Switch

**Switch** - ещё один вид переключателей, который появился в Android 4.0 (API 14). Находится в разделе **Widgets**. Фактически, это замена немного устаревшего **ToggleButton**. В новых проектах лучше использовать **Switch**.

Компонент используется в том случае, когда нужно переключиться на противоположное состояние - да/нет, вкл./выкл., открыть/закрыть. В реальной жизни это выглядит следующим образом.

Если нужна поддержка старых устройств (7+), то можно использовать **SwitchCompat** из библиотеки совместимости.

Компонент представляет собой полоску с двумя состояниями и сопровождающим текстом. Переключиться можно сдвиганием ползунка или касанием экрана в области компонента (не только на самом ползунке, но и на сопровождающем тексте).

По умолчанию виджет находится в выключенном состоянии. Чтобы установить его в включенном состоянии на этапе разработки, используйте атрибут **android:checked="true"**.

Сопровождающий текст задаётся в атрибуте **android:text**. А текст на самом переключателе задаётся при помощи атрибутов **android:textOn** (методы **getTextOn()** и **setTextOn()**) и**android:textOff** (методы **getTextOff()** и **setTextOff()**). Обратите внимание, что сопровождающий текст может быть очень большим и положение самого переключателя относительно этого текста можно регулировать при помощи атрибута **android:gravity** (смотри пример ниже). Если сопровождающий текст вам не нужен, то не используйте атрибут **android:text**.

Момент переключения можно отслеживать при помощи прослушки событий **setOnCheckedChangeListener**.

Зададим разметку для нескольких переключателей:

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical" >

<Switch

android:id="@+id/monitored\_switch"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dip"

android:text="Переключатель с собственным текстом"

android:textOff="Кошка"

android:textOn="Кот" />

<Switch

android:layout\_width="300dip"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dip"

android:gravity="top|left"

android:singleLine="false"

android:text="This is an example of a switch with a lot of text in it. It may end up wrapping to another line. The switch will be pinned at the top." />

<Switch

android:layout\_width="300dip"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dip"

android:gravity="center\_vertical|left"

android:singleLine="false"

android:text="This is an example of a switch with a lot of text in it. It may end up wrapping to another line. The switch will be vertically centered." />

<Switch

android:layout\_width="300dip"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dip"

android:gravity="bottom|left"

android:singleLine="false"

android:text="This is an example of a switch with a lot of text in it. It may end up wrapping to another line. The switch will be pinned at the bottom." />

<Switch

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dip"

android:text="Switch with match\_parent width" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Standalone switch below:" />

<Switch

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

</LinearLayout>

</ScrollView>

И простой код для отслеживания момента переключения у одного из переключателей:

public class MainActivity extends Activity implements CompoundButton.OnCheckedChangeListener {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Switch switch = (Switch) findViewById(R.id.monitored\_switch);

if (switch != null) {

switch.setOnCheckedChangeListener(this);

}

}

@Override

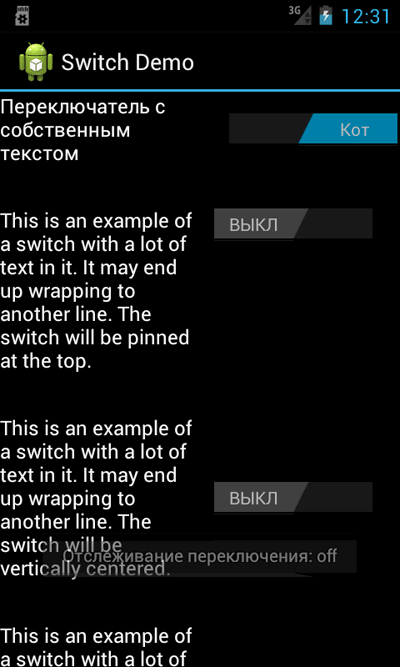
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

Toast.makeText(this, "Отслеживание переключения: " + (isChecked ? "on" : "off"),

Toast.LENGTH\_SHORT).show();

}

}



SwitchCompat

compile 'com.android.support:appcompat-v7:23.1.1'

Если вы разрабатываете приложение под Material Design, то используйте **SwitchCompat**, который специально переработали под новые требования. Сравните внешний вид двух компонентов, чтобы увидеть разницу.

...

xmlns:app="http://schemas.android.com/apk/res-auto"

...

<android.support.v7.widget.SwitchCompat

android:id="@+id/switch\_compat"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="50dp"

android:checked="false"

android:text="SwitchCompat"

android:textOff="OFF"

android:textOn="ON"

app:showText="true"/>

<Switch

android:id="@+id/switch1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/switch\_compat"

android:layout\_centerHorizontal="true"

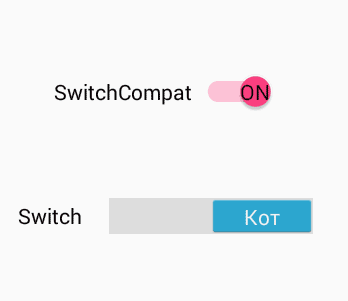
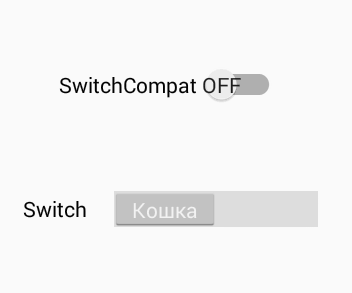
android:layout\_marginTop="55dp"

android:checked="false"

android:text="Switch"

android:textOff="Кошка"

android:textOn="Кот"/>

Как видите, новый компонент из библиотеки использует цвета темы (**colorAccent**).

Для показа текста на кнопке переключателя установите в **true** значение атрибута **app:showText**.

Не забывайте, что данный компонент можно использовать только в активностях типа **AppCompatActivity**.