Uniwersytet Wrocławski

Rafał Lewandków

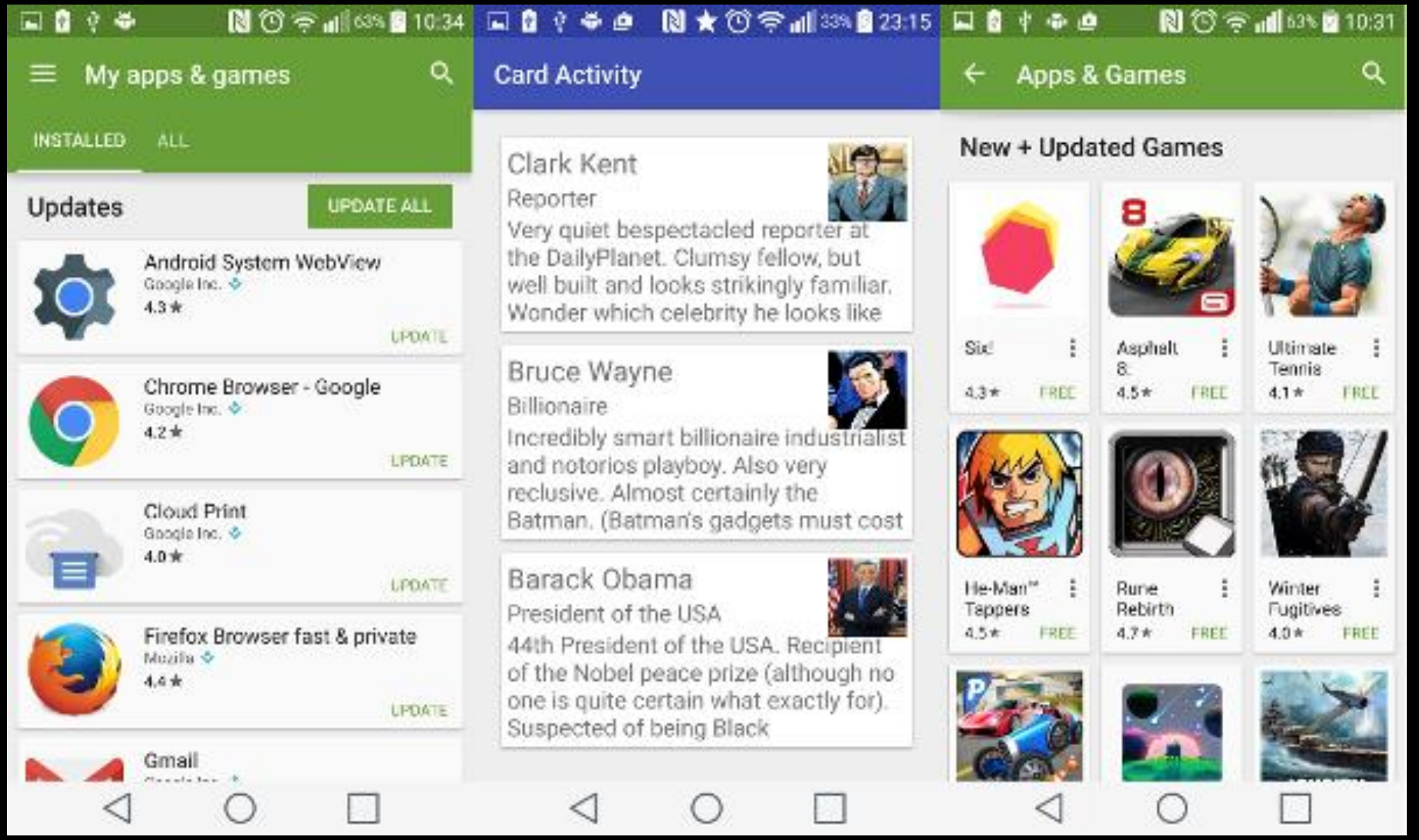
# PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

