Paixon

Recap

Recap

Activities:

- Komponente um einen einzelnen Screen anzuzeigen
- Deklaration UI im XML
- Logik in Java

Intent:

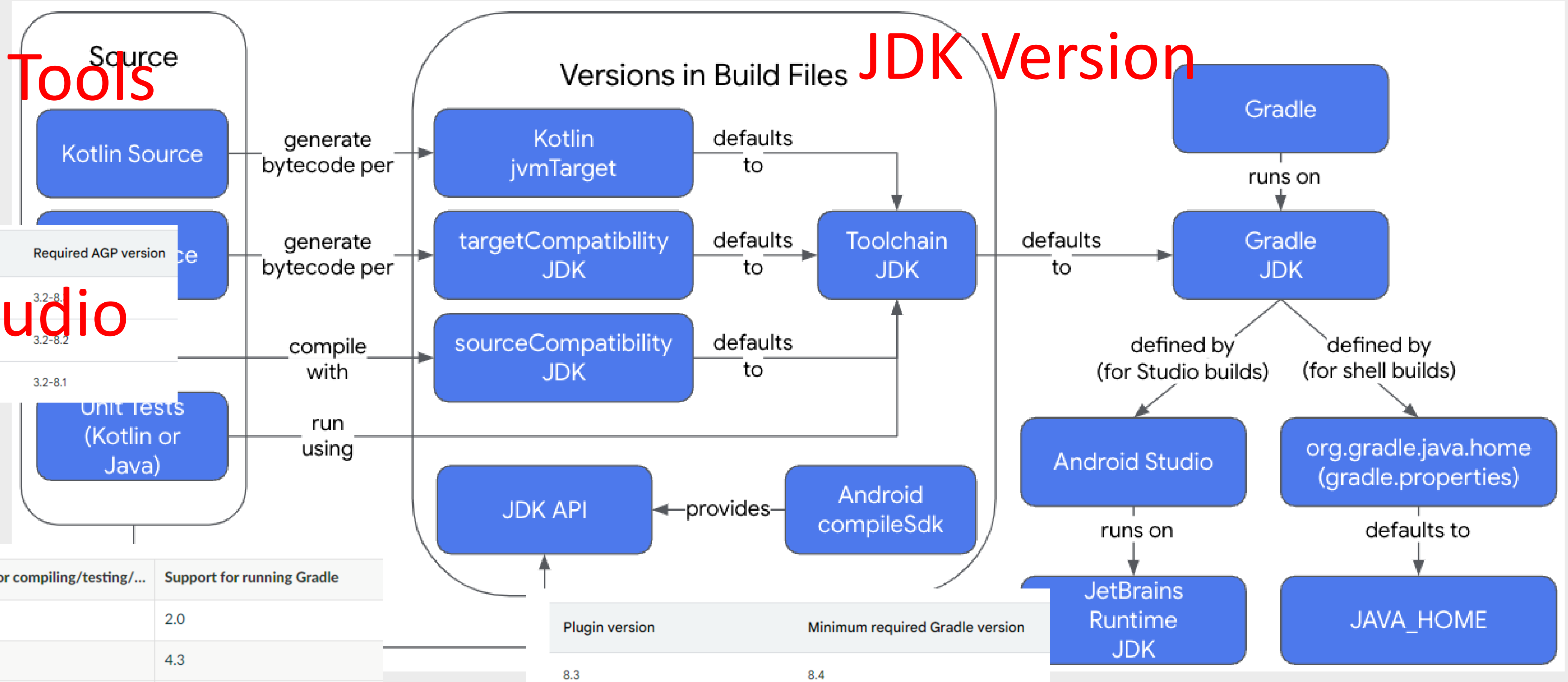
- “Absicht” eine Aufgabe auszuführen
- Explizite Intents um von A nach B zu navigieren
- Implizite Intents für generische Tasks

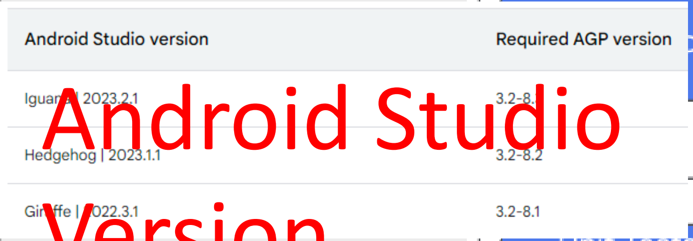
Recap

Paixon

SDK Build Tools

JDK Version





Android Studio version	Required AGP version
Iguana | 2023.2.1	3.2-8.[illegible]
Hedgehog | 2023.1.1	3.2-8.2
Giraffe | 2022.3.1	3.2-8.1

Android Studio Version

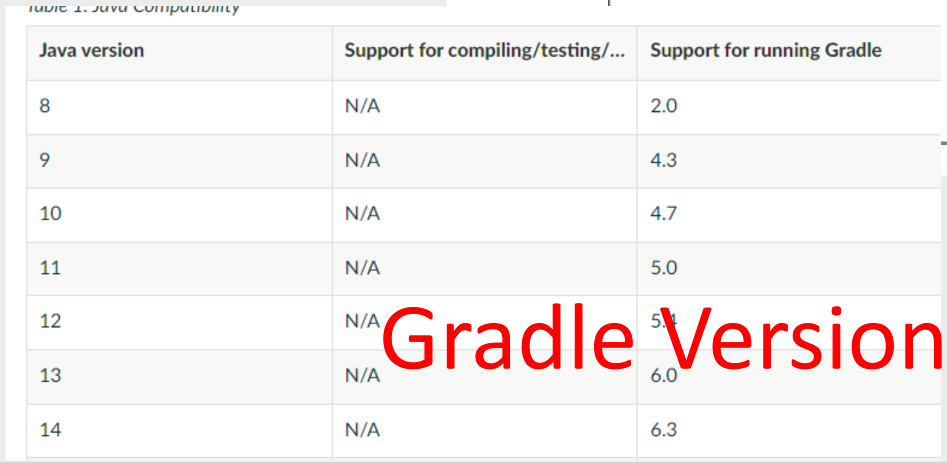
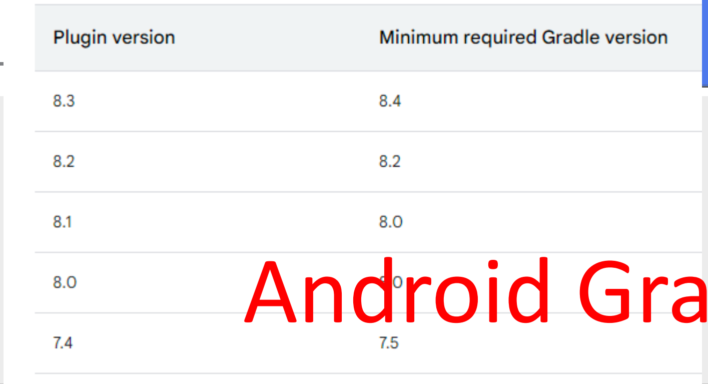


