# 4. Material Design, Support library, Анимации.

## Material Design

В Android 5.0 появился Material Design.

Основные принципы - расположение контента по z плоскости, тени, простота. Всем изучить <https://material.io/guidelines/material-design/introduction.html>

В реалиях аутсорсинга не все состояния дизайна отрисовываются. Это не значит, что их не нужно делать. Всегда помните что:

* Обращение к сети должно как-то отображаться в интерфейсе (прогресс-бар). Кроме случаев, когда юзеру не важен результат (лайк записи).
* Любая неудачная операция с сетью должна обрабатываться и не приводить к тупиковым ситуациям в приложении. Например, если поставили лайк и сразу его закрасили, а запрос не прошел, то нужно вернуть прошлое состояние кнопки лайка.
* Пользователь не должен оказываться в ситуации, когда он не может повторить загрузку данных, если первоначальная загрузка не прошла.
* Когда данных нет нужно обязательно показывать вью с соответствующей информацией. Не допускать пустых экранов.
* Интерфейс должен быть отзывчивым, т.е. реагировать на нажатия. Никаких деревянных интерфейсов!

Стоит иметь ввиду, что у типичных экранов с данными обычно 4 состояния:  
1. Данные загружаются

2. Данный загружены и отображаются

3. Данных нет, пустое состояние

4. Ошибка загрузки данных

Для совместимости гугл выпустила app compat design library. Обзор основных лейаутов

Пройтись по основным лейаутам

<https://www.sitepoint.com/material-design-android-design-support-library/>

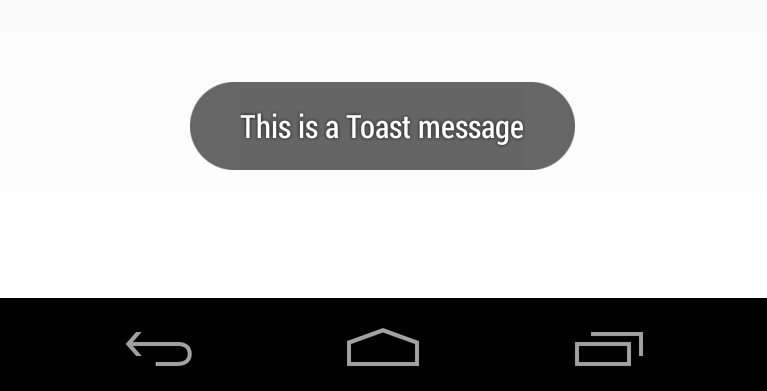
* BottomNavigation <https://material.io/design/components/bottom-navigation.html>

## Оповещение о событиях

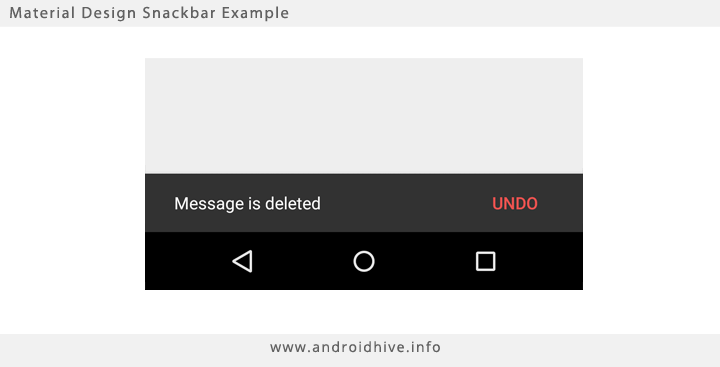
Alert dialog

Progress dialog

Toast



## Snackbar



## 

## RecyclerView

Виджет [RecyclerView](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html?hl=ru) служит для построения списков.