## WYKŁAD 3

- RecyclerView
- Adapter
- ViewHolder
- ViewPager2

Uniwersytet Wrocławski

# RecyclerView



# RecyclerView

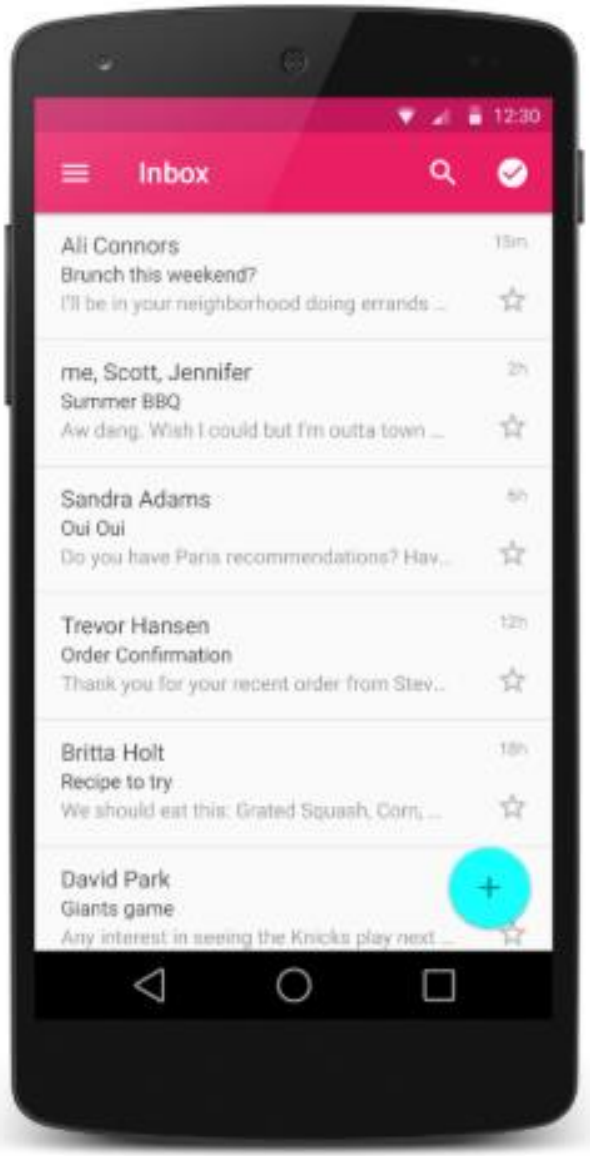