Table 1. Java Compatibility

Java version	Support for compiling/testing/...	Support for running Gradle
8	N/A	2.0
9	N/A	4.3
10	N/A	4.7
11	N/A	5.0
12	N/A	5.4
13	N/A	6.0
14	N/A	6.3

Gradle Version



Plugin version	Minimum required Gradle version
8.3	8.4
8.2	8.2
8.1	8.0
8.0	8.0
7.4	7.5

Android Gradle Plugin

Paixon

Task and Back Stack

Task and Back Stack

Task

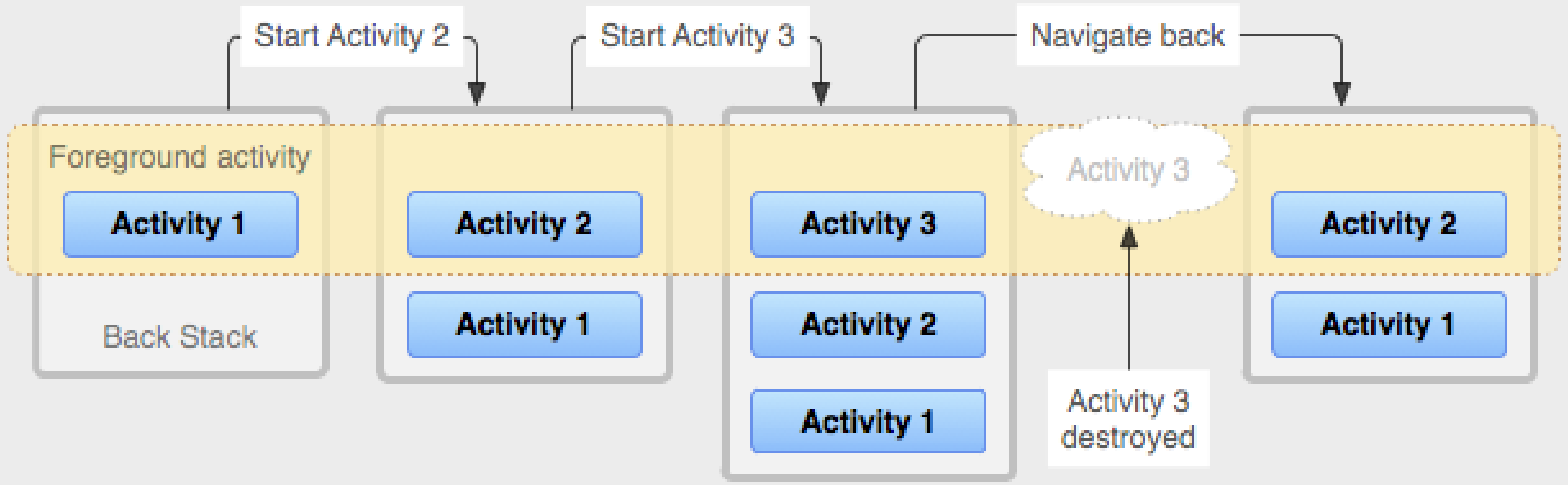
A task is a collection of activities that users interact with when performing a certain job.

Back Stack

The activities are arranged in a stack (the "back stack"), in the order in which each activity is opened.

Task and Back Stack

Paixon



Paixon

Resources

Resources

- Split resources from code
  - Layout/strings/images/…
- Resources are all kept in a “res”-folder
- Different resources for different languages or screen sizes
- Default-Resources (Fallback)
  - When not found → App crash (runtime)

Resources – Folder structure

drawable

Everything that has something todo with graphics (images, xml)

layout

Layout files for your activities, list items, ...

menu

Menu definitions

values

Simple values like strings for translations

mipmap

Launcher Icon

Resources – Providing alternative Resources

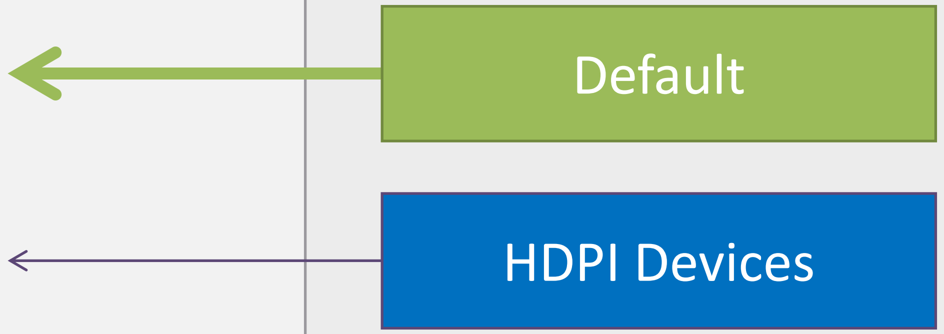
Same folder structure as default resources

Additional qualifiers

```
<resourcesfolder-name>-<qualifier>
```

Same filename as default resource

```
res/
  drawable/
     icon.png
     background.png
  drawable-hdpi/
     icon.png
     background.png
```



Resources – Providing alternative Resources

Qualifier Examples

Language / Region	values-en ; values-de-CH
Screen Size	layout-small ; layout-large
Screen Orientation	layout-land ; layout-port
Screen pixel density (dpi)	layout-mdpi ; layout-hdpi ; drawable-hdpi
Platform version (API Level)	values-v10 ; layout-v8

Other qualifiers

Mobile Country Code (MCC); Layout Direction; Smallest Width; Available Width; Available Height; Screen aspect; UI Mode; Night Mode; Touchscreen type; Keyboard availability; Primary text input method; Navigation key availability; Primary non touch navigation method

