**Обзор TableLayout 2**

Продолжаем говорить о контейнере компоновки TableLayout.  
В этом обзоре мы разберём атрибуты android:shrinkColumns, android:collapseColumns и напишем программное добавление строк таблицы.

**Атрибут android:shrinkColumns**

В предыдущем обзоре TableLayout, мы рассмотрели атрибут android:stretchColumns, который отвечал за то, что элементы будут растянуты по всей длине, но что делать если содержимое слишком длинное и занимает всю ширину? Решение есть! Необходимо установить атрибутandroid:shrinkColumns, чтобы переносить содержимое столбца или столбцов, если для других столбцов нужно больше места.

Начнём с примера. Возьмём за основу первый пример из предыдущего поста и изменим немного его, добавив новый атрибут и длинное название для первого и второго столбцов.

**Пример 1**

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

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

**android:background**=**"#eeeeee"**

**android:stretchColumns**=**"\*"**

**android:shrinkColumns**=**"0,1"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**>

<**TableRow** **android:background**=**"#cccccc"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Дата заказа"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceLarge"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Количество заказов"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceLarge"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Сумма"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceLarge"**

/>

</**TableRow**>

<**TableRow** **android:padding**=**"5dp"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"22 ноября"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"397"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"19039.21"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

</**TableRow**>

<**TableRow** **android:padding**=**"5dp"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"23 ноября"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"348"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"16759.28"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

</**TableRow**>

<**TableRow** **android:padding**=**"5dp"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"24 ноября"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"305"**

**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"14527.19"**

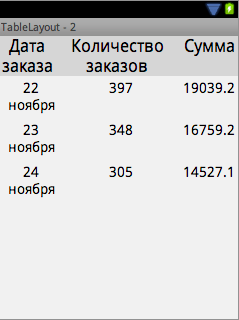
**android:gravity**=**"center\_horizontal"**

**android:textAppearance**=**"?android:attr/textAppearanceMedium"**

/>

</**TableRow**>

</**TableLayout**>



Как видно из скриншота, первый и второй столбец перенесли содержимое на следующую строку, а задали мы такое поведение в теге TableLayout

android:shrinkColumns="0,1"

Индекс столбцов считается с нуля. Чтобы задать перенос для всех столбцов, можно использовать значение звёздочка

android:shrinkColumns=**"\*"**

Программно на JAVA данный атрибут можно задать следующими методами:

tableLayout.setShrinkAllColumns(**true**); *// для всех столбцов (аналог android:shrinkColumns="\*")*

*// или (аналог android:shrinkColumns="0,1")*

tableLayout.setColumnShrinkable(0, **true**); *// для первого столбца*

tableLayout.setColumnShrinkable(1, **true**); *// для второго столбца*

**Атрибут android:collapseColumns**

Данный атрибут позволяет скрыть выбранные столбцы. Через запятую указываются порядковый номер столбца, начиная с нуля.  
Рассмотрим пример:

**Пример 2**

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

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

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:collapseColumns**=**"1,3,4"**

>

<**TableRow**>

<**Button** **android:text**=**"0"**/>

<**Button** **android:text**=**"1"**/>

<**Button** **android:text**=**"2"**/>

<**Button** **android:text**=**"3"**/>

<**Button** **android:text**=**"4"**/>

<**Button** **android:text**=**"5"**/>

<**Button** **android:text**=**"6"**/>

<**Button** **android:text**=**"7"**/>

</**TableRow**>

<**TableRow**>

<**Button** **android:text**=**"0"**/>

<**Button** **android:text**=**"1"**/>

<**Button** **android:text**=**"2"**/>

<**Button** **android:text**=**"3"**/>

<**Button** **android:text**=**"4"**/>

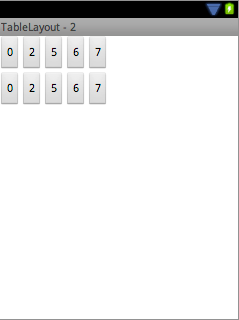
<**Button** **android:text**=**"5"**/>

<**Button** **android:text**=**"6"**/>

<**Button** **android:text**=**"7"**/>

</**TableRow**>

</**TableLayout**>



В коде мы разместили подряд в строку 8 кнопок (получилось 8 столбцов) и затем скрыли 3 столбца под индексами 1, 3 и 4, после чего, как видно из скриншота, у нас осталось 5 кнопок.

android:collapseColumns="1,3,4"

Программно на JAVA это можно сделать методом setColumnCollapsed(int index, boolean isCollapseColumns), вызванным для контейнера TableLayout.

tableLayout.setColumnCollapsed(1, **true**);

tableLayout.setColumnCollapsed(3, **true**);

tableLayout.setColumnCollapsed(4, **true**);

**Программное добавление строк TableRow в TableLayout**

Рассмотрим, как можно программно добавлять новые строки в таблицу.  
Создадим таблицу и простую форму для добавления строк. Для формы мы также воспользуемся табличной вёрсткой.

