Инструкция по использованию RecycleView

1. Добавить RecycleView в xml файл
   1. Добавить необходимый LayoutManager либо сразу в xml файле, либо в коде
2. Создать Адаптер
3. Создать ViewHolder по плану

class MarsPropertyViewHolder private constructor(val binding: GridViewItemBinding) :  
 RecyclerView.ViewHolder(binding.*root*) {  
 companion object {  
 fun from(parent: ViewGroup): MarsPropertyViewHolder =  
 MarsPropertyViewHolder(getBinding(parent))  
  
 private fun getBinding(parent: ViewGroup): GridViewItemBinding {  
 val layoutInflater = LayoutInflater.from(parent.*context*)  
 val binding = GridViewItemBinding.inflate(layoutInflater, parent, false)  
 return binding  
 }  
 }  
  
 fun bind(item: MarsProperty, clickListener: (String) -> Unit) =  
 fulfillBinding(item, clickListener)  
  
 private fun fulfillBinding(item: MarsProperty, clickListener: (String) -> Unit) {  
 binding.*apply* **{** *property* = item  
 executePendingBindings()  
 **}** itemView.setOnClickListener **{** clickListener(item.id) **}** }

1. DiffUtils

class MarsPropertyDiffCallback : DiffUtil.ItemCallback<MarsProperty>(){  
 override fun areItemsTheSame(oldItem: MarsProperty, newItem: MarsProperty): Boolean =  
 oldItem.id == newItem.id  
  
 override fun areContentsTheSame(oldItem: MarsProperty, newItem: MarsProperty): Boolean =  
 oldItem == newItem  
  
}

1. Создаем конструктор класса

class PhotoGridAdapter(val clickListener: (String) -> Unit):  
 ListAdapter<MarsProperty, PhotoGridAdapter.MarsPropertyViewHolder>(MarsPropertyViewHolder.MarsPropertyDiffCallback())

1. Имплементируем недостающие методы

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MarsPropertyViewHolder {  
 *TODO*("Not yet implemented")  
}  
  
override fun onBindViewHolder(holder: MarsPropertyViewHolder, position: Int) {  
 *TODO*("Not yet implemented")  
}

1. Изменяем их