Resources – Providing alternative Resources

Qualifier Rules

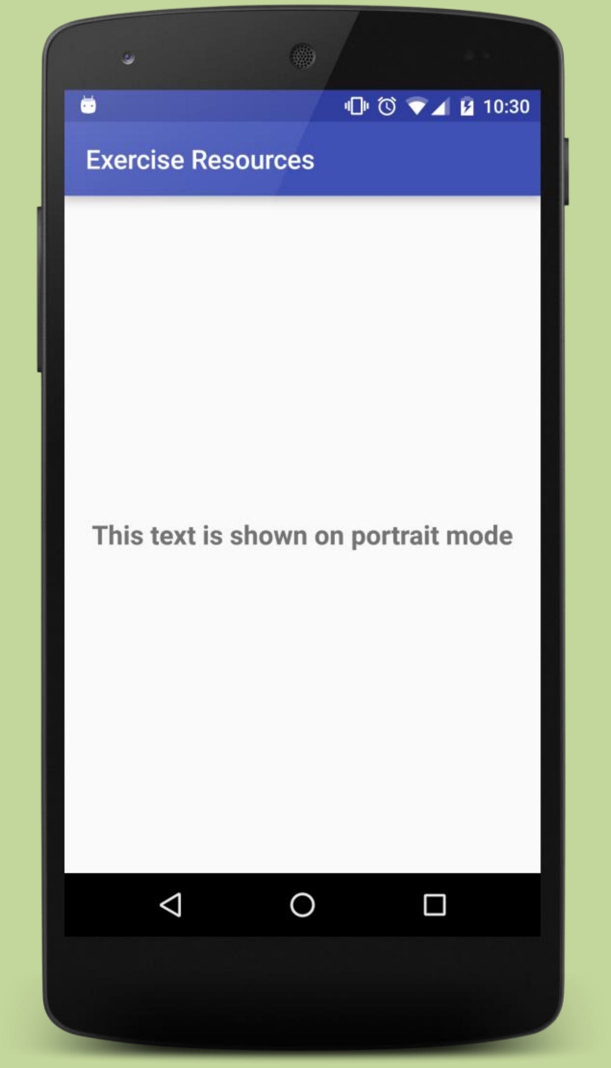
- Multiple qualifiers for single resource possible

  For example: layout-de-CH-hdpi
- Order of chained qualifiers is important
  http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources
- Case insensitive
- Multiple qualifier of same type not supported

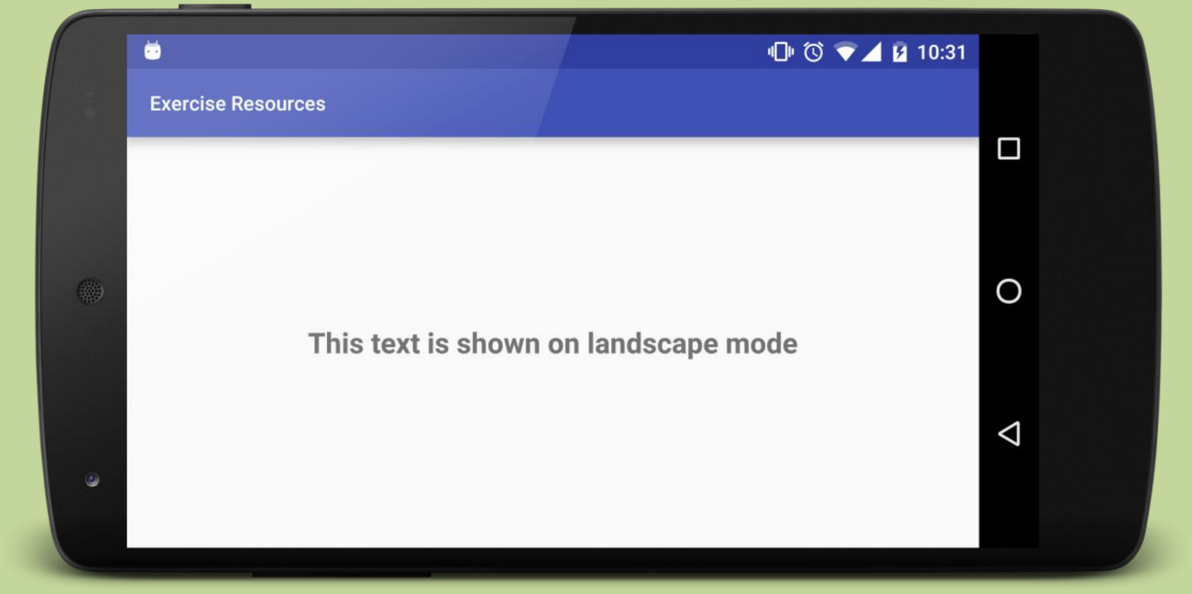
  For example: “layout-hdpi-mdpi” is not allowed

Übung Resources

Paixon



This text is shown on portrait mode



This text is shown on landscape mode

Übung Resources

Aufgaben:

1. Verstehe wie Android die korrekte Resource lädt
2. Verstehe warum und wann du Resourcen einsetzen musst
3. Stelle sicher dass die Demo-App in der Landscape-Ansicht einen anderen Text darstellt als im Portrait-Mode

Projekt: Exercise_Resources

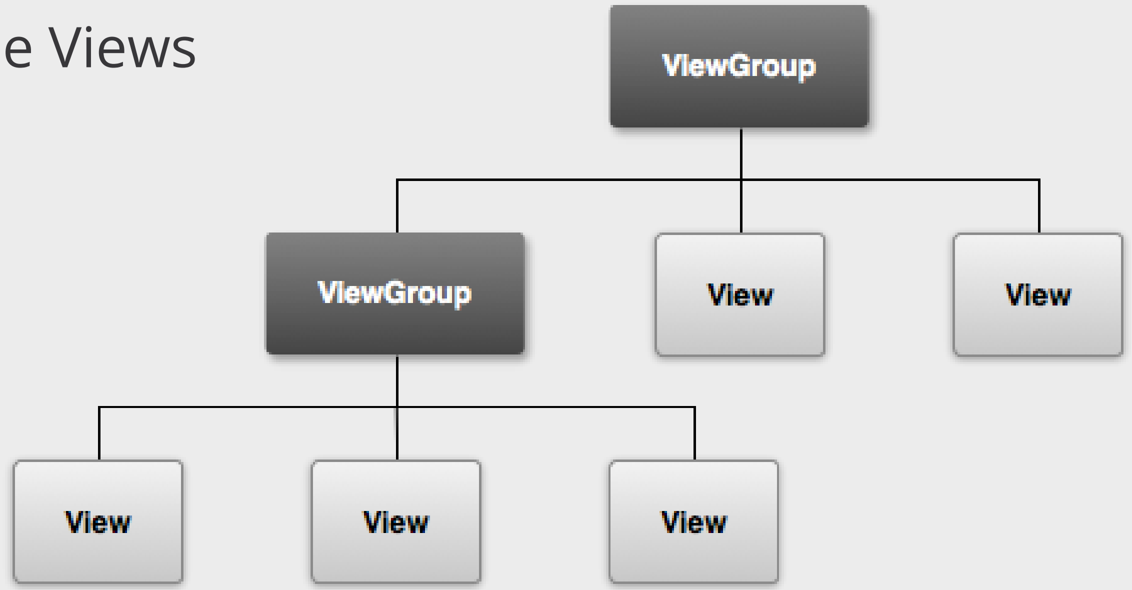
User Interface

User Interface

Paixon

- Layouts are created in XML
  - Placed in res/layouts-folder
- Everything is a View
  - TextView, EditText, Button, Spinner, etc.

