Лекция #13. Разметка

Графический интерфейс пользователя формируется с помощью объектов **View** (представление), **ViewGroup** (группа представлений). Класс **ViewGroup** является дочерним для класса **View**.

Класс **View** является основой для подклассов, которые называют виджетами. Виджеты – это элементы пользовательского интерфейса – текстовые поля, кнопки и т.д.

Разметка интерфейса пользователя – это процесс размещения элементов интерфейса для конкретного приложения (для конкретной деятельности). Разметку можно создать двумя способами. Первый заключается в редактировании **XML**-файла окна приложения. Второй способ – графический – это использование редактора разметки интерфейса пользователя.

XML-файл разметки

Каждый файл разметки должен содержать только один корневой элемент, который должен быть объектом **view** или **viewGroup**.

Основными контейнерами являются:

- ConstraintLayout
- FrameLayout
- LinearLayout
- TableLayout
- RelativeLayout
- GridLayout

После создания корневого элемента, можно добавить дополнительные объекты разметки или виджеты как дочерние элементы, которые определят требуемую разметку. Рассмотрим простой файл разметки «Hello, world»:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

     android:layout_width="match_parent"
     android:layout_height="match_parent"
     tools:context="ru.turbopro.hello.MainActivity">

     <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Hello World!"
         app:layout_constraintBottom_toBottomOf="parent"
         app:layout_constraintLeft_toLeftOf="parent"
         app:layout_constraintRight_toRightOf="parent"
         app:layout_constraintTop_toTopOf="parent" />

