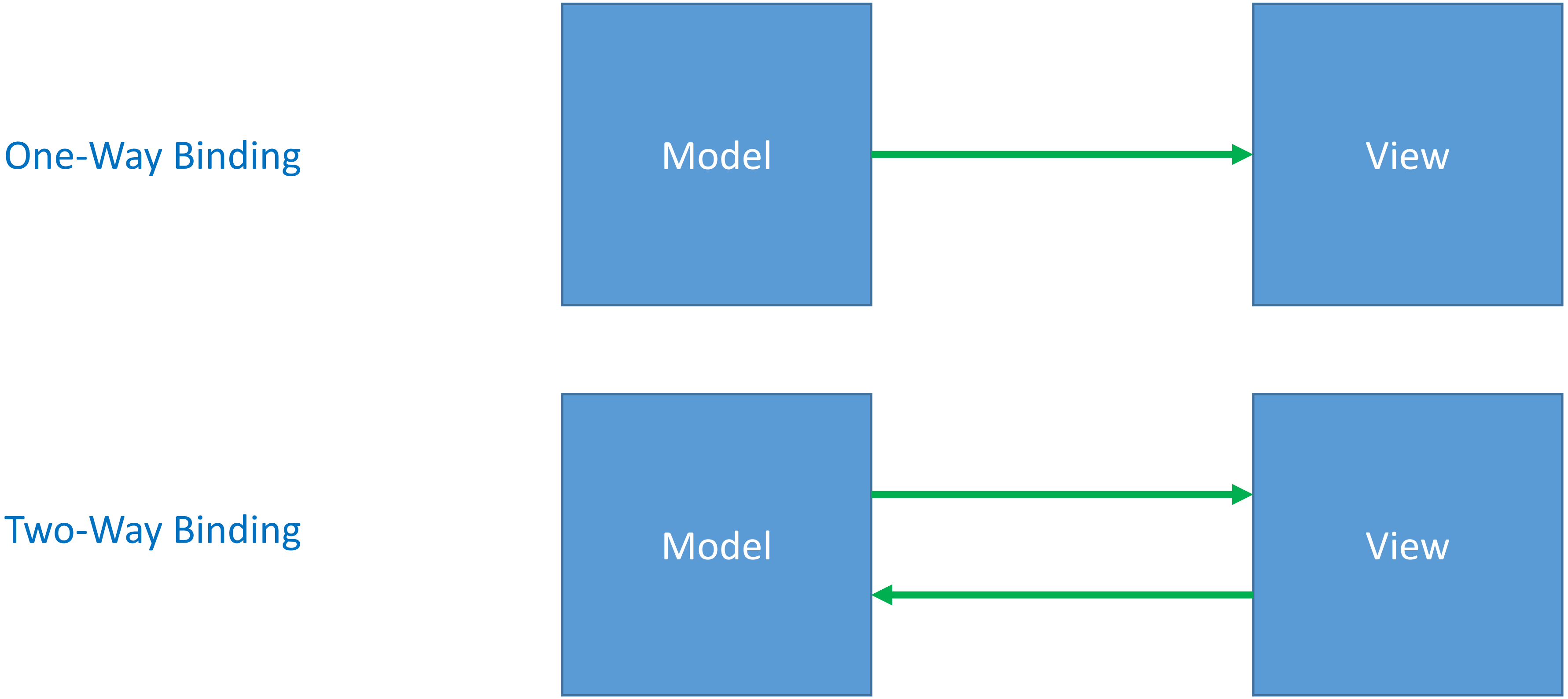
Databinding in Android

Shopping TechTalk 27.10.16

André Roß

Was ist Databinding?