- ViewGroup is a container for multiple Views



User Interface

Paixon

- Attributes
- Layout_width / layout_height mandatory
- Id-attribute to link the view

```
<TextView
        android:id="@+id/textViewDemo"
        android:textColor= "#F00"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Activity lifecycle demo" />
```

User Interface – ID Attribute

Paixon

Create a new Id with “@+id/NameOfTheId”

```
<TextView
        android:id="@+id/textViewDemo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
```

Use an existing Id with “@id/NameOfTheId”

```
<TextView
        android:layout_above="@id/theOtherTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
```

Layout Parameter

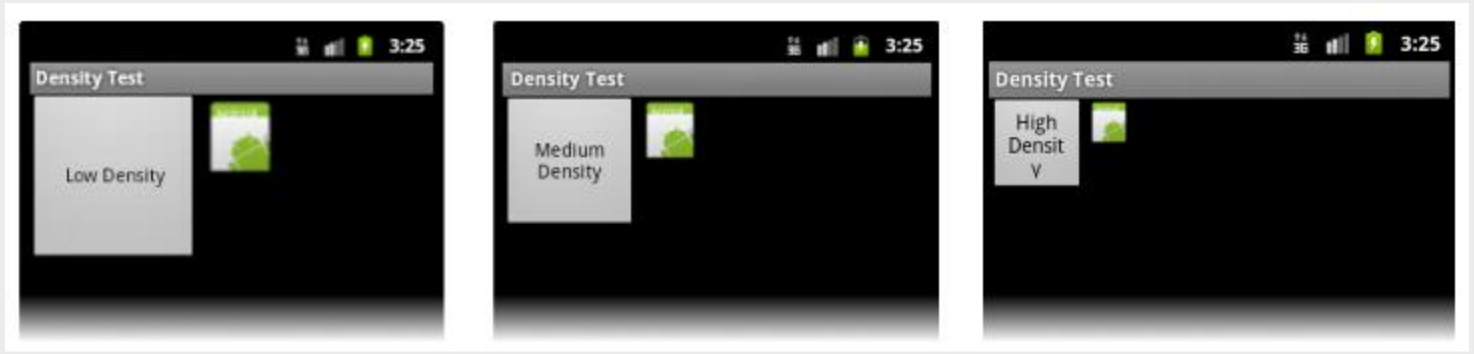
Layout Parameter

Paixon

- Define how views are measured
- Depend on the ViewGroup the view does belong to
- layout_width / layout_height are mandatory to every view
- Multiple measurement units available

Layout Parameters – Measurement units

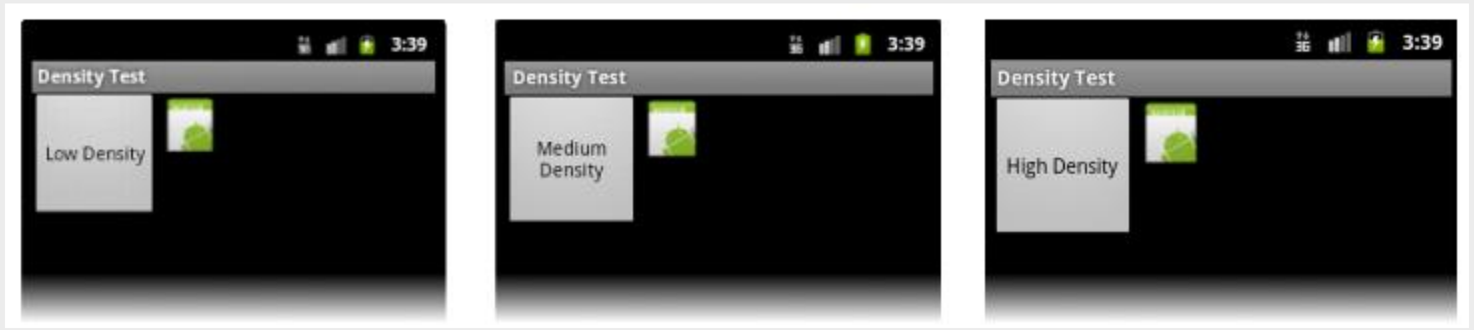
Shortcut	Name	Description
px	Pixels	Real pixel on screen
in	Inches	Size in inches
mm	Millimeters	Size in millimeters
pt	Points	1/72 in



Layout Parameters – Measurement units

Paixon

Shortcut	Name	Description
dp/dip	Density – independent pixel	Abstract unit based on the current screen density
sp	Scale – independent pixel	Same as dp but scaling with font preference



Layout Parameters – Generic Values

match_parent (früher fill_parent)

Die View nimmt soviel Platz ein, wie das Parent-Element zur Verfügung stellt.

wrap_content

Die View nimmt genau so viel Platz ein, damit der Inhalt dargestellt werden kann.