 </android.support.constraint.ConstraintLayout>
```

Элементы управления могут иметь множество атрибутов, которые определяют внешний вид и поведение элемента управления. В дальнейшем при изучении элементов управления мы постепенно будем с ними знакомиться.

Загрузка XML файла разметки из ресурсов

При компиляции приложения, каждый файл разметки (**XML-layout**-файл) компилируется в представление ресурсов в виде **R.layout.layout_имя_файла**. Загрузка выполняется путем вызова **setContentView()** в методе **onCreate()**, например:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Расположение относительно других элементов

Значением атрибута является **id** элемента, относительно которого выполнено расположение.

android:layout_above элемент располагается сверху от другого элемента

android:layout_toLeftOf элемент располагается слева от другого элемента

android:layout_toRightOf элемент располагается справа от другого элемента.

android:layout_below элемент располагается снизу от другого элемента.

android:layout_alignBaseline базовая линия элемента выравнивается по базовой линии другого элемента

android:layout_alignTop верхняя граница элемента выравнивается по верхней границе другого элемента

android:layout_alignLeft левая граница элемента выравнивается по левой границе другого элемента

android:layout_alignRight правая граница элемента выравнивается по правой границе другого элемента

android:layout_alignBottom нижняя граница элемента выравнивается по нижней границе другого элемента

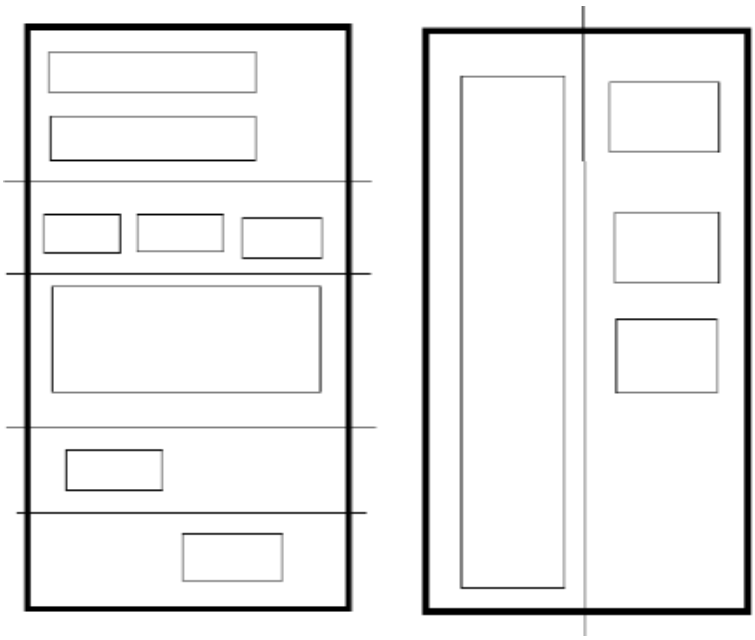
Стандартные разметки

Существуют следующие стандартные типы разметок, которые используются в создаваемых приложениях:

- ConstraintLayout (рассмотрена раннее)
- FrameLayout
- LinearLayout
- TableLayout
- RelativeLayout

Разметка ведёт себя как самостоятельный элемент управления, поэтому различные разметки можно группировать и использовать одну разметку внутри другой.

Например, наиболее часто используется сочетания **vertical** и **horizontal** линейной разметки **LinearLayout**.



FrameLayout

FrameLayout является самым простым типом разметки. Все дочерние элементы будут прикреплены к верхнему левому углу экрана, один над другим. Для создания полноценной разметки данный вариант не годится. **FrameLayout** часто используют внутри ячейки

FrameLayout удобно использовать для нескольких элементов, которые скрываются и показываются программно, таким образом, что в каждый момент времени видимым оказывается только один из них.

Для управления видимостью элемента используется атрибут **android:visibility** в **xml** или метод **setVisibility (int visibility)**, где **visibility** принимает одно из трех значений:

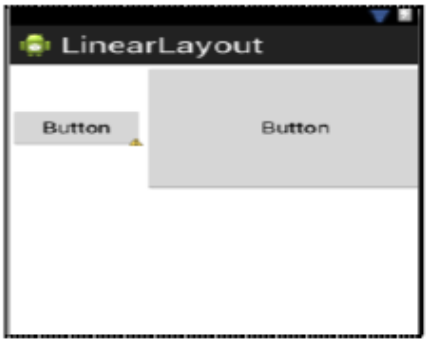
- VISIBLE — элемент видим
