Android Programmierung mit Java Studiengang MI

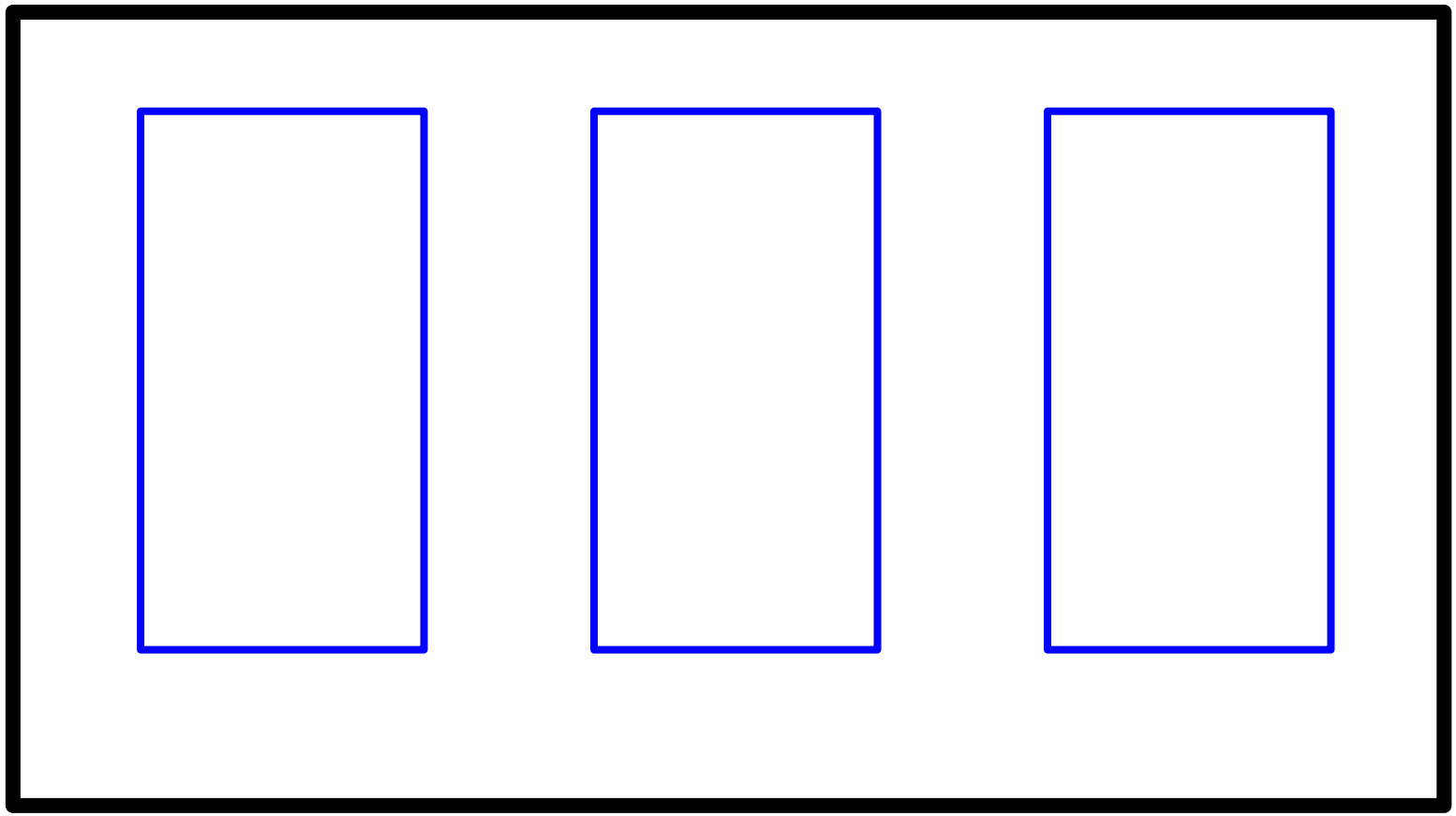
- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- http://mwilhelm.hs-harz.de
- Raum 2.202
- Tel. 03943 / 659 338

Gliederung

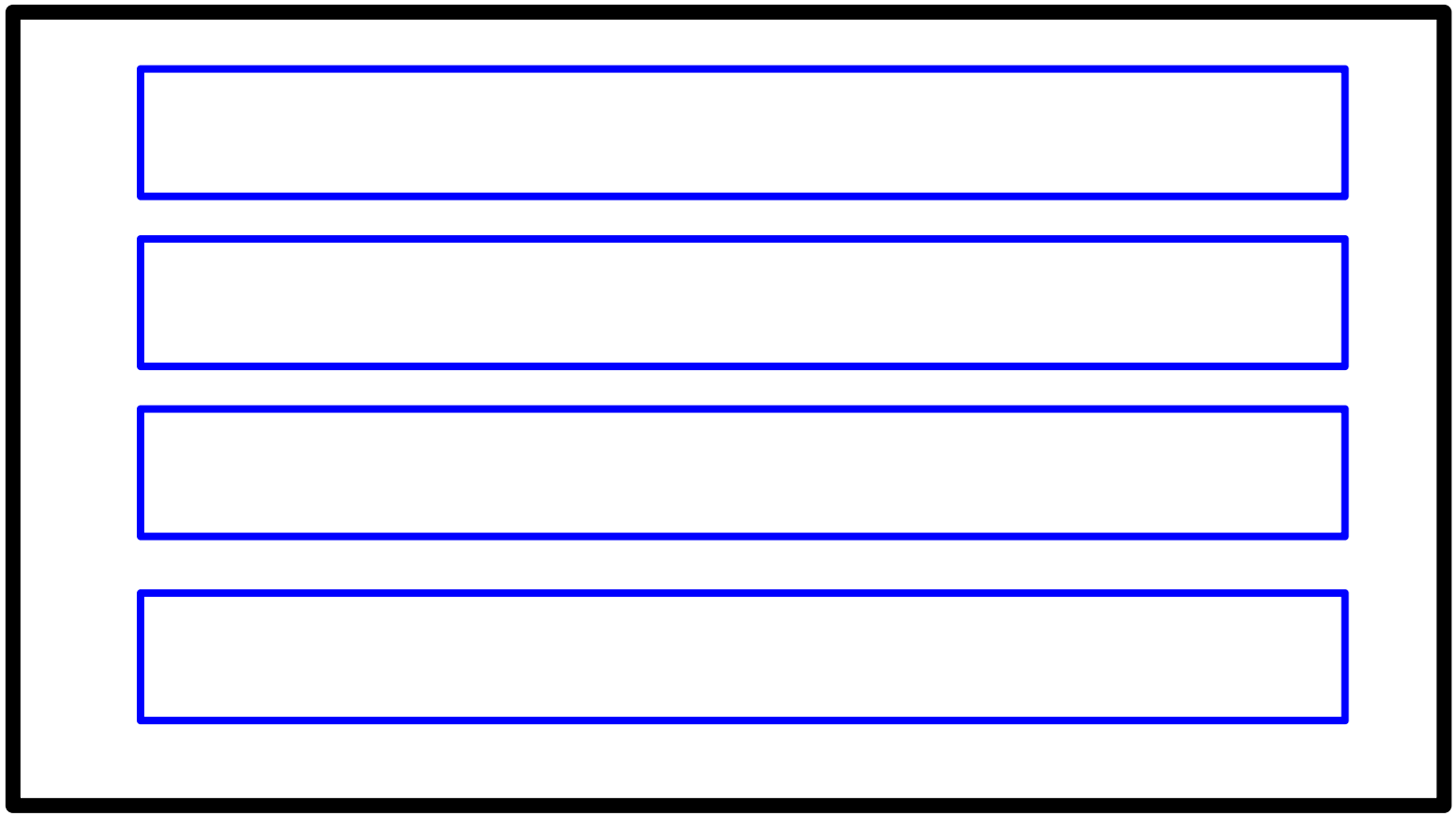
Überblick:

- Einleitung
- **Layouts**
  - LinearLayout
  - TableLayout
  - GridLayout
  - ConstraintLayout
  - RelativeLayout
  - Fragment
- Single page
- Multi-Page
- I/O
- Datenbanken
- Sensoren
- Web
- Thread, modale Dialoge