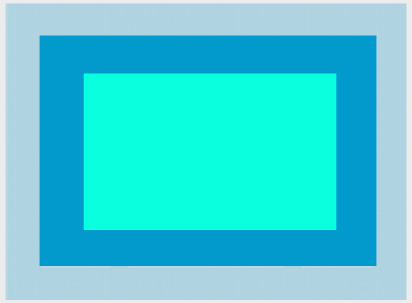
Layout Parameter

Paixon

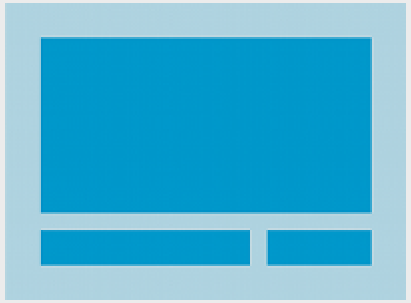
View Groups

View Groups

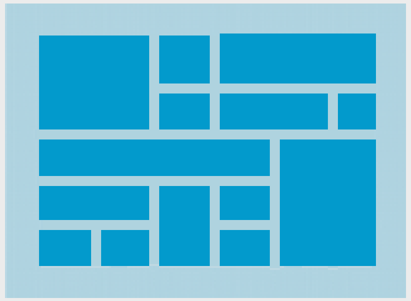
FrameLayout



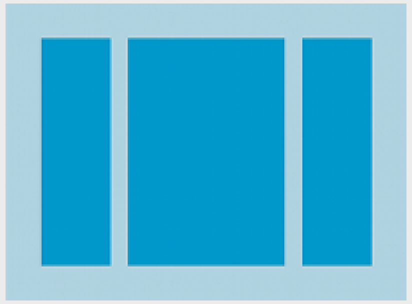
RelativeLayout



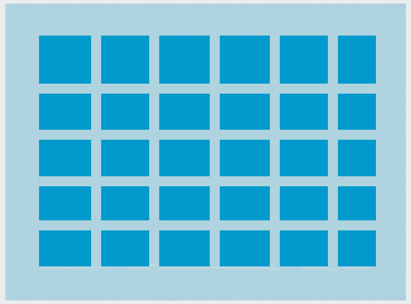
GridLayout



LinearLayout



TableLayout



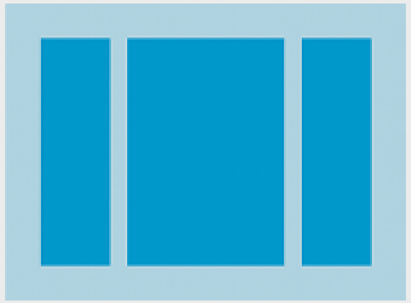
...

View Groups – LinearLayout

Paixon

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

    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />

    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Orientation (vertical/horizontal)



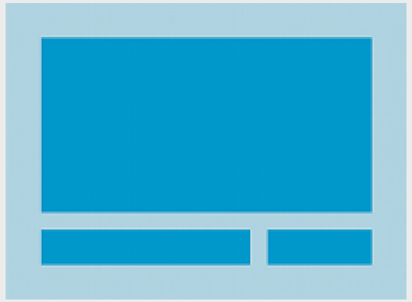
View Groups – RelativeLayout

Paixon

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/demoTextView"
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/secondDemoTextView"
        android:layout_toRightOf="@id/demoTextView"
        android:layout_alignParentBottom="true"
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

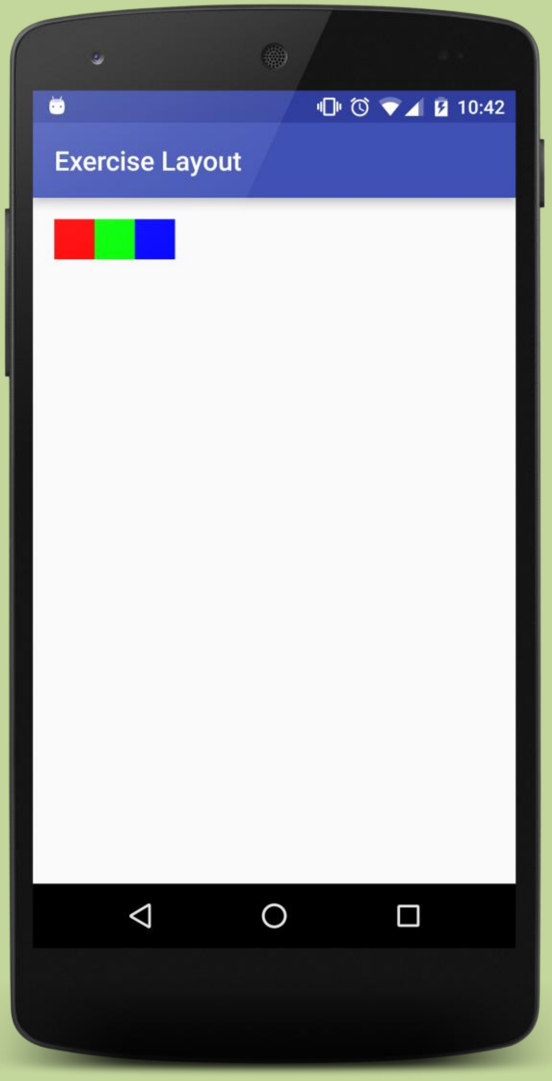
Alight right of “demoTextView”

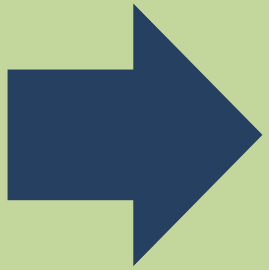
Align to bottom of parent (RelativeLayout itself)

