TableLayout

Последнее обновление:

Контейнер **TableLayout** структурирует элементы управления в виде таблицы по столбцам и строкам. Определим в файле **activity_main.xml** элемент TableLayout, который будет включать две строки и два столбца:

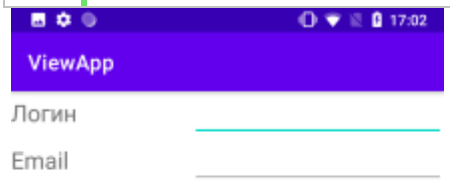
```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TableRow>
        <TextView
            android:layout_weight="0.5"
            android:text="Логин"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <EditText
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"  />
    </TableRow>

    <TableRow>
        <TextView
            android:layout_weight="0.5"
```

```
            android:text="Email"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <EditText
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>
</TableLayout>
```



Используя элемент **TableRow**, мы создаем отдельную строку. Как разметка узнает сколько столбцов надо создать? Android находит строку с максимальным количеством виджетов одного уровня, и это количество будет означать количество столбцов. Например, в данном случае у нас определены две строки и в каждой по два элемента. Если бы в какой-нибудь из них было бы три виджета, то соответственно столбцов было бы также три, даже если в другой строке осталось бы два виджета.

Причем элемент TableRow наследуется от класса LinearLayout, поэтому мы можем к нему применять тот же функционал, что и к LinearLayout. В частности, для определения пространства для элементов в строке используется атрибут `android:layout_weight`.

Если какой-то элемент должен быть растянут на ряд столбцов, то мы можем растянуть его с помощью атрибута **layout_span**, который указывает на какое количество столбцов надо растянуть элемент:

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <TableRow>
        <TextView
            android:textSize="22sp"
            android:text="Логин"
            android:layout_width="100dp"
            android:layout_height="wrap_content" />

        <EditText
            android:textSize="22sp"
            android:layout_width="200dp"
            android:layout_height="wrap_content"  />
    </TableRow>

    <TableRow>
        <TextView
            android:textSize="22sp"
            android:text="Email"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <EditText
            android:textSize="22sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>
    <TableRow>
        <Button
            android:text="Отправить"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="2"/>
    </TableRow>
</TableLayout>
```

Также можно растянуть элемент на всю строку, установив у него атрибут `android:layout_weight="1"`:

```
<TableRow>
    <Button
        android:text="Отправить"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />
</TableRow>
```

Программное создание TableLayout

Создадим TableLayout программным образом, переложив на код java самый первый пример из данной статьи:

```
package com.example.viewapp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
14          super.onCreate(savedInstanceState);
15
16          TableLayout tableLayout = new TableLayout( this);
17
18          // первая строка
19          TableRow tableRow1 = new TableRow(this);
20
21          TextView textView1 = new TextView(this);
22          textView1.setText("Логин");
23          tableRow1.addView(textView1, new TableRow.LayoutParams(
24                  TableRow.LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WI
25
26          EditText editText1 = new EditText(this);
27          tableRow1.addView(editText1, new TableRow.LayoutParams(
28                  TableRow.LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WI
29
30          // вторая строка
31          TableRow tableRow2 = new TableRow(this);
32
33          TextView textView2 = new TextView(this);
34          textView2.setText("Email");
35          tableRow2.addView(textView2, new TableRow.LayoutParams(
36                  TableRow.LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WI
37
38          EditText editText2 = new EditText(this);
39          tableRow2.addView(editText2, new TableRow.LayoutParams(
40                  TableRow.LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WI
41
42          tableLayout.addView(tableRow1);
43          tableLayout.addView(tableRow2);
44          setContentView(tableLayout);
45      }
46  }
```