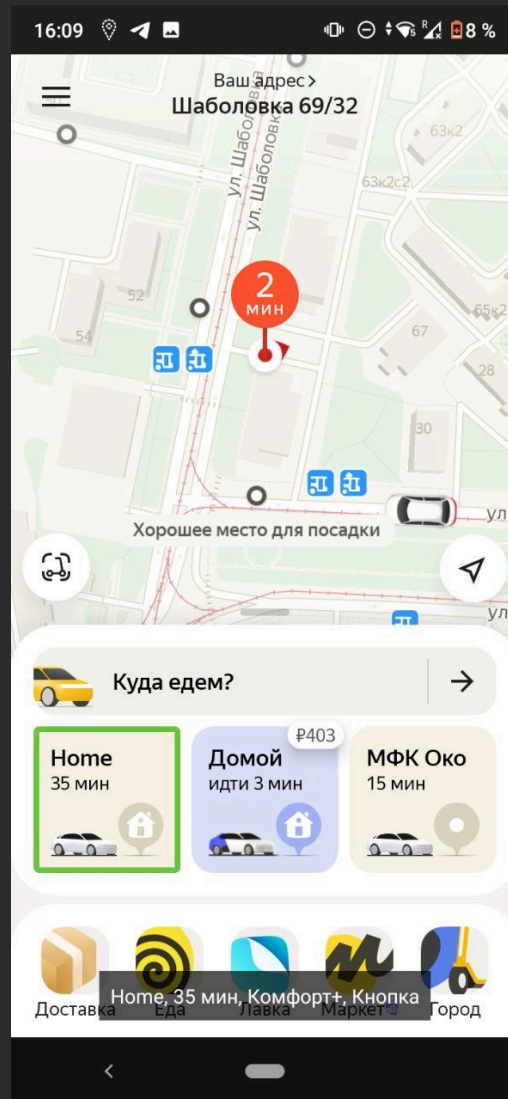
```
    android:importantForAccessibility="no"
```

```
/>
```

```
setImportantForAccessibility(View.IMPORTANT_FOR_ACCESSIBILITY_NO);
```

Что делать с изображениями?

Изображение дает больше контекста

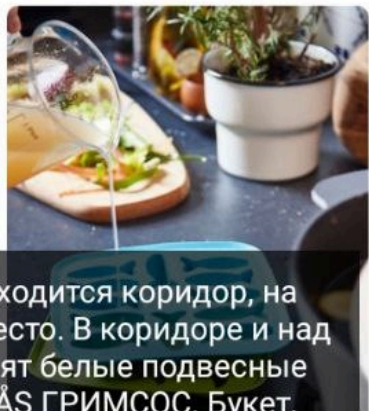
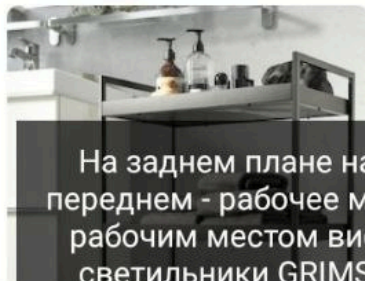
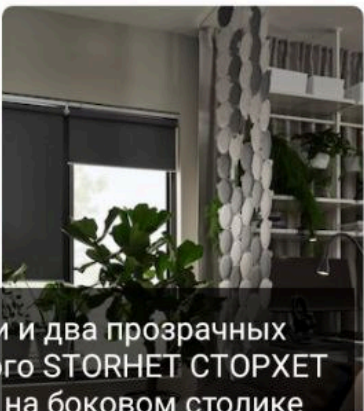
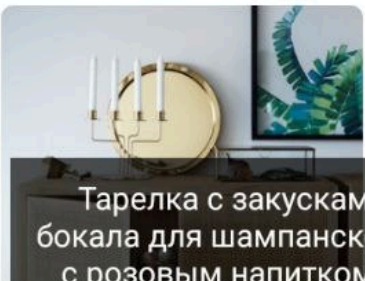


x



Увеличенное изображение, Изображение

Изображение занимает весь экран



Тарелка с закусками и два прозрачных бокала для шампанского **STORNET** **СТОПХЕТ** с розовым напитком на боковом столике у дивана-кровати. За воскресную ночь!



🔍 Кнопка, Столбец 1. ❤️



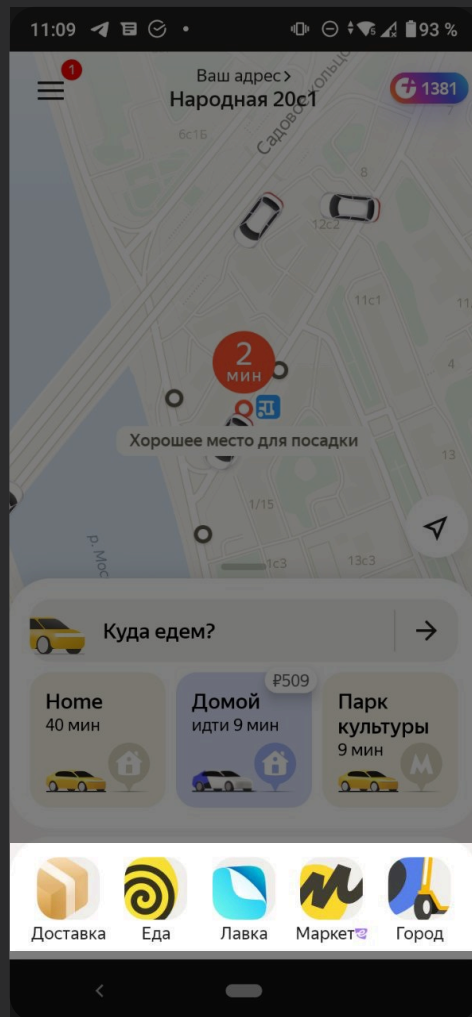
На заднем плане находится коридор, на переднем - рабочее место. В коридоре и над рабочим местом висят белые подвесные светильники **GRIMSÅS** **ГРИМСОС**. Букет, который сделает каждый ваш день ярче,



🔍 Кнопка, Столбец 2. ❤️



Нельзя скрывать функционал

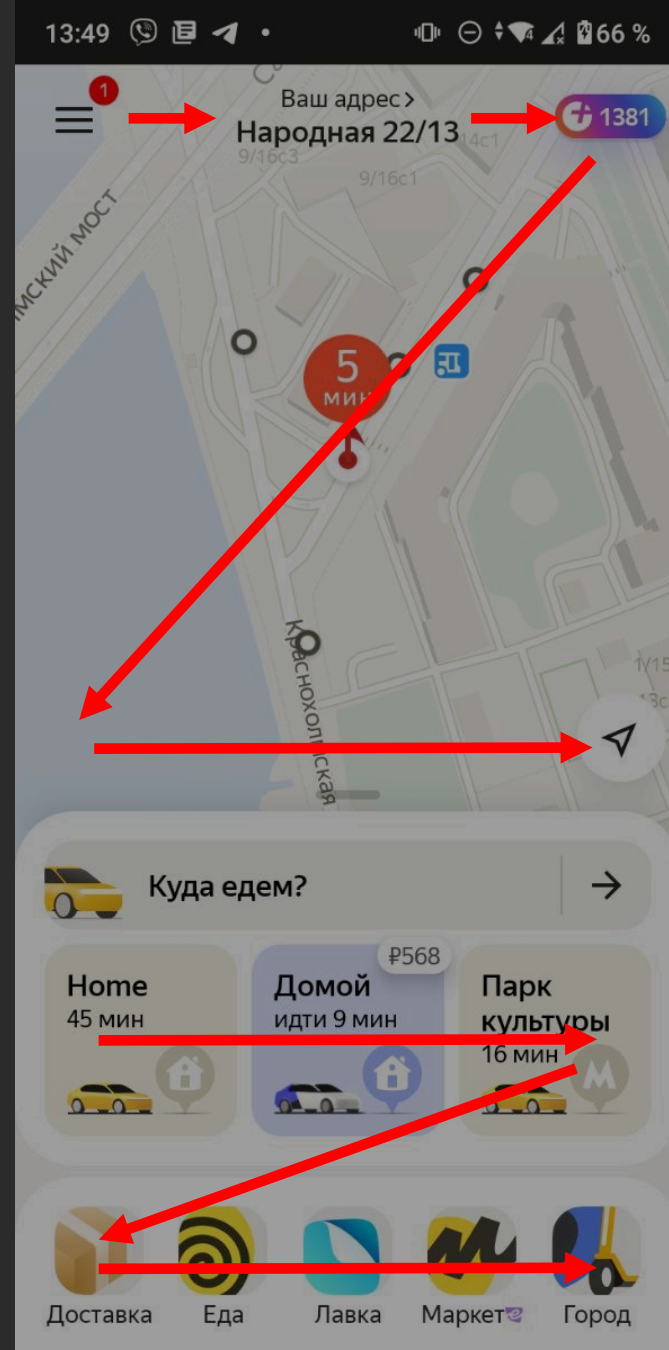


Важно не додумывать за пользователя

Придерживаться правила, что функционал должен быть одинаковым

Навигация. Порядок обхода

Порядок обхода



```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```
<TextView
```

```
    android:id="@+id/date_title"
```

```
    android:text="Дата доставки"
```

```
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/date_value"
```

```
    android:text="28 апреля"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@id/date_title" />
```

```
<TextView
```

```
    android:id="@+id/address_title"
```

```
    android:text="Адрес доставки"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/address_value"
```

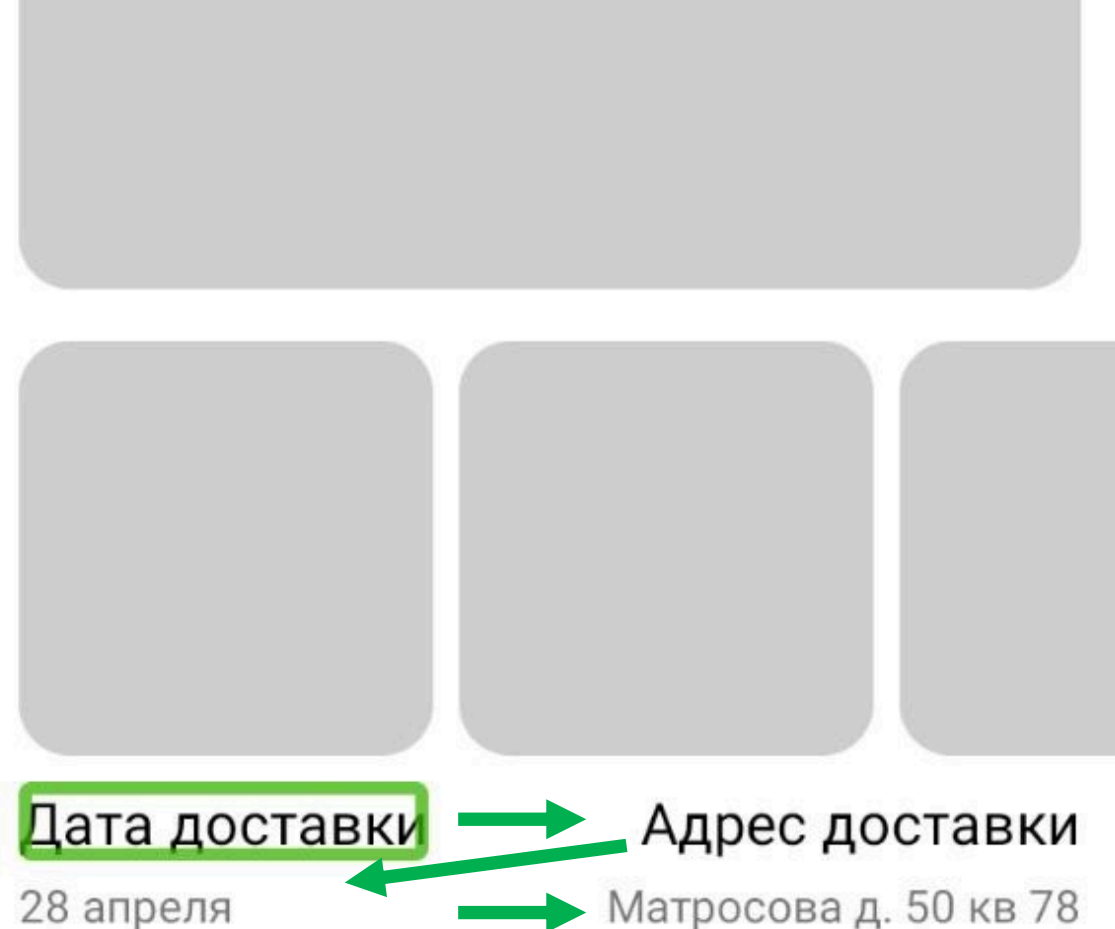
```
    android:text="Матросова д. 50 кв 78"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@id/address_title" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```



accessibilityTraversalAfter