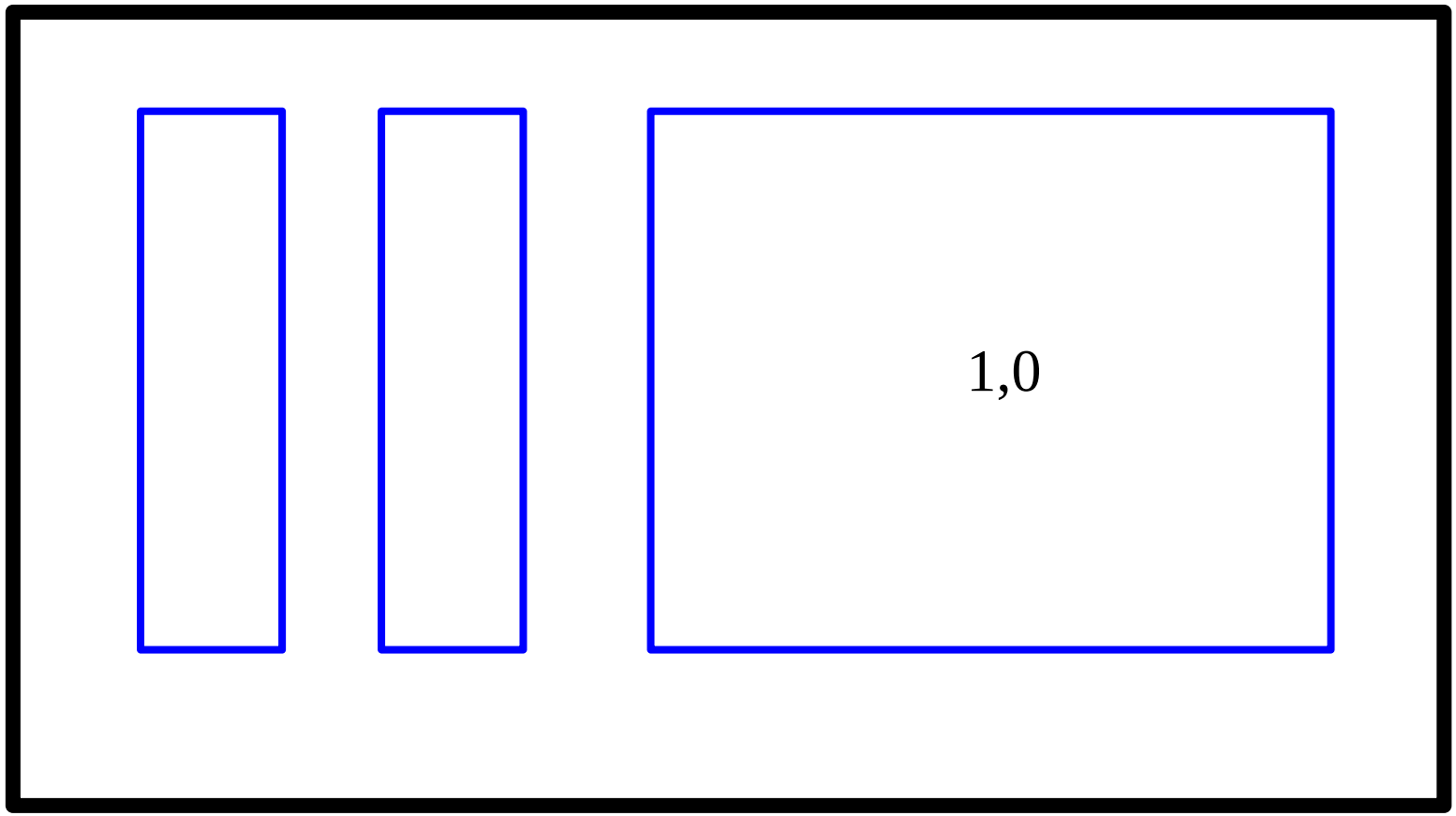
LinearLayout



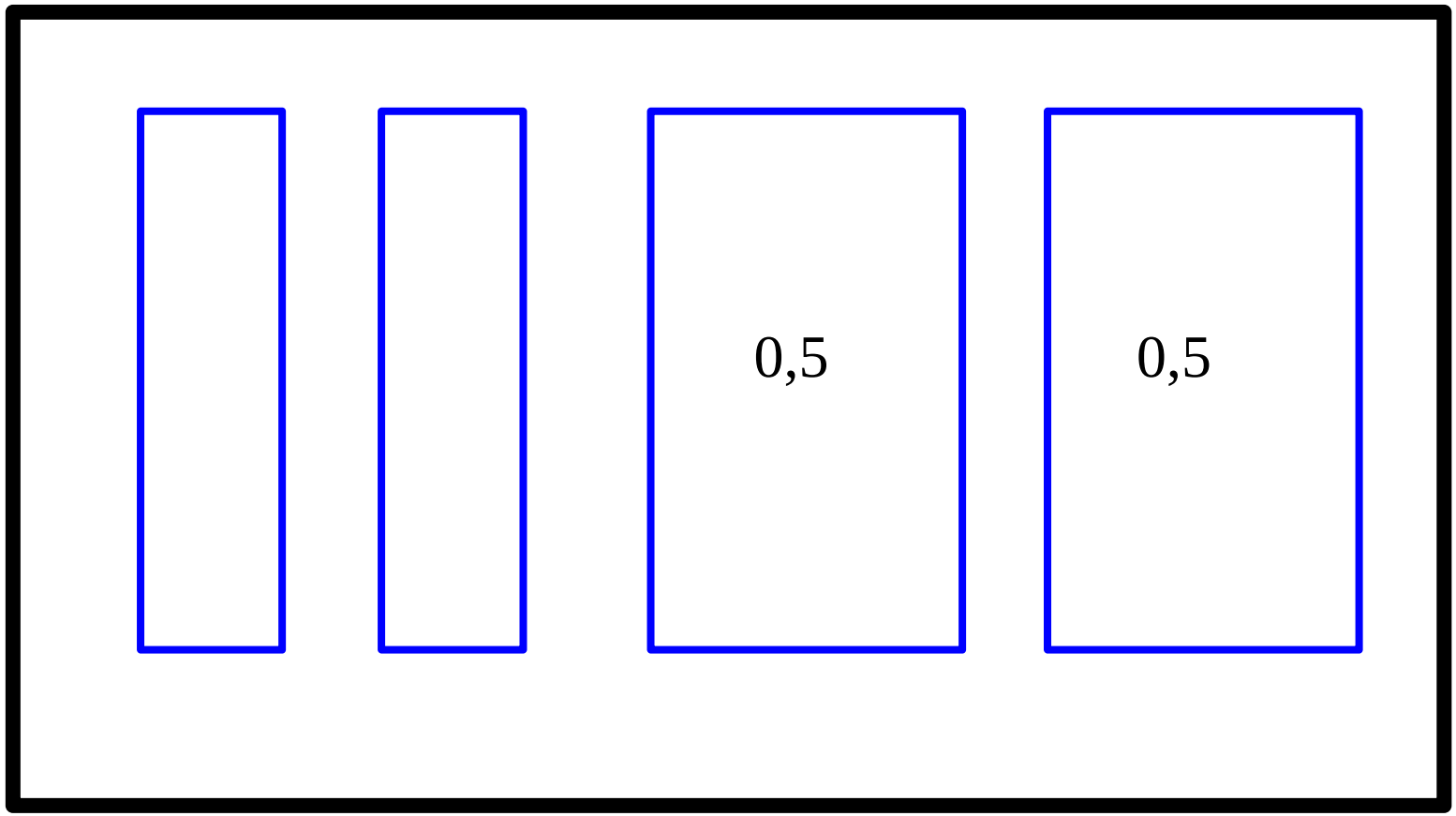
LinearLayout



LinearLayout: weight=1.0



LinearLayout: weight=0.5, wight=0.5



LinearLayout: Attribute

- Wie ein Regal, à la StackPanel in WPF
- Orientation
  - horizontal
  - vertikal
- **Geht auch verschachtelt**
- **android:layout_width=**
  - "match_parent" komplette Breite
  - "wrap_content" nur benötigte Breite
- **android:layout_height=**
  - "match_parent" komplette Höhe
  - "wrap_content" nur benötigte Höhe

LinearLayout: Attribute

android:divider

- Möglichst unsichtbarer Trennstrich.
- Funktioniert aber auch mit Margin

android:gravity

- Definiert die Ausrichtungen in der Zelle.
- fill_horizontal, fill, bottom, center, center_horizontal, right,
- clip_horizontal, clip_vertical, end, fill_vertical,left, start, top

android:measureWithLargestChild

- Wenn true, dann erhalten alle gewichteten Elemente die Minimalgröße des größten Elementes.

android:weightSum

- Normalwerweise muss die Summe der Gewichtigungen 1.0 ergeben. Hiermit kann man die Summe ändern.

LinearLayout: Aufbau

```
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:id="@+id/activity_main"
   android:orientation="horizontal">

        // Elemente
    <TextView    />
    <EditText     />
    <Switch        />

</LinearLayout>
```

LinearLayout: Zwei Editoren mit Gewichtung

```
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:id="@+id/activity_main"
   android:orientation="horizontal">

   <TextView   … />
    <EditText
      android:layout_weight="0.67"
      android:inputType="textMultiLine"  />
   <EditText
      android:layout_weight="0.33"
      android:inputType="textMultiLine"/>
</LinearLayout>
```

LinearLayout: Zwei Editoren mit Gewichtung

```
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:weightSum="30"
   android:orientation="horizontal">

   <TextView   … />
    <EditText
      android:layout_weight="20"
      android:inputType="textMultiLine"  />
   <EditText
      android:layout_weight="10"
      android:inputType="textMultiLine"/>
</LinearLayout>
```

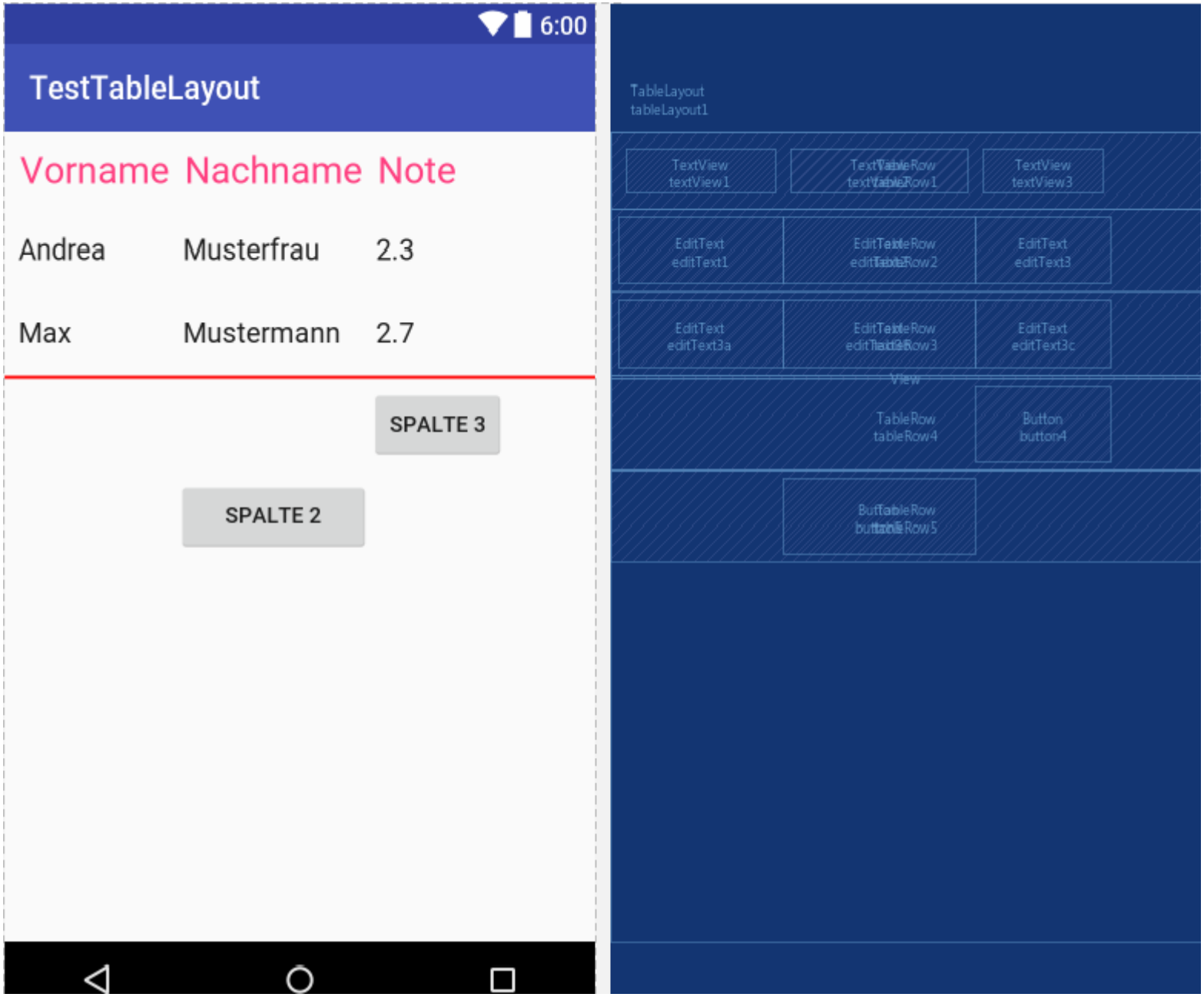
LinearLayout: Gravity-Konstanten

Konstante	Value	Beschreibung
top	0x30	Objekt oben verankert, Größe bleibt.
bottom	0x50	Objekt unten verankert, Größe bleibt.
left	0x03	Objekt links verankert, Größe bleibt.
right	0x05	Objekt rechts verankert, Größe bleibt.
center_vertical	0x10	Objekt vertikal zentriert, Größe bleibt.
fill_vertical	0x70	Objekt wird komplett gestreckt.
center_horizontal	0x01	Objekt horizontal zentriert, Größe bleibt.
fill_horizontal	0x07	Objekt wird komplett gestreckt.
center	0x11	Objekt vertikal und horizintal zentriert, Größe bleibt.
fill	0x77	Objekt wird komplett gestreckt.
clip_vertical	0x80	Zusätzliche Option. Die Ecken verschmelzen mit dem Parent.
clip_horizontal	0x08	Zusätzliche Option. Die Ecken verschmelzen mit dem Parent.
start	0x00800003	Setzt Objekt ganz oben hin, Größe bleibt.
end	0x00800005	Setzt Objekt ganz unten hin, Größe bleibt.

TableLayout

- Das TableLayout funktioniert wie die HTML-Tabelle
- Man definiert jeweils eine Zeile mit TableRow
- Spalten zusammenfügen
  - android:layout_span="2„
- Spalten direkt adressieren
  - android:layout_column="2"

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@+id/tableLayout1"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent" >

  <TableRow
      android:id="@+id/tableRow1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="5dip" >

      <TextView …   />
      <TextView …   />
      <TextView …   />

   </TableRow>

</TableLayout>
```


6:00
TestTableLayout
Vorname Nachname Note
Andrea Musterfrau 2.3
Max Mustermann 2.7
SPALTE 3
SPALTE 2
TableLayout
tableLayout1
TextView
textView1
TextView
textView3
EditText
editText1
EditText
editText3
EditText
editText3a
EditText
editText3c
TableRow
tableRow4
Button
button4

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tableLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <!-- 2 columns -->
    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >

        <TextView
            android:id="@+id/textView1"
            android:text="Vorname"
            android:layout_margin="5dp"
            android:textSize="24dp"
            android:textColor="@color/colorAccent"
            android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<TextView
        android:id="@+id/textView2"
        android:text="Nachname"
        android:layout_margin="5dp"
        android:textSize="24dp"
        android:textColor="@color/colorAccent"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView
        android:id="@+id/textView3"
        android:text="Note"
        android:layout_margin="5dp"
        android:textSize="24dp"
        android:textColor="@color/colorAccent"
        android:textAppearance="?android:attr/textAppearanceLarge" />
  </TableRow>
```

```
<TableRow
      android:id="@+id/tableRow2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="5dip" >

      <EditText
         android:id="@+id/editText1"
         android:text="Andrea" />
      <EditText
         android:id="@+id/editText2"
         android:text="Musterfrau" />
      <EditText
         android:id="@+id/editText3"
         android:text="2.3" />
   </TableRow>
```

```
<TableRow
        android:id="@+id/tableRow3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >

        <EditText
            android:id="@+id/editText3a"
            android:text="Max" />
        <EditText
            android:id="@+id/editText3b"
            android:text="Mustermann" />
        <EditText
            android:id="@+id/editText3c"
            android:text="2.7" />
    </TableRow>
    <!-- just draw a red line -->
    <View
        android:layout_height="2dip"
        android:background="#FF0000" />
```

```
    <TableRow
        android:id="@+id/tableRow4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >
        <Button
            android:id="@+id/button4"
            android:layout_column="2"
            android:text="Spalte 3" />
    </TableRow>
    <TableRow
        android:id="@+id/tableRow5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >
        <Button
            android:id="@+id/button5"
            android:layout_column="1"
            android:text="Spalte 2" />
    </TableRow>
</TableLayout>
```