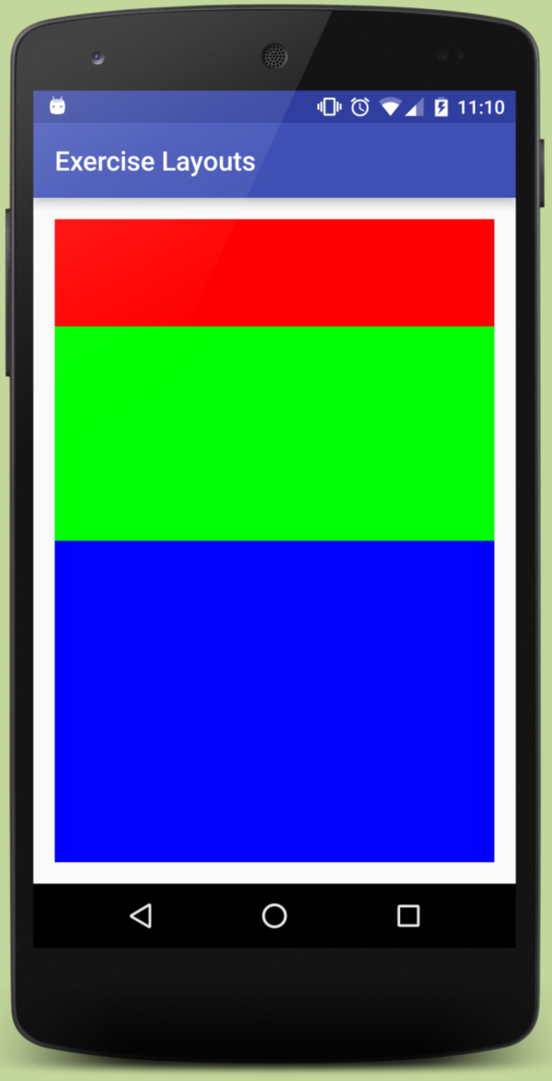


Übung 1: Layouts - Linear Layout

Paixon

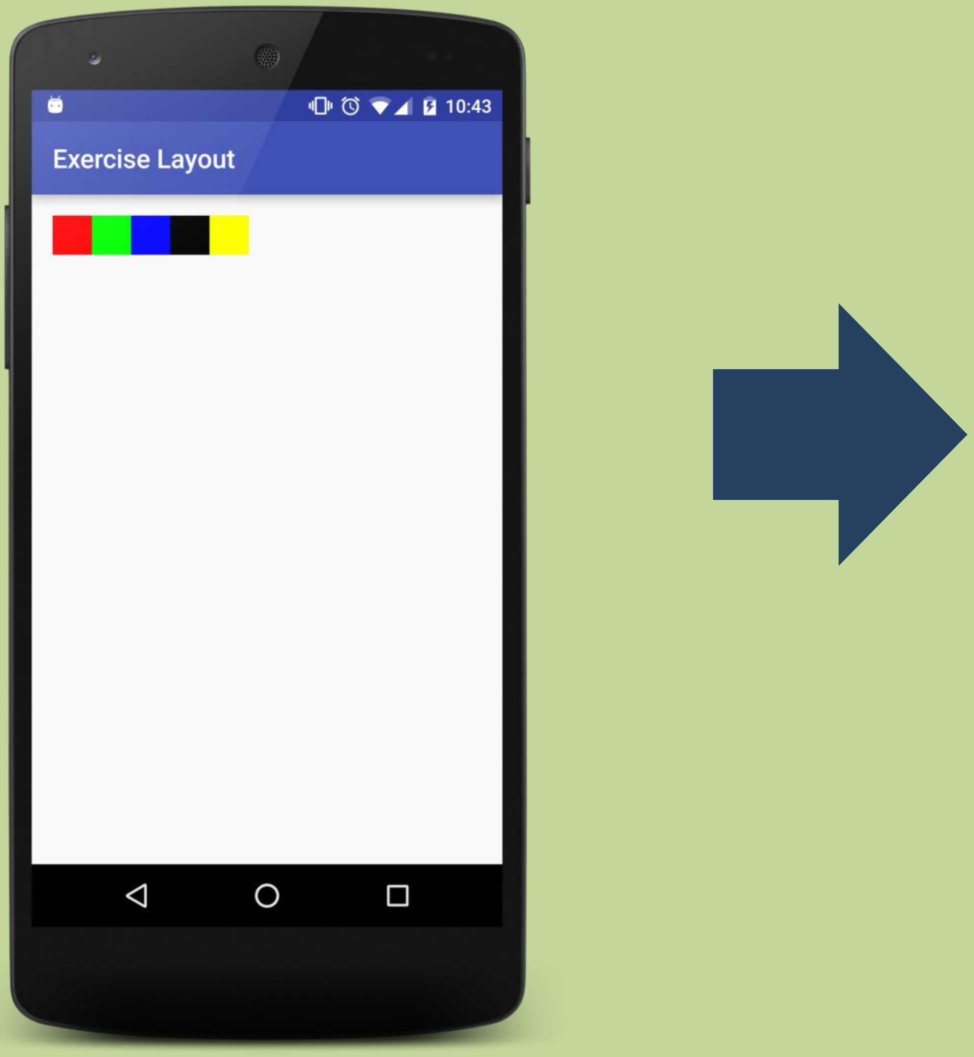


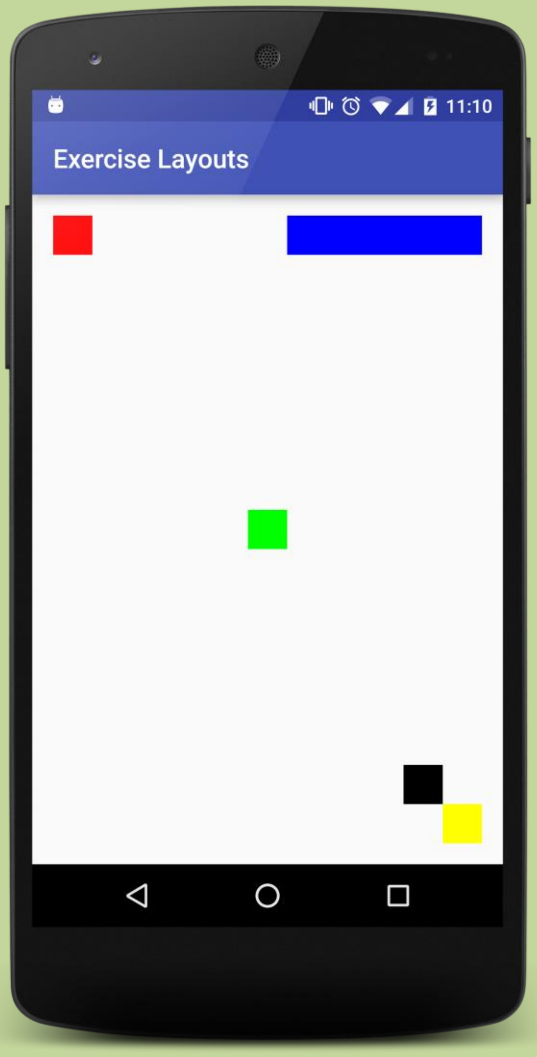




Übung 2: Layouts – Relative Layout

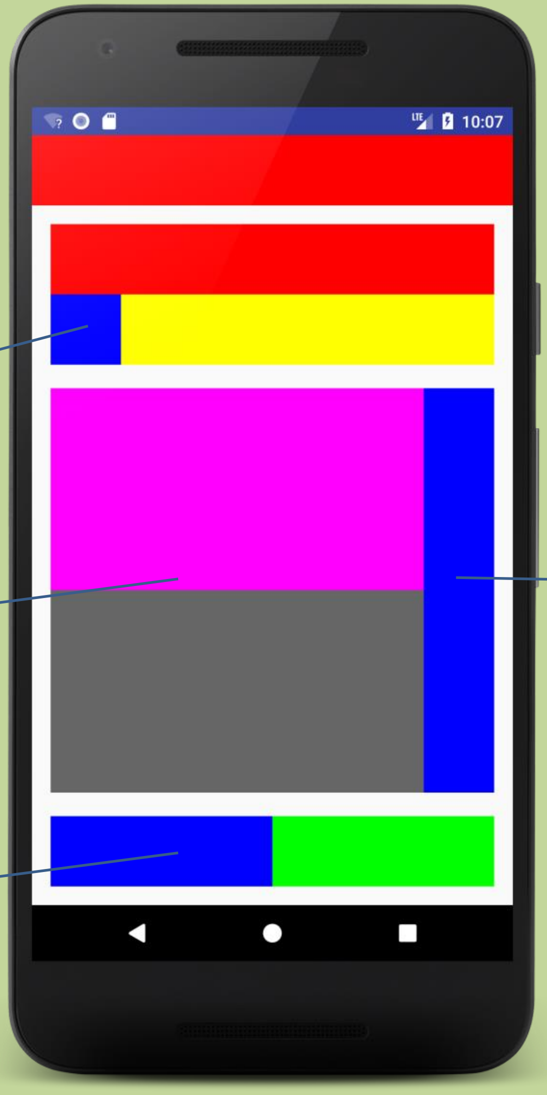
Paixon





Übung 3: Layouts - App Layout

Paixon



Fixe Breite und Höhe von 60dp

Flächen habe je 50% der Höhe

Flächen haben je 50% der Breite

Füllt die Höhe zwischen den Flächen oben und unten auf jedem Gerät aus.