```
<TextView  
    android:id="@+id/date_value"  
    android:text="28 апреля"  
    android:accessibilityTraversalAfter="@id/date_title"/>
```

accessibilityTraversalBefore

```
<TextView  
    android:id="@+id/date_value"  
    android:text="28 апреля"  
    android:accessibilityTraversalBefore="@id/date_title"/>
```

*Атрибуты доступны в API 22 и выше

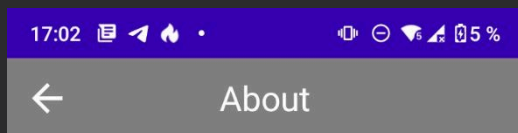


Виды навигации

- По символам
- По словам
- По строкам
- По абзацам
- По заголовкам
- По ссылкам
- По элементам управления

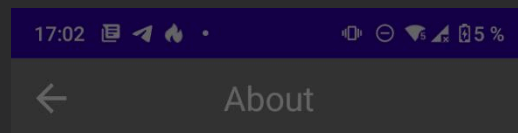
Используется альтернативный жест, например, свайп вверх/вниз

Навигация по тексту



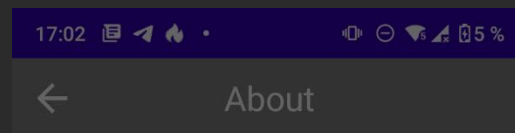
Android apps should aim to be usable by everyone, including people with accessibility needs. People with impaired vision, color blindness, impaired hearing, impaired dexterity, cognitive disabilities, and many other disabilities use Android devices to complete tasks in their day-to-day lives. When you develop apps with accessibility in mind, you make the user experience better, particularly for users with these and other accessibility needs. This document presents guidelines for implementing key elements of accessibility so that everyone can use your app more easily. For more in-depth guidance on how to make your app more accessible, visit the principles for improving app accessibility page.

Android apps should aim to be usable by everyone, including people with accessibility needs. People with impaired vision, color blindness, impaired hearing, impaired dexterity, cognitive disabilities, and many other disabilities use Android devices to complete tasks in their day-to-day lives. When you develop apps with accessibility in mind, you make the user experience better, particularly for users with these and other accessibility needs. This document presents guidelines for implementing key elements of accessibility so that everyone can use your app more easily. For more in-depth guidance on how to make your app more accessible, visit the principles for improving app accessibility page.



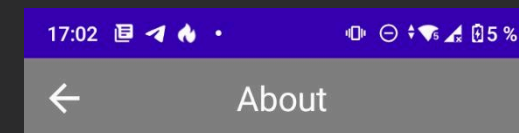
- Меню TalkBack
- Навигация
 - Читать со следующего элемента
 - Копировать последнюю фразу
 - Язык озвучивания
 - Поиск по экрану
 - Отключить экран
 - Голосовые команды
 - Настройки TalkBack
 - Настройки озвучивания текста
- Отмена

Навигация, 1 из 9, В списке, 9 пунктов



- Навигация
- Символы
 - Слова
 - Строки
 - Абзацы
 - Заголовки
 - Ссылки
 - Элементы управления
- Отмена

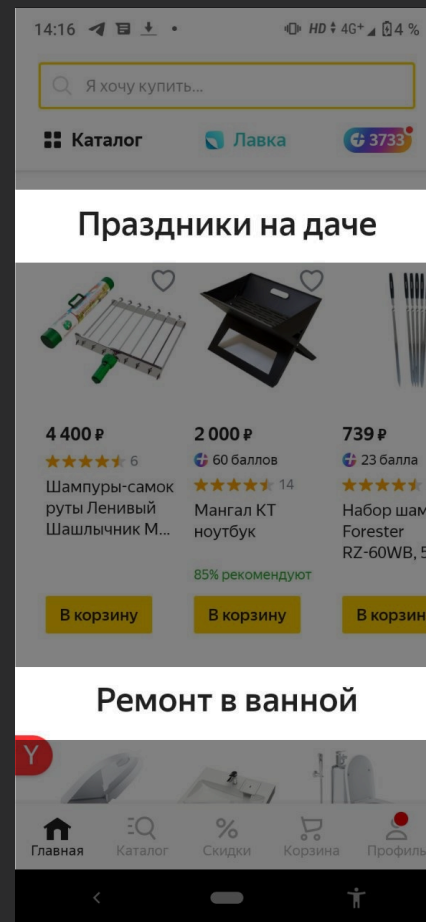
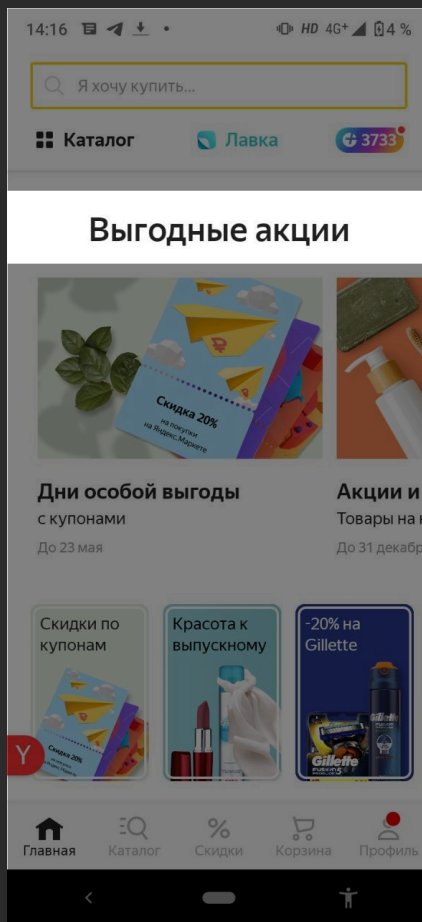
Строки, 3 из 7



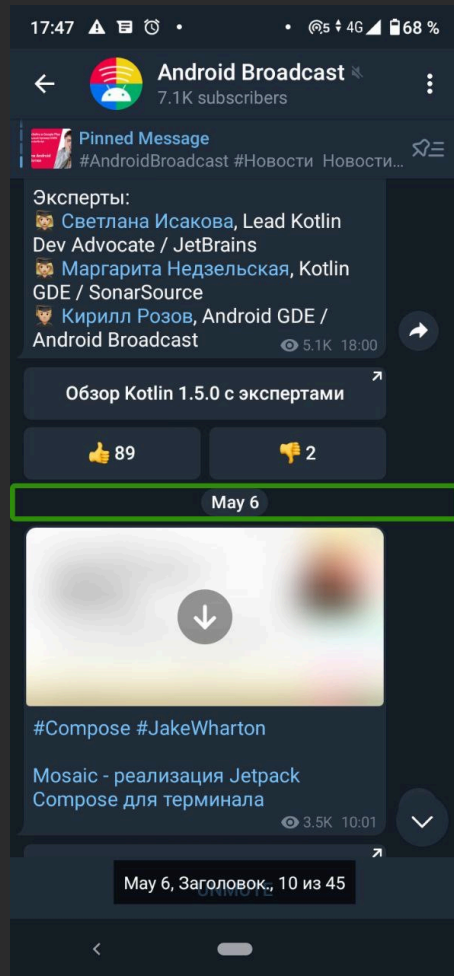
Android apps should aim to be usable by everyone, including people with accessibility needs. People with impaired vision, color blindness, impaired hearing, impaired dexterity, cognitive disabilities, and many other disabilities use Android devices to complete tasks in their day-to-day lives. When you develop apps with accessibility in mind, you make the user experience better, particularly for users with these and other accessibility needs. This document presents guidelines for implementing key elements of accessibility so that everyone can use your app more easily. For more in-depth guidance on how to make your app more accessible, visit the principles for improving app accessibility page.

Android apps should aim to be usable by

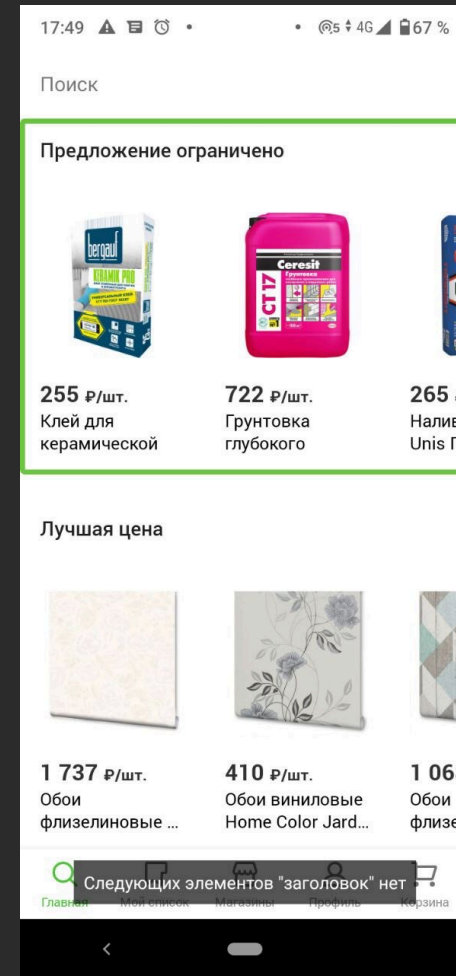
Заголовки



Навигация по заголовкам



May 6, Заголовок, 10 из 45



Следующих элементов
"заголовок" нет

Атрибут accessibilityHeading

```
android:accessibilityHeading="true"
```

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {  
    title.isAccessibilityHeading = true  
}
```


*Атрибут доступен в API 28 и выше

Объединение элементов

Части одного элемента
Разделены из за верстки

01:26 HD 18%

Current address > Бирюлёво-Тов. Восточное
Михневская 8 17976



Григорий

5.0 рейтинг 20 лет стажа 0 км 43 поездки

Муравьев Григорий Дорофеевич


Автомобиль
белый Ford Mondeo, У 776 МО 750

Партнёр
Индивидуальный предприниматель
Юдаков Максим Викторович

Запросить

01:26 HD 18%

Current address > Бирюлёво-Тов. Восточное
Михневская 8 17976



Григорий

5.0 рейтинг 20 лет стажа 0 км 43 поездки

Муравьев Григорий Дорофеевич

Автомобиль
белый Ford Mondeo, У 776 МО 750

Партнёр
Индивидуальный предприниматель
Юдаков Максим Викторович

Рейтинг

Объединение в группу

- Ухудшается целостное восприятие
- Избыточное количество переключений фокуса

осталось:

Купить 30 из 30

Оценить 15 из 15

Вы помогаете делать ассортимент магазинов "ВкусВилл" лучше!



Оценить товары



Купленные товары



Что еще купить?

