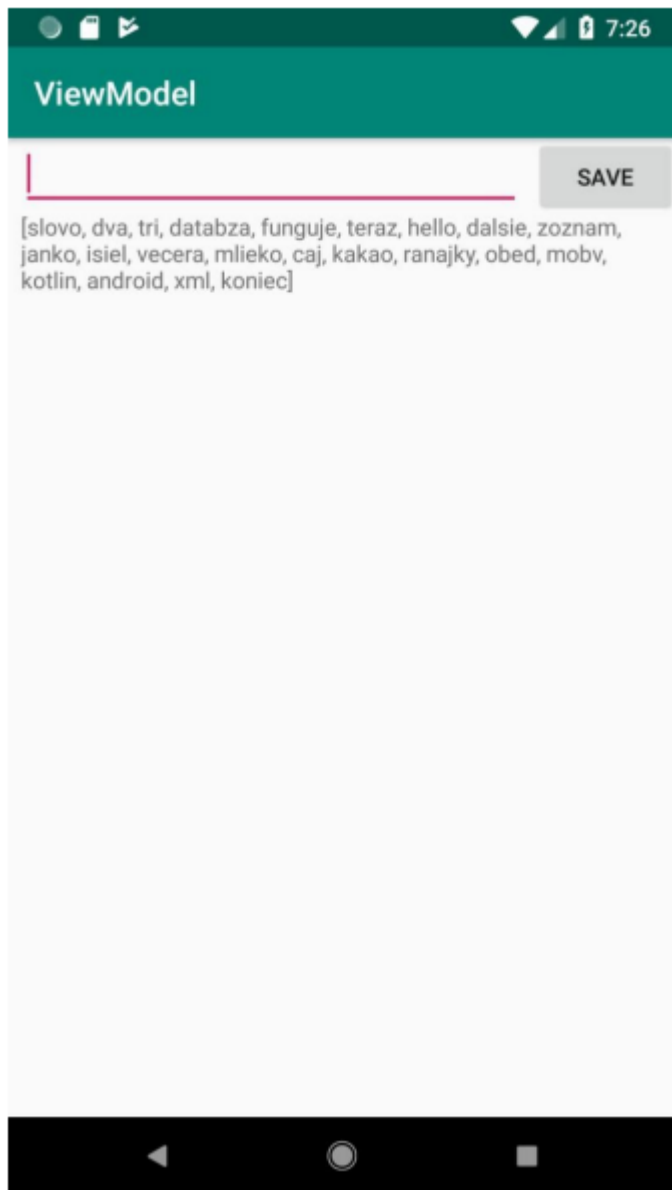




Mobilné výpočty

Maroš Čavojský



Zobrazenie údajov



RecyclerView vs. ListView vs. GridView

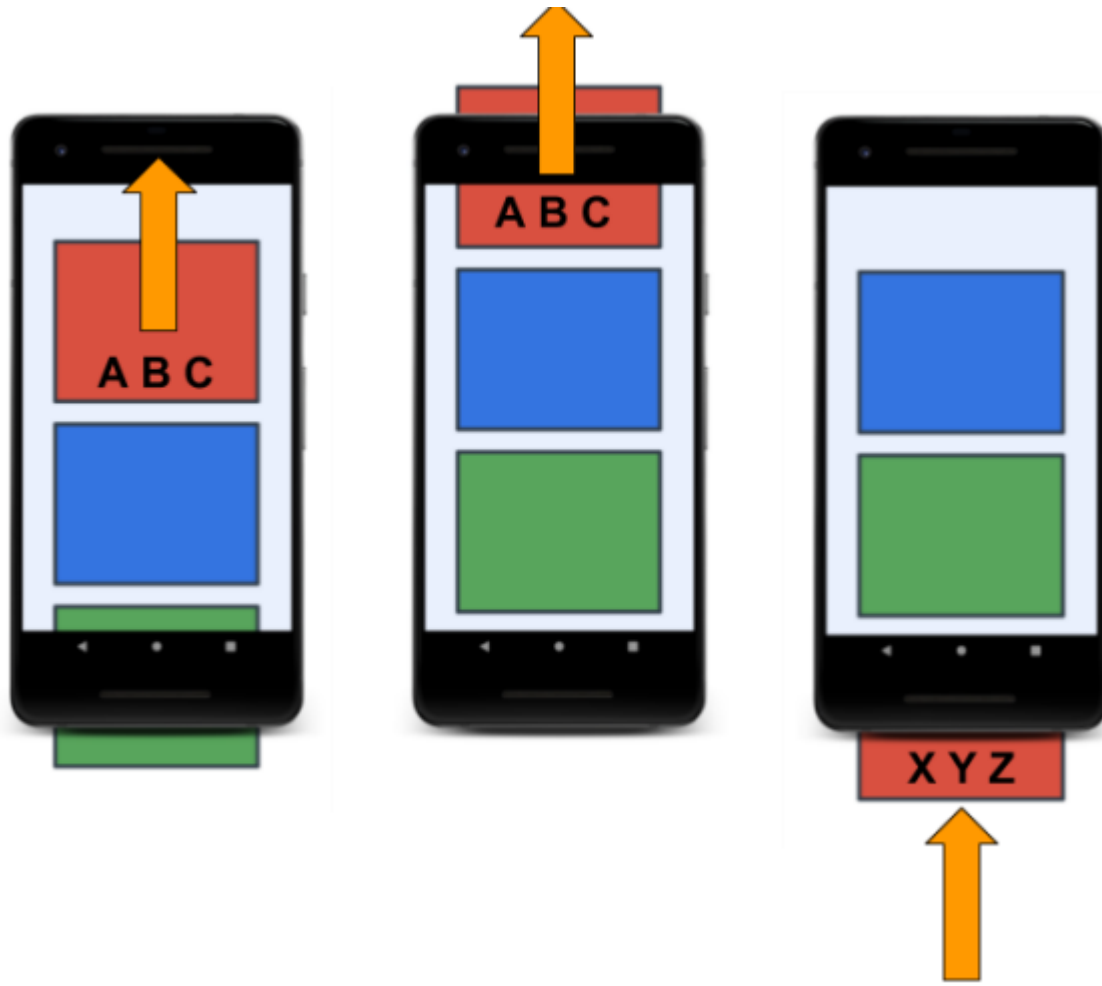


RecyclerView.**Adapter**<VH extends RecyclerView.ViewHolder>

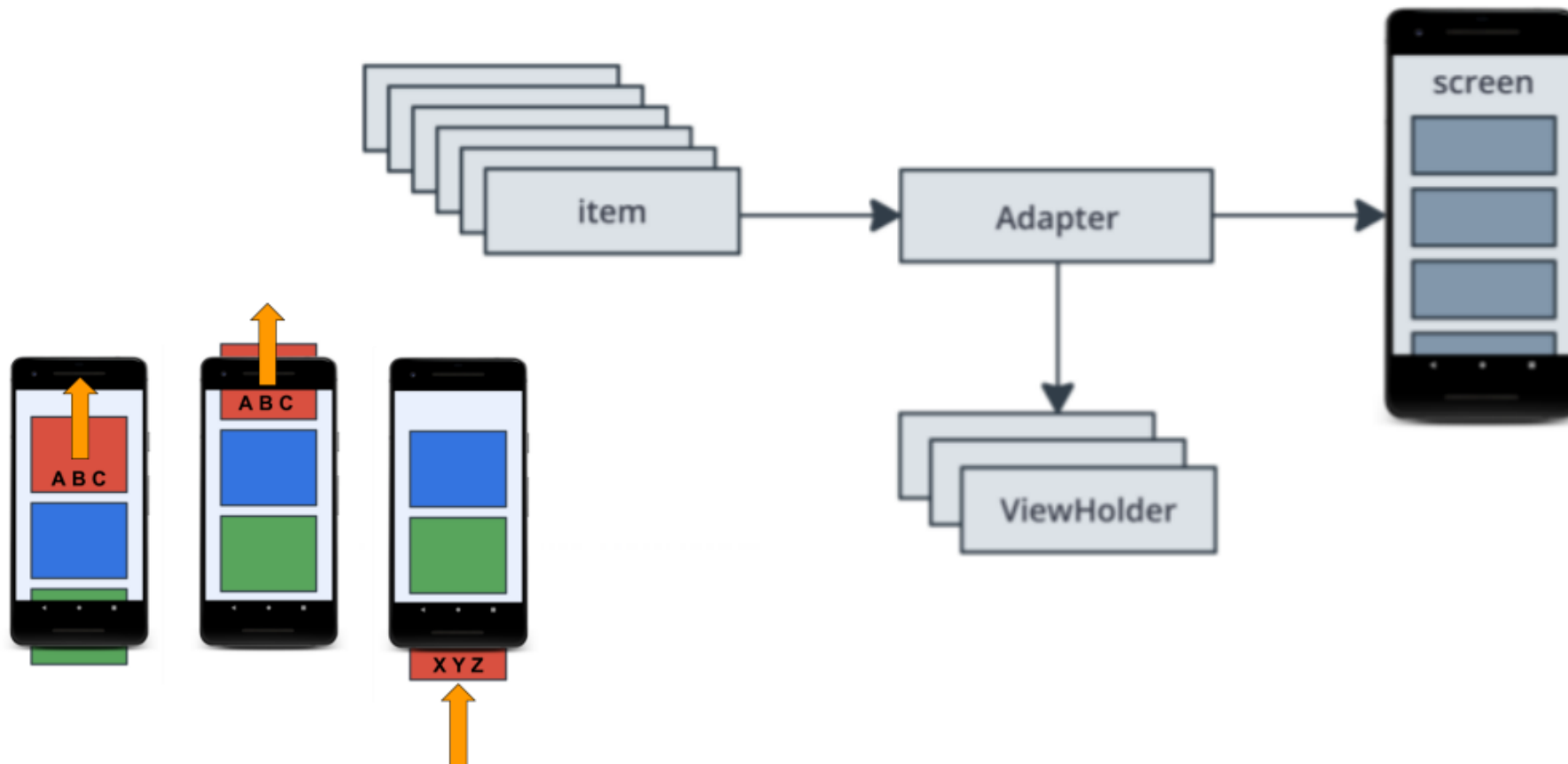
ListAdapter

Zdroj: <https://developer.android.com/>