**Пример 3 - XML макет**

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

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**>

<**LinearLayout**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"fill\_parent"**

**android:orientation**=**"vertical"**

**android:background**=**"#fff"**>

<**TableLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:stretchColumns**=**"\*"**

**android:id**=**"@+id/table"**>

<**TableRow** **android:id**=**"@+id/header"**

**android:padding**=**"5dp"**

**android:background**=**"#ccc"**>

<**TextView** **android:text**=**"Пользователь"**

**android:layout\_gravity**=**"center"**

/>

<**TextView** **android:text**=**"Кол-во баллов"**

**android:layout\_gravity**=**"center"**

/>

</**TableRow**>

</**TableLayout**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:textSize**=**"16sp"**

**android:textStyle**=**"bold"**

**android:layout\_marginTop**=**"30dp"**

**android:layout\_marginBottom**=**"10dp"**

**android:gravity**=**"center\_horizontal"**

**android:text**=**"Добавить запись:"**/>

<**TableLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:stretchColumns**=**"\*"**>

<**TableRow**>

<**TextView** **android:text**=**"Имя пользователя:"**

**android:layout\_marginLeft**=**"10dp"**/>

<**EditText** **android:id**=**"@+id/add\_user"**

**android:layout\_marginRight**=**"10dp"**/>

</**TableRow**>

<**TableRow**>

<**TextView** **android:text**=**"Кол-во баллов:"**

**android:layout\_marginLeft**=**"10dp"**/>

<**EditText** **android:id**=**"@+id/add\_point"**

**android:inputType**=**"number"**

**android:layout\_marginRight**=**"10dp"**/>

</**TableRow**>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Добавить"**

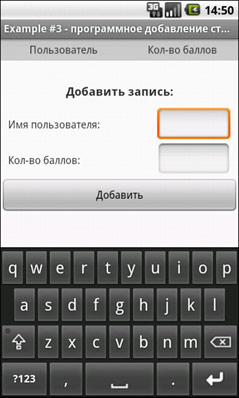
**android:id**=**"@+id/add\_button"**/>

</**TableLayout**>

</**LinearLayout**>

</**ScrollView**>

Вот, что у нас получилось:



Как видно из кода,  мы создали таблицу с id *table*и теперь нам нужно её заполнить с помощью формы под таблицей.  
Для этого напишем JAVA код, обрабатывающий нажатие кнопки добавить.

**public** **class** TableActivity **extends** Activity {

**private** TableLayout table;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.table\_ex3);

setTitle(**"Example #3 - программное добавление строк"**);

Button buttonAdd = (Button) findViewById(R.id.add\_button); *// находим кнопку в xml макете*

table = (TableLayout) findViewById(R.id.table); *// находим таблицу для заполнения в xml макете*

buttonAdd.setOnClickListener(**new** View.OnClickListener() { *// обрабатываем нажатие на кнопку*

**public** **void** onClick(View view) {

*// находим текстовое поле для добавления пользователя в xml макете*

EditText addUser = (EditText) findViewById(R.id.add\_user);

*// находим текстовое поле для добавления баллов в xml макете*

EditText addPoint = (EditText) findViewById(R.id.add\_point);

String addUserValue = addUser.getText().toString(); *// получаем значение поля добавления пользователя*

String addPointValue = addPoint.getText().toString(); *// получаем значение поля добавления баллов*

**if** (addUserValue.equals(**""**) || addPointValue.equals(**""**)) { *// проверяем заполненность полей*

Toast.makeText(getApplicationContext(), **"Не заполнены все поля!"**, Toast.LENGTH\_SHORT).show();

} **else** {

*// создаём текстовое предстввление с введённым именем пользователя*

TextView textUser = **new** TextView(getApplicationContext());

textUser.setGravity(Gravity.CENTER\_HORIZONTAL);

textUser.setTextColor(0xff000000);

textUser.setText(addUserValue);

*// создаём текстовое предстввление с введённым кол-вом баллов*

TextView textPoint = **new** TextView(getApplicationContext());

textPoint.setGravity(Gravity.CENTER\_HORIZONTAL);

textPoint.setTextColor(0xff000000);

textPoint.setText(addPointValue);

*// создаём строку для таблицы*

TableRow row = **new** TableRow(getApplicationContext());

row.addView(textUser); *// добавляем в строку столбец с именем пользователя*

row.addView(textPoint); *// добавляем в строку столбец с кол-вом баллов*

addUser.setText(**""**); *// очищаем значения в полях ввода*

addPoint.setText(**""**); *// очищаем значения в полях ввода*

table.addView(row); *// добавляем в таблицу новую строку*

}

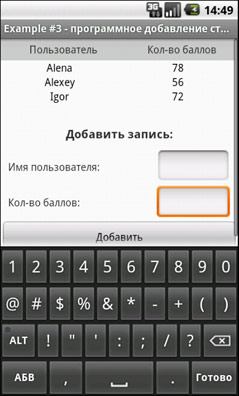
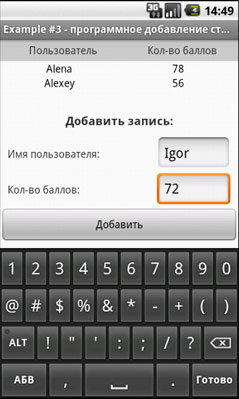
}

});

}

}

Код снабжён довольно подробными комментариями. Отдельно работу пояснять не буду.  
Приложу лишь скриншоты с результатами добавлений.



Исходный код примеров из статьи на гитхабе - <https://github.com/devpad/Android-TableLayout-2>