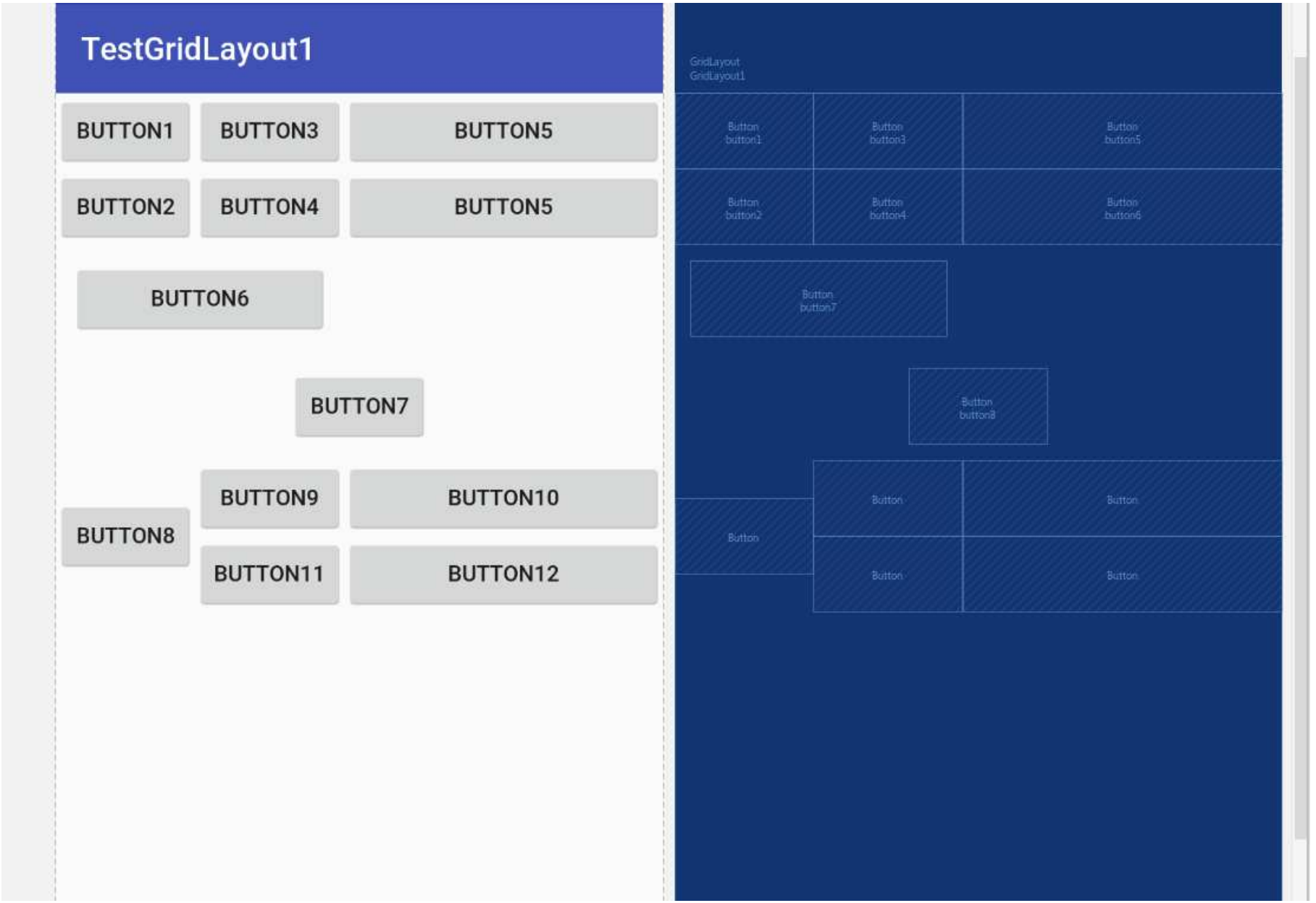
GridLayout

- Ähnlich wie TabletLayout
- **Vorab** werden aber die Spalten und Zeilen definiert
- Die Zelle jedes UI-Elementes wird über column und row definiert. Ähnlich dem GridBagLayout.
- Mit dem Attribut „gravity“ bestimmt man die Platzierung innerhalb der Zelle. Ähnlich dem GridBagLayout.
  - Dazu gibt es Konstanten
- Das Zusammenfassen von Zellen geht mit:
  - columnSpan
  - rowSpan

GridLayout: Gravity-Konstanten

Konstante	Value	Beschreibung
top	0x30	Objekt oben verankert, Größe bleibt.
bottom	0x50	Objekt unten verankert, Größe bleibt.
left	0x03	Objekt links verankert, Größe bleibt.
right	0x05	Objekt rechts verankert, Größe bleibt.
center_vertical	0x10	Objekt vertikal zentriert, Größe bleibt.
fill_vertical	0x70	Objekt wird komplett gestreckt.
center_horizontal	0x01	Objekt horizontal zentriert, Größe bleibt..
fill_horizontal	0x07	Objekt wird komplett gestreckt.
center	0x11	Objekt vertikal und horizintal zentriert, Größe bleibt.
fill	0x77	Objekt wird komplett gestreckt.
clip_vertical	0x80	Zusätzliche Option. Die Ecken verschmelzen mit dem Parent.
clip_horizontal	0x08	Zusätzliche Option. Die Ecken verschmelzen mit dem Parent.
start	0x00800003	Setzt Objekt ganz oben hin, Größe bleibt.
end	0x00800005	Setzt Objekt ganz unten hin, Größe bleibt.

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="3"
    android:rowCount="7"
    android:orientation="vertical"
    tools:context=".GridXMLActivity" >

        <Button …
            android:layout_column="0"
            android:layout_row="0"
            android:layout_gravity="fill_horizontal"/>

</GridLayout>
```


TestGridLayout1
BUTTON1
BUTTON3
BUTTON5
BUTTON2
BUTTON4
BUTTON5
BUTTON6
BUTTON7
BUTTON8
BUTTON9
BUTTON10
BUTTON11
BUTTON12
GridLayout
GridLayout1
Button
button1
Button
button3
Button
button5
Button
button2
Button
button4
Button
button6
Button
button7
Button
button8
Button
Button
Button
Button
Button
Button

GridLayout

Bemerkungen zum Beispiel

- Der Button „Button8“ hat ein RowSpan.
- Wenn aber keine weitere Reihe danach kommt, geht dieser Schalter dann bis zum Ende des Views.
- Abhilfe:
  - Man trägt eine zusätzliche Reihe ein.
  - Diese braucht kein Element haben !
- Button8:
  - android:layout_gravity="center_vertical"

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="3"
    android:rowCount="7"
    android:orientation="vertical"
    tools:context=".GridXMLActivity" >
        <Button
            android:id="@+id/button1"
            android:layout_column="0"
            android:layout_row="0"
            android:layout_gravity="fill_horizontal"
            android:text="Button1" />
        <Button
            android:id="@+id/button2"
            android:layout_column="0"
            android:layout_row="1"
            android:layout_gravity="fill_horizontal"
            android:text="Button2" />
```

```
<Button
   android:id="@+id/button3"
   android:layout_column="1"
   android:layout_row="0"
   android:layout_gravity="fill_horizontal"
   android:text="Button3" />
<Button
   android:id="@+id/button4"
   android:layout_column="1"
   android:layout_row="1"
   android:layout_gravity="fill_horizontal"
   android:text="Button4" />

<Button
   android:id="@+id/button5"
   android:layout_column="2"
   android:layout_row="0"
   android:layout_gravity="fill_horizontal"
   android:text="Button5" />
<Button
   android:id="@+id/button7"
   android:layout_column="0"
   android:layout_row="2"
   android:layout_columnSpan="2"
   android:layout_gravity="fill_horizontal"
   android:layout_margin="10dp"
   android:layout_width="wrap_content"
   android:text="Button6" />

<Button
   android:id="@+id/button8"
   android:layout_column="0"
   android:layout_row="3"
   android:layout_columnSpan="3"
   android:layout_gravity="center_horizontal"
   android:layout_margin="10dp"
   android:layout_width="wrap_content"
   android:text="Button7" />
```

```
<Button
    android:layout_column="0"
    android:layout_row="4"
    android:layout_gravity="center_vertical"
    android:layout_rowSpan="2"
    android:text="Button8" />

<Button
    android:layout_column="1"
    android:layout_row="4"
    android:layout_gravity="fill_horizontal"
    android:text="Button9" />

<Button
    android:layout_column="2"
    android:layout_row="4"
    android:layout_gravity="fill_horizontal"
    android:text="Button10" />

<Button
    android:layout_column="1"
    android:layout_row="5"
    android:layout_gravity="fill_horizontal"
    android:text="Button11" />

<Button
    android:layout_column="2"
    android:layout_row="5"
    android:layout_gravity="fill_horizontal"
    android:text="Button12" />

</GridLayout>
```

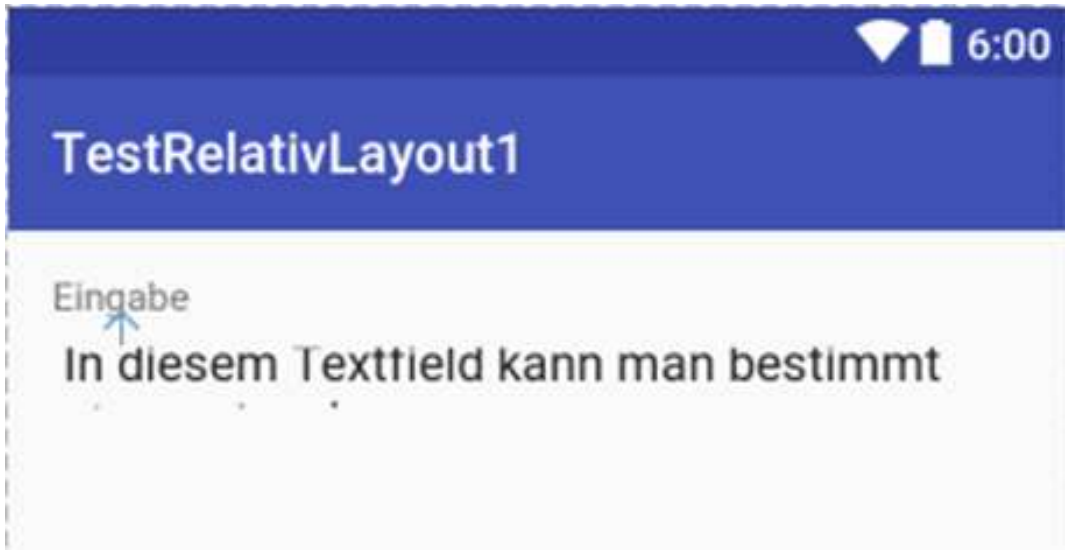
RelativeLayout

- Ähnlich dem Layout in iOS
- Alle UI-Elemente orientieren sich am
  - parent
  - sibling
- Die Verknüpfung geschieht durch die Id‘s

RelativeLayout

- android:layout_above
- android:layout_alignBaseline
- android:layout_alignBottom
- android:layout_alignEnd
- **android:layout_alignLeft**
- android:layout_alignParentBottom
- android:layout_alignParentEnd
- **android:layout_alignParentLeft**
- android:layout_alignParentRight
- android:layout_alignParentStart
- android:layout_alignParentTop
- android:layout_alignRight
- android:layout_alignStart
- android:layout_alignTop
- android:layout_alignWithParentIfMissing
- android:layout_below
- android:layout_centerHorizontal
- android:layout_centerInParent
- android:layout_centerVertical
- android:layout_toEndOf
- android:layout_toLeftOf
- android:layout_toRightOf
- android:layout_toStartOf

1. Aufgabe




```
<TextView
    android:id="@+id/textview1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Eingabe"  />
<EditText
    android:id="@+id/bn2"
    android:layout_width="…"
    android:layout_height="..."
    android:text="In diesem Textfield kann man bestimmt etwas
eingeben" />
```