RecyclerView



RecyclerView



RecyclerView

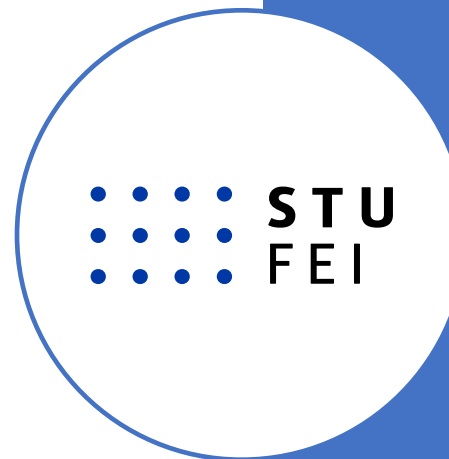
- ❑ Dáta
- ❑ RecyclerView v layout-e
- ❑ Layout/y pre konkrétny prvok/prvky
- ❑ Layout manager
- ❑ ViewHolder
- ❑ Adapter

```
dependencies {  
    implementation("androidx.recyclerview:recyclerview:1.2.1")  
  
    // For control over item selection of both touch and mouse driven selection  
    implementation("androidx.recyclerview:recyclerview-selection:1.1.0")  
}
```

Zdroj: <https://developer.android.com/>

Dáta

Kolekcia : List<Any>



RecyclerView

- ☒ Dáta
- ☐ RecyclerView v layout-e
- ☐ Layout/y pre konkrétny prvok/prvky
- ☐ Layout manager
- ☐ ViewHolder
- ☐ Adapter