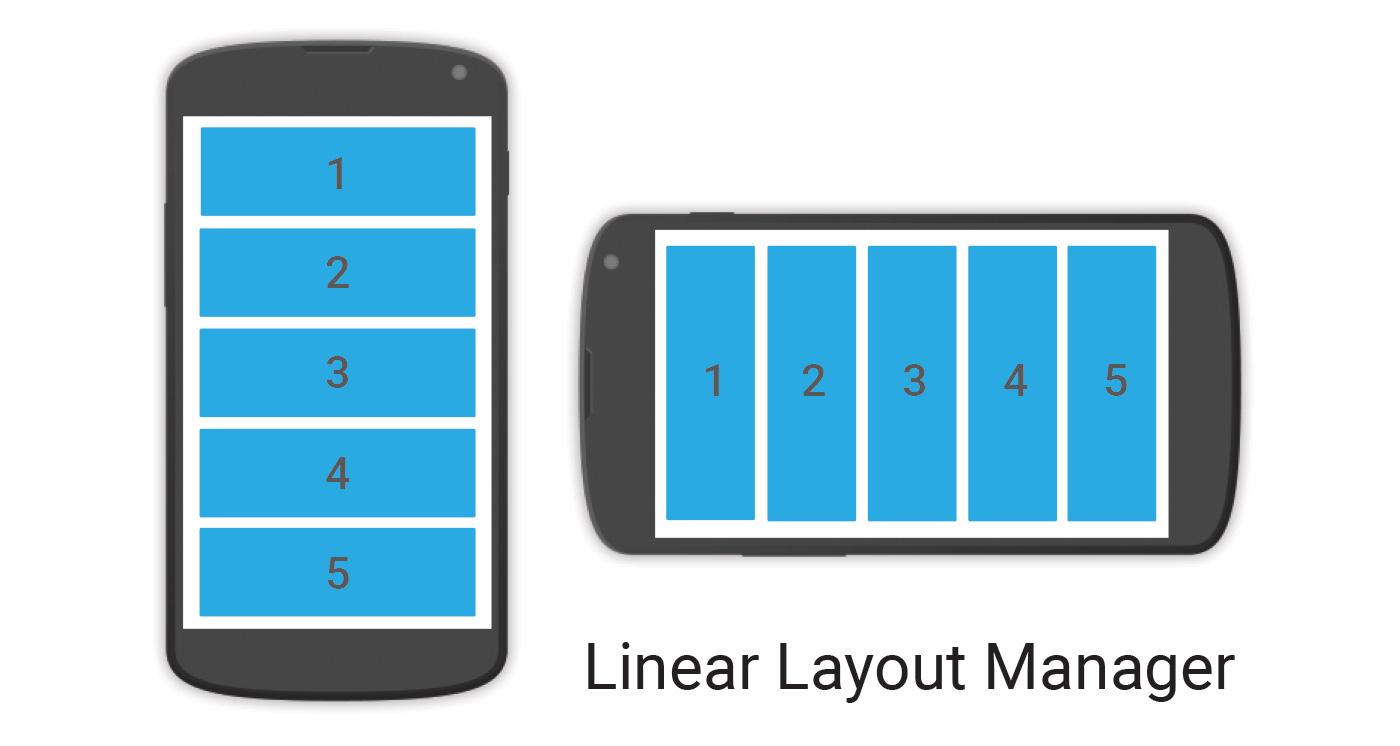
Объяснить почему такое название

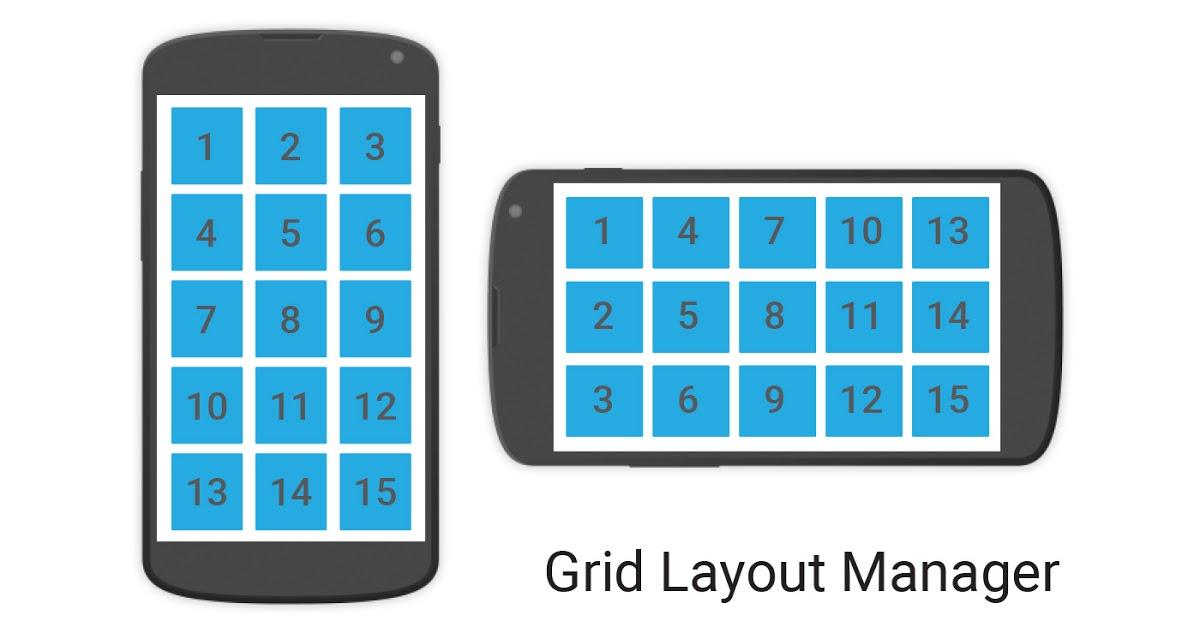
Нарисовать схему уничтожения