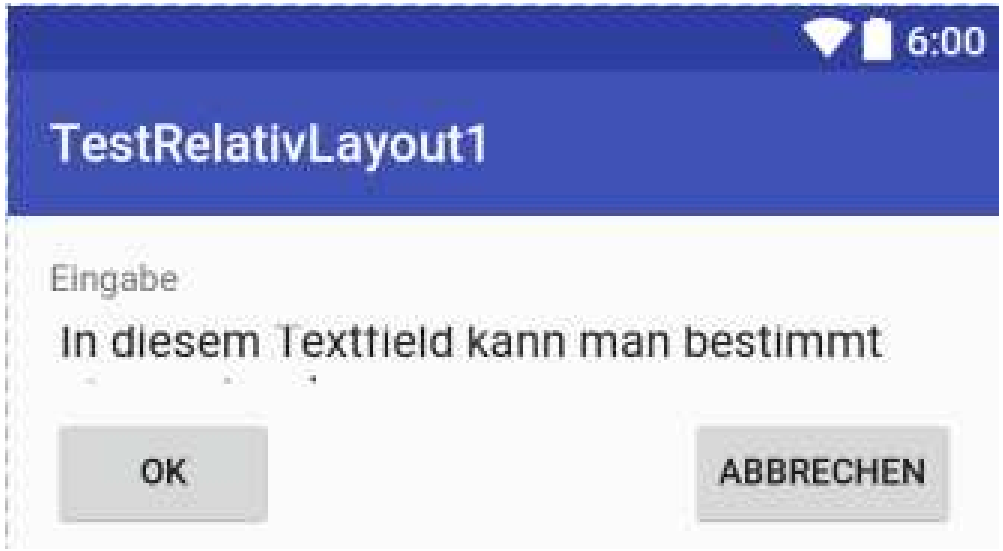
2. Aufgabe


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Klick mich"
/>
```

3. Aufgabe


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Klick mich"
/>
```

4. Aufgabe




```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Abbrechen"
    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ok"
/>
```

5. Aufgabe


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Abbrechen"
    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ok"
/>
```

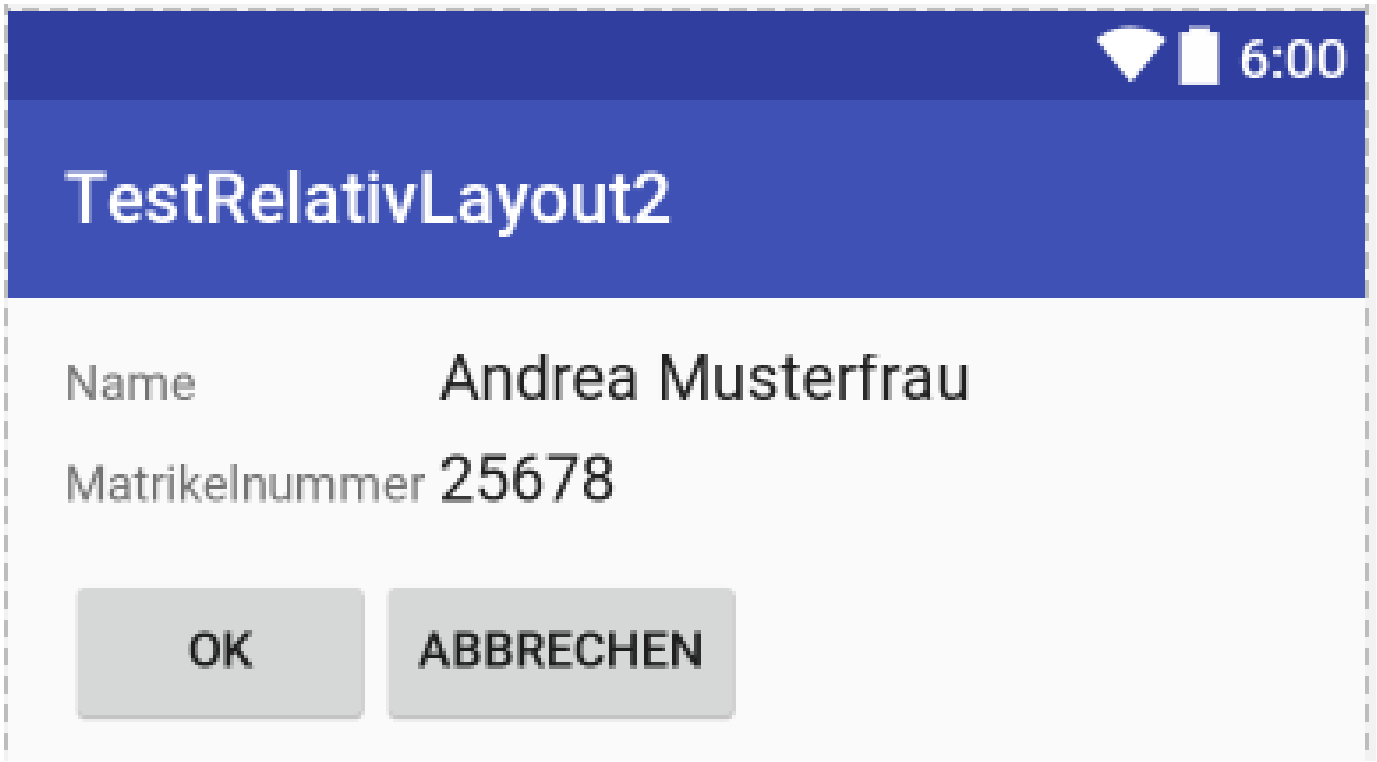
6. Aufgabe


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Abbrechen"
    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ok"
/>
```

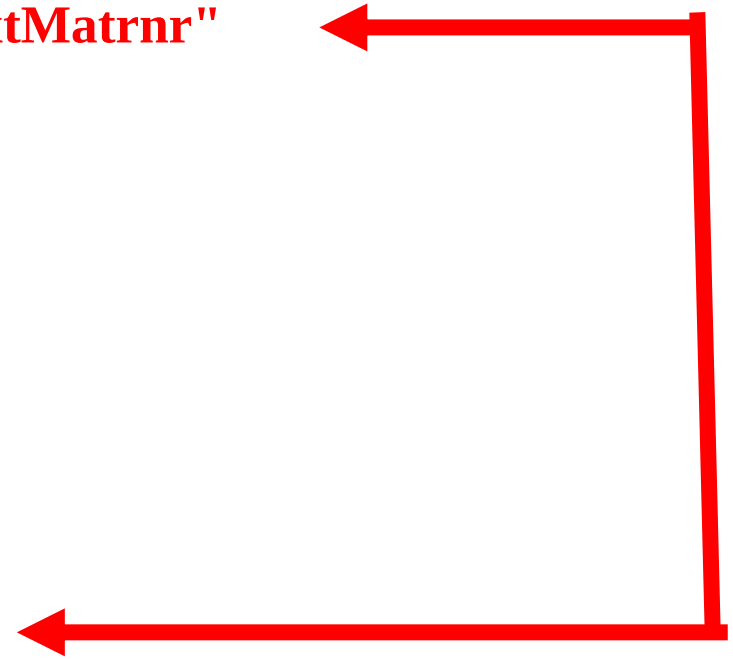
7. Aufgabe


TestRelativLayout2
Name Andrea Musterfrau
Matrikelnummer 25678
OK
ABBRECHEN

8. Aufgabe


6:00
TestRelativLayout2
Name
Andrea Musterfrau
Matrikelnummer
25678
OK
ABBRECHEN

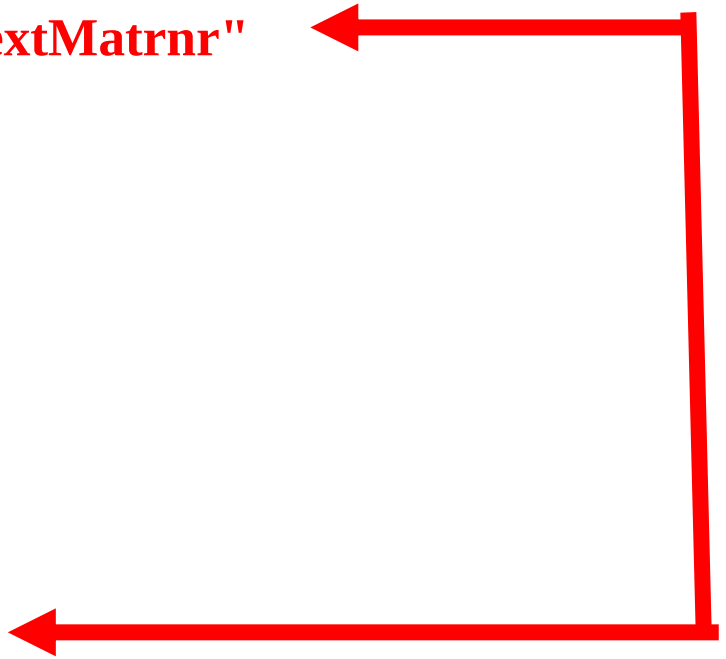
```
<TextView
    android:id="@+id/textview1"
    />
<EditText
    android:id="@+id/edittextName"
    android:layout_toRightOf="@id/textview1"
    android:layout_alignBaseline="@id/textview1"
    android:layout_alignLeft="@id/edittextMatrnr"
    android:lines="1"
    android:text="Andrea Musterfrau"
    />
<TextView
    android:id="@+id/textview2"
 />
<EditText
    android:id="@+id/edittextMatrnr"
/>
```



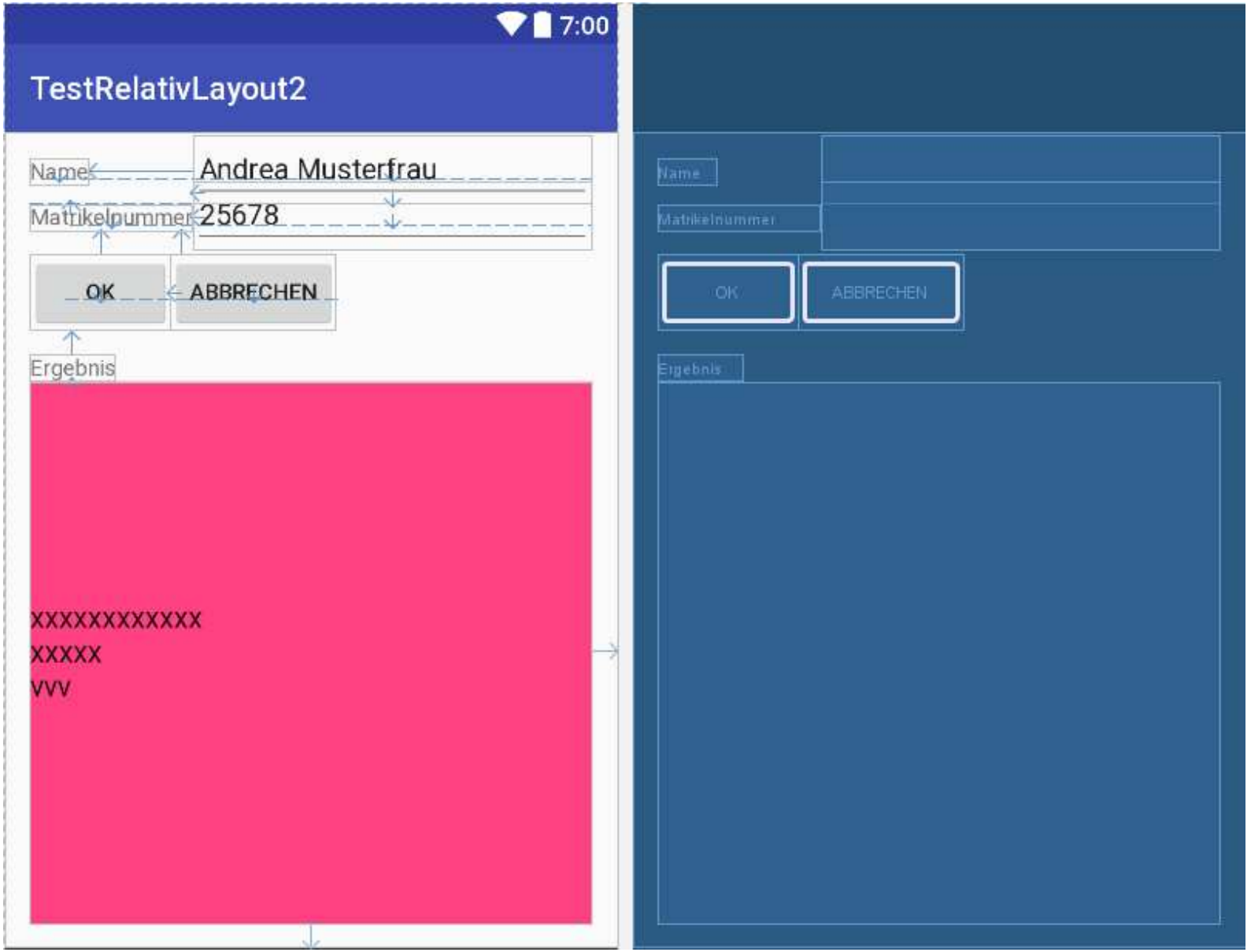
RelativeLayout: Problem

- Wenn man auf eine id verweisen will, so muss diese **VORER** bekannt sein. Also oberhalb definiert sein.
- Manchmal kann diese aber zu Problemen führen.
- **Regel:**
  - Wenn man vorher eine „id“ als Referenz benötigt, so kann man diese oberhalb „an der Verweisstelle“ definieren (mit Pluszeichen).
    - **@+id/edittextMatrnr**
  - Am Originalelement benötigt man dann nur eine Referenz (ohne Pluszeichen)
    - **@id/edittextMatrnr**

```
<TextView
   android:id="@+id/textview1"
   />
<EditText
   android:id="@+id/edittextName"
   android:layout_toRightOf="@id/textview1"
   android:layout_alignBaseline="@id/textview1"
   android:layout_alignLeft="@+id/edittextMatrnr"
   android:lines="1"
   android:text="Andrea Musterfrau"
   />
<TextView
   android:id="@+id/textview2"
 />
<EditText
   android:id="@id/edittextMatrnr"
/>
```