Прогресс

Скидка 3%

Купите 30 разных товаров



Корзина 158Р



Моя карта



Каталог



Магазины



Профиль



Поддержка

Объединение в группу

```
<androidx.constraintlayout.widget.ConstraintLayout  
    android:id="@+id/designer_item"
```

```
...
```

```
    android:focusableInTouchMode="true">
```

```
    <TextView
```

```
        ...
```

```
        android:focusableInTouchMode="false"/>
```

```
    <TextView
```

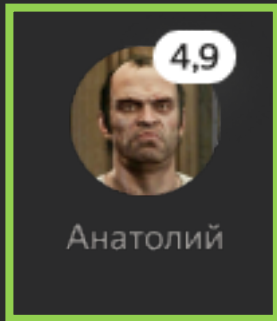
```
        android:id="@+id/designer"
```

```
        ...
```

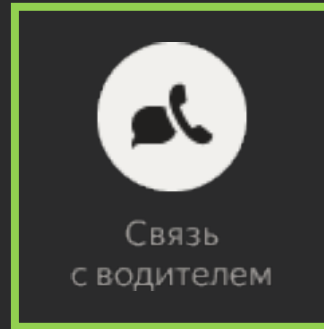
```
        android:focusableInTouchMode="false"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

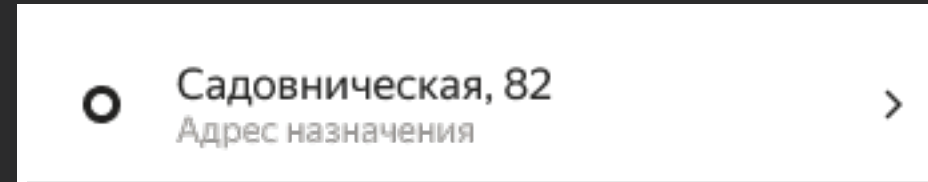
Группировка элементов



Анатолий, 4.9 кнопка



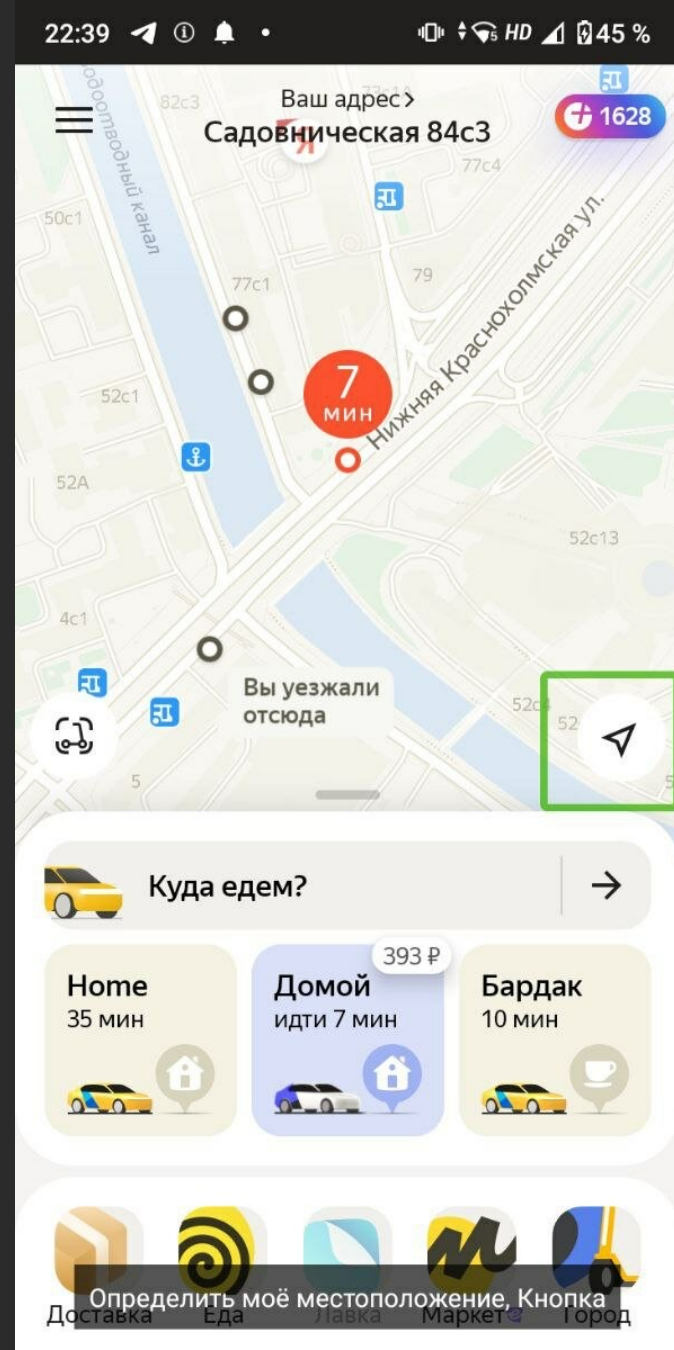
Связь с водителем, Кнопка



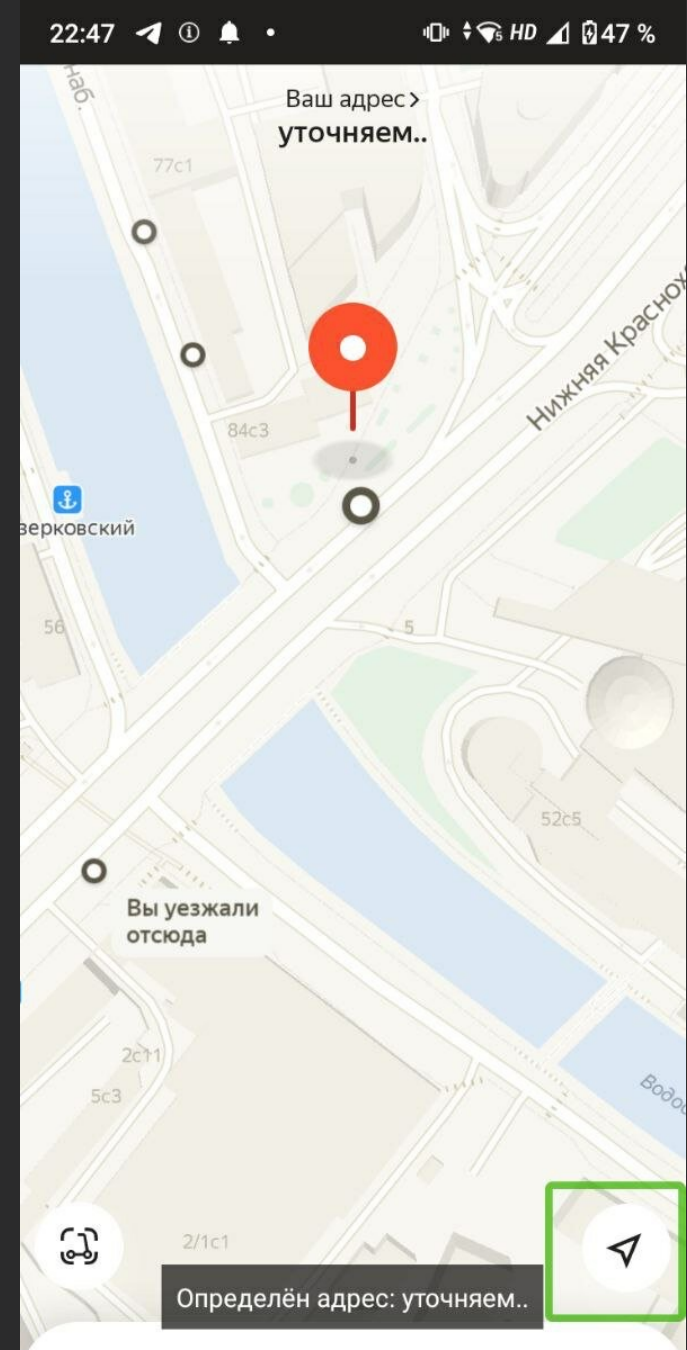
Садовническая, 82 адрес назначения, Кнопка

Динамическое обновление данных

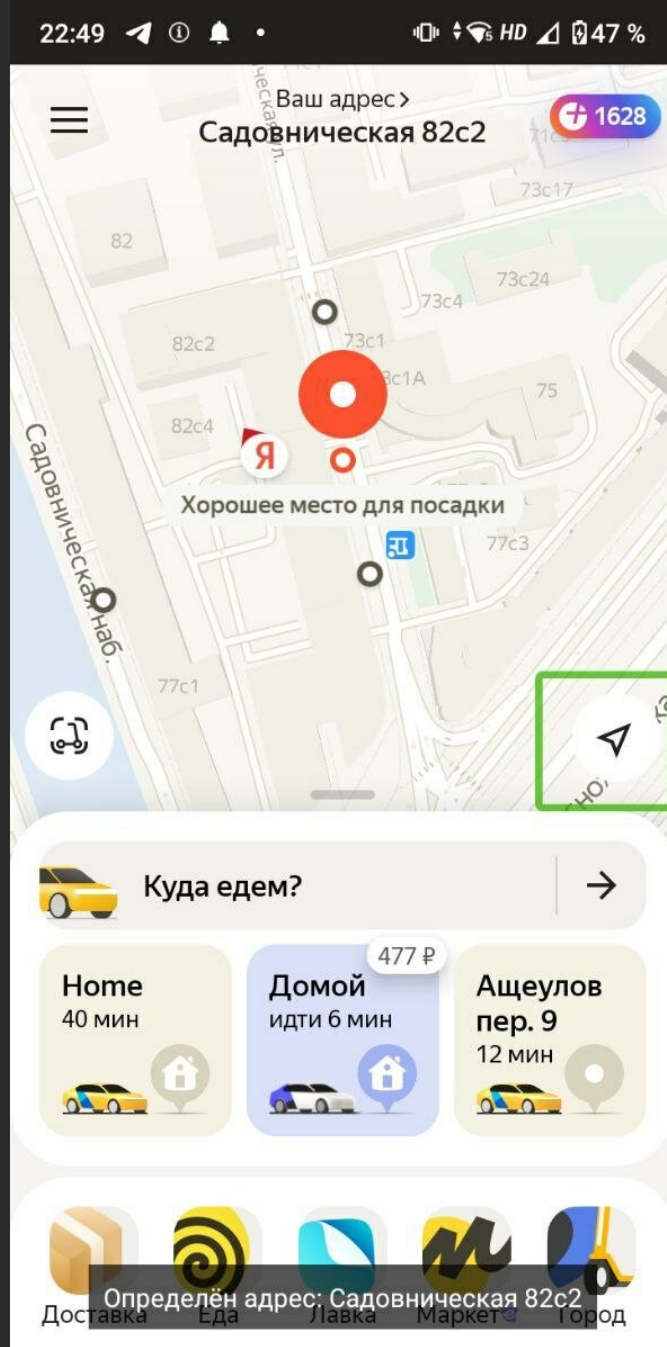
Тап на местоположение



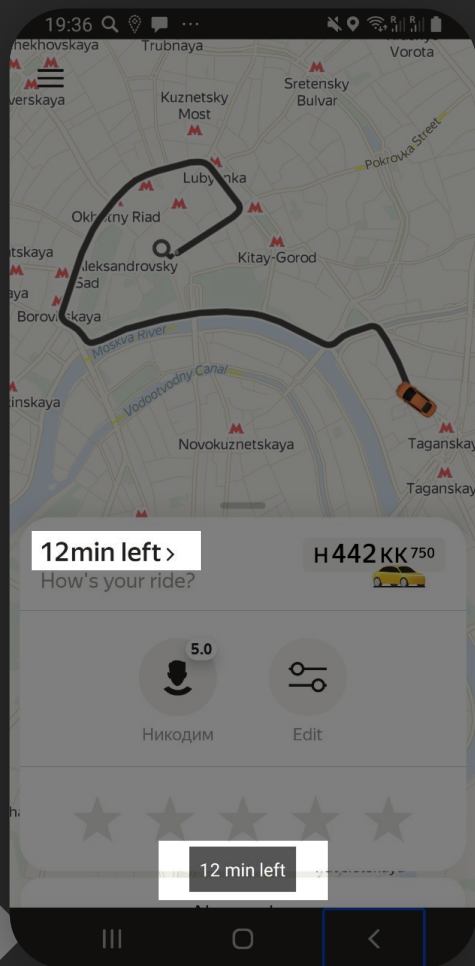
Адрес определяется



Адрес изменяется



Динамическое обновление данных



`ViewCompat.setAccessibilityLiveRegion(this, liveRegion);`

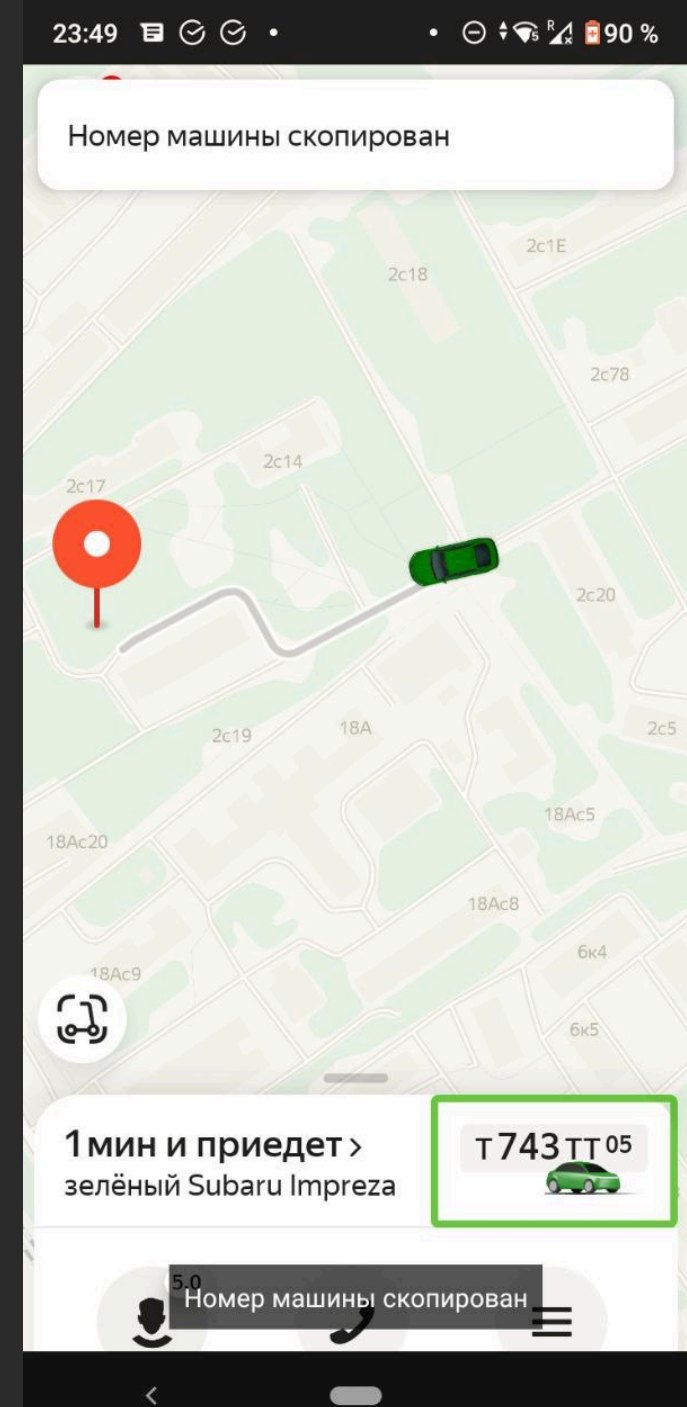
`ACCESSIBILITY_LIVE_REGION_NONE` (по умолчанию)

`ACCESSIBILITY_LIVE_REGION_POLITE`

`ACCESSIBILITY_LIVE_REGION_ASSERTIVE` (прерывающий)

Нотификации

`announceForAccessibility(notificationTitle);`

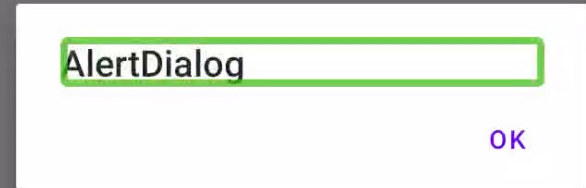


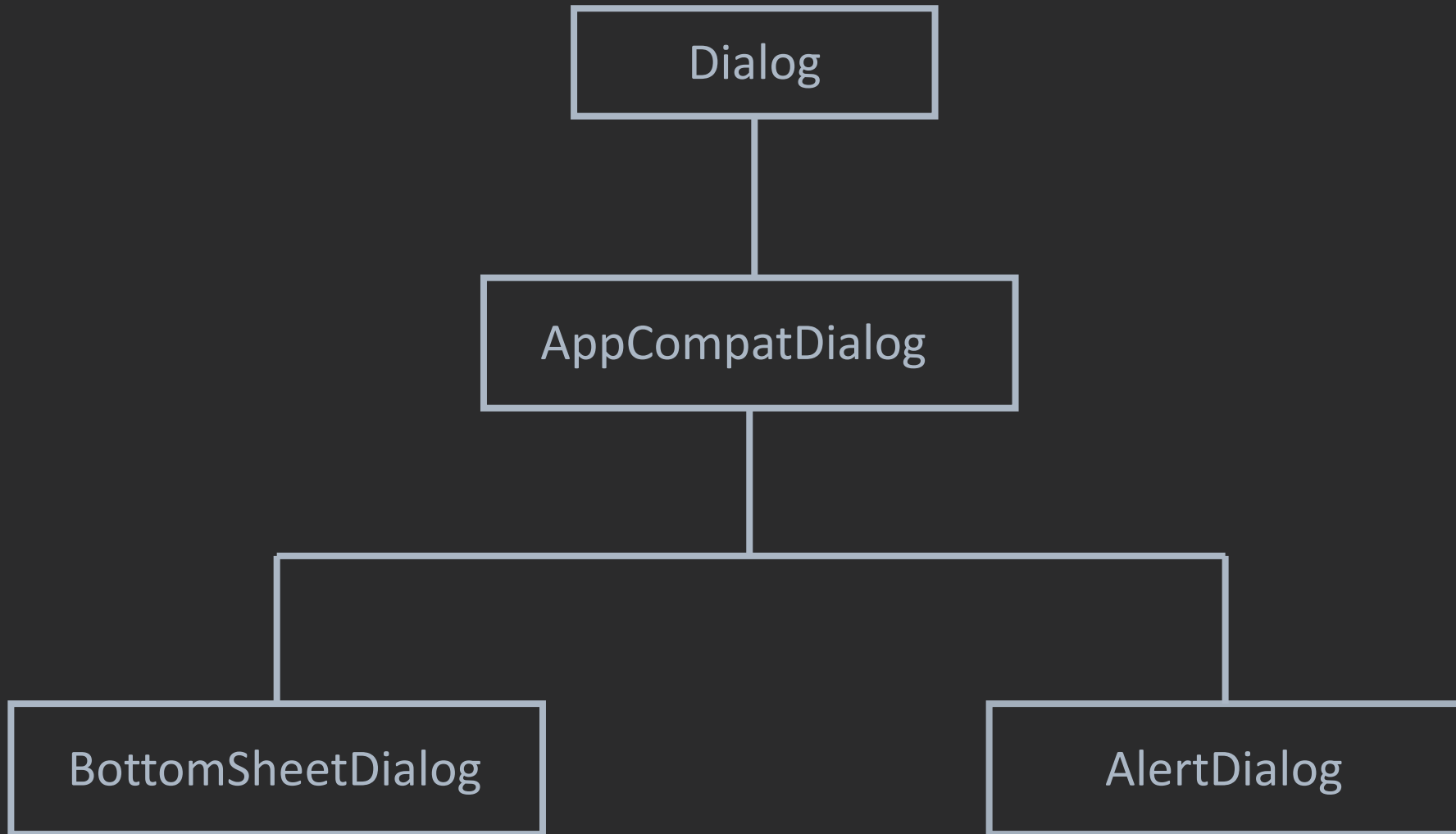
Модальные окна AlertDialog, BottomSheetDialog

AlertDialog