**Figure 1.** A list using `RecyclerView`

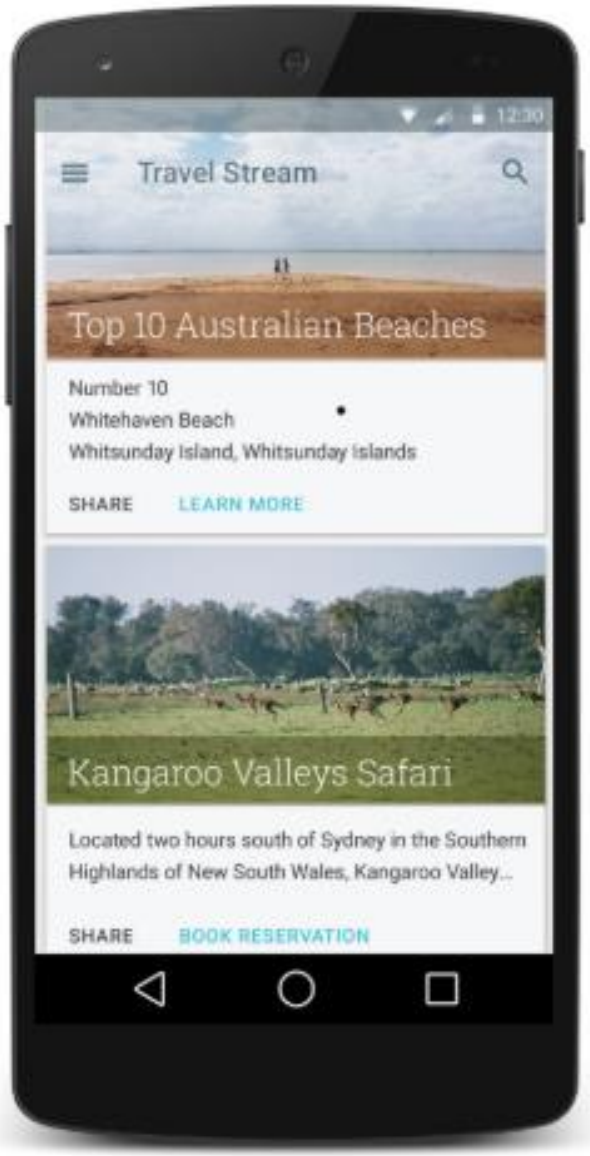
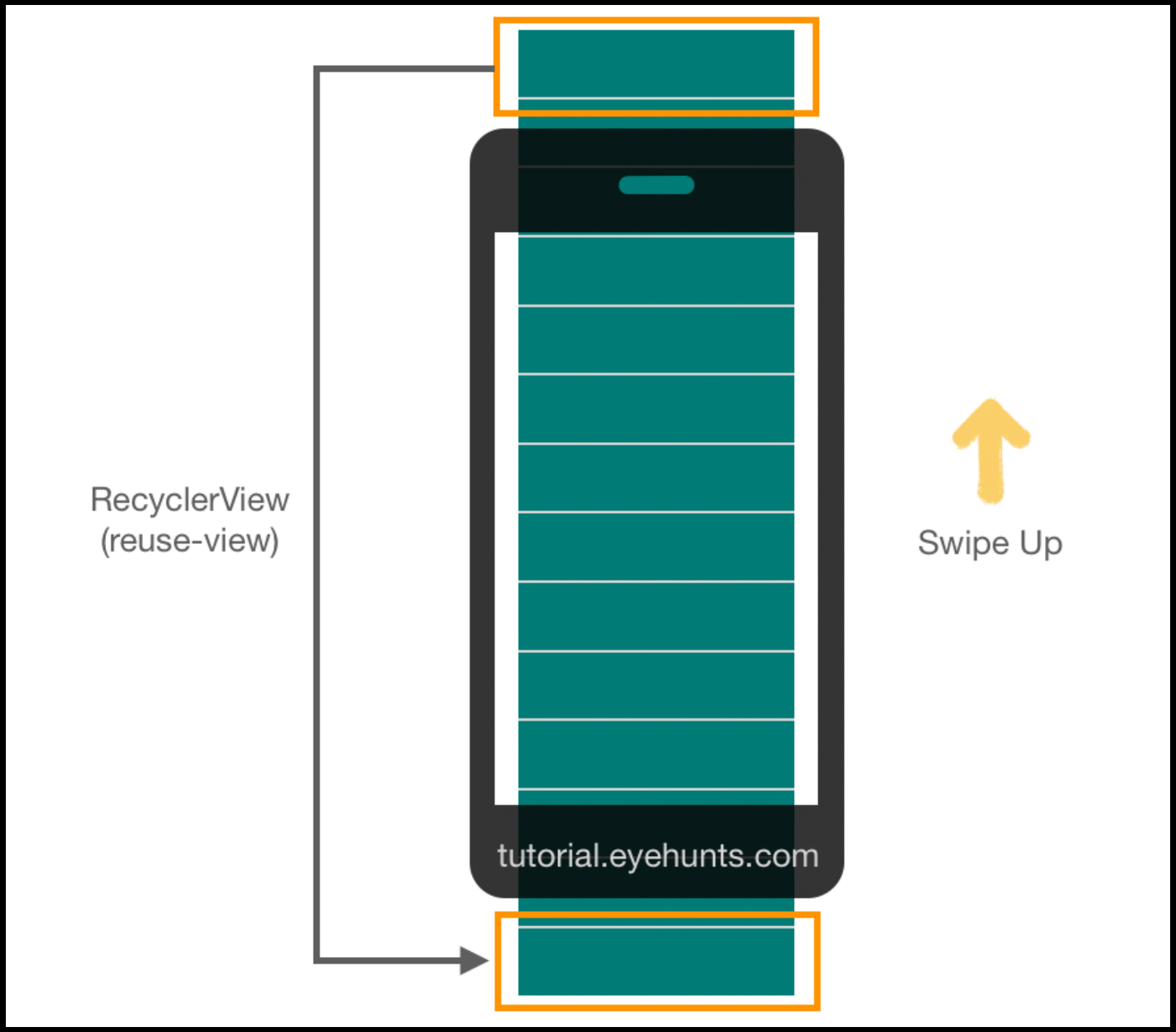



