



## Лекция #14. GridLayout

В **Android 4** появился новый вид макета под именем **GridLayout**, который имеет такую же разметку как **TableLayout** - ряды, колонки и клетки, но гораздо удобнее и функциональнее: элементы могут занимать несколько клеток.

### Автоматическое позиционирование элементов

Количество строк и столбцов устанавливают атрибуты

- **android:rowCount** = "число" - количество строк
- **android:columnCount** = "число" - количество столбцов

На практике количество строк явно не указывается. В этом случае **Android** автоматически создаст столько строк, сколько потребуется для отображения в таблице всех элементов. Если установить данные атрибуты количества строк и столбцов в общем описании

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:rowCount="2"
    android:columnCount="2">
```

то, **GridLayout** автоматически может позиционировать вложенные элементы управления по строкам. Элементы размещаются в макете в порядке следования в **XML** файле: первый элемент будет помещен в первую ячейку, второй во вторую ячейку и так далее. В нашем случае 2 строки и 2 колонки – четыре ячейки. При добавлении, например, 4 **Button** они займут первую, вторую, третью и четвертую ячейки. При этом ширина столбцов устанавливается автоматически по ширине самого широкого элемента.

Для элементов, размещаемых в **GridLayout**, не нужно указывать атрибуты **layout\_height** и **layout\_width**, по умолчанию они устанавливаются в значение **wrap\_content**.

Основными инструментами, которые придают гибкость работе с **GridLayout** является выравнивание и **layout\_gavity** (гравитации) компонентов в столбцах.

Значения для **layout\_gravity** могут быть:

- Top
- bottom
- left
- right
- center\_horizontal
- center
- fill\_horizontal
- fill
- center\_vertical
- fill\_vertival
- start
- end

▼ layout_gravity	[center]
top	<input type="checkbox"/>
bottom	<input type="checkbox"/>
left	<input type="checkbox"/>
right	<input type="checkbox"/>
center_vertical	<input type="checkbox"/>
fill_vertical	<input type="checkbox"/>
center_horizontal	<input type="checkbox"/>
fill_horizontal	<input type="checkbox"/>
center	<input checked="" type="checkbox"/>
fill	<input type="checkbox"/>
clip_vertical	<input type="checkbox"/>
clip_horizontal	<input type="checkbox"/>
start	<input type="checkbox"/>
end	<input type="checkbox"/>

## Создание табличного макета GridLayout

У автоматического размещения элементов имеется серьезный недостаток: удаление одного элемента из макета может привести к недопустимому изменению внешнего вида макета. Для того чтобы избежать подобных проблем используется явное указание, где должны находиться элементы.

Для этого используются следующие атрибуты:

**android:layout\_column**="номер столбца" (отсчет идет от нуля)

**android:layout\_row**= "номер строки"

**android:layout\_columnSpan**="кол-ство столбцов", которые занимает элемент

**android:layout\_rowSpan** = "кол-ство строк", которые занимает элемент

## Создание макета калькулятора

Общий вид представлен ниже



Макет состоит из 4 столбцов и 6 строк. Строки будут создаваться автоматически, поэтому мы их указывать не будем.

Поэтому заголовок может быть записан так

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:columnCount="4" >
```

В первой строке содержится **TextView**, который занимает все 4 столбца. Вот его **XML** описание.

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_columnSpan="4"
    android:text="00"
    android:textAlignment="textEnd"
    android:textSize="26sp"/>
```

Атрибут **android:layout\_columnSpan="4"** заставляет **TextView** занять все 4 столбца. Выравнивание текста в **TextView** по правому краю обеспечивает атрибут **android:textAlignment="textEnd"**. Размер шрифта равен **26sp**.

Далее идет вторая строка. Она будет создана автоматически при добавлении элементов. В этой строке первые две ячейки пустые, в третьей расположен символ «С», в четвертой символ «/». Вот **XML** код.

```
<Button
    android:layout_column="2"
    android:id="@+id/Button111"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="С"
/>
<Button
    android:layout_column="3"
    android:id="@+id/Button112"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="/" />
```

Далее идут кнопки. Вот часть **XML** кода

```
<Button
    android:id="@+id/Button01"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="7" />
<Button
    android:id="@+id/Button02"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="8" />
<Button
    android:id="@+id/Button03"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="9" />
<Button
    android:id="@+id/Button04"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="-" />
```

В четвертой строке располагается кнопка «+», которая занимает две строки. Начало кнопки идет в третьей строке после кнопки «3». Вот код.

```
<Button
    android:id="@+id/Button10"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:text="3"/> кнопка 3
<Button
    android:id="@+id/Button12"
    android:layout_height="wrap_content"
    android:layout_width="50dp"
    android:layout_rowSpan="2"
    android:layout_gravity="fill"
    android:text="+"/>
```

Атрибут **android:layout\_rowSpan="2"** указывает что кнопка занимает две строки, а атрибут **android:layout\_gravity="fill"** заставляет кнопку полностью заполнить две ячейки. Расположение кнопок «=» и «0» выполните самостоятельно.