```
AlertDialog.Builder(this)
    .setTitle("AlertDialog")
    .setPositiveButton("OK") { _, _ -> }
    .show()
```

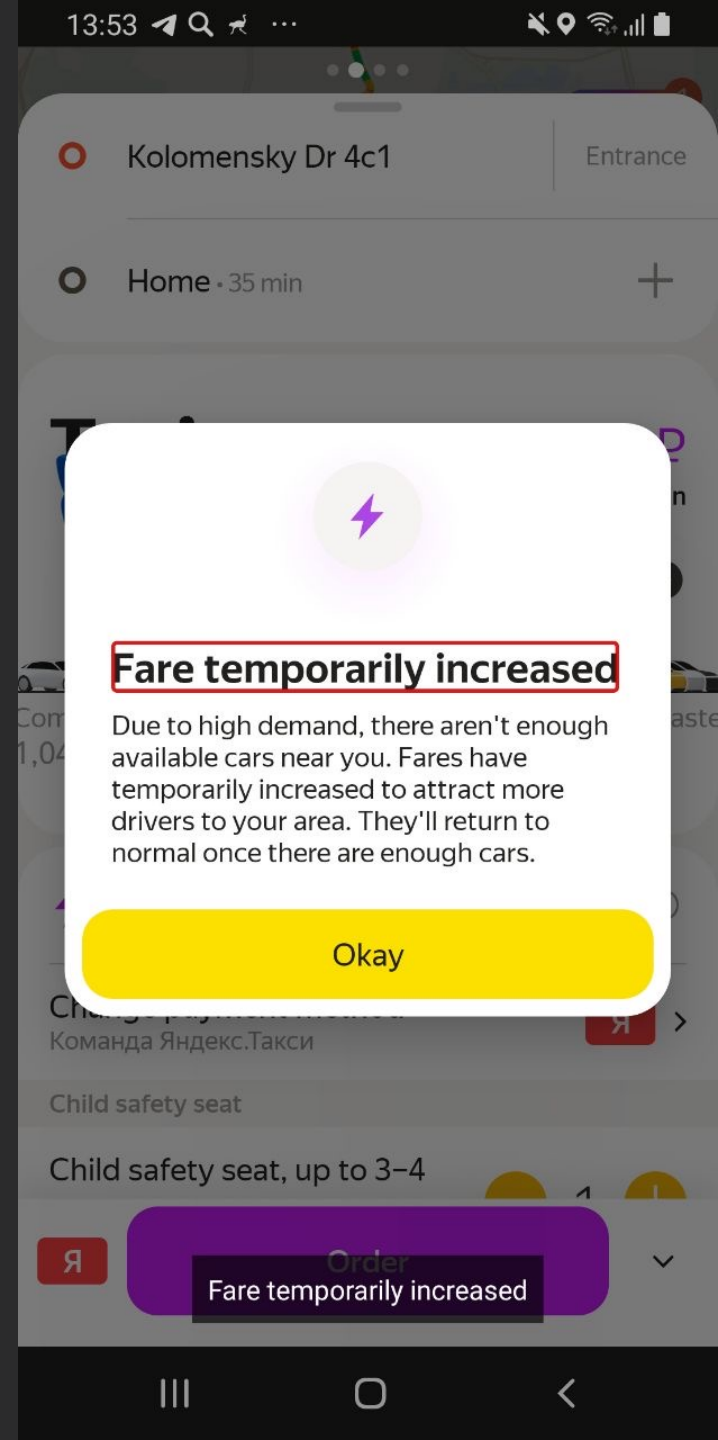
Element under dialog
cover





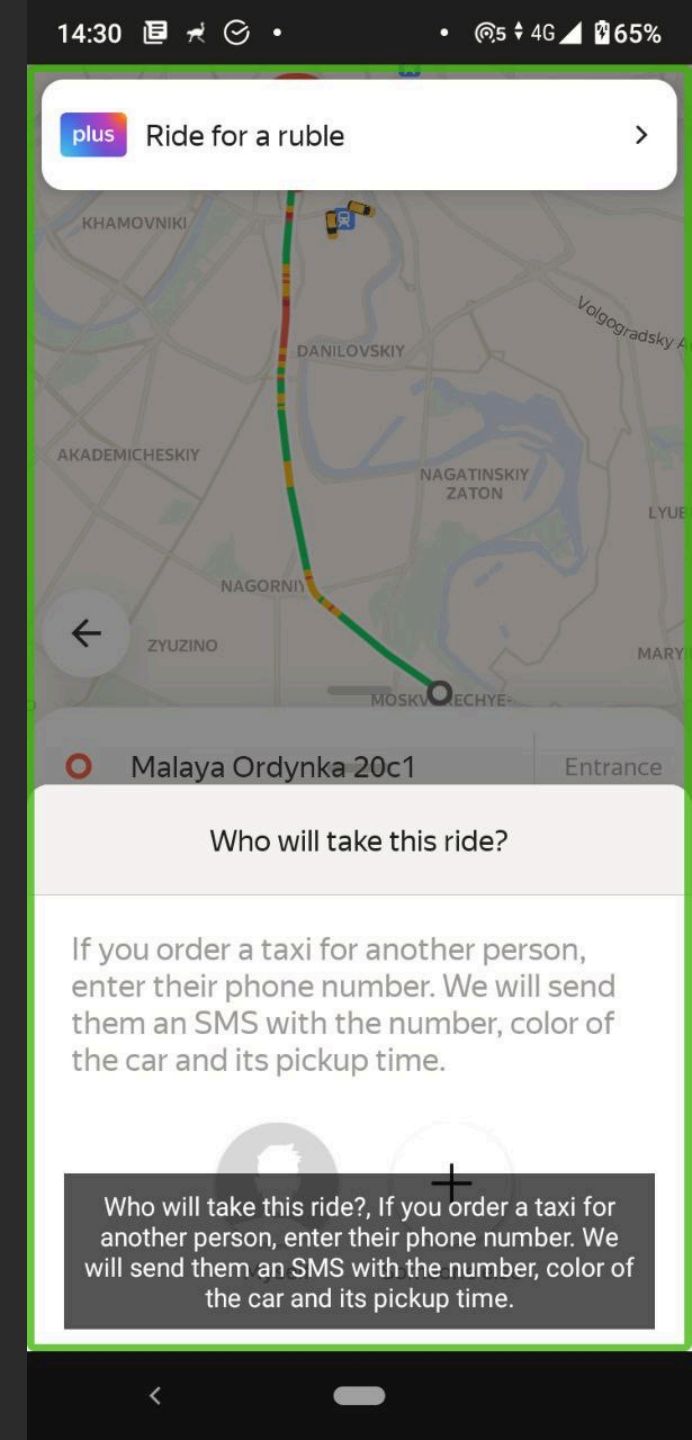
Custom AlertDialog