RecyclerView v layout-e

src/main/res/layout/activty_main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        ★ android:scrollbars="vertical"
        ★ app:layoutManager="LinearLayoutManager" />

</FrameLayout>
```



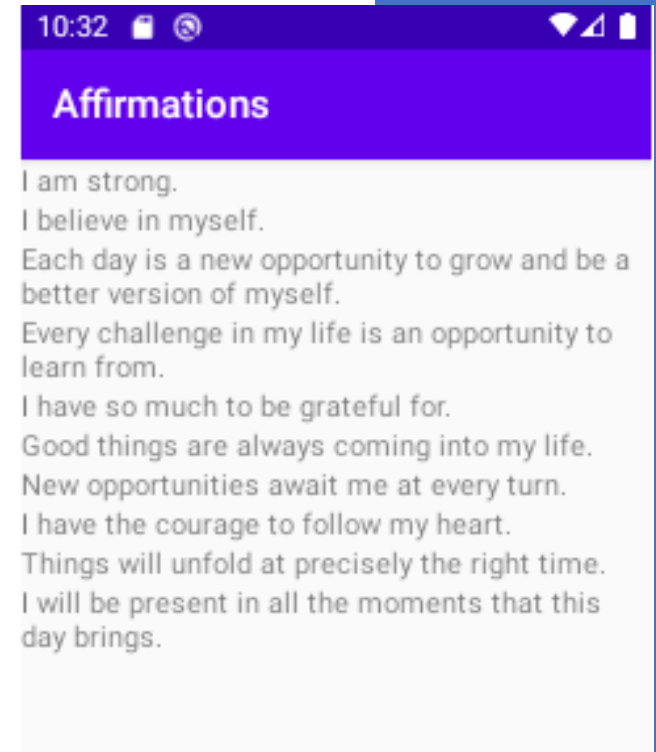
RecyclerView

- ☒ Dáta
- ☒ RecyclerView v layout-e
- ☐ Layout/y pre konkrétny prvok/prvky
- ☐ Layout manager
- ☐ ViewHolder
- ☐ Adapter

Layout pre jeden prvok

src/main/res/layout/list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/item_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```



Layout pre jeden prvok

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">

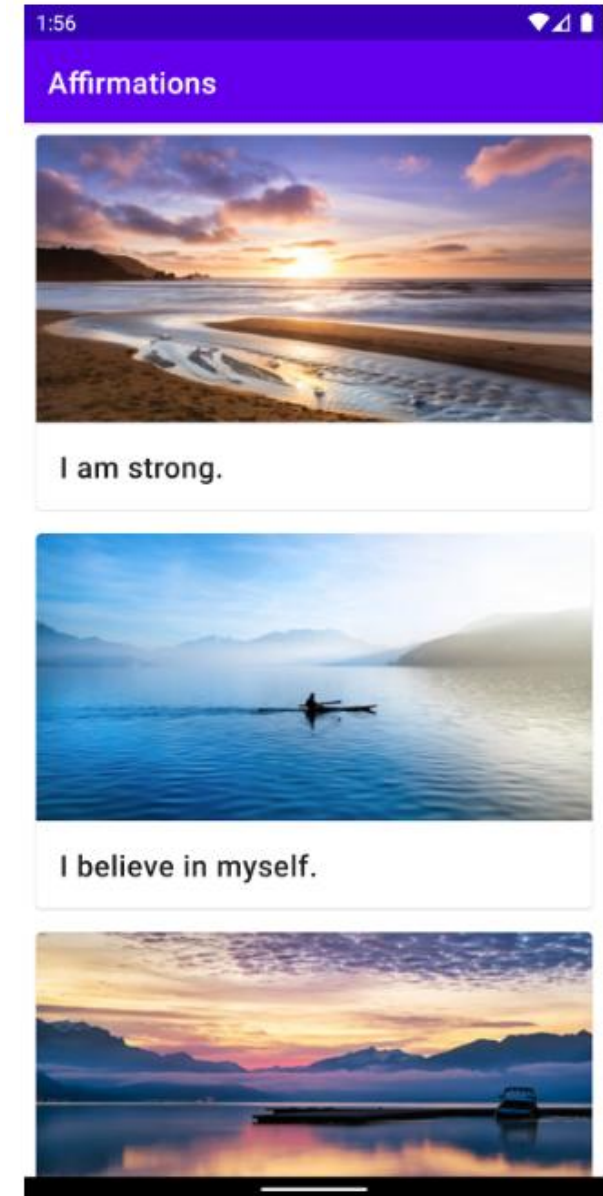
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/item_image"
            android:layout_width="match_parent"
            android:layout_height="194dp"
            android:importantForAccessibility="no"
            android:scaleType="centerCrop" />

        <TextView
            android:id="@+id/item_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="16dp"
            android:textAppearance="?attr/textAppearanceHeadline6" />

    </LinearLayout>

</com.google.android.material.card.MaterialCardView>
```