вьюх (onCreateViewHoler, onBindViewHolder).

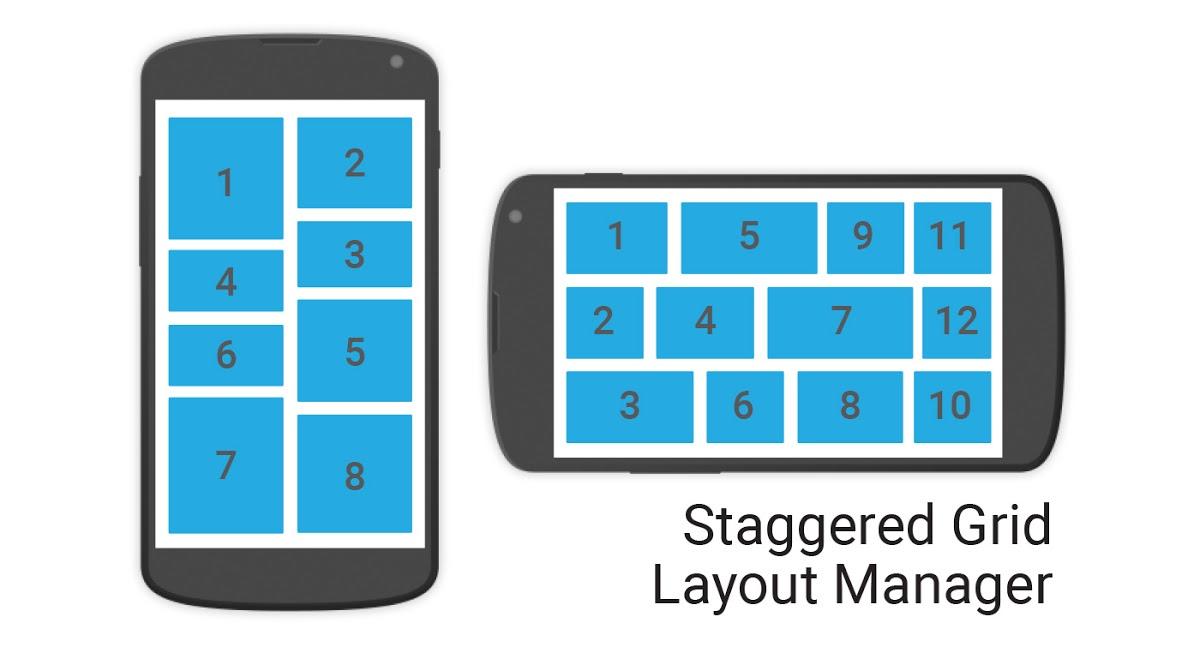
Основные элементы, необходимые для работы RecyclerView