```
public class BaseDialog extends FrameLayout
```

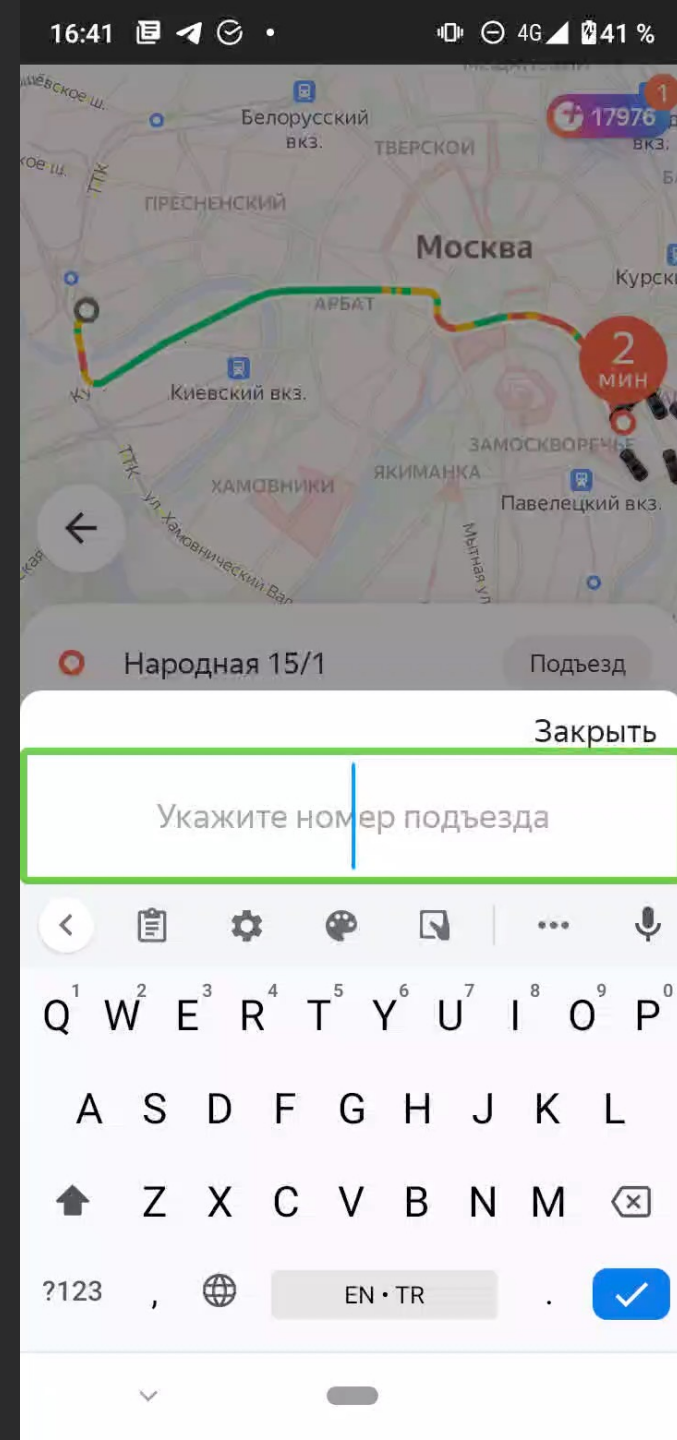


Custom BottomDialog

public abstract class BottomDialog extends FrameLayout



Фокус уходит с контекста диалога

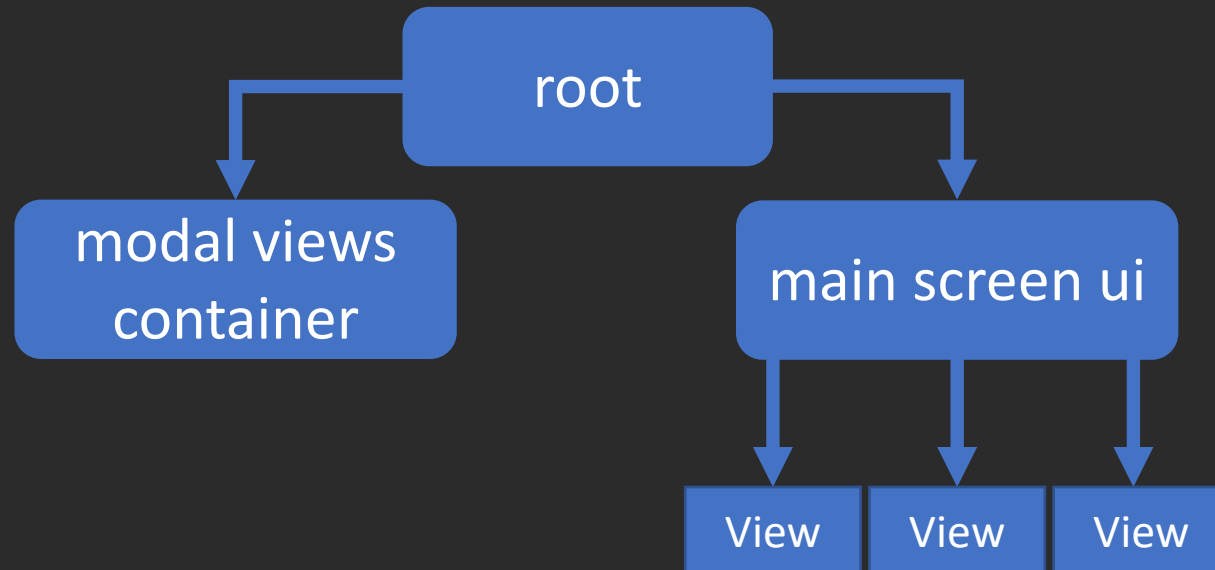


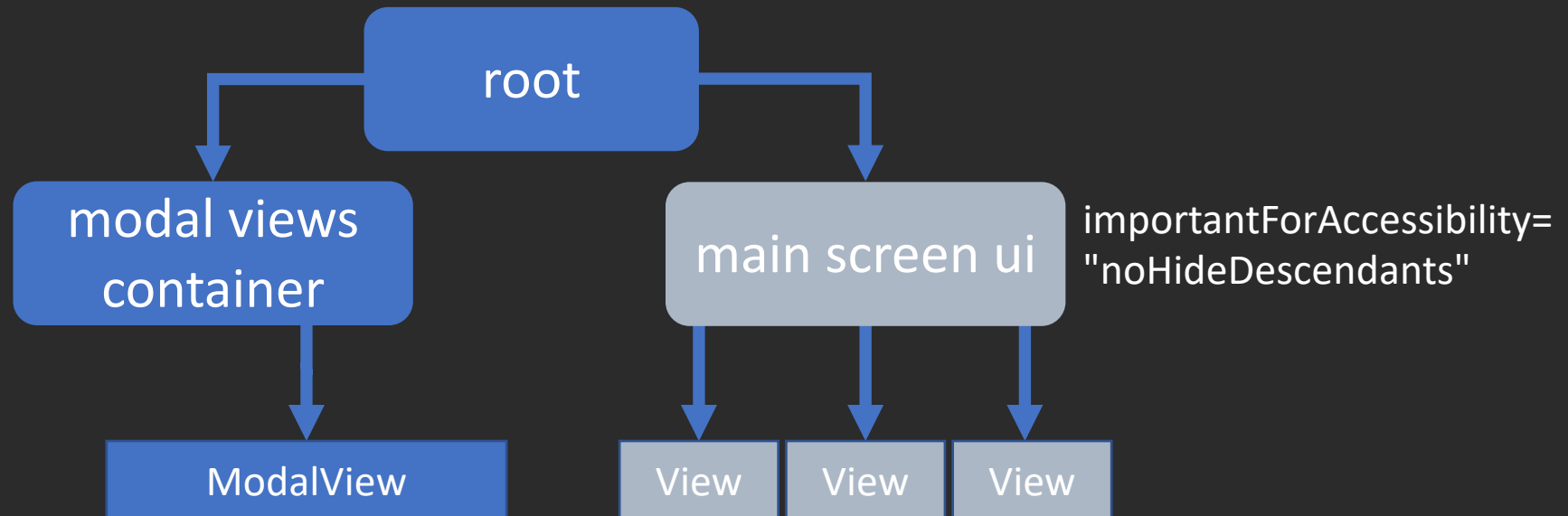
В это время в iOS

`accessibilityViewsModal`

A Boolean value indicating whether VoiceOver should ignore the elements within views that are siblings of the receiver.

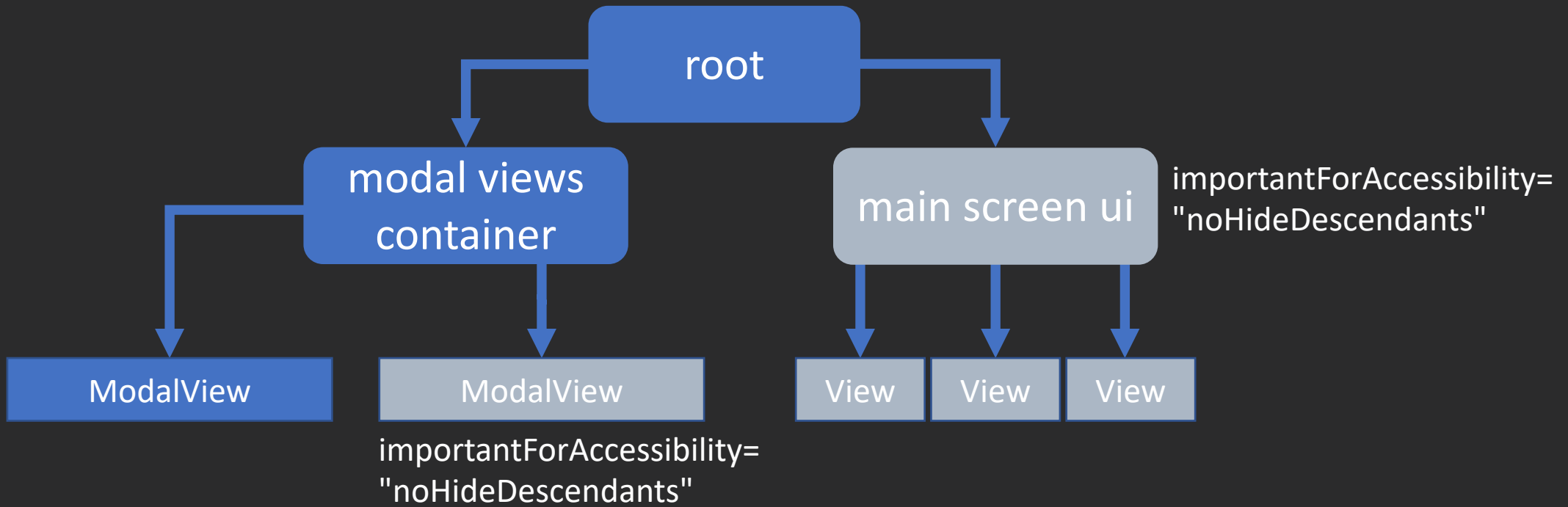
Блокирование соседних элементов

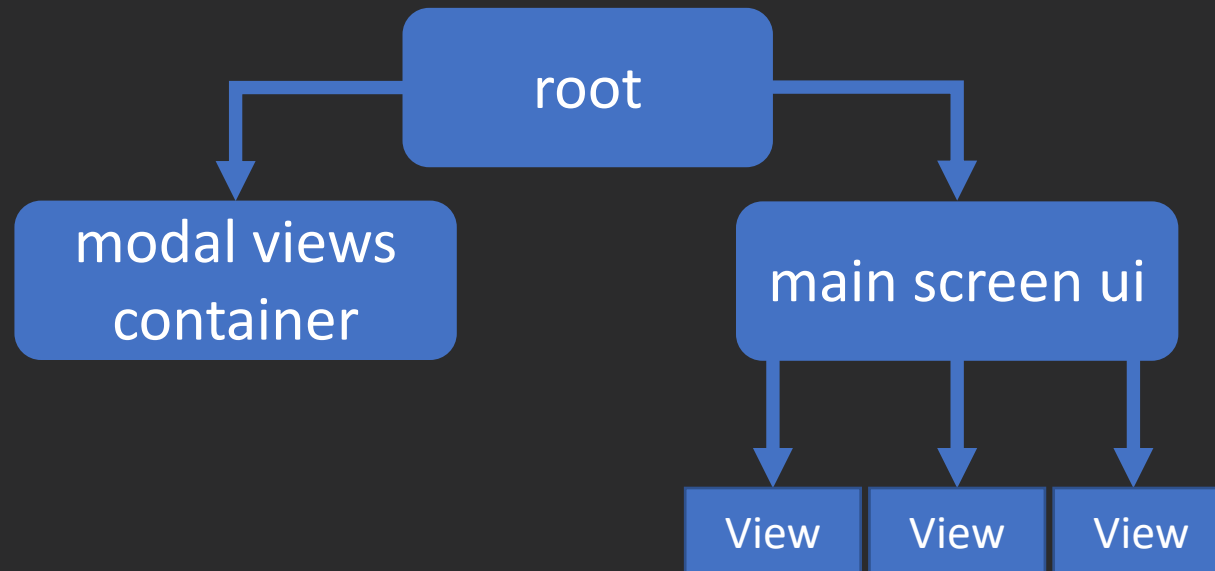




Записываем значение атрибута `importantForAccessibility`

