

ПРИМЕНЕНИЕ ФУНКЦИИ RECYCLERVIEW ДЛЯ ОС АНДРОИД ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ ПРОДУКТОВ (НА ЯЗЫКЕ КОТЛИН)

Б.Т.Нурымбетов

Нукусский филиал ТУИТ, ассистент

З.О.Юсупбаева

Нукусский филиал ТУИТ, студент

На сегодняшний день Информационные технологий прочно закрепились к нашей жизни. Разработка более совершенных систем и более качественных устройств благополучно влияет к человеческой среде и между тем качественно обеспечивает социальные бытия нашей жизни. Принятие системы Android положило начало разработок более качественных программных продуктов. Операционная система Андроид предназначена для смартфонов, смарттелевизоров, смартотчов, для систем управления автомобилей и для умных домов. Благодаря таким устройствам нашими программистами созданы многочисленные программные продукты. Все эти возможности напрямую воздействуя к человеческой жизни, экономить человеческой силы, и выполняет функций в короткое промежутки времени. [1]

При применении программных продуктов ход и действие выполнение работы обеспечивается в короткое промежутки времени и благодаря этим возможностям все более возрастает спрос к таким программным продуктам. Благодаря возрастанием спроса к программным продуктам все более усовершенствуется качества этих благ. Один из основных требования к программным продуктам- это в один момент времени без усилий доставит многочисленные информации к пользователю. [2]

В этой статье рассматривается применение функции RV программной среды Андроид в разработке мобильных приложений.

RecyclerView-очень удобная функция отображающий элементов списка. В чем состоит этот пользовательский компонент? - В первую очередь интерфейс программы, информации которые передается к пользователю реализуется через список. В программировании Андроид функция ListView обеспечивает работы со списками. Если, информации в списке многочисленно, то оперативный память будет занятым и это приводит к нагрузке устройств. На рисунке показан принцип работы RecyclerView.

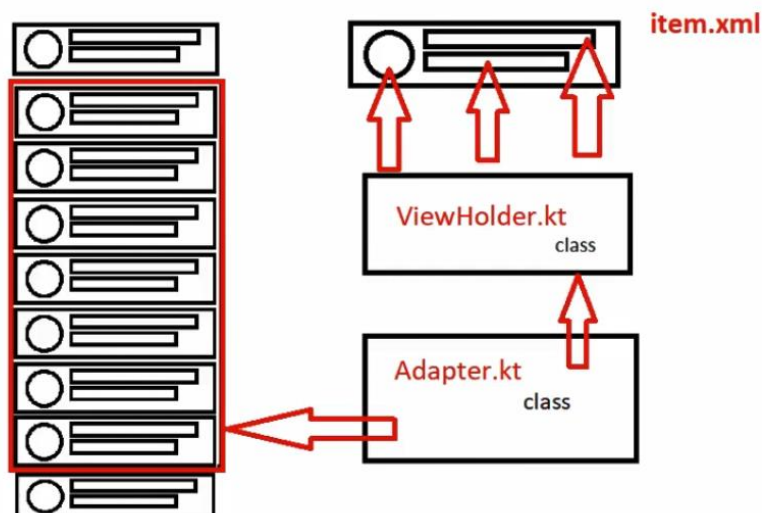


Рисунок 1. Структура RecyclerView

В нашем примере наверно элементы списка многочисленна, но благодаря RecyclerView, в оперативной памяти имеет место лишь те элементы которые отображается на экране и те элементы которые на верху и внизу.

Для этого сперва мы создаем элемент в виде файла item.xml. С помощью класса ViewHolder, управляем данными принимаемые через класса адаптер, выводимые на экране.

```
class MyViewHolder(private val view: View):
RecyclerView.ViewHolder(view)
{

fun populate (data: Model, listener: OnClickListener, pos: Int) {

view.tvtitle.text = data.title

view.tvDescription.text = data.description
}
}
```

Через класса адаптера данных отправляем следующим порядке:

```
class MyAdapter():RecyclerView.Adapter<MyViewHolder> () {

var models: MutableList<Model> = mutableListOf()

set(value) {

field = value
}
```

```
        notifyDataSetChanged()
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
    MyViewHolder {
        val item =
        LayoutInflater.from(parent.context).inflate(R.layout.item, parent, false)

        return MyViewHolder(item)
    }

    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        holder.populate(models[position], listener, position)
    }

    override fun getItemCount(): Int {
        return models.size
    }
}
```

Применяя этих классов можно управлять тысячи и миллионы список. Преимущество функции RecyclerView от функции ListView, в том что ListView предназначена только для список вертикальных расположении, а RecyclerView при помощи функции layoutManager располагает список в нужном форме. И так, используя linearLayoutManager и настраивая ориентацию вертикальной форме, имеем вертикальный список, а при помощи GridLayoutManager имеем горизонтальный список, вводя функцию spanCount строки списка выводятся в горизонтальном виде.

Нами применяемая функция RecyclerView является оптимальный вариант самых сложных программ.

Литература

1. Rick Rogers. Android Application Development - М.: КРТ Полиграф, 2009.
2. Barry Burd. Android Application Development All-in-One For Dummies® - Свердловск: Клип, 2011.