**Figure 2.** A list also using `CardView`

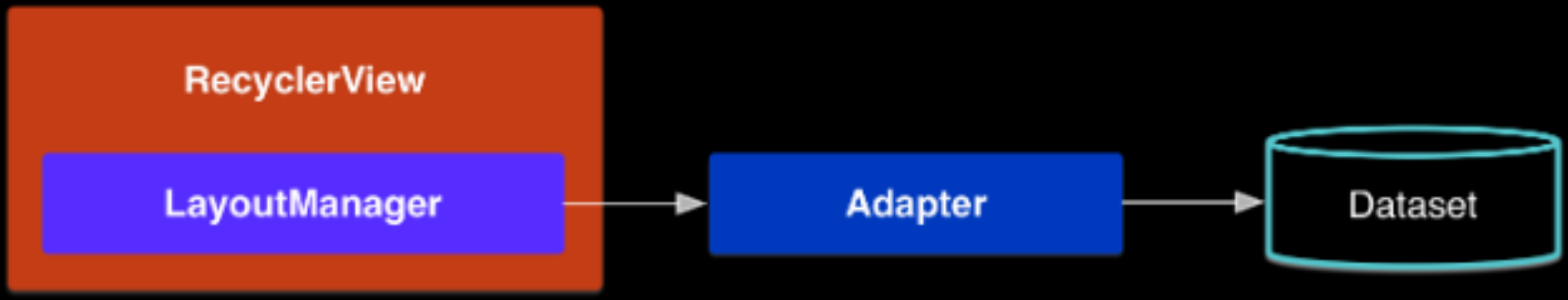
# RecyclerView



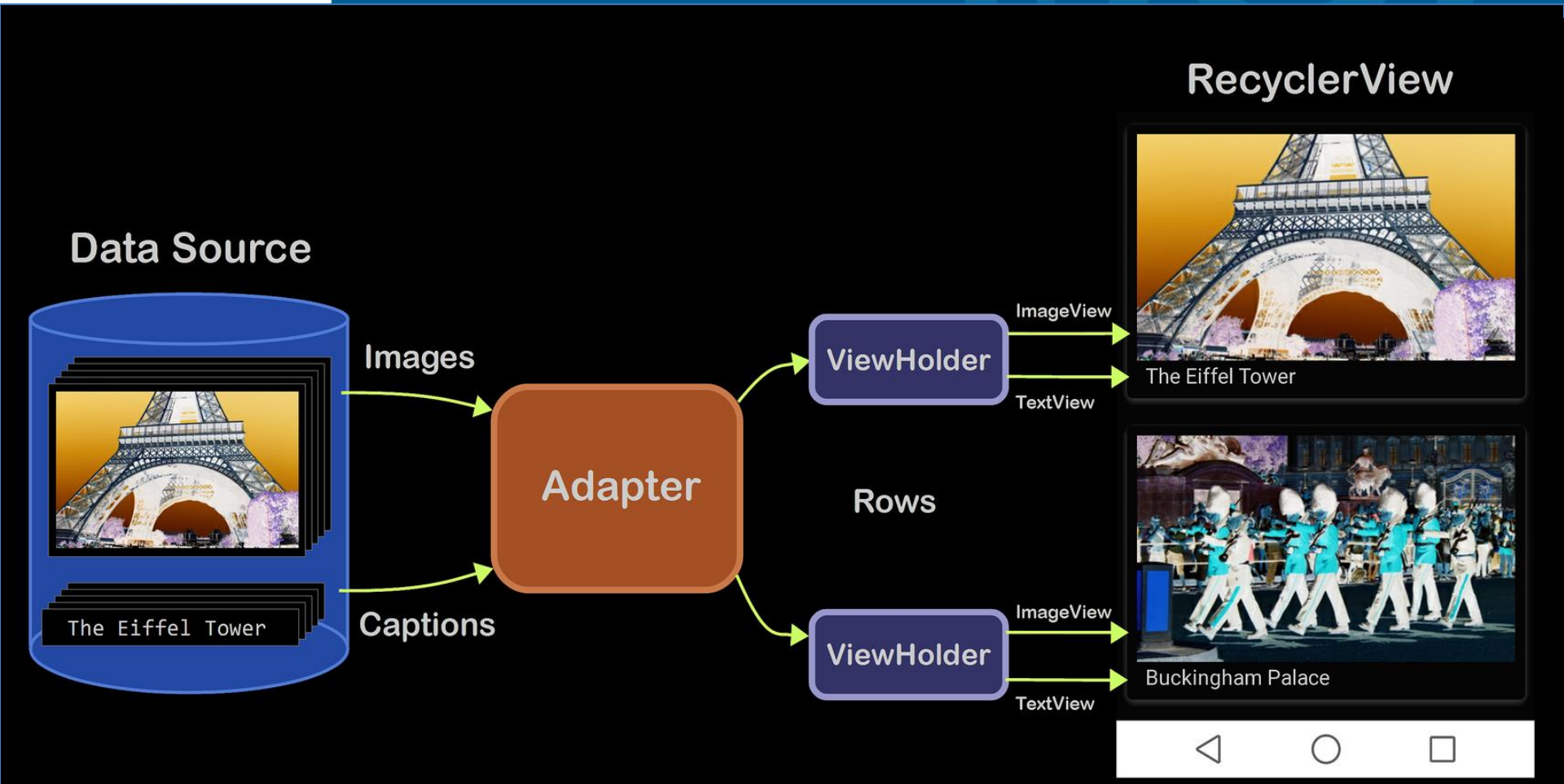
# RecyclerView

## The most important classes of the RecyclerView API

| Class | Usage |
|---|---|
| Adapter | Wraps the data set and creates views for individual items |
| ViewHolder | Holds all sub views that depend on the current item's data |
| LayoutManager | Places items within the available area |
| ItemDecoration | Draws decorations around or on top of each item's view |
| ItemAnimator | Animates items when they are added, removed or reordered |