```
Map<View, Integer> importanceMap = new HashMap<>();
```





Восстанавливаем значения атрибутам из сохраненного массива

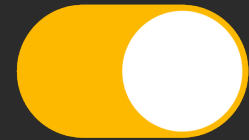
```
Map<View, Integer> importanceMap = new WeakHashMap<>();
```

Custom View

Custom Switch

```
abstract class Switcher @JvmOverloads constructor(  
    ...  
    ) : View(context, attrs, defStyleAttr) {
```

```
    open var isChecked = true  
        protected set
```



Без ярлыка

Custom Switch

```
abstract class Switcher @JvmOverloads constructor(  
    ...  
    ) : View(context, attrs, defStyleAttr) {
```

```
    open var isChecked = true  
        protected set
```

```
    override fun getAccessibilityClassName(): CharSequence {  
        return Switch::class.java.name  
    }
```

```
open var isChecked = true
    protected set
```

```
override fun getAccessibilityClassName(): CharSequence {
    return Switch::class.java.name
}
```



Переключатель, не отмечено.



Переключатель, не отмечено.

```
open var isChecked = true
    protected set
```

```
override fun getAccessibilityClassName(): CharSequence {
    return Switch::class.java.name
}
```

```
override fun onInitializeAccessibilityNodeInfo(
    info: AccessibilityNodeInfo
) {
    super.onInitializeAccessibilityNodeInfo(info)
    info.isChecked = isChecked
}
```

```
open var isChecked = true
    protected set
```

```
override fun getAccessibilityClassName(): CharSequence {
    return Switch::class.java.name
}
```

```
override fun onInitializeAccessibilityNodeInfo(
    info: AccessibilityNodeInfo
){
    super.onInitializeAccessibilityNodeInfo(info)
    info.isChecked = isChecked
}
```



Переключатель, не отмечено.



Переключатель, отмечено.

```
open var isChecked = true
    protected set
```

```
override fun getAccessibilityClassName(): CharSequence {
    return Switch::class.java.name
}
```

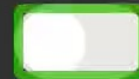
```
override fun onInitializeAccessibilityNodeInfo(
    info: AccessibilityNodeInfo
) {
    super.onInitializeAccessibilityNodeInfo(info)
    info.isCheckable = true
    info.isChecked = isChecked
}
```

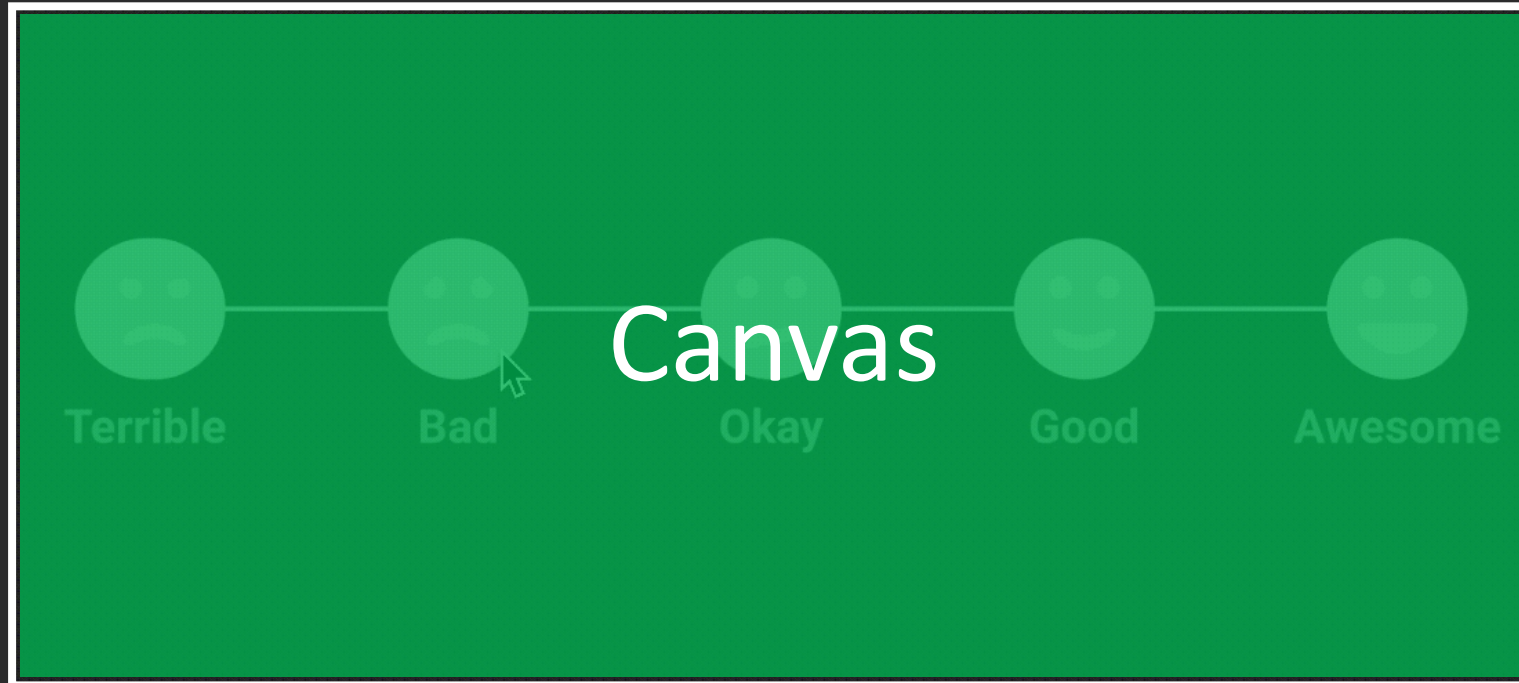


```
open var isChecked = true
    protected set
```

```
override fun getAccessibilityClassName(): CharSequence {
    return Switch::class.java.name
}
```

```
override fun onInitializeAccessibilityNodeInfo(
    info: AccessibilityNodeInfo
) {
    super.onInitializeAccessibilityNodeInfo(info)
    info.isCheckable = true
    info.isChecked = isChecked
}
```





View

ExploreTouchHelper

```
protected abstract void getVisibleVirtualViews(List<Integer> virtualViewIds);
```