- INVISIBLE — элемент невидим, но все еще занимает место в разметке
- GONE — элемент невидим и не занимает место в разметке

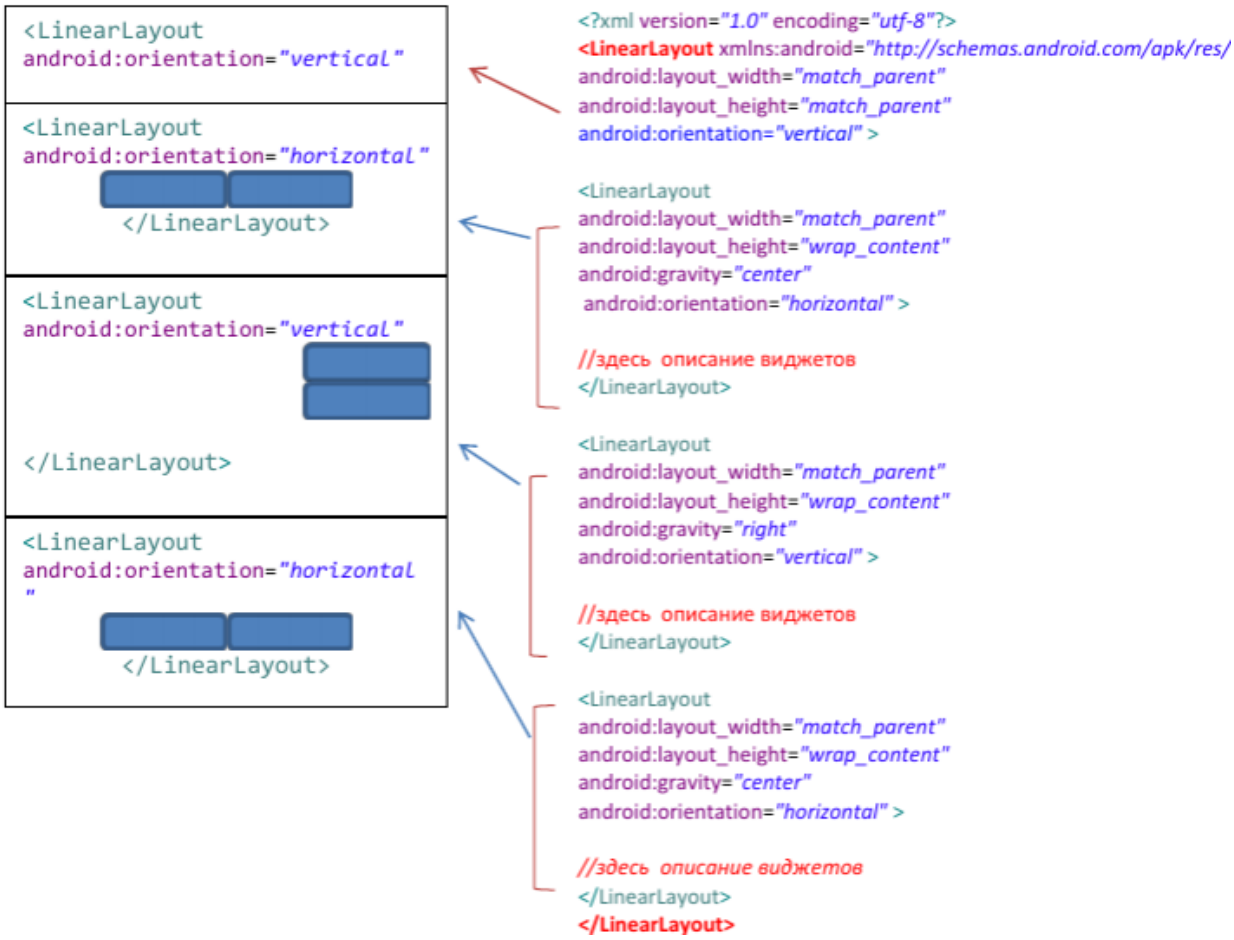
LinearLayout

Разметка для отображения одного или нескольких элементов в одну линию, горизонтально или вертикально. Для выбора ориентации используется атрибут **android:orientation** с двумя возможными значениями «**horizontal**» и «**vertical**». Вот пример, показывающий различие расположений двух кнопок в зависимости от значения атрибута **android:orientation**.

`android:orientation="vertical"`

`android:orientation="horizontal"`





TableLayout

TableLayout это разметка для расположения элементов в виде таблицы. Ряды задаются в **xml** с помощью тега **TableRow**, а ячейки в каждом ряду создаются автоматически для каждого элемента. Количество колонок в таблице будет равно максимальному количеству элементов в рядах. То есть, если в одном ряду 3 элемента, а во втором 1, то колонок в **TableLayout** будет 3. Ширина колонки определяется по самому широкому элементу в ней. Ячейки таблицы можно оставлять пустыми, или объединять. Атрибут **android:layout_column** — задает в какой столбец поместить данный элемент (нумерация столбцов начинается с 0).

Атрибут **android:layout_span**— позволяет объединить указанное количество столбцов (**ряды объединить нельзя**).

RelativeLayout

Разметка для расположения элементов относительно родителя или друг друга. Элементы начинают располагаться в указанном порядке, поэтому необходимо чтобы элемент был описан до того, как другой элемент будет на него ссылаться.

Атрибут	Описание
android:layout_above	Располагает элемент над элементом с указанным ID
android:layout_alignBottom	Выравнивает нижнюю границу элемента относительно нижней границы другого элемента с указанным ID
android:layout_alignLeft	Выравнивает левую границу элемента относительно левой границы другого элемента с указанным ID
android:layout_alignParentBottom	Если атрибут имеет значение true, то его нижняя граница соответствует нижней границе родительского контейнера
android:layout_alignParentRight	Если атрибут имеет значение true, то его правая граница соответствует правой границе родительского контейнера
android:layout_alignParentTop	Если атрибут имеет значение true, то его верхняя граница соответствует верхней границе родительского контейнера
android:layout_below	Располагает элемент под элементом с указанным ID
android:layout_centerInParent	Если атрибут имеет значение true, то элемент располагается по центру родительского контейнера
android:layout_toLeftOf	Правая граница элемента проходит по левой границе другого элемента с указанным ID
android:layout_toRightOf	Левая граница элемента проходит по правой границе другого элемента с указанным ID