

3.1. Создание списков элементов с помощью RecyclerView

Сайт: [Samsung Innovation Campus](https://innovationcampus.ru/)

Курс: Мобильная разработка на Kotlin

Книга: 3.1. Создание списков элементов с помощью RecyclerView

Напечатано:: Murad Rezvan

Дата: понедельник, 3 июня 2024, 17:49

Оглавление

[Списки на платформе Android](#)

[LayoutManager](#)

Списки на платформе Android

Отображение информации в виде прокручиваемого списка или сетки элементов – один из наиболее часто встречающихся элементов UI на платформе Android. Списки встречаются в подавляющем большинстве приложений. Это могут быть достаточно простые списки: контактов в телефонном справочнике, покупок в приложении какого-либо интернет магазина, фотографий. Или более сложные списки, которые отображают много дополнительной информации

Из-за повсеместного использования списков крайне важно, чтобы

- * их отображение было быстрым,
- * задействование ресурсов устройства было минимально необходимым,
- * при изменении элементов не происходило обновление всего списка, а лишь тех элементов, которым действительно необходимо обновление.

Все выше перечисленные «пожелания» учтены в виджете RecyclerView:

- * по умолчанию RecyclerView выполняет работу по отрисовке только тех элементов списка, которые видны на экране в данный момент времени. Например, если Ваш список состоит из нескольких тысяч элементов, но в текущий момент времени на экране видны лишь 10, RecyclerView выполнит работу достаточную для отображения именно видимых на данный момент 10 элементов;
- * одной из самых «дорогих» операций является создание элементов интерфейса и процесс надувания разметки в объект класса, который отвечает за эту разметку. В тот момент, когда элемент списка исчезает с экрана устройства, RecyclerView не уничтожает элементы интерфейса участвующие в отображении элемента списка, а, как следует из названия виджета, «перерабатывает» их. Это значит, что эти элементы могут быть использованы повторно в будущем при отображении других элементов набора данных;
- * при изменении элементов набора данных, RecyclerView позволяет обновлять лишь те элементы списка, которые в этом действительно нуждаются.

RecyclerView -- очень мощный виджет, позволяющий кастомизировать списки для многих сценариев. Однако это достигается путём делегирования ответственности отдельным вспомогательным классам:

- * `LayoutManager` (размещение элементов в списке),
- * `ItemDecoration` (добавление разделителей между элементами, группировки и т.п.);
- * `ItemAnimator` (анимации добавления, удаления, перемещения элементов списка);
- * `SmoothScroller` (плавная прокрутка);
- * `SnapHelper` (выравнивание элементов при прокрутке);
- * `Selection` (выбор одного или нескольких элементов);
- * `DiffUtil` (обновление отдельных элементов, а не всего списка).

Для использования RecyclerView необходимо добавить зависимость в `build.gradle` модуля:

```
dependencies {  
    ...  
    implementation 'androidx.recyclerview:recyclerview:1.2.1'  
    ...  
}
```

LayoutManager

Основные варианты расположения элементов в списке:

1. LinearLayoutManager (аналог ListView)
2. GridLayoutManager (аналог GridView)
3. StaggeredGridLayoutManager (сетка, в которой элементы могут иметь различные размеры)

Пример установки StaggeredGridLayoutManager с тремя колонками и вертикальной прокруткой:

```
recyclerView = findViewById(R.id.recycler_view)
staggeredGridLayoutManager = StaggeredGridLayoutManager(3, StaggeredGridLayoutManager.VERTICAL)
recyclerView.layoutManager = staggeredGridLayoutManager
recyclerView.setHasFixedSize(true)
```

[Начать тур для пользователя на этой странице](#)