Verknüpfung zwischen Model und View



One-Way Binding

Model

View

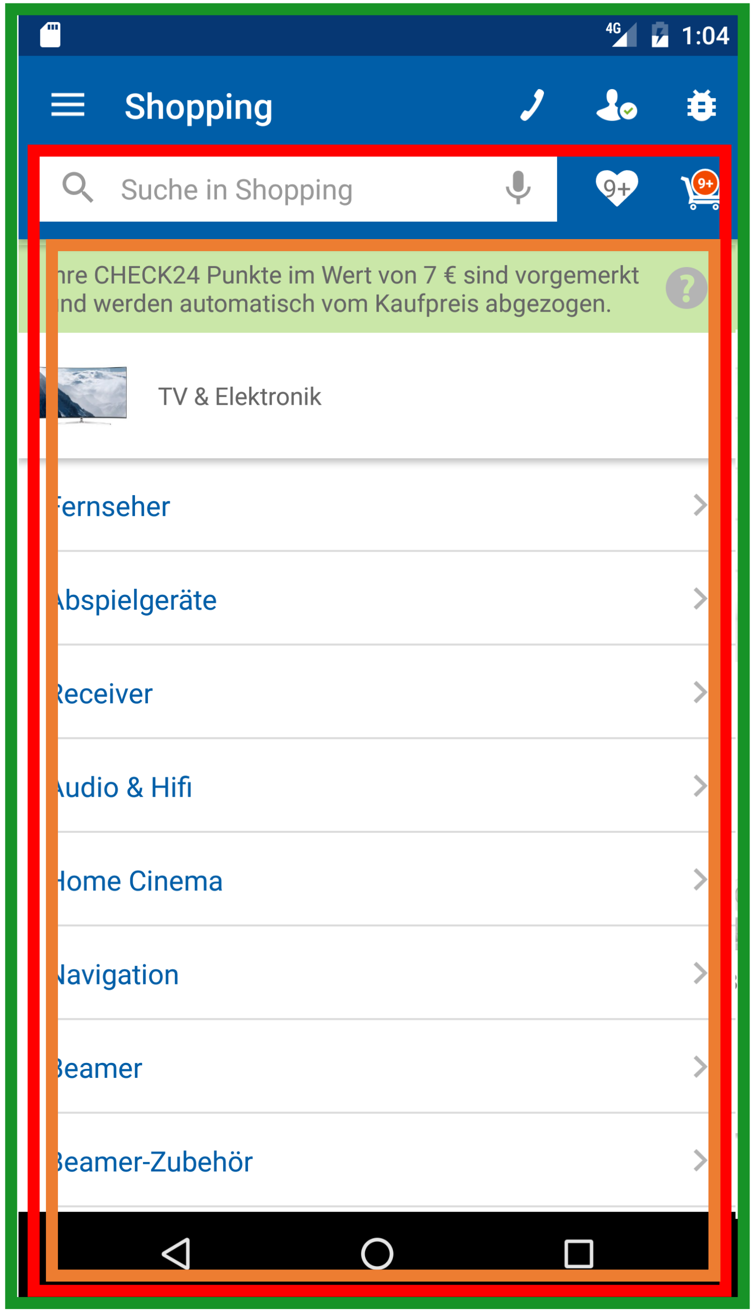
Two-Way Binding

Model

View

Exkurs Android

- **Activity**: eine Bildschirmseite in der App
- **Fragment**: Teil einer Bildschirmseite
- **Layout**: in xml deklariert