Основные LayoutManager ы:

* [LinearLayoutManager](https://developer.android.com/reference/android/support/v7/widget/LinearLayoutManager.html?hl=ru) для отображения элементов в виде списка с вертикальной или горизонтальной прокруткой;  
  RecyclerView rvSomeThing = ...  
  rvSomeThing.setLayoutManager(**new** LinearLayoutManager(**this**, LinearLayoutManager.HORIZONTAL, **false**));  
  ...
* 
* [GridLayoutManager](https://developer.android.com/reference/android/support/v7/widget/GridLayoutManager.html?hl=ru) для отображения элементов в виде сетки;  
  RecyclerView rvSomeThing = ...  
  rvSomeThing.setLayoutManager(**new** GridLayoutManager(**this**, 3));



* [StaggeredGridLayoutManager](https://developer.android.com/reference/android/support/v7/widget/StaggeredGridLayoutManager.html?hl=ru) для отображения элементов в виде шахматной сетки  
  

### Анимация в списках

По умолчанию анимация для добавления и удаления элементов включается в виджете [RecyclerView](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html?hl=ru). Для настройки такой анимации следует наследовать класс [RecyclerView.ItemAnimator](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.ItemAnimator.html?hl=ru) и воспользоваться методом [RecyclerView.setItemAnimator()](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html?hl=ru#setItemAnimator(android.support.v7.widget.RecyclerView.ItemAnimator)) .

### Пример (удобнее мотреть тут): <https://developer.android.com/training/material/lists-cards.html?hl=ru> ):

<!-- A RecyclerView with some commonly used attributes -->

<android.support.v7.widget.RecyclerView

android:id="@+id/my\_recycler\_view"

android:scrollbars="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

public class MyActivity extends Activity {

private RecyclerView mRecyclerView;

private RecyclerView.Adapter mAdapter;

private RecyclerView.LayoutManager mLayoutManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.my\_activity);

mRecyclerView = (RecyclerView) findViewById(R.id.my\_recycler\_view);

// use this setting to improve performance if you know that changes

// in content do not change the layout size of the RecyclerView

mRecyclerView.setHasFixedSize(true);

// use a linear layout manager

mLayoutManager = new LinearLayoutManager(this);

mRecyclerView.setLayoutManager(mLayoutManager);

// specify an adapter (see also next example)

mAdapter = new MyAdapter(myDataset);

mRecyclerView.setAdapter(mAdapter);

}

...

}

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {

private String[] mDataset;

// Provide a reference to the views for each data item

// Complex data items may need more than one view per item, and

// you provide access to all the views for a data item in a view holder

public static class ViewHolder extends RecyclerView.ViewHolder {

// each data item is just a string in this case

public TextView mTextView;

public ViewHolder(TextView v) {

super(v);

mTextView = v;

}

}

// Provide a suitable constructor (depends on the kind of dataset)

public MyAdapter(String[] myDataset) {

mDataset = myDataset;

}

// Create new views (invoked by the layout manager)

@Override

public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,

int viewType) {

// create a new view

View v = LayoutInflater.from(parent.getContext())

.inflate(R.layout.my\_text\_view, parent, false);

// set the view's size, margins, paddings and layout parameters

...