9. Aufgabe: mit Multiline-Editor



android:layout_below="@id/button1
"

/>

<EditText

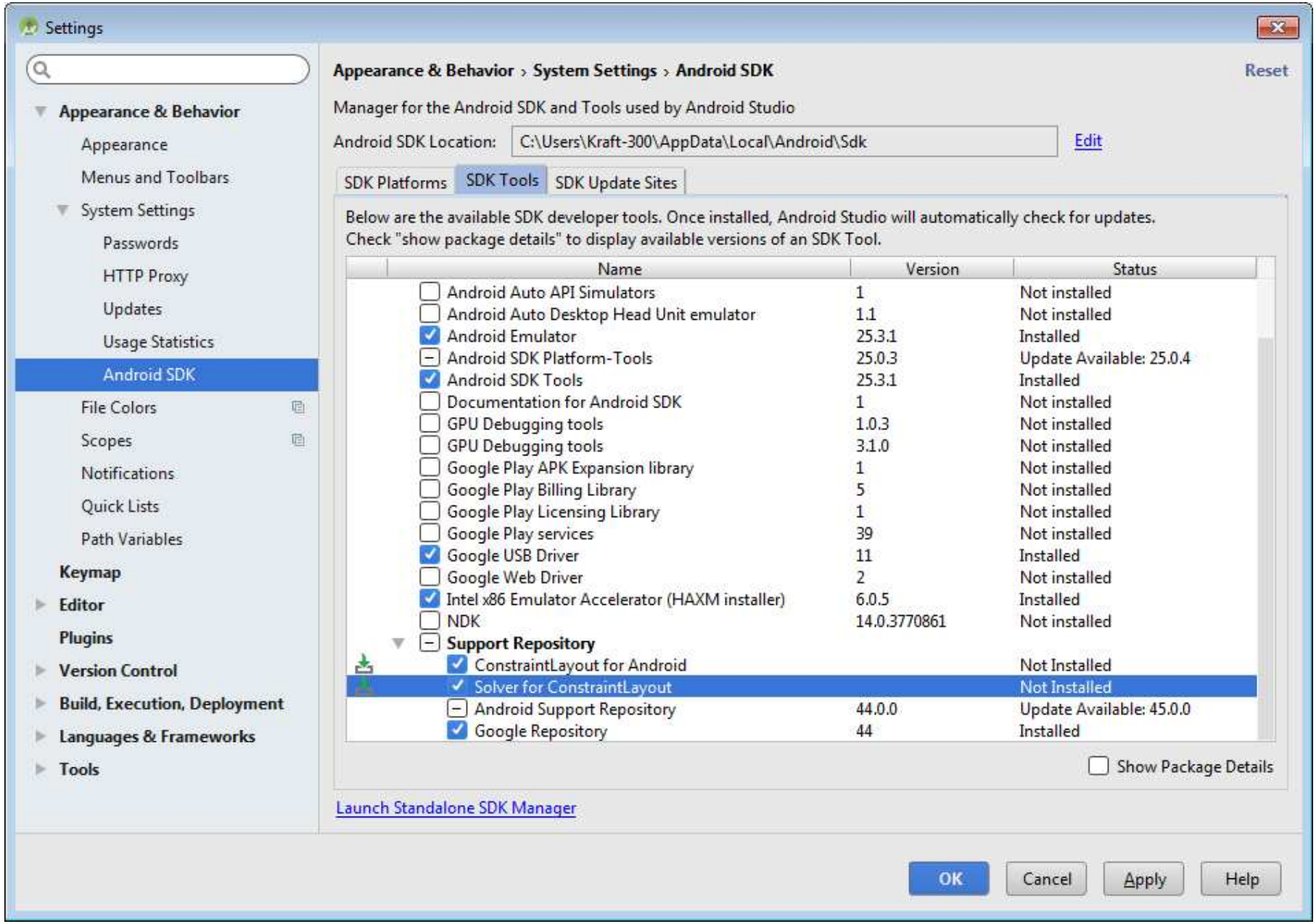
android:id="@+id/ergebnis"

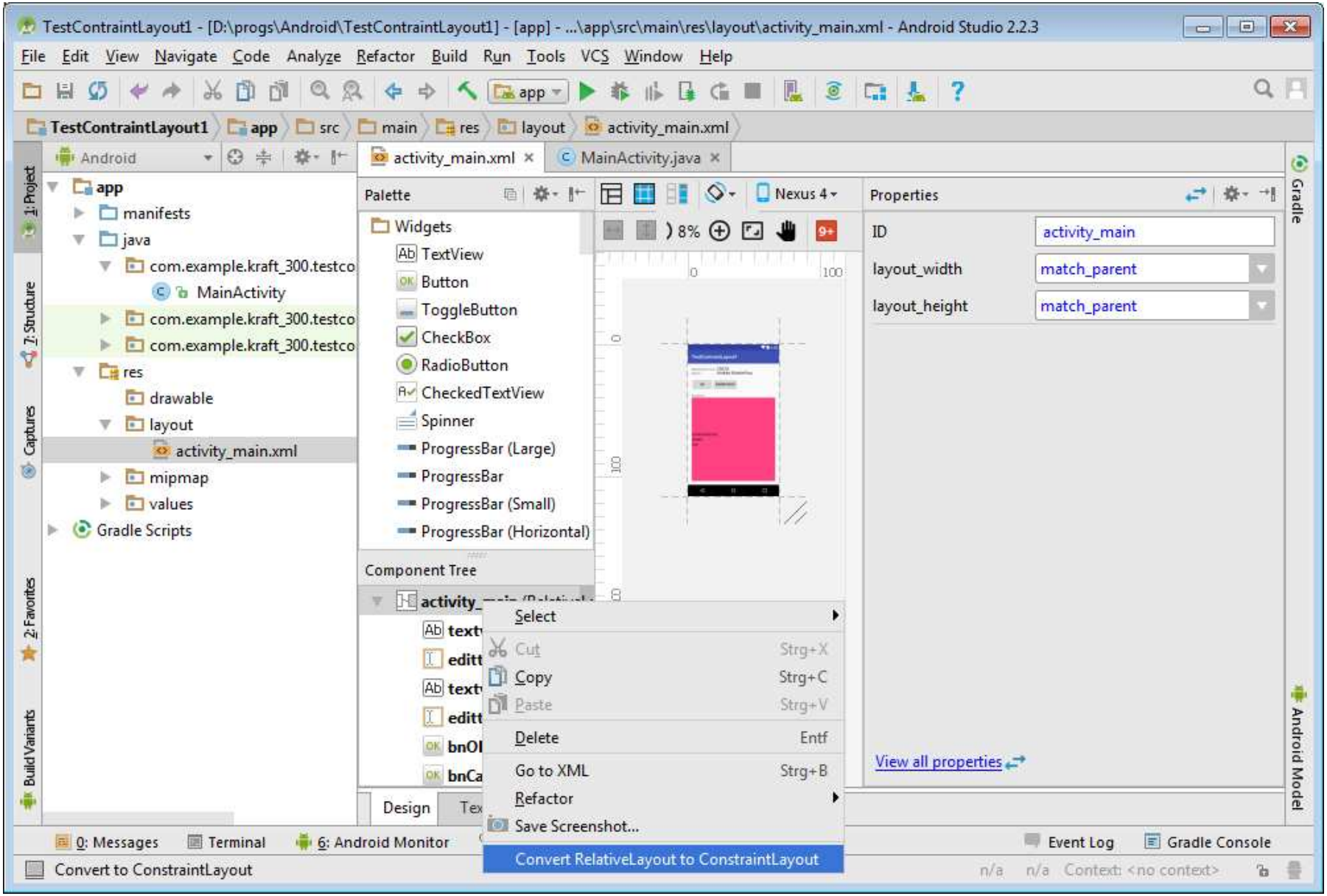
android:layout_width="match_pare
nt"

android:layout_height="wrap_cont
ent"

android:layout_below="@id/textvi
ewergebnis"

android:layout_alignParentBottom
="true"


Settings
Appearance & Behavior
Appearance
Menus and Toolbars
System Settings
Passwords
HTTP Proxy
Updates
Usage Statistics
Android SDK
File Colors
Scopes
Notifications
Quick Lists
Path Variables
Keymap
Editor
Plugins
Version Control
Build, Execution, Deployment
Languages & Frameworks
Tools
Appearance & Behavior › System Settings › Android SDK
Reset
Manager for the Android SDK and Tools used by Android Studio
Android SDK Location: C:\Users\Kraft-300\AppData\Local\Android\Sdk
Edit
SDK Platforms
SDK Tools
SDK Update Sites
Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates.
Check "show package details" to display available versions of an SDK Tool.
Name
Version
Status
Android Auto API Simulators 1 Not installed
Android Auto Desktop Head Unit emulator 1.1 Not installed
Android Emulator 25.3.1 Installed
Android SDK Platform-Tools 25.0.3 Update Available: 25.0.4
Android SDK Tools 25.3.1 Installed
Documentation for Android SDK 1 Not installed
GPU Debugging tools 1.0.3 Not installed
GPU Debugging tools 3.1.0 Not installed
Google Play APK Expansion library 1 Not installed
Google Play Billing Library 5 Not installed
Google Play Licensing Library 1 Not installed
Google Play services 39 Not installed
Google USB Driver 11 Installed
Google Web Driver 2 Not installed
Intel x86 Emulator Accelerator (HAXM installer) 6.0.5 Installed
NDK 14.0.3770861 Not installed
Support Repository
ConstraintLayout for Android Not Installed
Solver for ConstraintLayout Not Installed
Android Support Repository 44.0.0 Update Available: 45.0.0
Google Repository 44 Installed
Show Package Details
Launch Standalone SDK Manager
OK
Cancel
Apply
Help


TestContraintLayout1 - [D:\progs\Android\TestContraintLayout1] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 2.2.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
TestContraintLayout1 app src main res layout activity_main.xml
Android
app
manifests
java
com.example.kraft_300.testco
MainActivity
com.example.kraft_300.testco
com.example.kraft_300.testco
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MainActivity.java
Palette
Widgets
TextView
Button
ToggleButton
CheckBox
RadioButton
CheckedTextView
Spinner
ProgressBar (Large)
ProgressBar
ProgressBar (Small)
ProgressBar (Horizontal)
Component Tree
Nexus 4
8%
Properties
ID
activity_main
layout_width
match_parent
layout_height
match_parent
View all properties
Select
Cut Strg+X
Copy Strg+C
Paste Strg+V
Delete Entf
Go to XML Strg+B
Refactor
Save Screenshot...
Convert RelativeLayout to ConstraintLayout
Design
0: Messages
Terminal
6: Android Monitor
Event Log
Gradle Console
Convert to ConstraintLayout
n/a n/a Context: <no context>
1: Project
7: Structure
Captures
2: Favorites
Build Variants
Gradle
Android Model