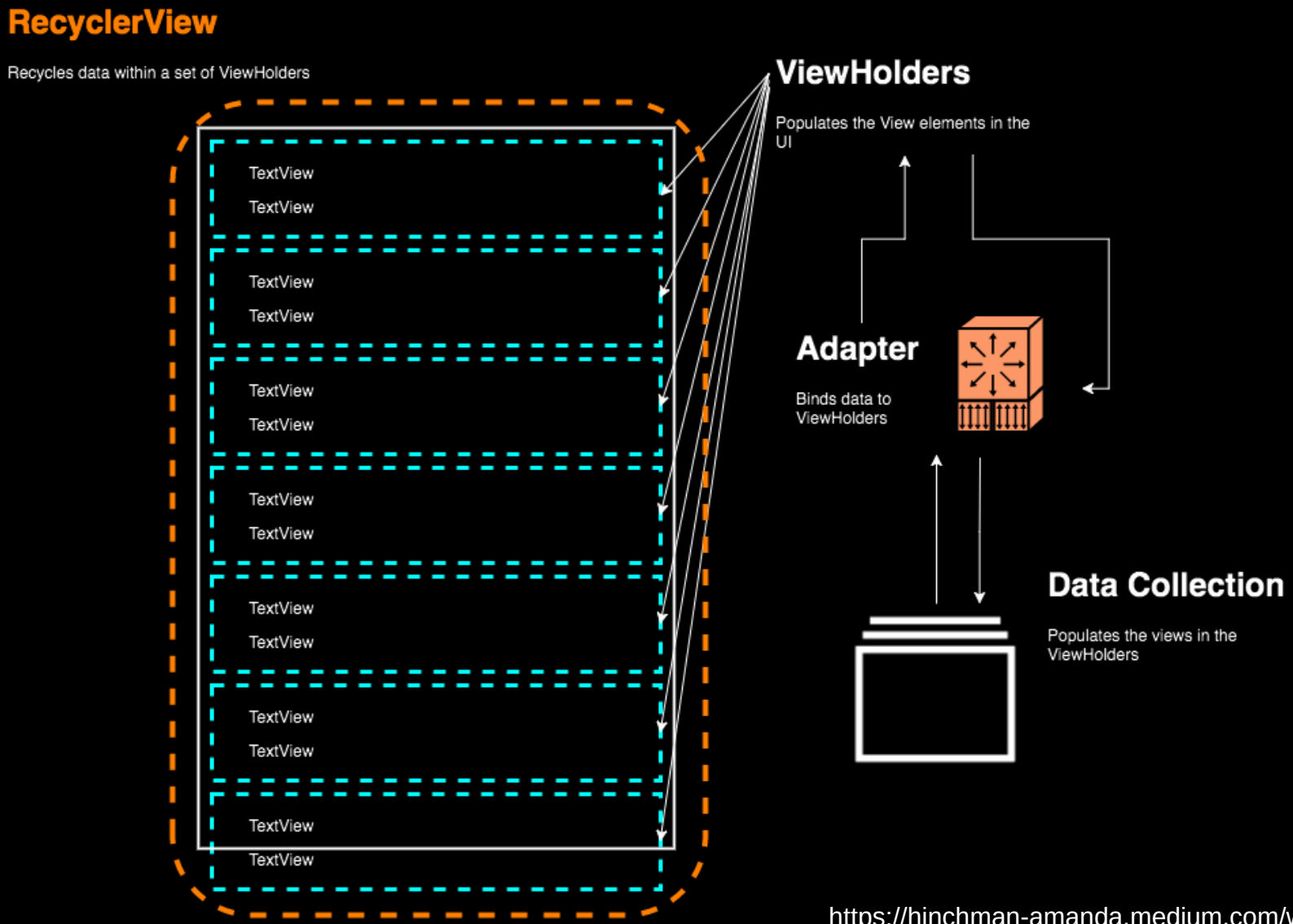


# RecyclerView




https://www.google.com/url?sa=i&url=https%3A%2F%2Flearn.microsoft.com%2Fpl-pl%2Fxamarin%2Fandroid%2Fuser-interface%2Flayouts%2Frecycler-view%2Fparts-and-functionality&psig=AOvVaw15sXcomaaNh3j6e9kHPA07&ust=1697714574289000&source=images&cd=vfe&opi=89978449&ved=0CBMQjhxqFwoTCIiSnem9_4EDFQAAAAAdAAAAABAE

# RecyclerView




https://hinchman-amanda.medium.com/working-with-recyclerview-in-android-kotlin-84a62aef94ec

# RecyclerView

## Adapter

1. Return information, e.g., about the number of items
getItemCount()

RecyclerView

Adapter

Data source

ViewHolder

2. Create new view instances
onCreateViewHolder()

3. Populate view items with data
onBindViewHolder()

| id | itemName |
|---|---|
| 100411 | LED Green 568 nm, 5mm |
| 101119 | Aluminium Capacitor 4.7µF |
| 101629 | Potentiometer 500kΩ |