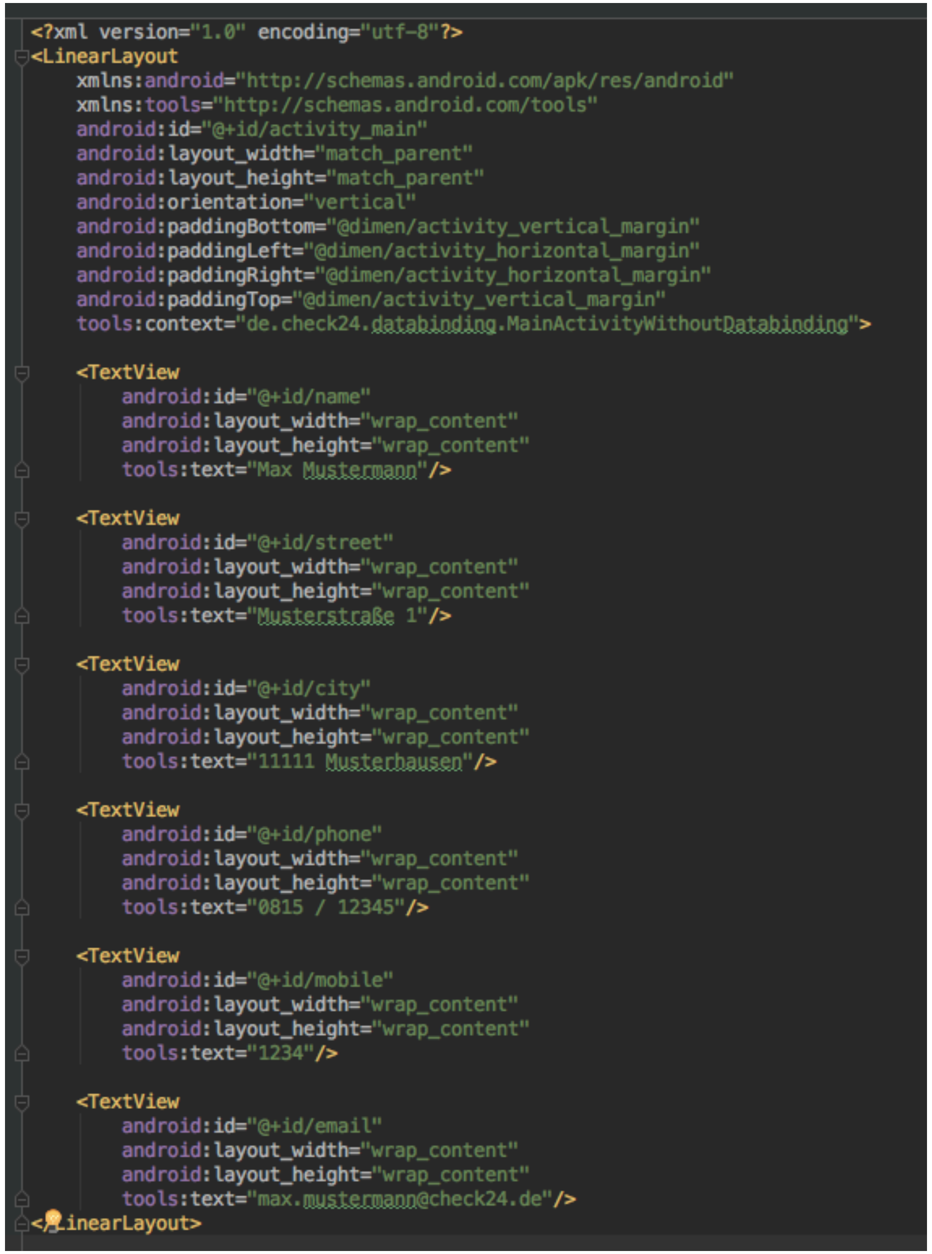


C24CoreActivity: Activity

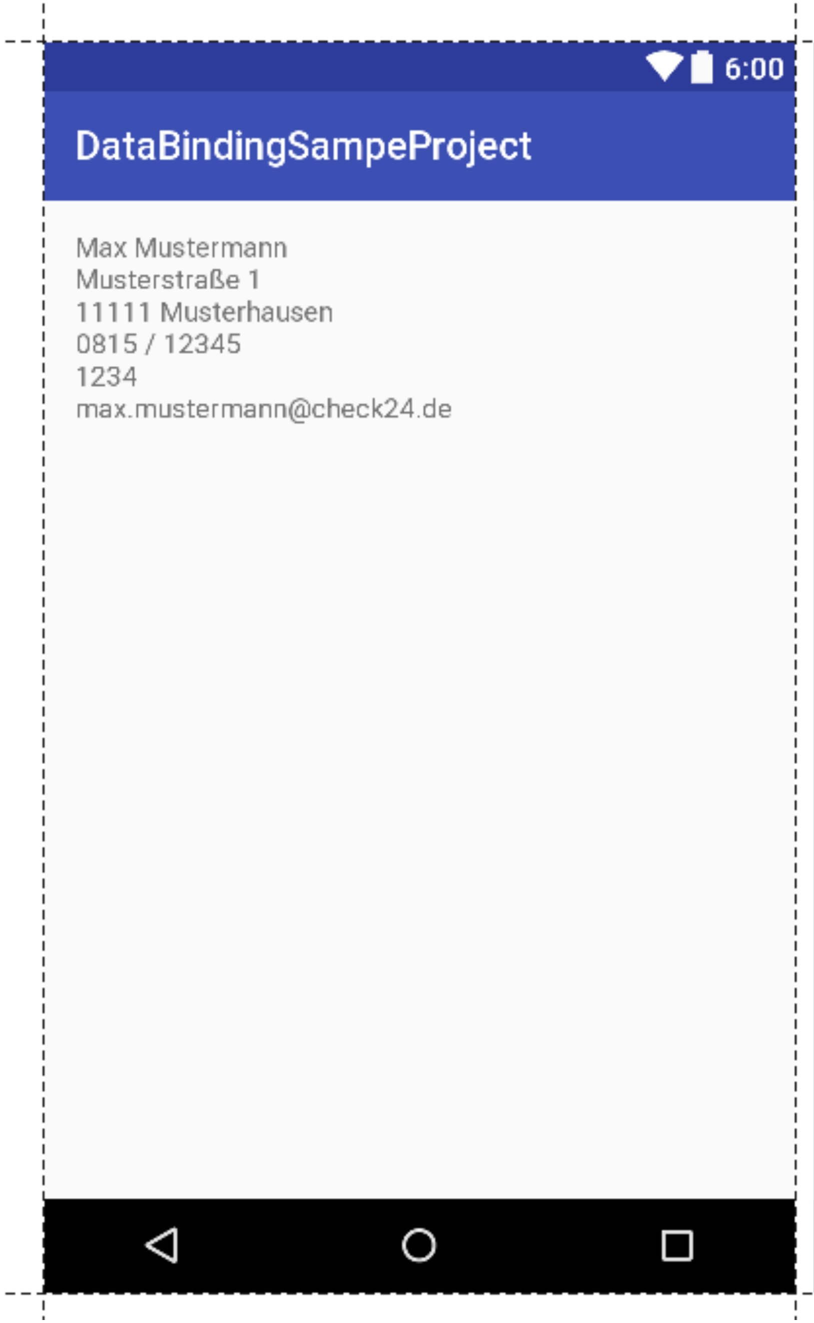
ShoppingFragment : Fragment

FrontpageFragment: (Child-) Fragment

hre CHECK24 Punkte im Wert von 7 € sind vorgemerkt nd werden automatisch vom Kaufpreis abgezogen.