ConstraintLayout

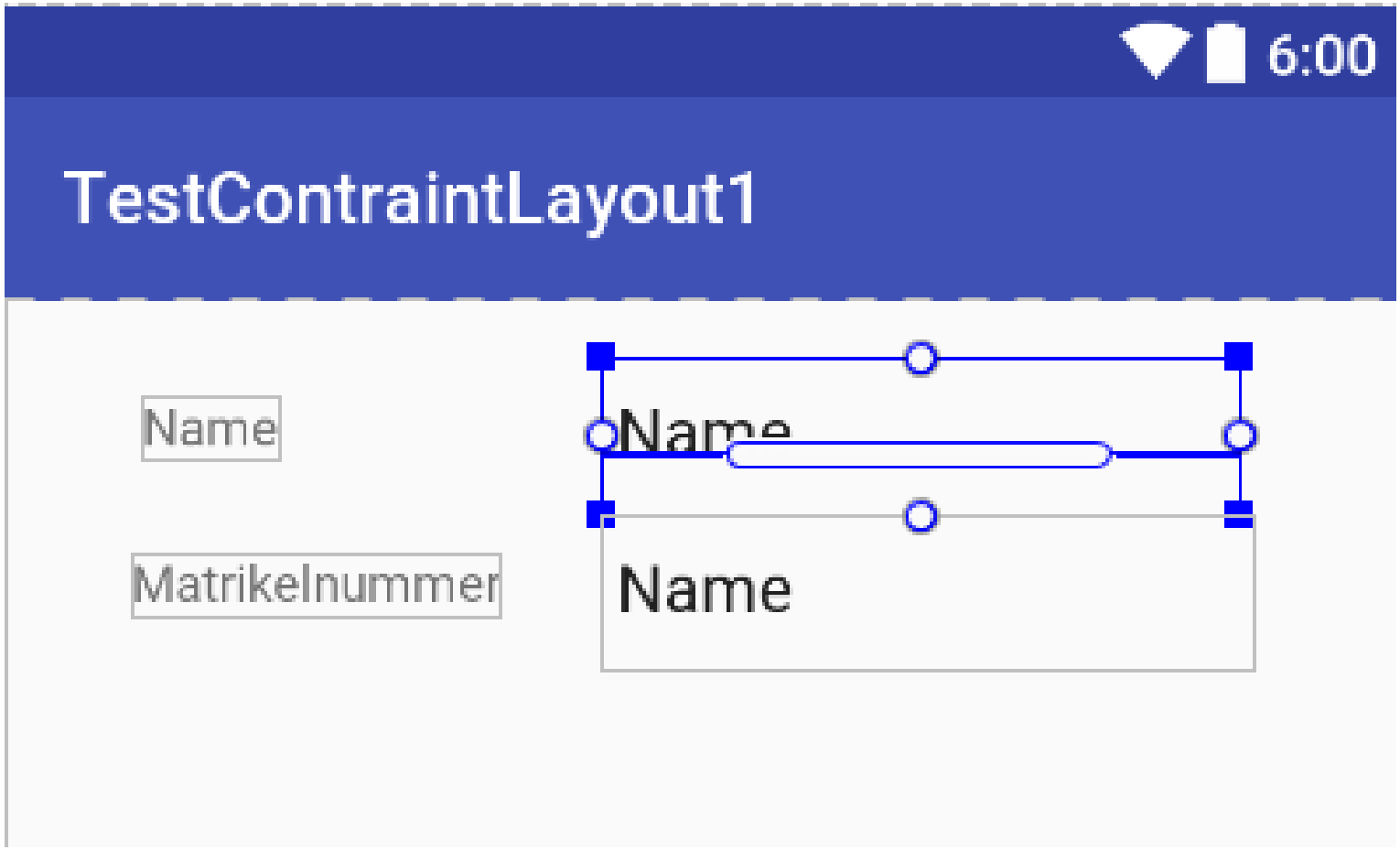
- Neues Layout
- Stamm von ViewGroup
- Ist sehr flexibel
- Erlaubt das Erstellen und Ändern mittels Drag & Drop & Maus
- https://developer.android.com/reference/android/support/constraint/ConstraintLayout.html

ConstraintLayout

Arten der Constraints:

- Relative positioning
- Margins
- Centering positioning
- Visibility behavior
- Dimension constraints
- Chains
- Virtual Helpers objects