Uniwersytet Wrocławski

# RecyclerView

# RecyclerView

word_list_item.xml

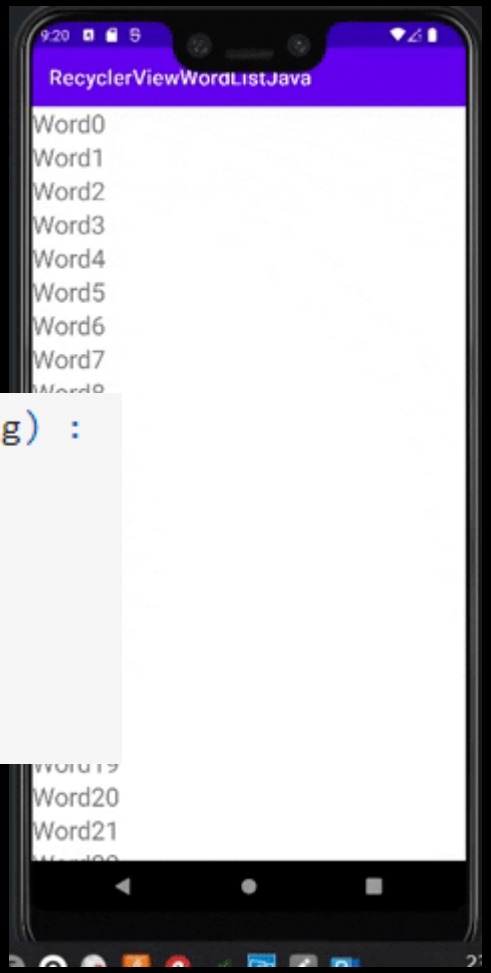
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

```kotlin
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```



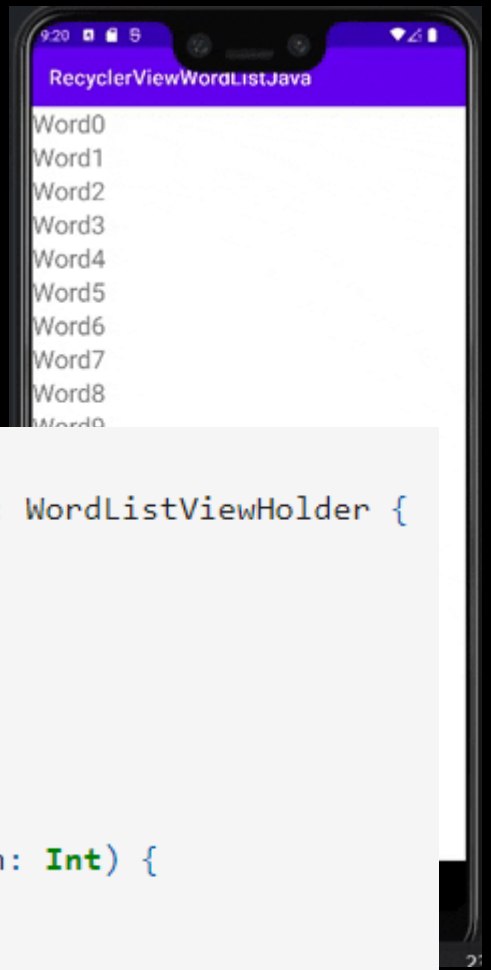
# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

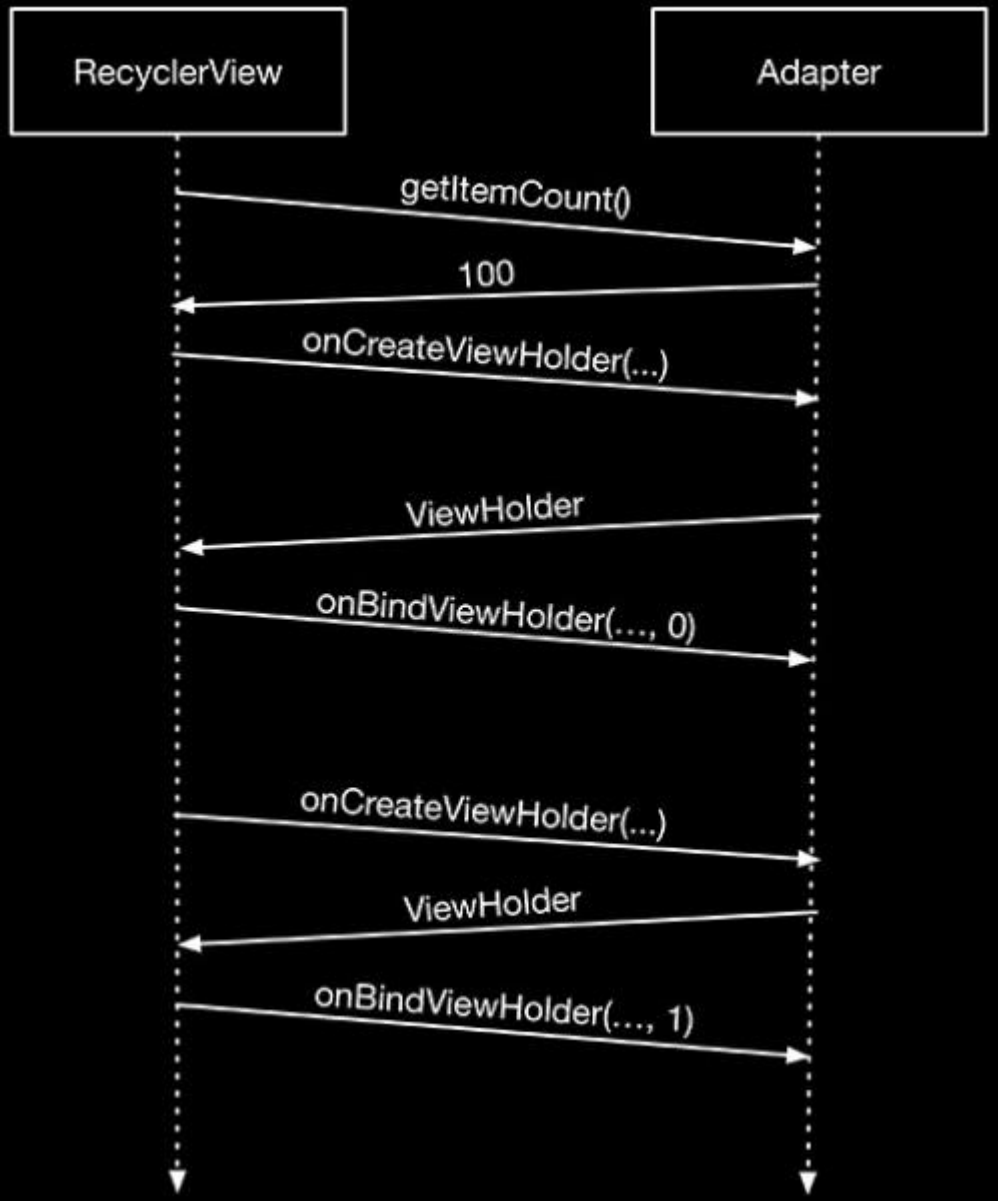
WordListViewHolder