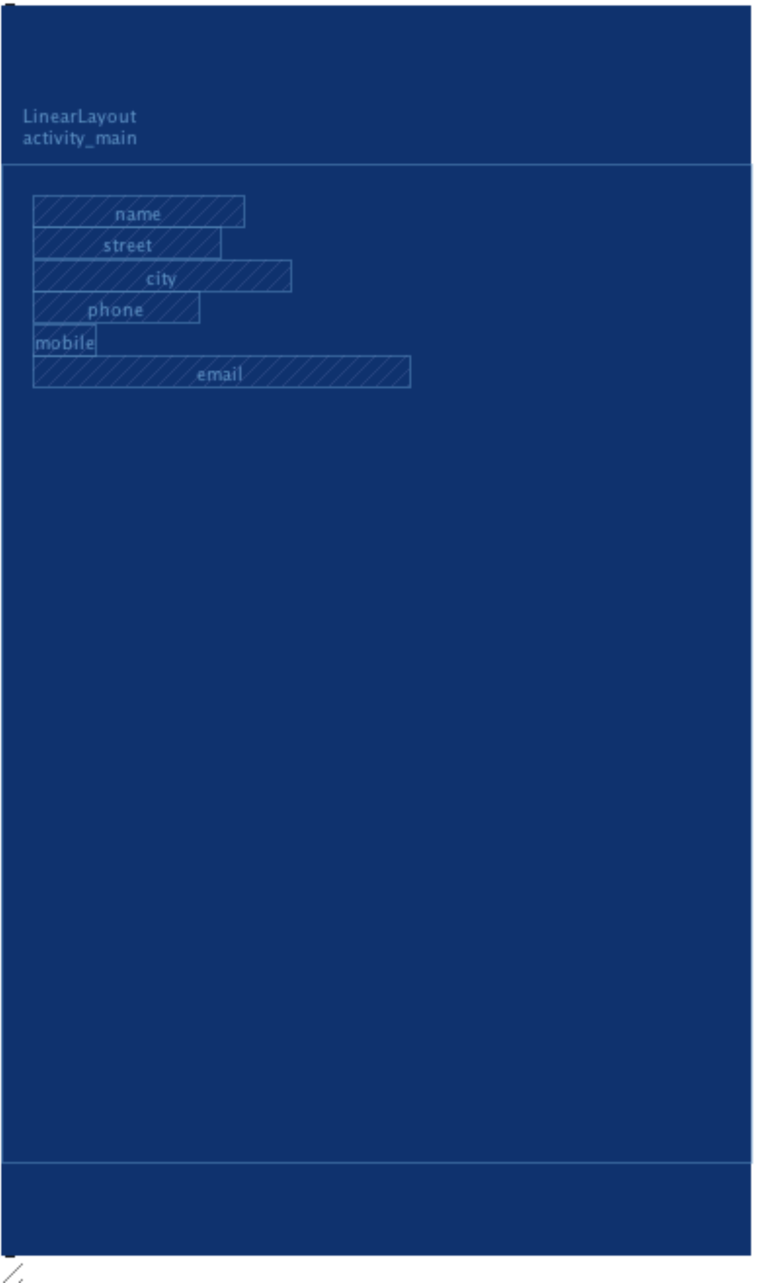


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="de.check24.databinding.MainActivityWithoutDatabinding">

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Max Mustermann"/>

    <TextView
        android:id="@+id/street"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Musterstraße 1"/>

    <TextView
        android:id="@+id/city"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="11111 Musterhausen"/>

    <TextView
        android:id="@+id/phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="0815 / 12345"/>

    <TextView
        android:id="@+id/mobile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="1234"/>