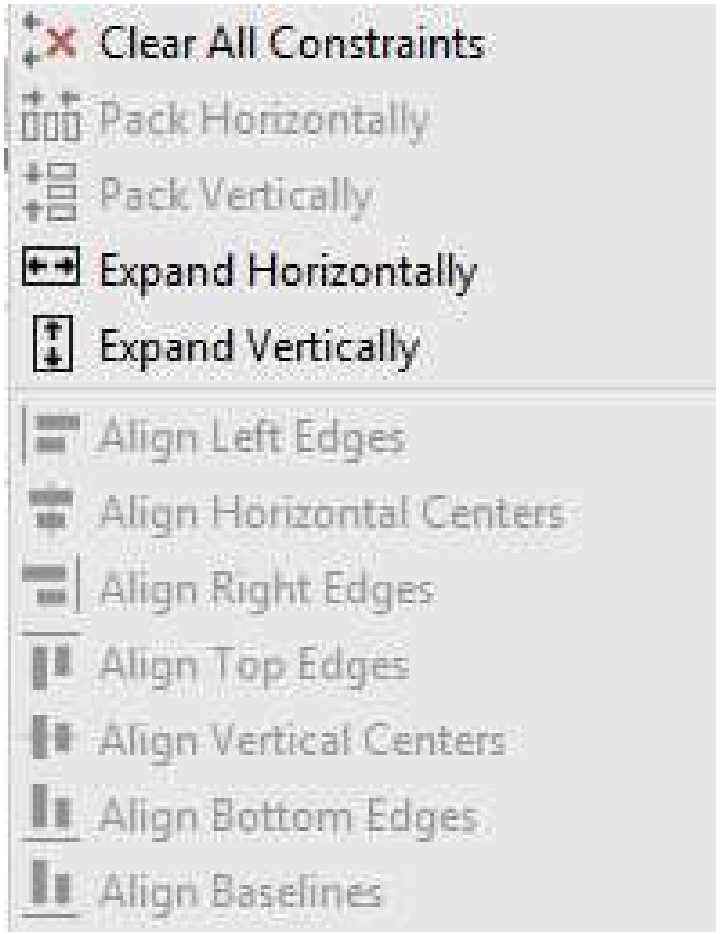
ConstraintLayout

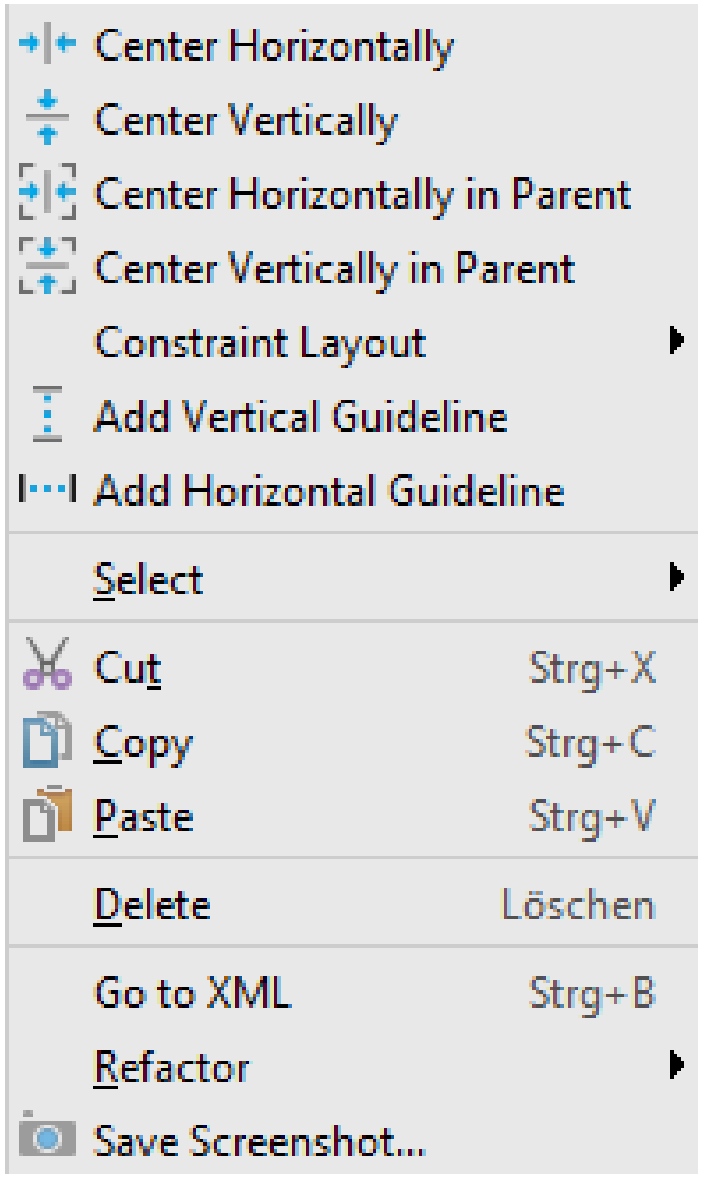


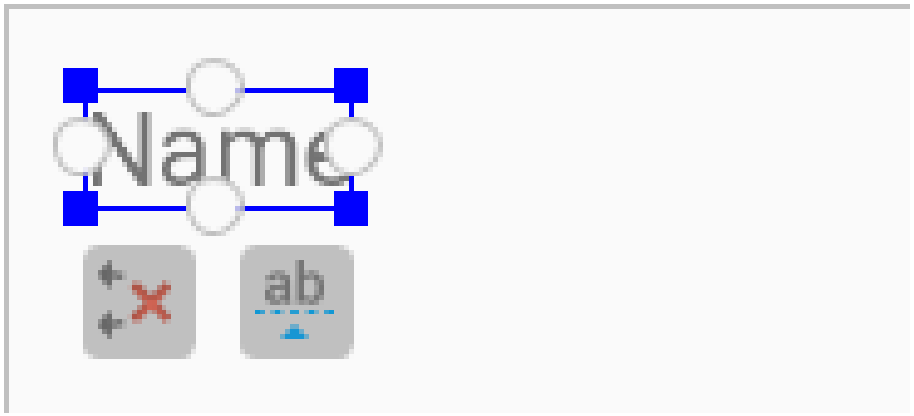
ConstraintLayout: Grundstruktur

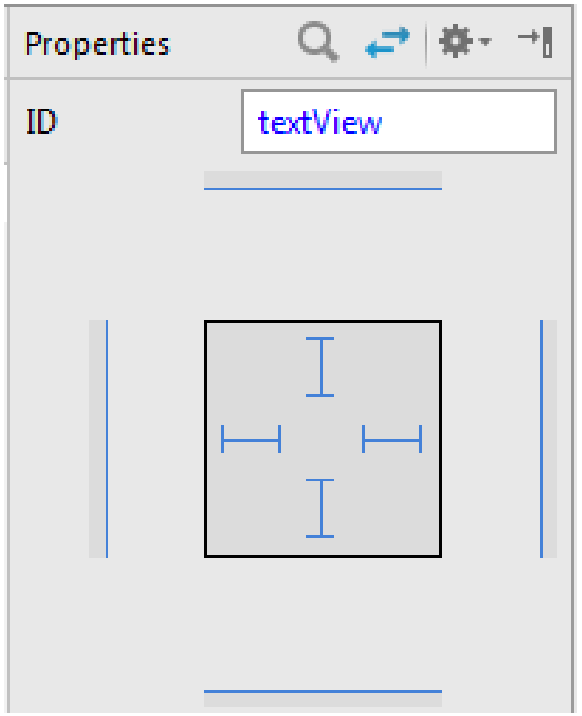
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</android.support.constraint.ConstraintLayout>
```

ConstraintLayout: Funktionen





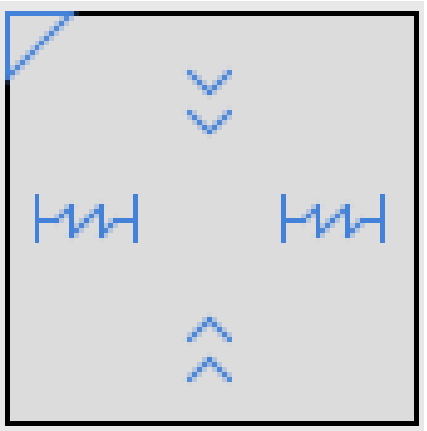




Fixed: Feste Breite /Höhe

AnySize: Gewichtung 1, Skalierung

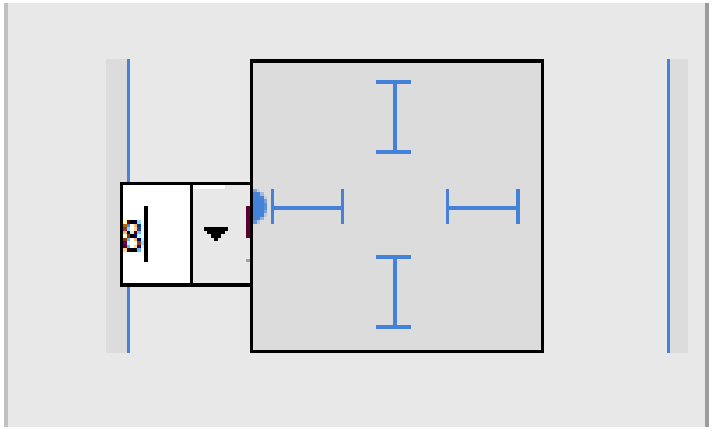
Wrap Content: This option only expands to fill the widget with the contained element such as text or a drawable.



ConstraintLayout: Linker Rand

Drag & Drop zum linken Rand

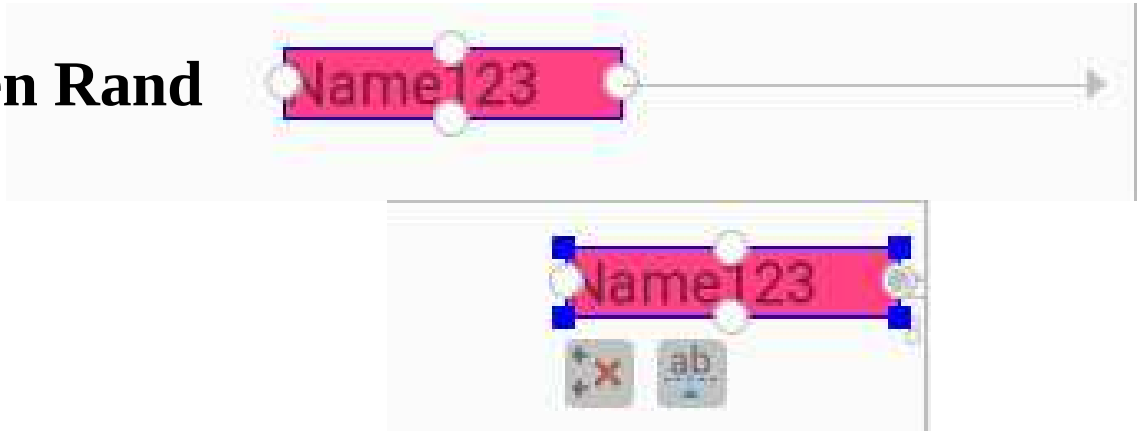
Ändern der Breite



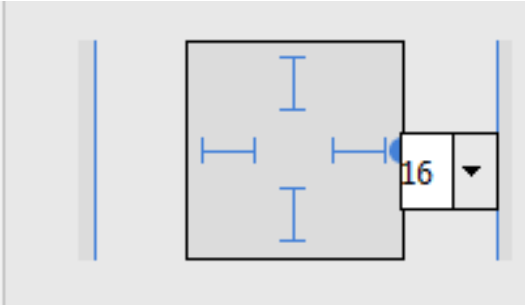
```
android:layout_marginLeft="16dp"
```

ConstraintLayout: Rechter Rand

Drag & Drop zum rechten Rand



Ändern der Breite

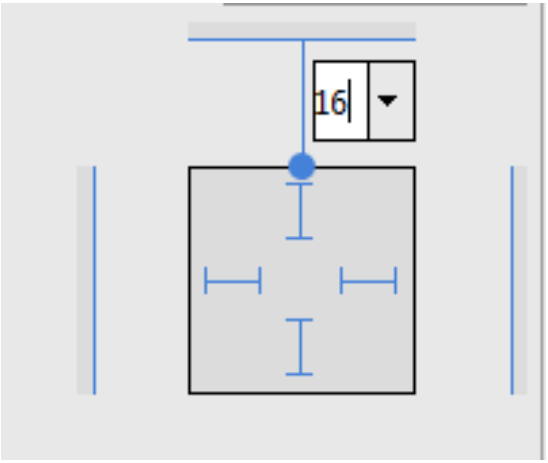



```
android:layout_marginRight="16dp"
```

ConstraintLayout: Oberer Rand

Drag & Drop zum oberen Rand

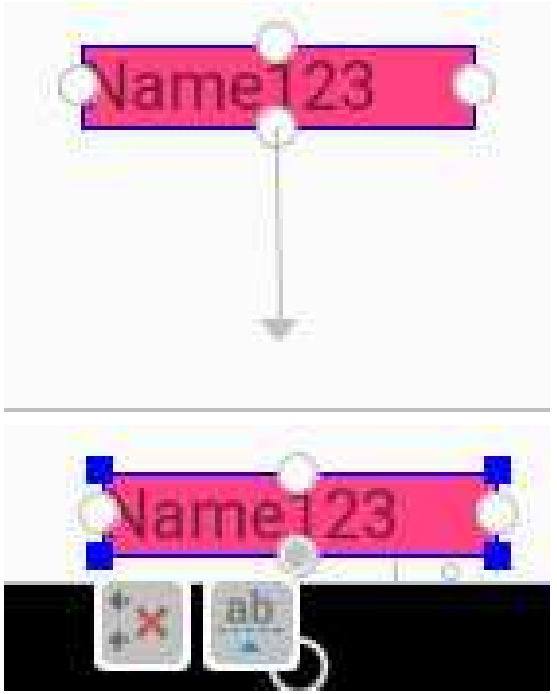
Ändern des Abstands



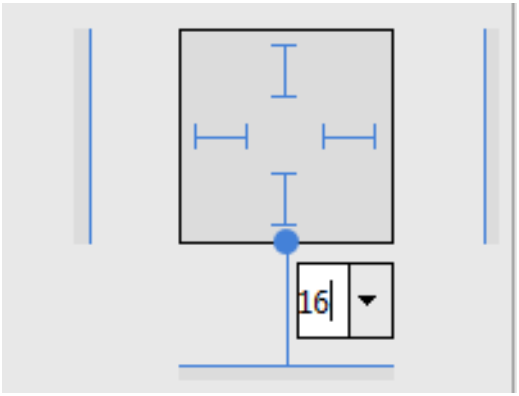

```
android:layout_marginTop="16dp"
```

ConstraintLayout: Unterer Rand

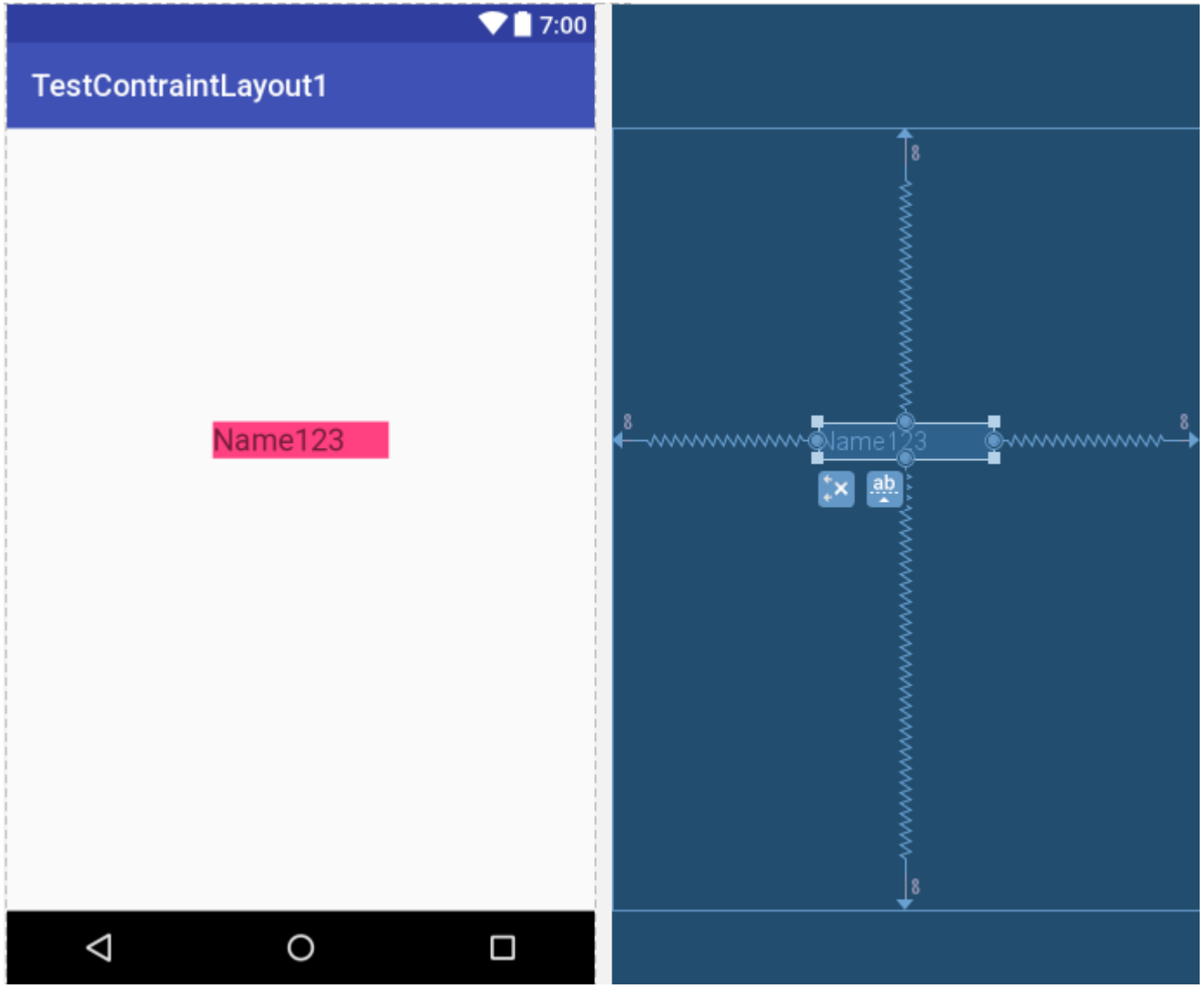
Drag & Drop zum unteren Rand



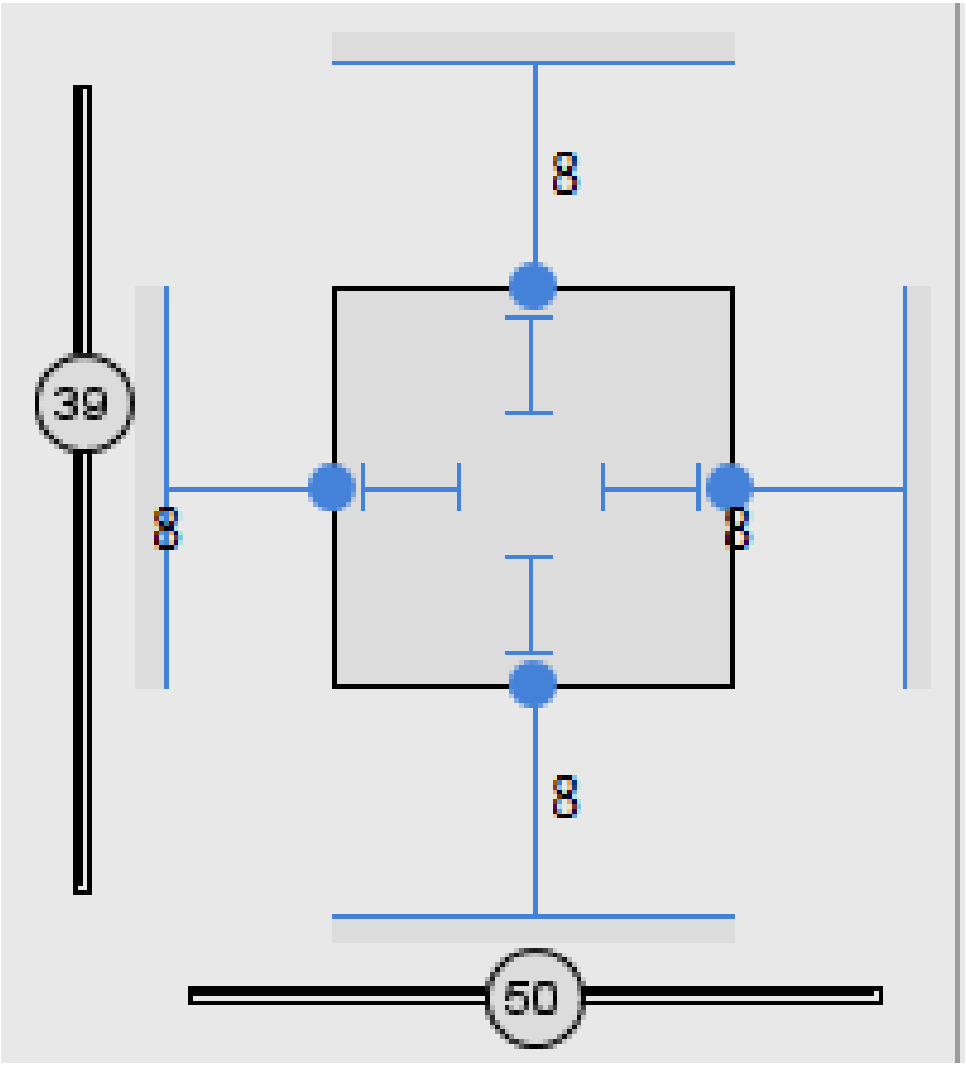
Ändern des Abstands



ConstraintLayout: Ausrichtung / Ränder

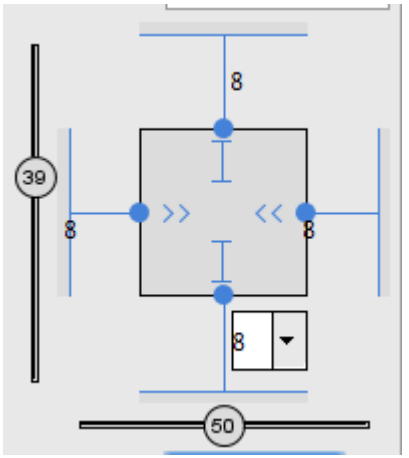


ConstraintLayout: Ausrichtung / Ränder



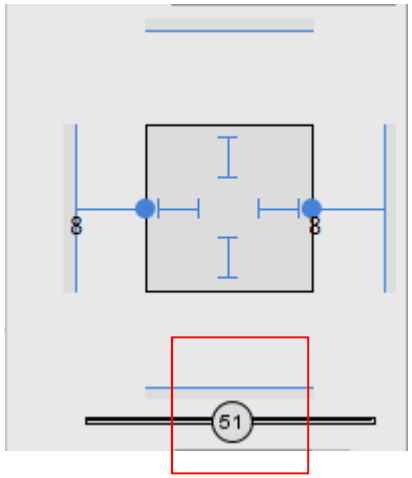
Größe des Inhalts