ViewHolder vh = new ViewHolder(v);

return vh;

}

// Replace the contents of a view (invoked by the layout manager)

@Override

public void onBindViewHolder(ViewHolder holder, int position) {

// - get element from your dataset at this position

// - replace the contents of the view with that element

holder.mTextView.setText(mDataset[position]);

}

// Return the size of your dataset (invoked by the layout manager)

@Override

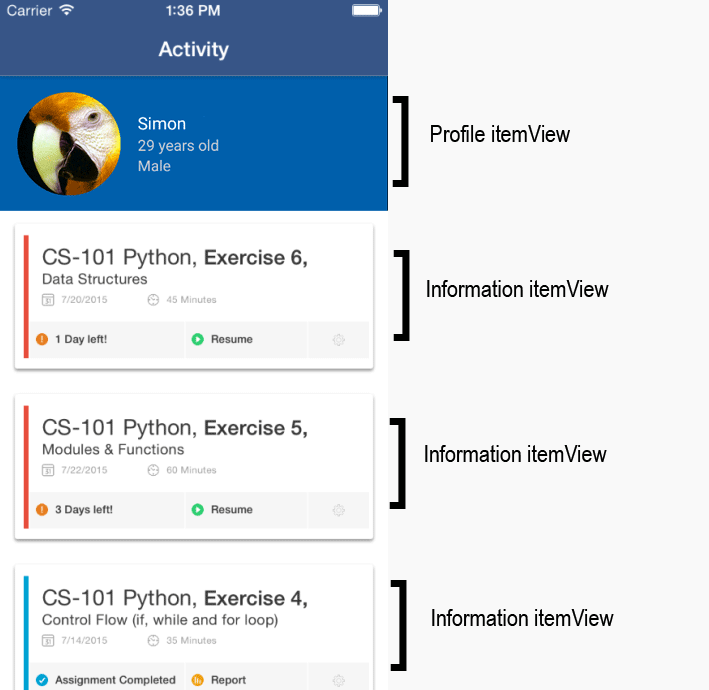
public int getItemCount() {

return mDataset.length;