Übung Layouts

Tipps LinearLayout:

- Layout orientation attribute
- Use android:layout_weight

Tipps RelativeLayout:

- Use android:layout_XXX-Attributes
- Reference other views by id “@id/otherViewName”
- Use example in UIElements-project as reference
- Use intellisense to find all possible attributes
- Maybe you have to reorder the elements

Tipps App Layout

- Kombinieren sie das LinearLayout und das RelativeLayout wo nötig.

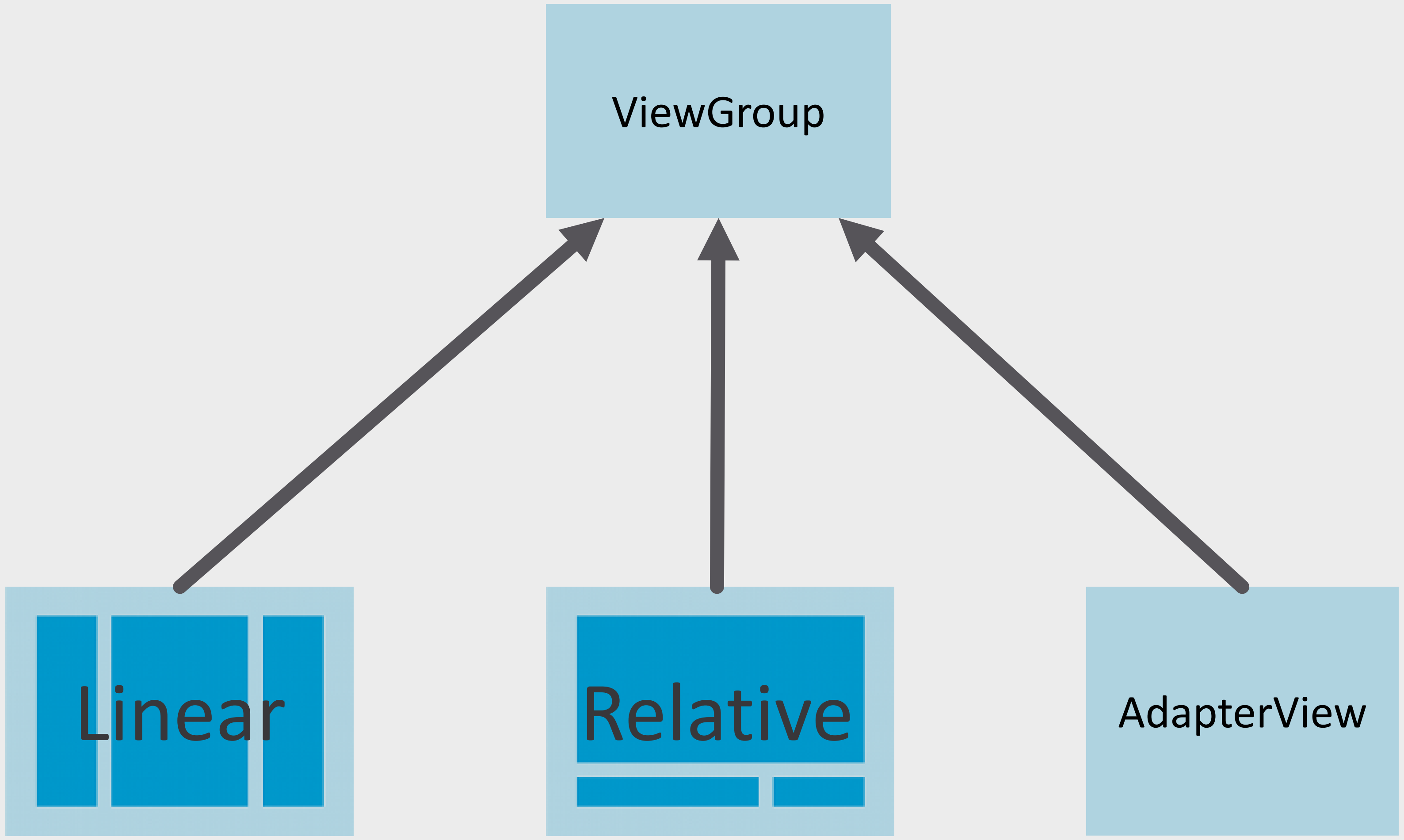
Projekt: Exercise_Layouts

Paixon

Adapter View

Adapter View

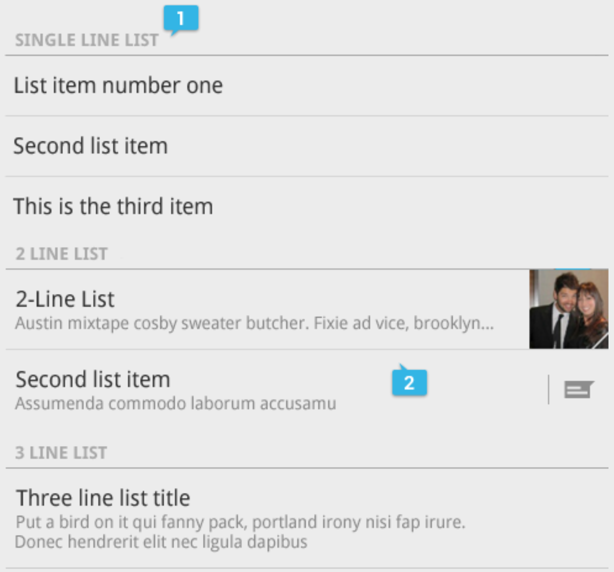
Paixon



Adapter View

Paixon

- Display data in a List/Grid
- Dynamic views



SINGLE LINE LIST

List item number one

Second list item

This is the third item

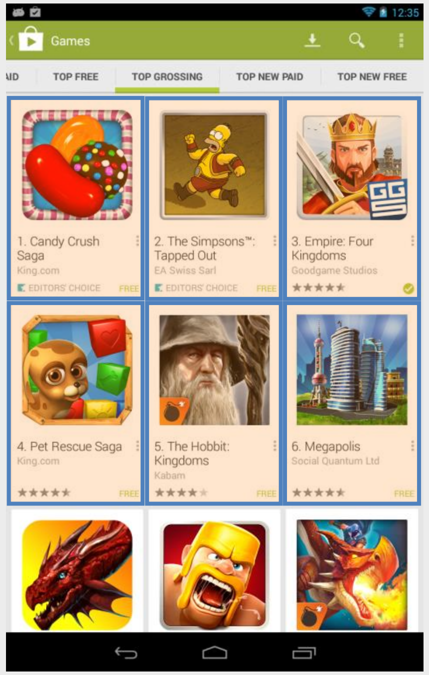
2 LINE LIST

2-Line List
Austin mixtape cosby sweater butcher. Fixie ad vice, brooklyn...