    <TextView
        android:id="@+id/email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="max.mustermann@check24.de"/>
</LinearLayout>
```



DataBindingSampeProject

Max Mustermann
Musterstraße 1
11111 Musterhausen
0815 / 12345
1234
max.mustermann@check24.de

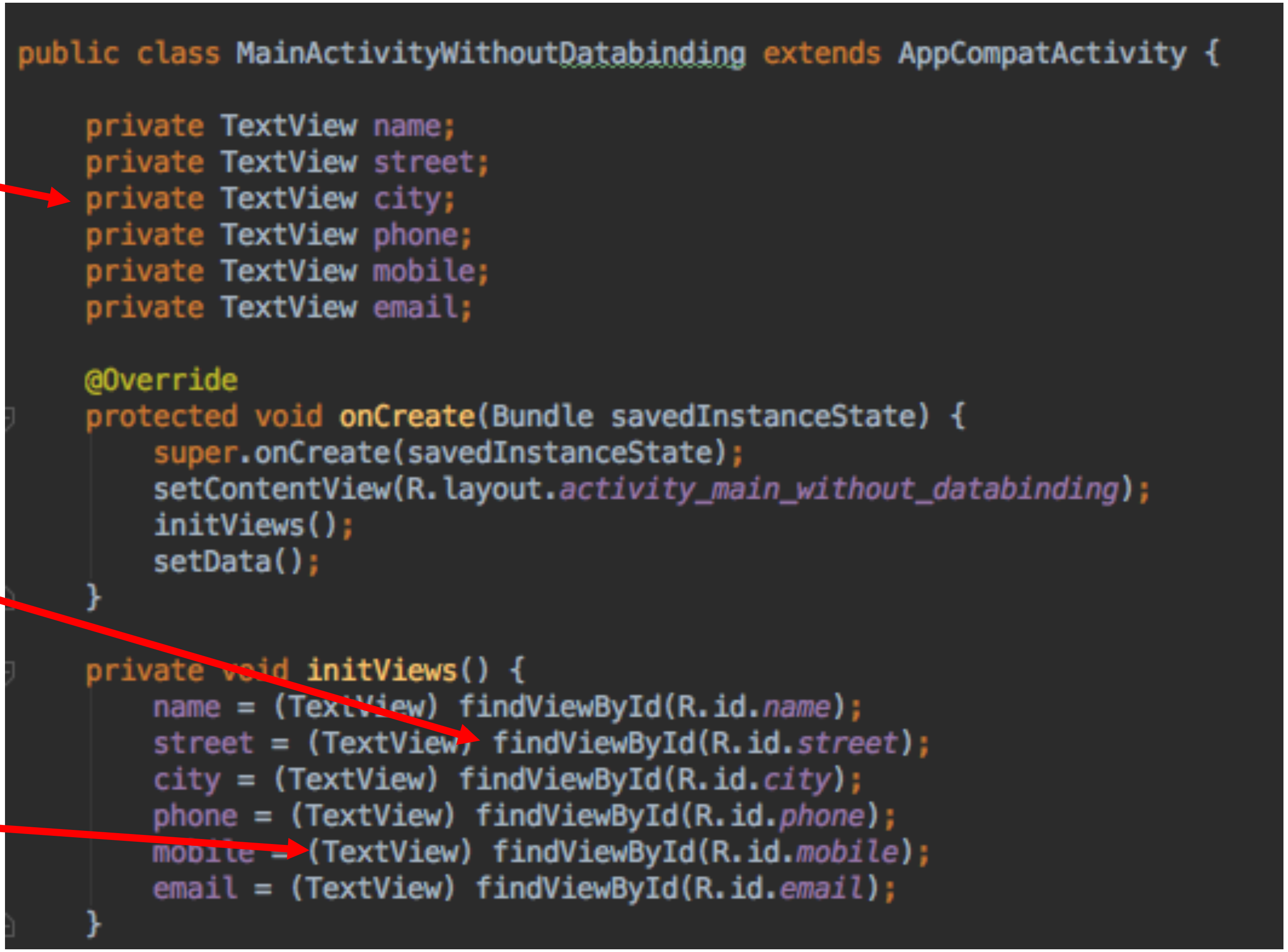


Motivation

Member pro View Element

findViewById()

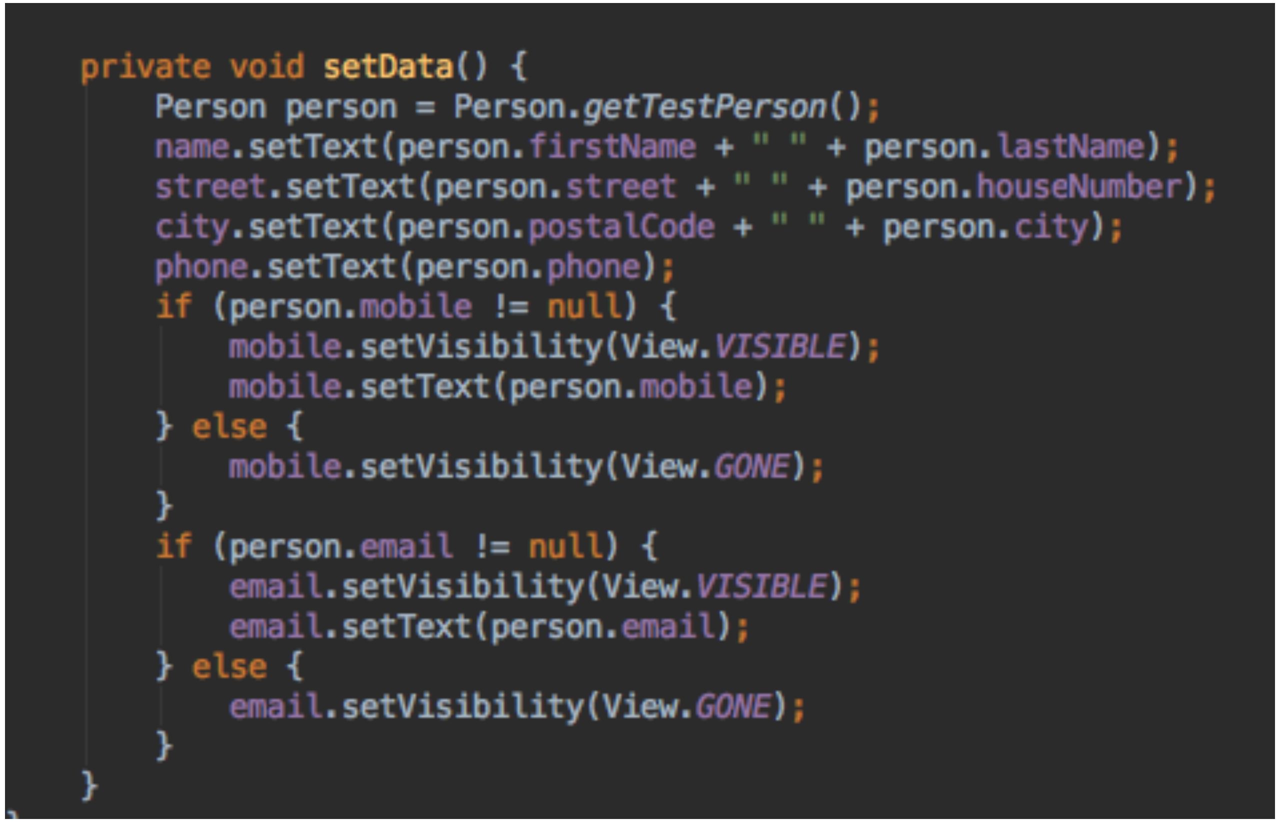
Unschöner Cast