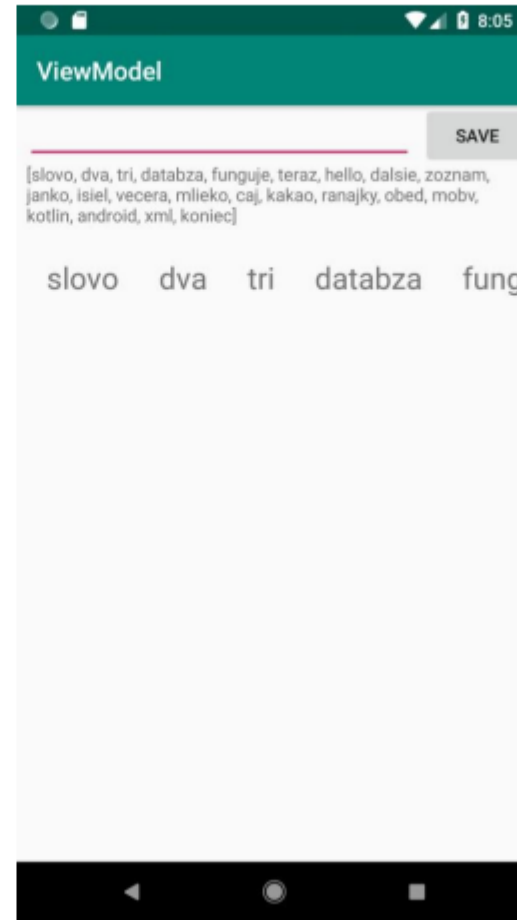


RecyclerView

- ☒ Dáta
- ☒ RecyclerView v layout-e
- ☒ Layout/y pre konkrétny prvok/prvky
- ☐ Layout manager
- ☐ ViewHolder
- ☐ Adapter

LayoutManager

LinearLayoutManager



GridLayoutManager



LayoutManager

binding.wordsList.layoutManager =

LinearLayoutManager(context, RecyclerView.VERTICAL, false)

LinearLayoutManager(context, RecyclerView.VERTICAL, true)

LinearLayoutManager(context, RecyclerView.HORIZONTAL, true)

LinearLayoutManager(context, RecyclerView.HORIZONTAL, false)

GridLayoutManager(context, 3, RecyclerView.VERTICAL, false)



RecyclerView

- ☒ Dáta
- ☒ RecyclerView v layout-e
- ☒ Layout/y pre konkrétny prvok/prvky
- ☒ Layout manager
- ☐ ViewHolder
- ☐ Adapter

ViewHolder

```
class ViewHolder private constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {  
  
    fun bind(item: WordItem) {  
        (itemView as TextView).text = item.word  
    }  
  
    companion object {  
        fun from(parent: ViewGroup): ViewHolder {  
            val inflater = LayoutInflater.from(parent.context)  
            val view = inflater  
                .inflate(R.layout.text_item, parent, false)  
  
            return ViewHolder(view)  
        }  
    }  
}
```