```
protected abstract int getVirtualViewAt(float x, float y);
```

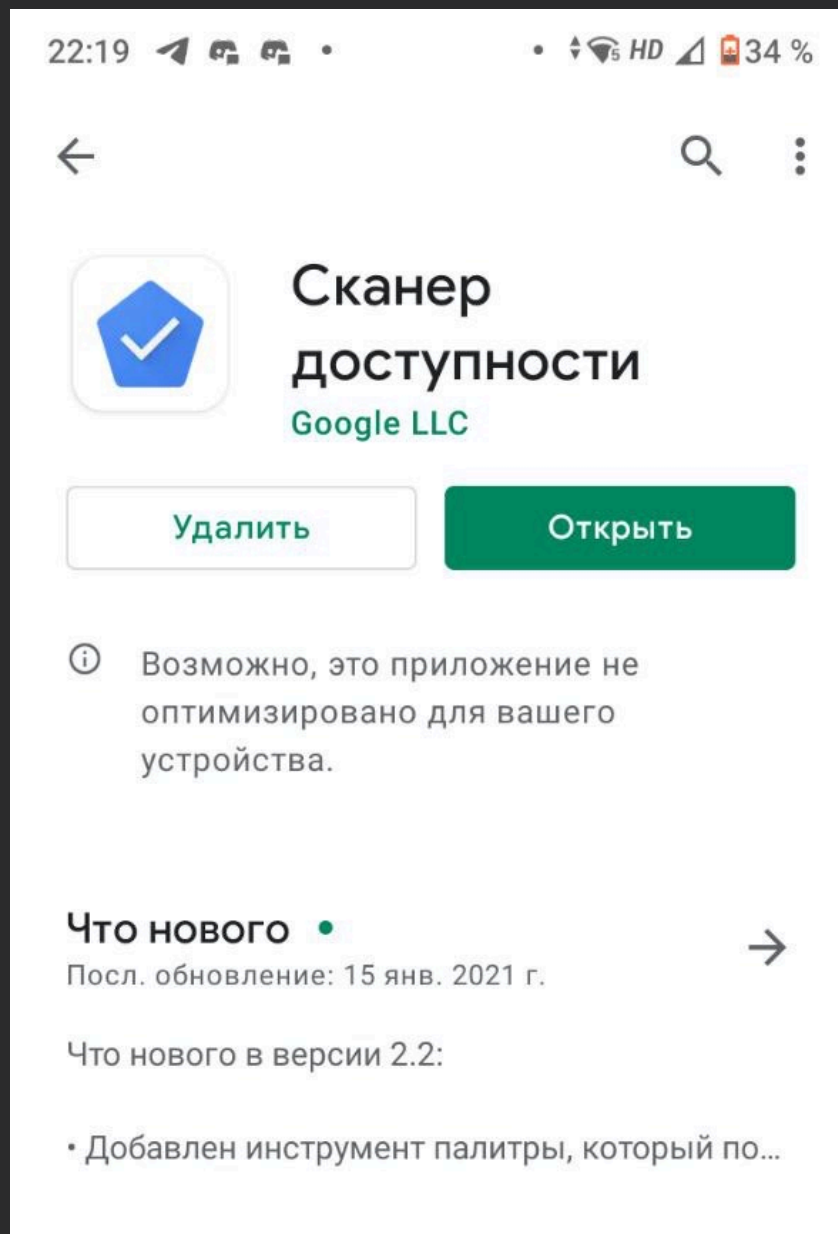
```
protected abstract void onPopulateNodeForVirtualView(  
    int virtualViewId, @NonNull AccessibilityNodeInfoCompat node);
```