```
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```




```
class WordListAdapter() : RecyclerView.Adapter<WordListViewHolder>(){
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordListViewHolder {
        TODO("Not yet implemented")
    }

    override fun getItemCount(): Int {
        TODO("Not yet implemented")
    }

    override fun onBindViewHolder(holder: WordListViewHolder, position: Int) {
        TODO("Not yet implemented")
    }

}
```

# RecyclerView



# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

WordListViewHolder

```
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```

```
class WordListAdapter(private val wordList: MutableList<String>) :
    RecyclerView.Adapter<WordListViewHolder>(){
```

# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

WordListViewHolder

```
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```

```
class WordListAdapter(private val wordList: MutableList<String>) :
    RecyclerView.Adapter<WordListViewHolder>(){
```

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

# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

WordListViewHolder

```
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```

```
class WordListAdapter(private val wordList: MutableList<String>) :
    RecyclerView.Adapter<WordListViewHolder>(){
```

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

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordListViewHolder {
    return WordListViewHolder(
        WordListItemBinding.inflate(
        LayoutInflater.from(parent.context), parent, false
    ))
}
```

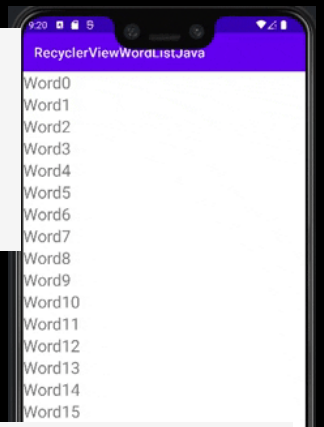
# RecyclerView

word_list_item.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/singleWord"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"/>

</LinearLayout>
```

WordListViewHolder

```
class WordListViewHolder(private val binding: WordListItemBinding) :
    RecyclerView.ViewHolder(binding.root){

        fun bind(item: String){
            binding.singleWord.text = item
        }
}
```




```
class WordListAdapter(private val wordList: MutableList<String>) :
    RecyclerView.Adapter<WordListViewHolder>(){
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordListViewHolder {
        return WordListViewHolder(
            WordListItemBinding.inflate(
            LayoutInflater.from(parent.context), parent, false
        ))
    }

    override fun getItemCount(): Int = wordList.size

    override fun onBindViewHolder(holder: WordListViewHolder, position: Int) {
        val currentItem = wordList[position]
        holder.bind(currentItem)
    }