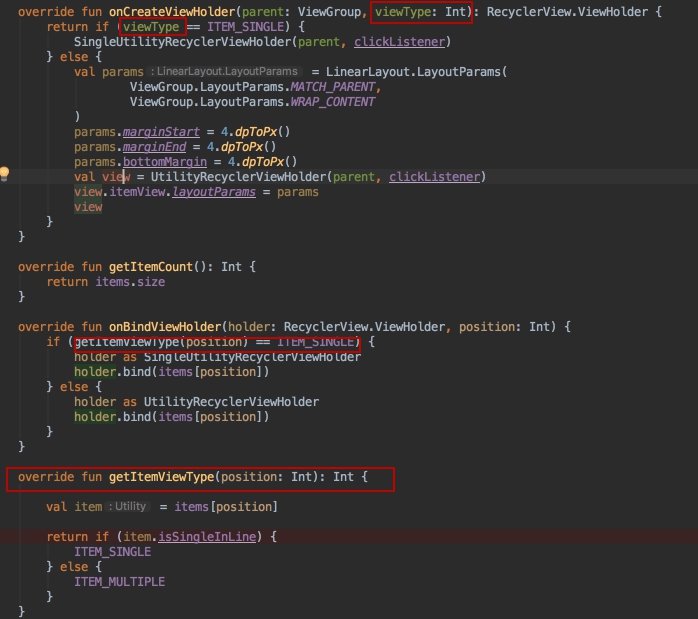
}

}

**Различные типы ячеек.**

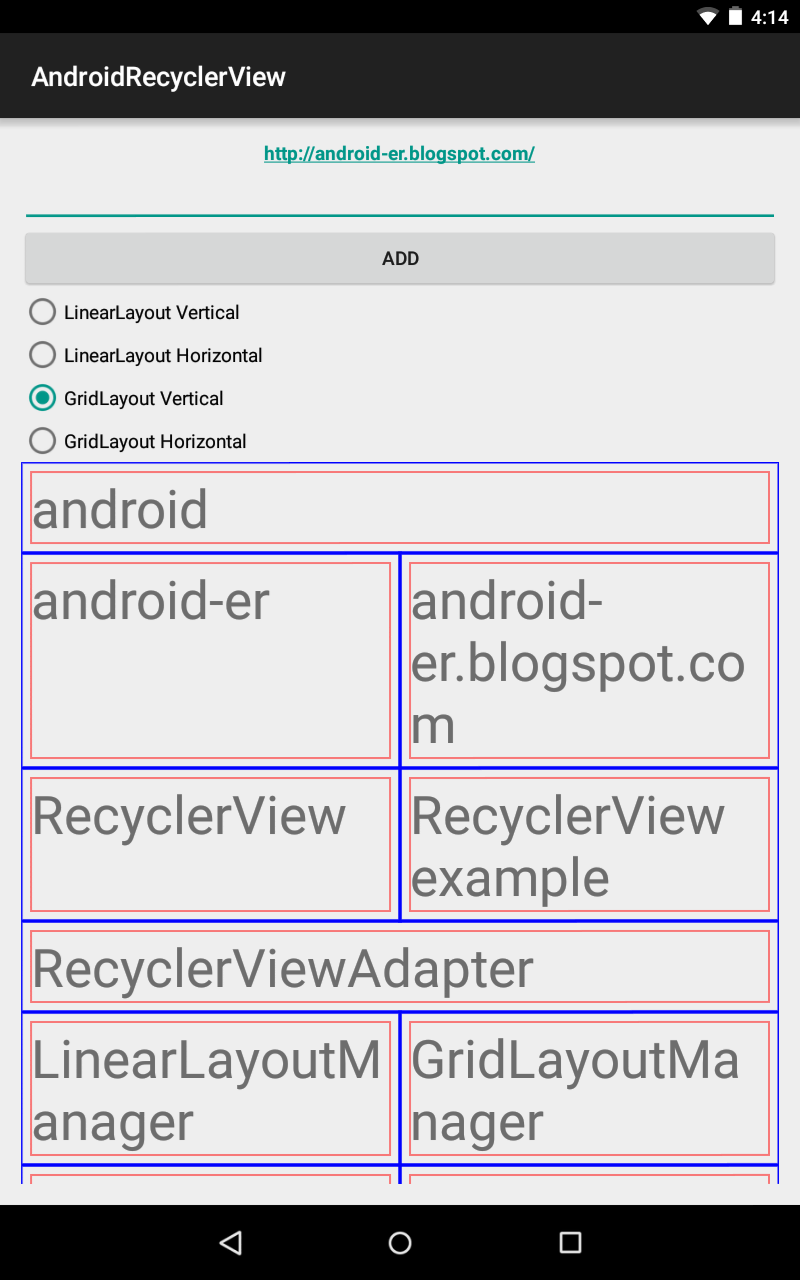
****

<http://joxi.ru/J2bY38dT4kMogm>



Объединение ячеек в 1

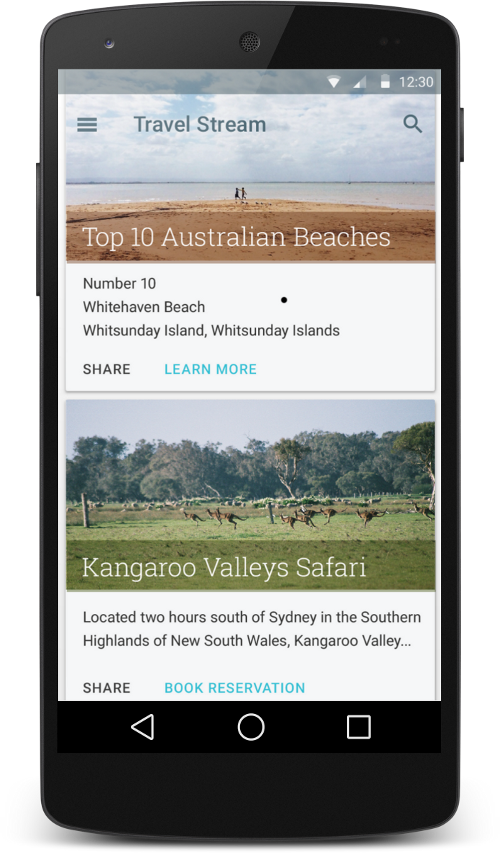
Бывают случаи, когда ячейка должна занимать место двух и более ячеек. Для этого есть специальный механизм.