```
protected abstract boolean onPerformActionForVirtualView(  
    int virtualViewId, int action, @Nullable Bundle arguments);
```

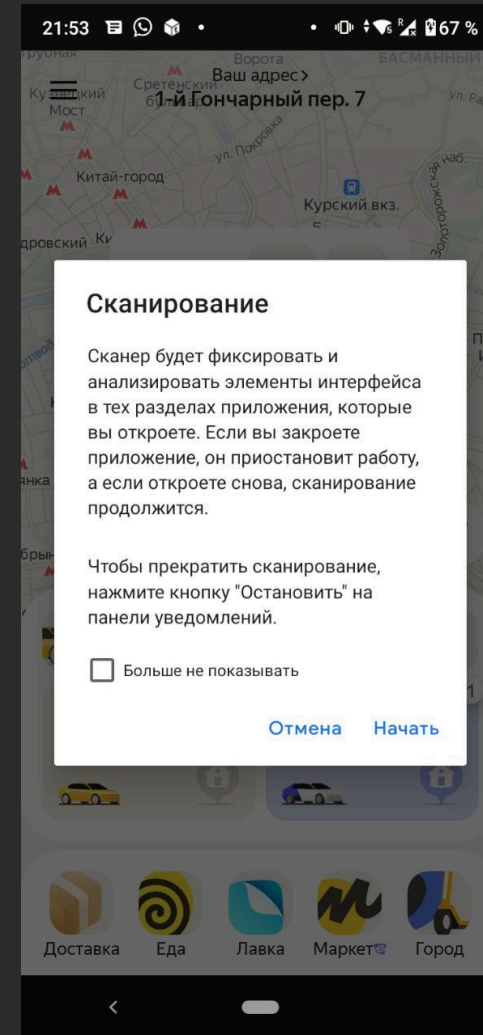
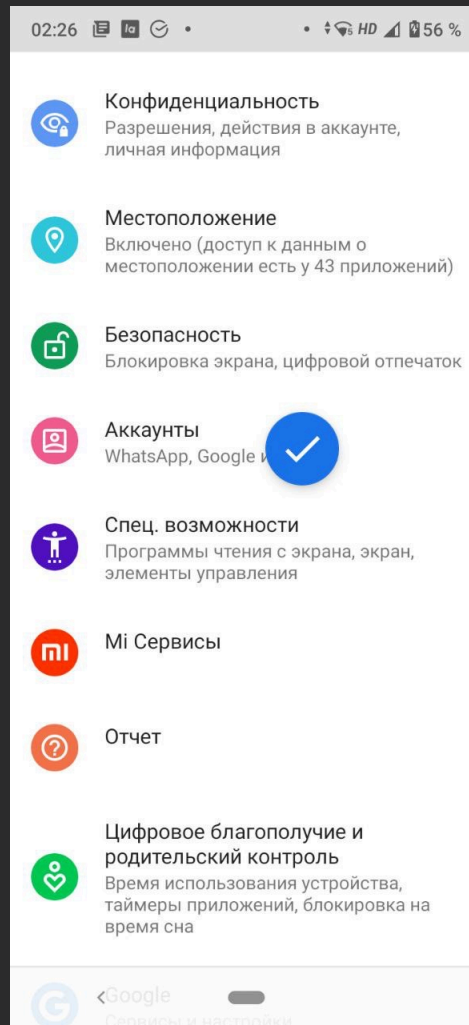
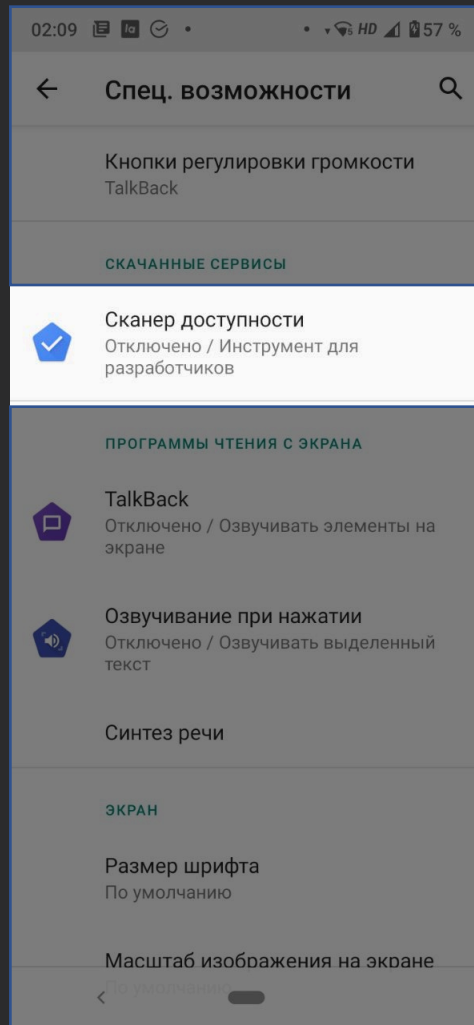

Сканер доступности

Сканер доступности

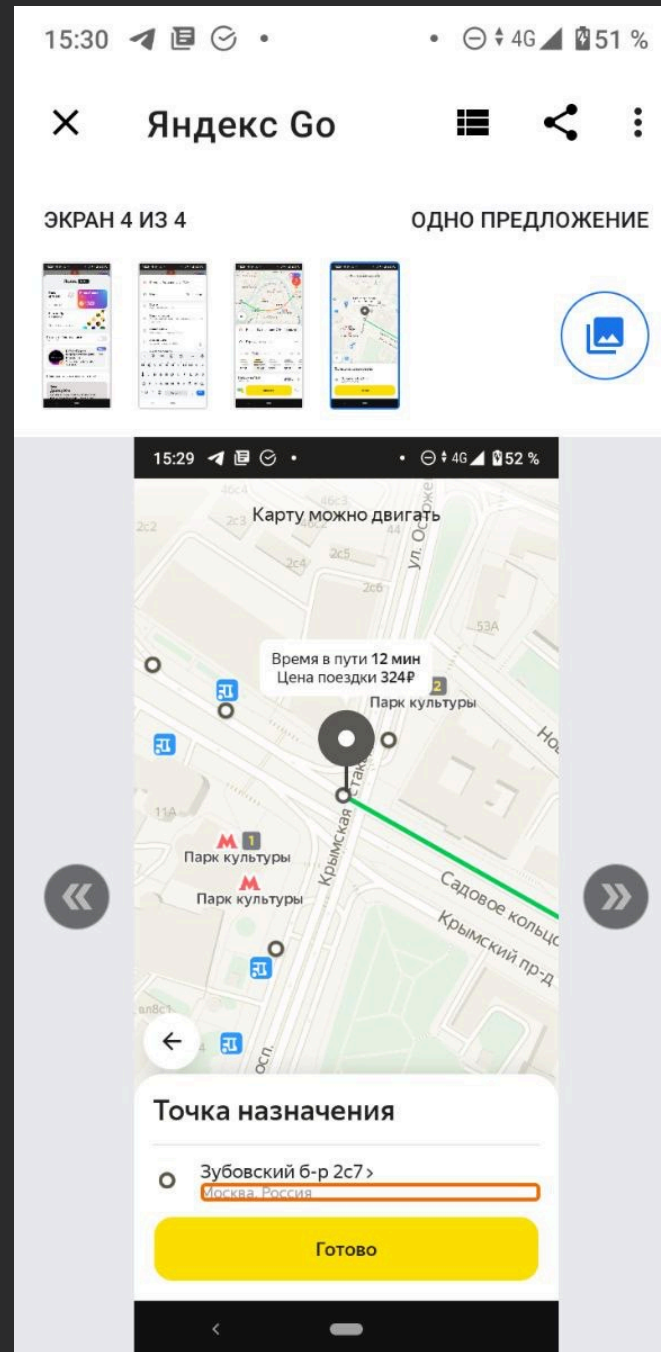
Accessibility scanner – тестирует приложения и дает рекомендации по улучшению доступности



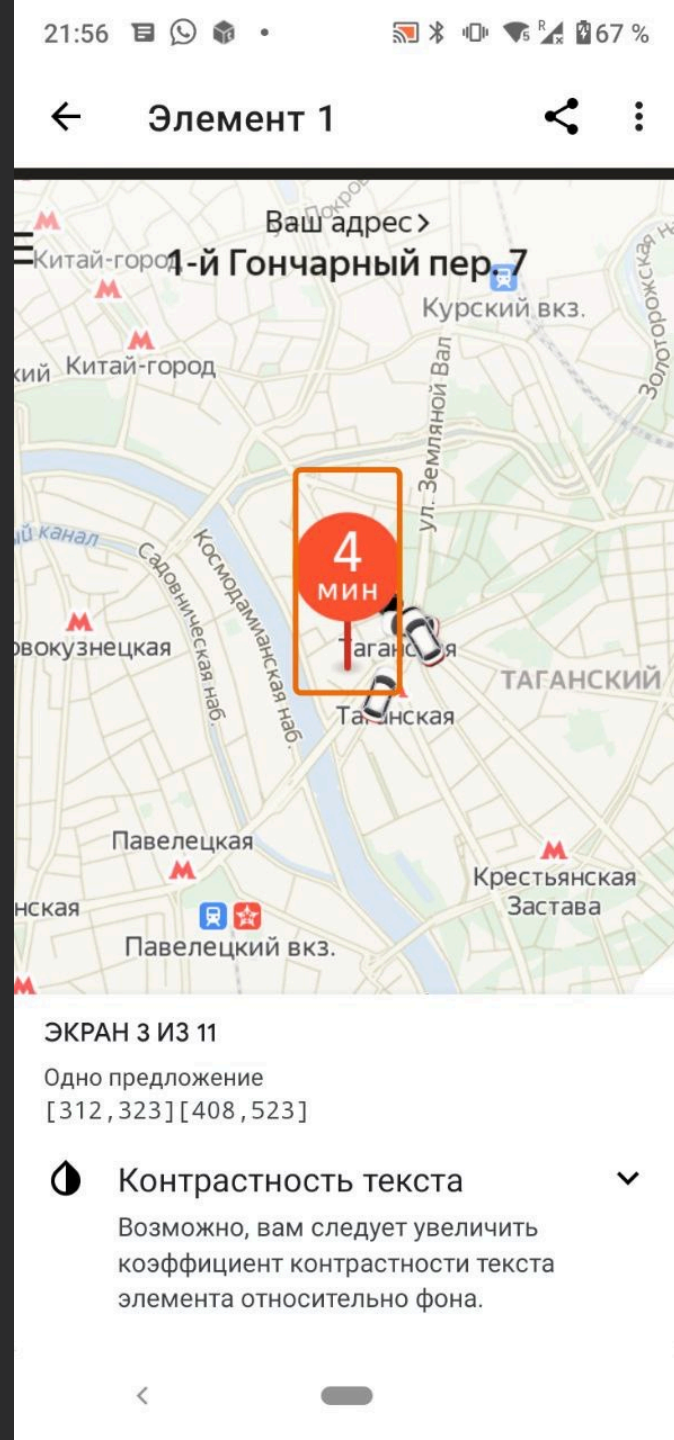
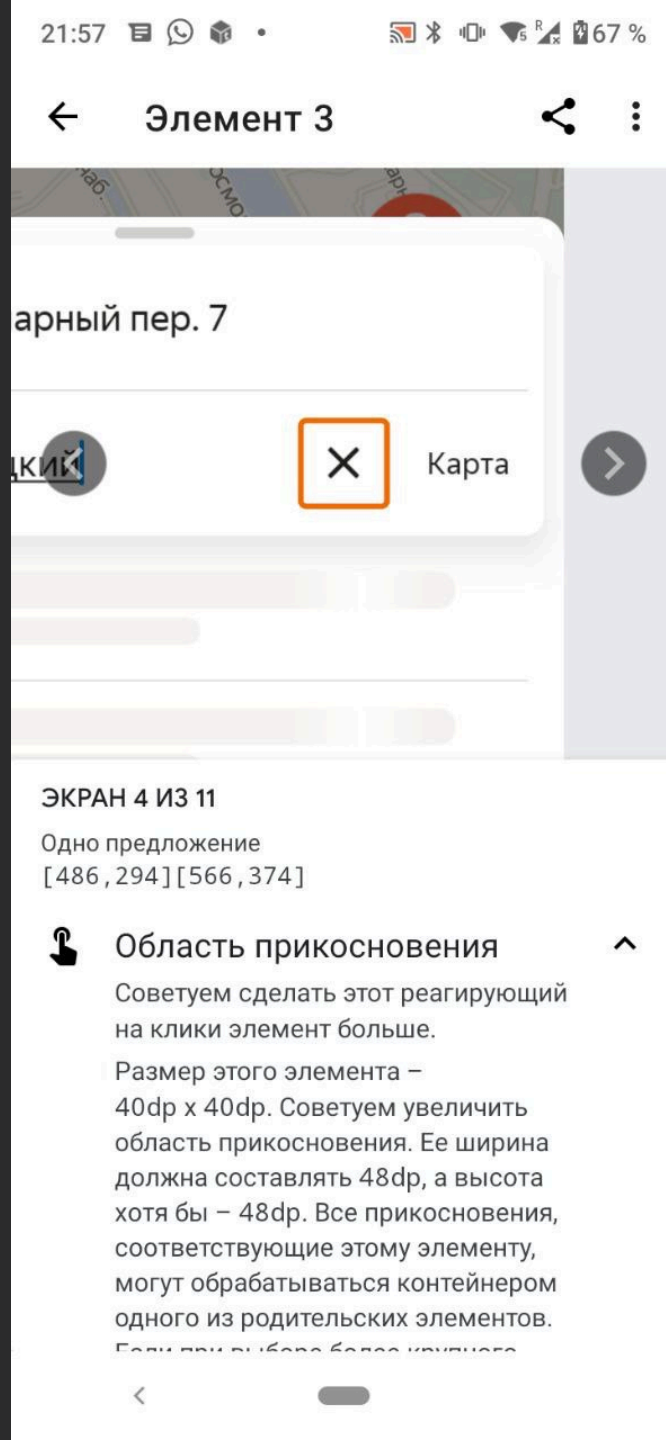
Accessibility Scanner



Список экранов



Отчеты



Опыт Яндекс Go

Дизайн система



Римская

Россия, Москва, Международная улица



Получить скидку



Arriving in 12min >

Yellow Skoda Octavia

МК582⁷⁷

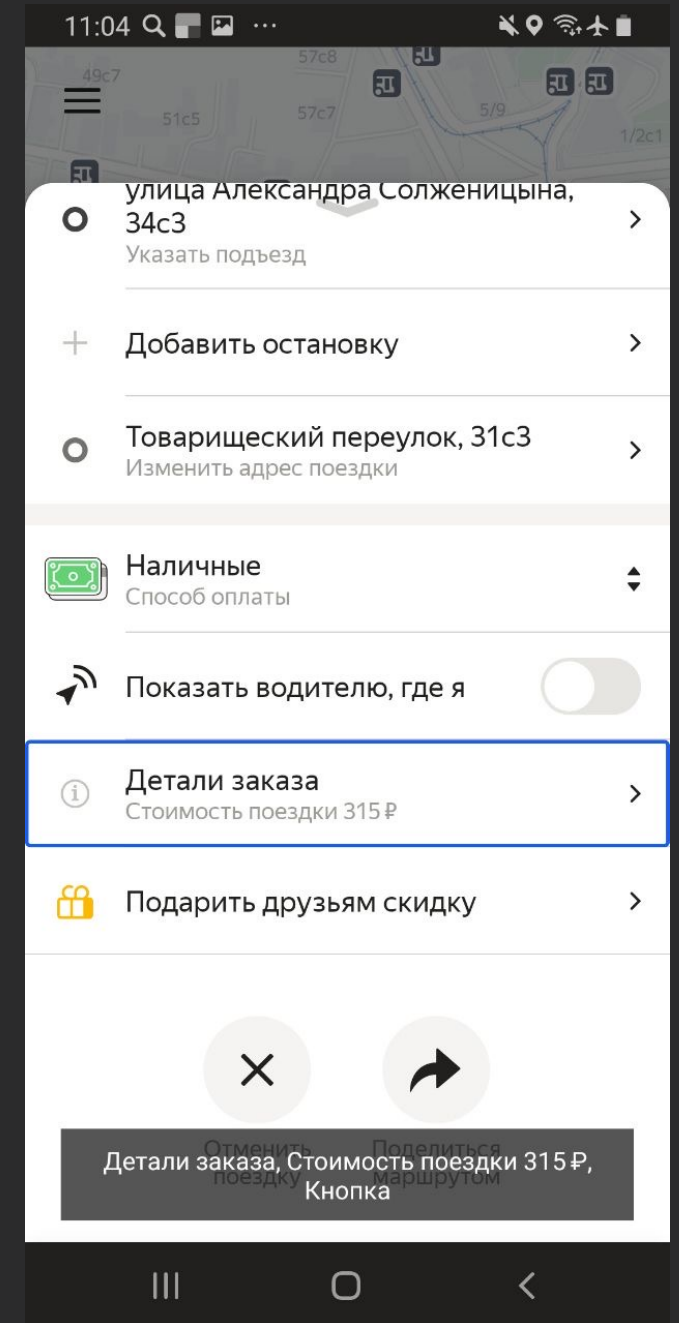
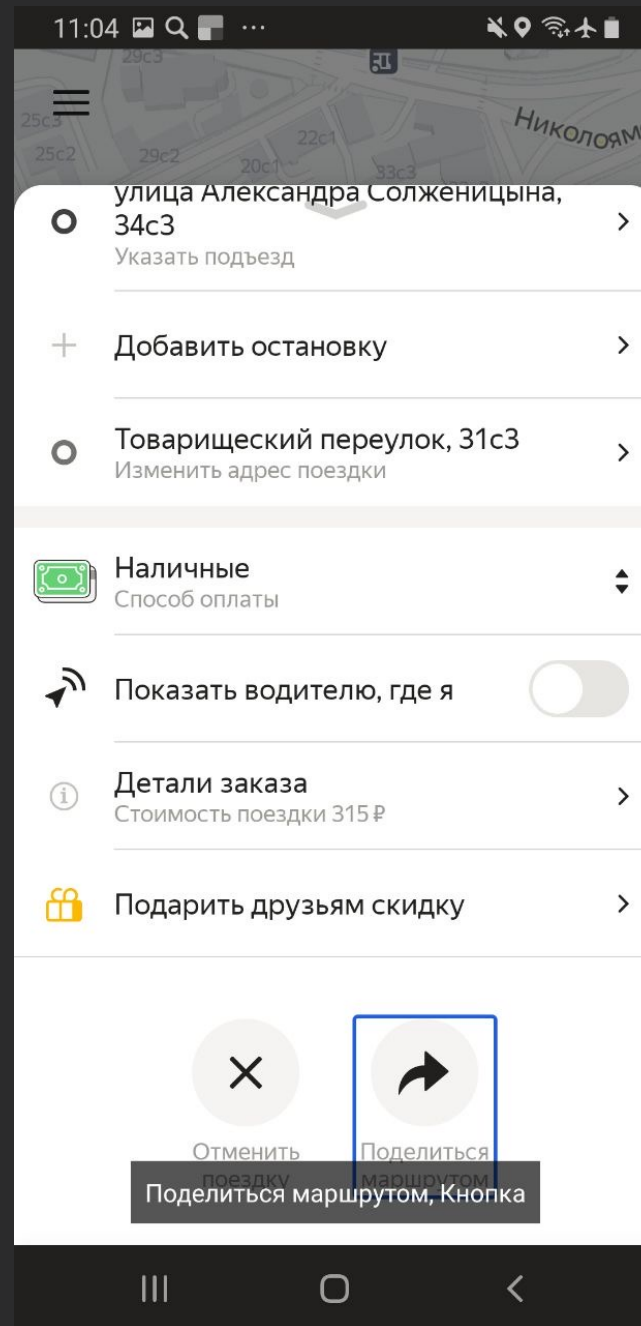


Сменить оплату

Наличные



Дизайн система



Что оказалось полезным?

Регулярное тестирование доступности

Есть эксперт по доступности в штате

Задействуем всю команду



Спасибо

Вихров Виктор

Android разработчик Яндекс Go

 victorvikhrov@yandex-team.ru

 @vvikhrov