Second list item
Assumenda commodo laborum accusamu

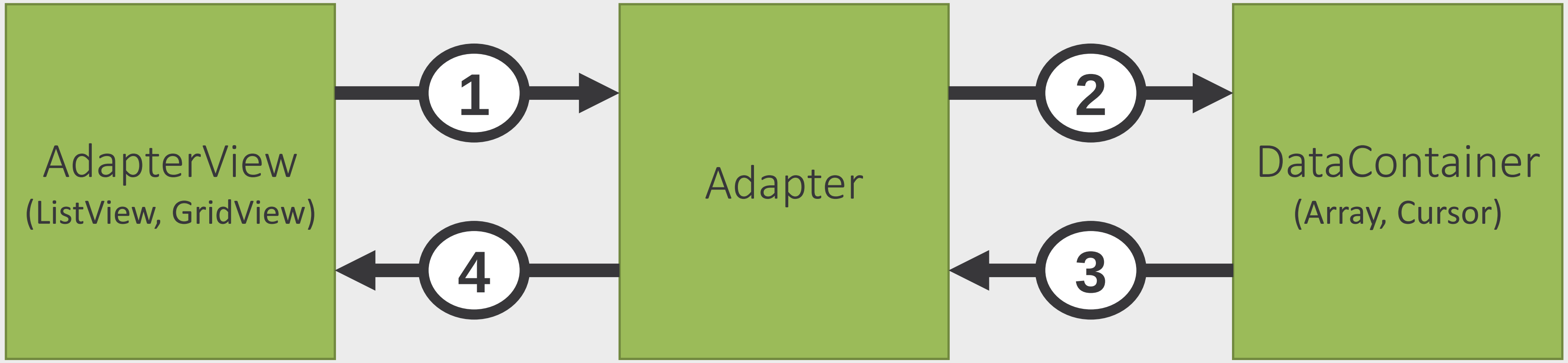
3 LINE LIST

Three line list title
Put a bird on it qui fanny pack, portland irony nisi fap irure.
Donec hendrerit elit nec ligula dapibus



Adapter View - Connect data to AdapterView

Paixon



1. AdapterView requests Views to be shown
2. Adapter requests data to be shown
3. DataContainer delivers data entries
4. Adapter creates views for every data entry

Adapter

Implement interface Adapter

- ArrayAdapter, BaseAdapter, SimpleAdapter, SpinnerAdapter, …

Methods to implement

- getCount()
- getItem(int position)
- getItemId(int position)
- getViewType(int position)
- getViewTypeCount()
- getView(int position, View convertView, ViewGroup parent)

Adapter - Signature of getView(...)

Paixon

```
public abstract View getView (int position, View convertView, ViewGroup parent)
```

position	Position of the item within the adapters data-set
convertView	Cached view to be reused
parent	Reference to the parent (AdapterView)

Was has to be done in this method?

- Create the view when “convertView” is null
- Set data to the given view

How can multiple types of views be implemented?

- Use getViewTypeCount() to define how many different View-Types you have
- Use getViewType() to tell android which View-Type the current item should be

Adapter - Create Views in getView(...)

Paixon

How can views defined as XML-Resource be instantiated?

- Use a LayoutInflater (System-Service)
- Use inflate(int layout, ViewGroup root, bool attachToRoot)

```
String name = Context.LAYOUT_INFLATER_SERVICE;
LayoutInflater infl = (LayoutInflater) context.getSystemService(name);
View view = infl.inflate(R.layout.spaced_list_item, parent, false);
```

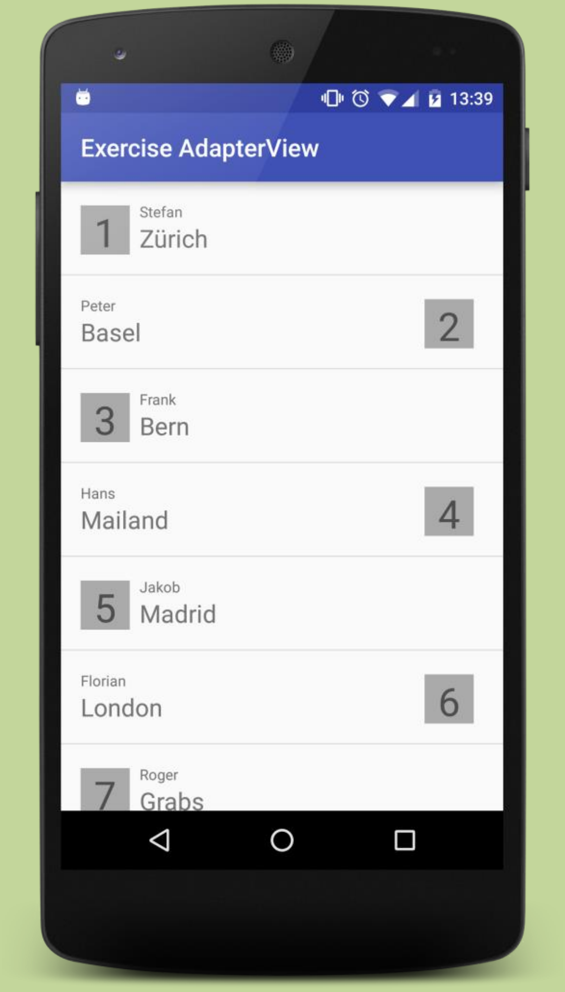
Übung AdapterView

Paixon

Aufgaben

1. Sämtliche Klassen analysieren und Zusammenhang zwischen MainActivity (main_activity.xml), PersonAdapter und der ListView verstehen
2. Bug beheben, dass der Name und die Adresse nicht angezeigt werden
3. Bei jedem zweiten Eintrag soll die ID auf der rechten Seite angezeigt werden.
4. Warum muss bei “idTextView.setText(person.getId() + “”)” die entsprechende ID in einen String umgewandelt werden?

Projekt: Exercise_AdapterView



13:39

Exercise AdapterView

1	Stefan Zürich	
	Peter Basel	2
3	Frank Bern	
	Hans Mailand	4
5	Jakob Madrid	
	Florian London	6
7	Roger Grabs	