



03

Основные компоненты Android: Часть 2

Nikolay Moskvina
moskvina@sibext.com

24 октября 2013 г.

1 Учимся находить информацию

2 Intent

3 View и ViewGroup

4 Android Widget

5 Собственный View

- Google!
- Stack Overflow – <http://stackoverflow.com>
- Читаем javadoc – <SDK>/docs/reference/packages.html

- **Category** – Дополнительная информация, может включать в себя несколько категорий.
 - **CATEGORY_LAUNCHER** – Activity является начальной и находится в верхнем уровне у launcher.
 - **CATEGORY_BROWSABLE** – Activity может выполнять роль обработчика ссылок в браузере (изображения, сообщения электронной почты)
 - **CATEGORY_HOME** – Activity будет отображаться на главном экране, первом экране после запуска и по кнопке Home
 - **CATEGORY_PREFERENCE** – Activity является экраном настроек
 - Описание остальных можно найти здесь
<http://developer.android.com/reference/android/content/Intent.html>

В тот момент, когда Activity получает фокус, все layout получают приказ отрисоваться.

- **draw(canvas)** и **onDraw(canvas)** – методы, которые занимаются рисованием View
- **measure(width, height)** и **onMeasure(width, height)** – первый проход, который измеряет View и его содержание на предмет изменений ширины и высоты
- **layout(l, t, r, b)** и **onLayout(l,t,r,b)** – второй проход, который размещает всех детей, используя измерения из первого прохода.

- Если View видит, что не вписывается в существующий размеры, то может вызвать у родителя **requestLayout()**
- После прохода **measure** измерения будут доступны с помощью **getMeasuredWidth()** и **getMeasuredHeight()**
- ViewGroup может вызывать **measure** несколько раз
- Если дети не договорятся между собой о том, как много места каждый из них получит, родитель будет устанавливать правила на втором проходе.

Методы set

- 1 Id
- 2 Tag
- 3 BackgroundResource
- 4 Alpha
- 5 Clickable

XML и android:

- 1 id
- 2 tag
- 3 background
- 4 alpha
- 5 clickable

Описание

- 1 уникальный идентификатор (findViewById)
- 2 возможность указывать данные, которые зависят от логики приложения
- 3 фон View из ресурса
- 4 0 – полностью прозрачная, 1 – полностью не прозрачная
- 5 возможность обработки кликов по View

Методы set

XML и android:

Описание

- | | | |
|------------------|------------------|---|
| 1 Text | 1 text | 1 текст из строки или ресурсов |
| 2 TextColor | 2 textColor | 2 цвет из класса Color или ресурсов |
| 3 SingleLine | 3 singleLine | 3 если true, то строка в одну линию, по умолчанию false |
| 4 TextAppearance | 4 textAppearance | 4 можно задать цвет, размер текста из style |
| 5 Gravity | 5 gravity | 5 выравнивание текста |

Другие атрибуты можно увидеть в XML Attributes таблице [docs/reference/android/widget/TextView.html](https://developer.android.com/reference/android/widget/TextView.html)

Методы set	XML и android:	Описание
1 ScaleType	1 scaleType	1 алгоритм расположения картинки (класс ScaleType)
2 ImageResource	2 src	2 указываем путь к ресурсу до Drawable

Другие атрибуты можно увидеть в XML Attributes таблице [docs/reference/android/widget/ImageView.html](https://developer.android.com/reference/android/widget/ImageView.html)

XML и android:

- 1 `layoutAnimation`
- 2 `addStatesFromChildren`
- 3 `animateLayoutChanges`

Описание

- 1 Задаем анимацию при первом появлении ViewGroup
- 2 Используется к примеру, когда требуется выделить группу по выделению кнопки
- 3 Задаем анимацию в случаях добавления и удаления объектов из ViewGroup

LayoutParams

- `android:layout_height`
- `android:layout_width`

Другие атрибуты можно увидеть в XML Attributes таблице
[docs/reference/android/view/ViewGroup.html](https://developer.android.com/reference/android/view/ViewGroup.html)

Методы set

- 1 Gravity
- 2 Orientation
- 3 нет

XML и android:

- 1 gravity
- 2 orientation
- 3 weightSum

Описание

- 1 Расположение содержания layout по X и Y
- 2 Горизонтальная либо вертикальная линейка
- 3 Устанавливает общий вес по умолчанию 1.0f

LayoutParams

- android:layout_weight
- android:layout_gravity

Другие атрибуты можно увидеть в XML Attributes таблице [docs/reference/android/widget/LinearLayout.html](https://developer.android.com/reference/android/widget/LinearLayout.html)

RelativeLayout

Методы set	XML и android:	Описание
1 Gravity	1 gravity	1 Расположение содержания layout по X и Y
2 IgnoreGravity	2 ignoreGravity	2 Указываем View, которые не должны изменять своё местоположение от gravity

LayoutParams

- android:layout_above
- android:layout_below
- android:layout_center*
- android:layout_align*
- android:layout_to*

Другие атрибуты можно увидеть в XML Attributes таблице [docs/reference/android/widget/RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout).

[LayoutParams.html](#)

Необходимо реализовать подкласс View, переопределив соответствующие методы:

```
public class LabelView extends View {  
    ...  
  
    @Override  
    protected void onMeasure(int widthMeasureSpec,  
        int heightMeasureSpec) {  
        setMeasuredDimension(measureWidth(widthMeasureSpec),  
            measureHeight(heightMeasureSpec));  
    }  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        canvas.drawText(text,  
            getPaddingLeft(),  
            getPaddingTop() - ascent, textPaint);  
    }  
}
```

Можно объявить собственные атрибуты для View, создав XML definition file в res/values:

```
<resources>
    ...
    <declare-styleable name="LabelView">
        <attr name="text" format="string"/>
        <attr name="textColor" format="color"/>
        <attr name="textSize" format="dimension"/>
    </declare-styleable>
</resources>
```

Использование при объявлении layout в XML:

```
<com.sibext.view.LabelView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@drawable/blue"
    app:text="Blue"
    app:textSize="20dp"/>
```

В следующей лекции

- Drawables
- Рассмотрим понятие Adapter
- Научимся обрабатывать большие данные