}
```

Uniwersytet Wrocławski

# RecyclerView

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="wrap_content"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

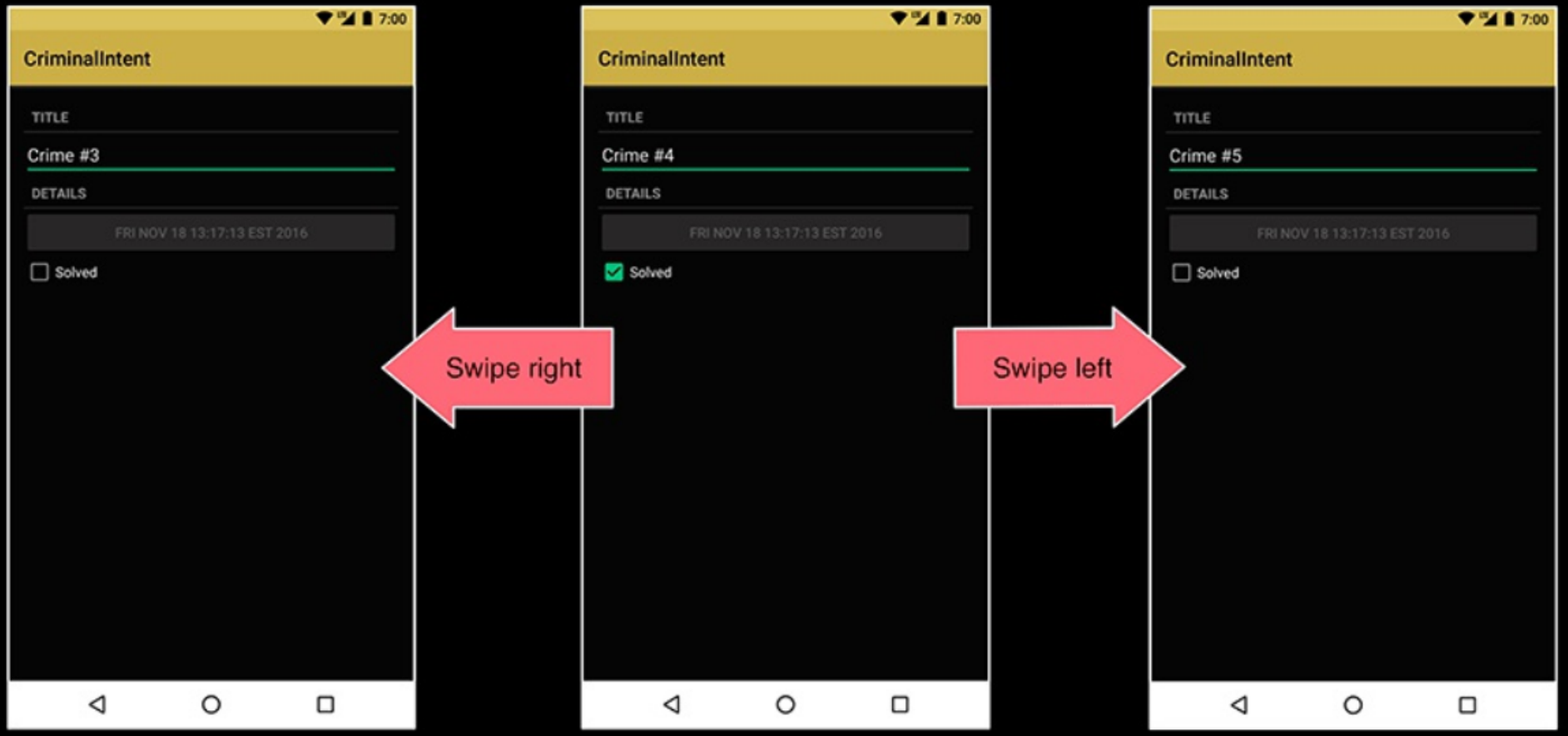
```
binding.recyclerView.apply {
    adapter = WordListAdapter(wordList)
    layoutManager = LinearLayoutManager(this@MainActivity)
}
```

Uniwersytet Wrocławski

# RecyclerView

```
fun bind(item: String) {
    binding.singleWord.text = item
    binding.root.setOnClickListener {
        val element = binding.singleWord.text
        binding.singleWord.text = "Clicked $element"
    }
}
```

# ViewPager2


7:00
CriminalIntent
TITLE
Crime #3
DETAILS
FRI NOV 18 13:17:13 EST 2016
Solved
Swipe right
7:00
CriminalIntent
TITLE
Crime #4
DETAILS
FRI NOV 18 13:17:13 EST 2016
Solved
Swipe left
7:00
CriminalIntent
TITLE
Crime #5
DETAILS
FRI NOV 18 13:17:13 EST 2016
Solved

Uniwersytet Wrocławski

# ViewPager2

```
class PagerAdapter : RecyclerView.Adapter<PagerAdapter.ViewHolder>() {

    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {

        val textView: TextView = view.findViewById(R.id.textView)

        fun bind(text: String){
            textView.text = text
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(parent.context).inflate(
                R.layout.item_view, parent, false
            )
        )
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.bind(MainActivity.values[position])
    }

    override fun getItemCount(): Int = MainActivity.values.size
}
```

Uniwersytet Wrocławski

# ViewPager2

```
class PagerAdapter(fragmentActivity: FragmentActivity) : FragmentStateAdapter(fragmentActivity) {

    override fun getItemCount(): Int = MainActivity.num

    override fun createFragment(position: Int): Fragment {
        return TemplateFragment(position)
    }
}
```

Uniwersytet Wrocławski

# ViewPager2 + TabLayout