ViewHolder

```
class ViewHolder private constructor(itemView: View) : RecyclerView.ViewHolder(itemView) {  
  
    fun bind(item: WordItem) {  
        (itemView.findViewById(R.id.item_title) as TextView).text = item.word  
        (itemView.findViewById(R.id.item_image) as ImageView)  
            .setImageResource(item.imageResourceId)  
    }  
  
    companion object {  
        fun from(parent: ViewGroup): ViewHolder {  
            val inflater = LayoutInflater.from(parent.context)  
            val view = inflater.inflate(R.layout.text_item, parent, false)  
  
            return ViewHolder(view)  
        }  
    }  
}
```



RecyclerView

- ☒ Dáta
- ☒ RecyclerView v layout-e
- ☒ Layout/y pre konkrétny prvok/prvky
- ☒ Layout manager
- ☒ ViewHolder
- ☐ Adapter

Adapter

```
class MessagesAdapter : RecyclerView.Adapter<MessagesAdapter.ViewHolder>() {  
    var data =  
  
    override fun getItemCount() = data.size  
  
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
        val item = data[position]  
        holder.bind(item)  
    }  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
        return ViewHolder.from(parent)  
    }  
  
    class ViewHolder ... {}  
}
```



Adapter

- ***notifyDataSetChanged()***
- notifyItemChanged(int position)
- notifyItemInserted(int position)
- notifyItemMoved(int fromPosition, int toPosition)
- notifyItemRangeChanged(int positionStart, int itemCount)
- notifyItemRangeInserted(int positionStart, int itemCount)
- notifyItemRangeRemoved(int positionStart, int itemCount)
- notifyItemRemoved(int position)

Adapter

```
class MessagesAdapter : RecyclerView.Adapter<MessagesAdapter.ViewHolder>() {  
    var data = listOf<WordItem>()  
    set(value) {  
        field = value  
        notifyDataSetChanged()  Neoptimálne, ale funkčné !!!  
    }  
}
```

```
override fun getItemCount() = data.size
```

```
override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
    val item = data[position]  
    holder.bind(item)  
}
```

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
    return ViewHolder.from(parent)  
}
```

```
class ViewHolder ... {}
```

```
}
```



```

class MessageAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    var data = ....

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {
        return when (viewType) {
            MyItem.TYPE_VIDEO-> VideoViewHolder(parent)
            MyItem.TYPE_IMAGE -> ImageViewHolder(parent)
            MyItem.TYPE_STICKER -> StickerViewHolder(parent)
            else -> TextViewHolder(parent)
        }
    }

    override fun getItemViewType(position: Int): Int {
        return items[position].type
    }

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

    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
        (holder as GeneralViewHolder).bind(position, items[position])
    }
}

```

```

interface GeneralViewHolder {
    fun bind(position: Int, item: GeneralItem)
}

```



RecyclerView

- ✓ ☒ Dáta
- ✓ ☒ RecyclerView v layout-e
- ✓ ☒ Layout/y pre konkrétny prvok/prvky
- ✓ ☒ Layout manager
- ✓ ☒ ViewHolder
- ✓ ☒ Adapter

Bonus: Vo Fragmente

```
class MyFragment : Fragment() {
```

```
....
```

```
binding.wordsList.layoutManager = LinearLayoutManager(context, RecyclerView.VERTICAL, false)
```