  
<https://developer.android.com/reference/android/support/v7/widget/GridLayoutManager.SpanSizeLookup.html>   
<https://stackoverflow.com/questions/26869312/set-span-for-items-in-gridlayoutmanager-using-spansizelookup>

gridLayoutManager.setSpanSizeLookup(new GridLayoutManager.SpanSizeLookup() { @Override public int getSpanSize(int position) { return position % 3 == 0 ? 2: 1 } });

## Создание карточек

Виджет [CardView](https://developer.android.com/reference/android/support/v7/widget/CardView.html?hl=ru) является расширением класса [FrameLayout](https://developer.android.com/reference/android/widget/FrameLayout.html?hl=ru) и позволяет отображать информацию внутри карточек, которые будут выглядеть одинаково в рамках платформы. Виджеты [CardView](https://developer.android.com/reference/android/support/v7/widget/CardView.html?hl=ru) могут отбрасывать тени и иметь закругленные углы.



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

xmlns:tools="http://schemas.android.com/tools"

xmlns:card\_view="http://schemas.android.com/apk/res-auto"

... >

<!-- A CardView that contains a TextView -->

<android.support.v7.widget.CardView

xmlns:card\_view="http://schemas.android.com/apk/res-auto"

android:id="@+id/card\_view"

android:layout\_gravity="center"

android:layout\_width="200dp"

android:layout\_height="200dp"

card\_view:cardCornerRadius="4dp">

<TextView

android:id="@+id/info\_text"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</android.support.v7.widget.CardView>

</LinearLayout>

Полный список атрибутов есть в документации: <https://developer.android.com/reference/android/support/v7/widget/CardView.html?hl=ru>

<https://codelabs.developers.google.com/codelabs/material-design-style-ru/index.html?index=..%2F..%2Findex#5>

#### Задание:

В зеплине.

На данном экране есть объекты двух типов.

Первый тип (DetailInfoItem) - с квитанций по охрану включительно. Второй(BaseInfoItem) - с контактов УК до памятки.   
Обратите внимание, что охрана является первым типом, но занимает всю ширину экрана. Это происходит потому, что у нее не нашлось пары нужного типа, в таком случае объект первого типа должен занимать всю ширину экрана.  
Т.о., с точки зрения view, получается, что у нас список состоит из ячеек двух типов

<http://joxi.ru/J2bY38dTX0D3em> ,т.е. “охрана” с точки зрения ява объекта будет типом 1, а с точки зрения view типом 2.

При нажатии на иконку ic\_info\_white сверху должен отображаться диалог с любым текстом и кнопкой ОК.  
  
При нажатии на домик сверху должен показываться toast  
  
При нажатии на любой из элементов списка должен показываться snackbar с заголовком нажатой карточки

#### Полезные ссылки:

* Принципы дизайна - <https://material.io/guidelines/material-design/introduction.html> обязательно к прочтению перед уроком.
* Про списки и карточки в списках <https://developer.android.com/training/material/lists-cards.html?hl=ru>
* Про расположение ячейки на месте двух: <https://developer.android.com/reference/android/support/v7/widget/GridLayoutManager.SpanSizeLookup.html> <https://stackoverflow.com/questions/26869312/set-span-for-items-in-gridlayoutmanager-using-spansizelookup>
* Подключение библиотек:

dependencies {  
 compile 'com.android.support:appcompat-v7:21.0.+'  
 compile 'com.android.support:cardview-v7:21.0.+'  
 compile 'com.android.support:recyclerview-v7:21.0.+'  
}  
Актуальные версии тут:

<https://developer.android.com/topic/libraries/support-library/revisions.html>

* xml атрибуты CardView можно посмотреть тут <https://developer.android.com/reference/android/support/v7/widget/CardView.html?hl=ru>
* Обязательно всем пройти <https://codelabs.developers.google.com/codelabs/material-design-style/#0>
* Обязательно всем прочитать <https://www.sitepoint.com/material-design-android-design-support-library/>