```
app:layout_constraintRight_toRightOf="parent"
android:layout_marginLeft="8dp"
app:layout_constraintLeft_toLeftOf="parent"
```



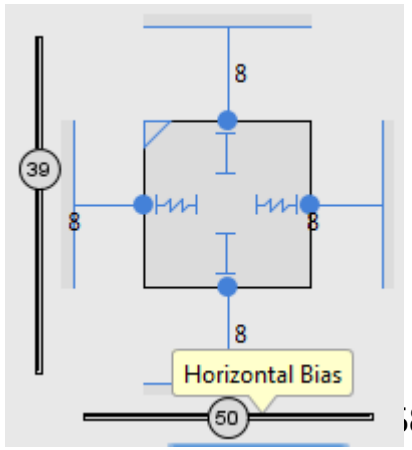
Größe wird definiert

```
android:layout_marginLeft="8dp"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="paren
app:layout_constraintHorizontal_bias="0.51"
```



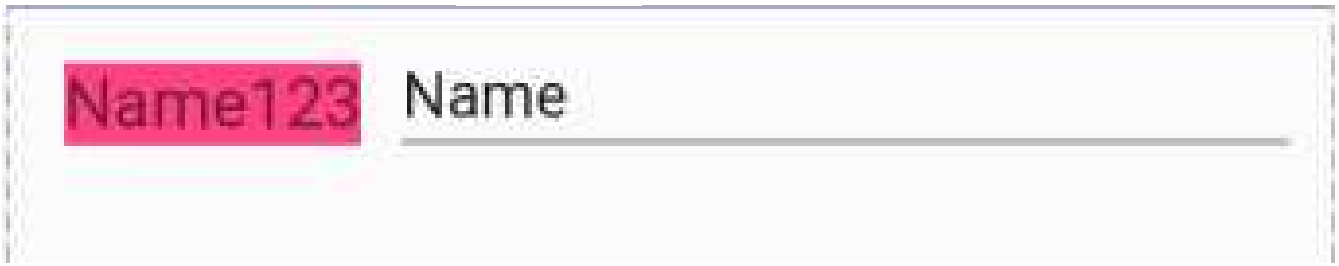
Komplette Breite: erst beide Seiten <- ->

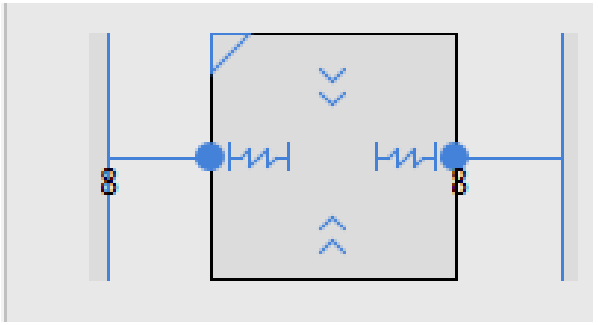
```
android:layout_marginLeft="8dp"
android:layout_marginRight="8dp"
app:layout_constraintLeft_toLeftOf="parent„
app:layout_constraintRight_toRightOf="parent"
```



TextView mit einem TextField

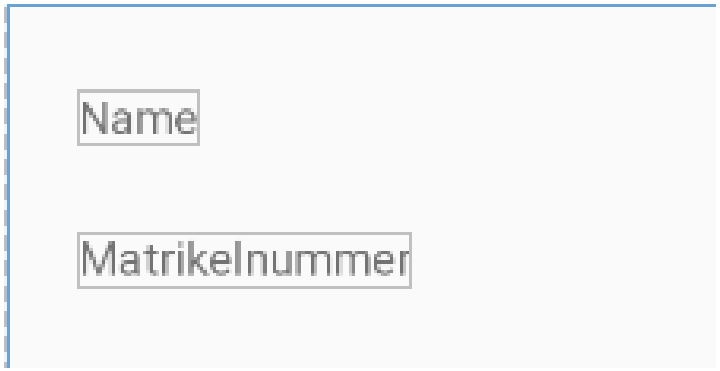
- Positionieren des Textfeldes neben dem TextView
- Anker auf beiden Seiten
- Type „anySize“





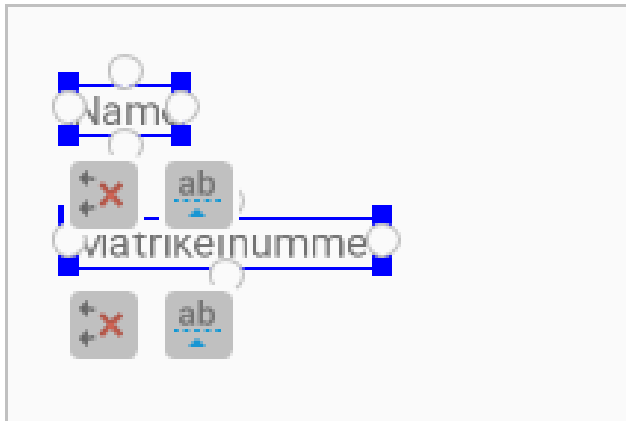
ConstraintLayout-Beispiel

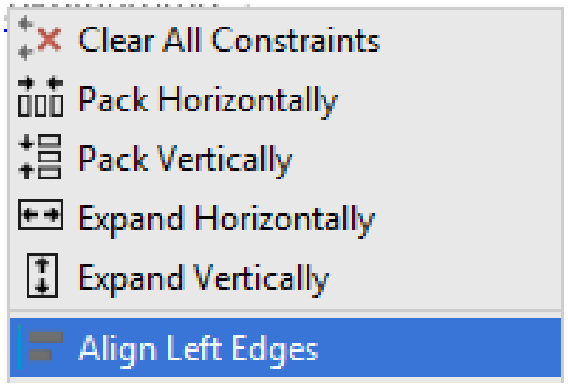
linksbünding



Ablauf

- Beide UI-Elemente markieren
- UI-Element „Name“ nach oben schieben
- Align Left Egdes

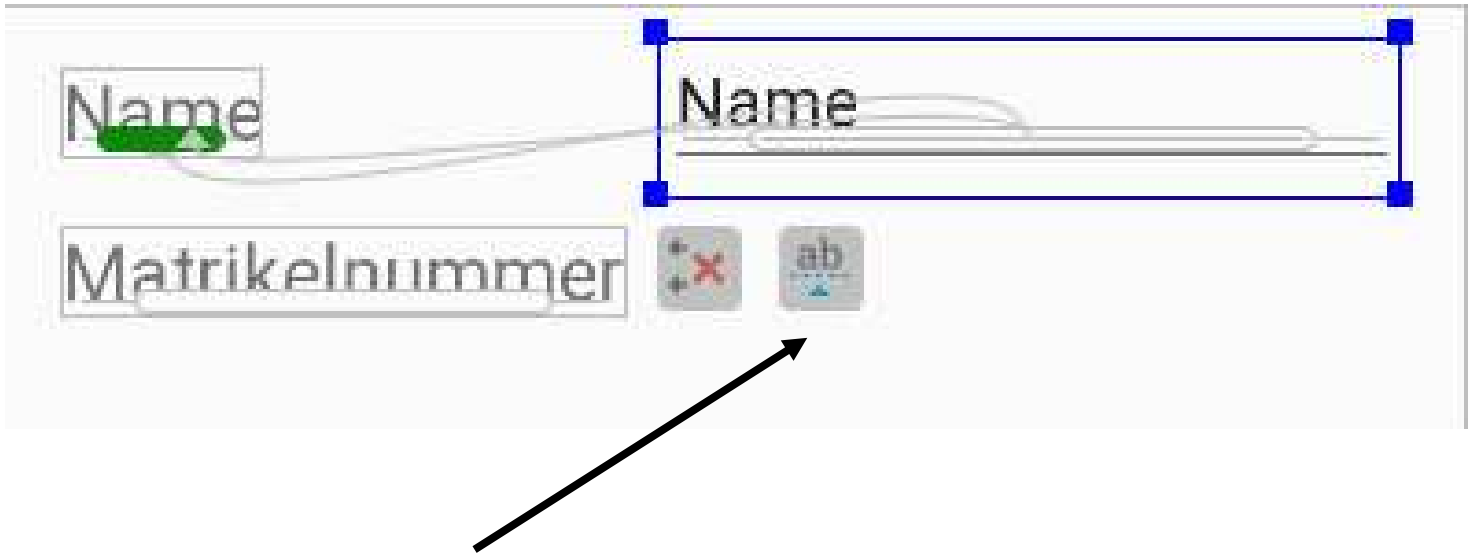




Beide Elemente werden linksbündig in der Mitte dargestellt

```
app:layout_constraintLeft_toLeftOf="@+id/textView"
```

ConstraintLayout-Beispiel



- Schalter „ab“ Baseline

Absolute Layout

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
            android:layout_x="10px"
            android:layout_y="110px"
            android:text="@string/username"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

</AbsoluteLayout>
```

FrameLayout: Übereinanderliegende UI-Elemente

```
<FrameLayout
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 xmlns:android="http://schemas.android.com/apk/res/android">

 <ImageView
  android:id="@+id/image1"
  android:src="@mipmap/ic_launcher"
  android:scaleType="fitCenter"
  android:layout_height="fill_parent"
  android:layout_width="fill_parent"
  android:gravity="center"
  android:onClick="image_click“
 />

 <TextView
   android:id="@+id/textview1"
   android:text="Dies ist ein Bild"
   android:layout_height="fill_parent"
   android:layout_width="fill_parent"
   android:textSize="30sp"
   android:gravity="center"
   android:onClick="textview_click"
 />

 </FrameLayout>
```

```
import android.media.Image;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

public class MainActivity extends AppCompatActivity {

    private TextView textview1 = null;
    private ImageView image1 = null;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textview1 = (TextView) findViewById(R.id.textview1);
    image1 = (ImageView)findViewById(R.id.image1);
}  // onCreate
```

```
    public void image_click(View view) {
        textview1.setVisibility(View.VISIBLE);
        image1.setVisibility(View.GONE);
    }

    public void textview_click(View view) {
        textview1.setVisibility(View.GONE);
        image1.setVisibility(View.VISIBLE);
    }
}
```

- https://codelabs.developers.google.com/codelabs/constraint-layout/index.html#4

- https://codelabs.developers.google.com/codelabs/constraint-layout/index.html#7