```
val adapter = MessagesAdapter()  
binding.wordsList.adapter = adapter
```

```
adapter.data = listOf()
```

```
.....
```

```
}
```



Diskusia

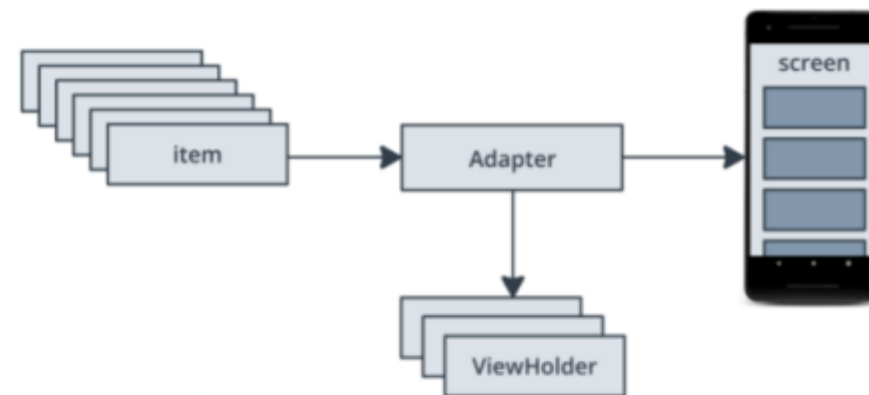
Otázky ?



RecyclerView - pozície

❖ Layout pozícia – [RecyclerView.LayoutManager](#)

❖ Adaptér pozícia - [RecyclerView.Adapter](#)



Rovnaké ???

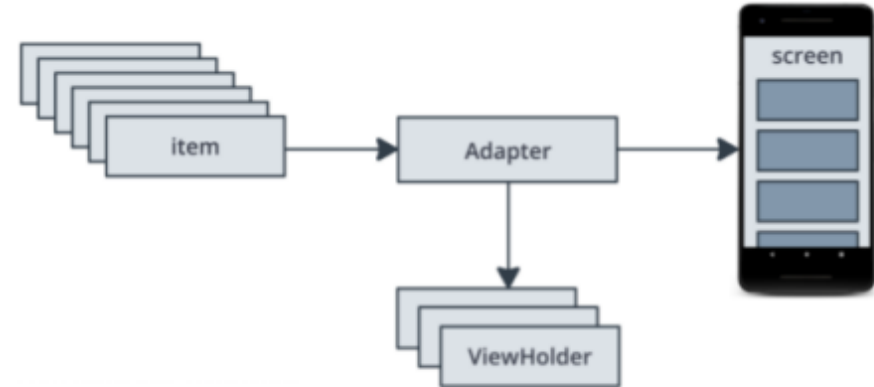
RecyclerView - pozície

❖ Layout pozícia – [RecyclerView.LayoutManager](#)

- [getLayoutPosition\(\)](#),
- [findViewHolderForLayoutPosition\(int\)](#)

❖ Adaptér pozícia - [RecyclerView.Adapter](#)

- [getAdapterPosition\(\)](#),
- [findViewHolderForAdapterPosition\(int\)](#)



*Rovnaké ??? Áno aj nie. Odlišné v čase medzi
adapter.notify.* a úpravou layoutu*

RecyclerView - binding

```
class ViewHolder(private var binding: ImageItemBinding) : RecyclerView.ViewHolder(binding.root) {

    fun bind(item: MarsItem) {
        binding.property = item
        binding.executePendingBindings()
    }

    companion object {
        fun from(parent: ViewGroup): ViewHolder {
            val view = ImageItemBinding.inflate(LayoutInflater.from(parent.context))

            return ViewHolder(view)
        }
    }
}
```

RecyclerView - binding

```
class VideoViewHolder(  
    private val parent: ViewGroup,  
    val binding: WidgetUserFollowingListItemBinding =  
        DataBindingUtil.inflate(  
            LayoutInflater.from(parent.context),  
            R.layout.widget_user_following_list_item,  
            parent,  
            false  
        )  
    ) : RecyclerView.ViewHolder(binding.root){  
  
    ...  
  
}
```

RecyclerView - binding

```
<layout>
  <data>
    <variable name="property" type="com.example.viewmodel.data.db.model.WordItem"/>
  </data>
  <TextView
    app:mojText="@{property}"
    android:textSize="24sp"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:layout_width="match_parent" android:layout_height="wrap_content"/>
</layout>
```

```
@BindingAdapter("mojText")
fun TextView.setMojText(item: WordItem) {
    text = "Slovo je ${item.word}"
}
```