```java
public class MainActivityWithoutDatabinding extends AppCompatActivity {

    private TextView name;
    private TextView street;
    private TextView city;
    private TextView phone;
    private TextView mobile;
    private TextView email;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_without_databinding);
        initViews();
        setData();
    }

    private void initViews() {
        name = (TextView) findViewById(R.id.name);
        street = (TextView) findViewById(R.id.street);
        city = (TextView) findViewById(R.id.city);
        phone = (TextView) findViewById(R.id.phone);
        mobile = (TextView) findViewById(R.id.mobile);
        email = (TextView) findViewById(R.id.email);
    }
```

Motivation

Daten setzen z.T. sehr unschön



```
private void setData() {
    Person person = Person.getTestPerson();
    name.setText(person.firstName + " " + person.lastName);
    street.setText(person.street + " " + person.houseNumber);
    city.setText(person.postalCode + " " + person.city);
    phone.setText(person.phone);
    if (person.mobile != null) {
        mobile.setVisibility(View.VISIBLE);
        mobile.setText(person.mobile);
    } else {
        mobile.setVisibility(View.GONE);
    }
    if (person.email != null) {
        email.setVisibility(View.VISIBLE);
        email.setText(person.email);
    } else {
        email.setVisibility(View.GONE);
    }
}
```

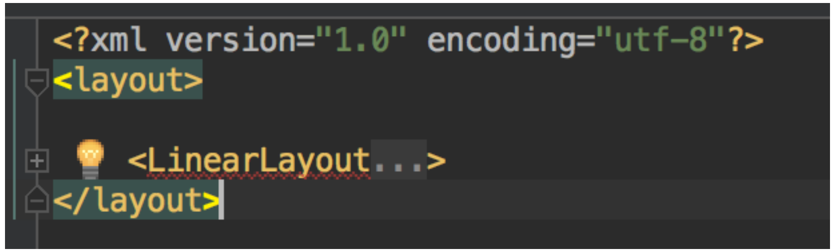
Databinding in Android

Seit Android Marshmallow im SDK verfügbar

```
android {
    ....
    dataBinding {
        enabled = true
    }
}
```

Databinding in Android - Schritt 1

<layout> tag im xml als root



```
<?xml version="1.0" encoding="utf-8"?>
<layout>

    <LinearLayout...>
</layout>
```

=> Autogeneriertes Binding Objekt

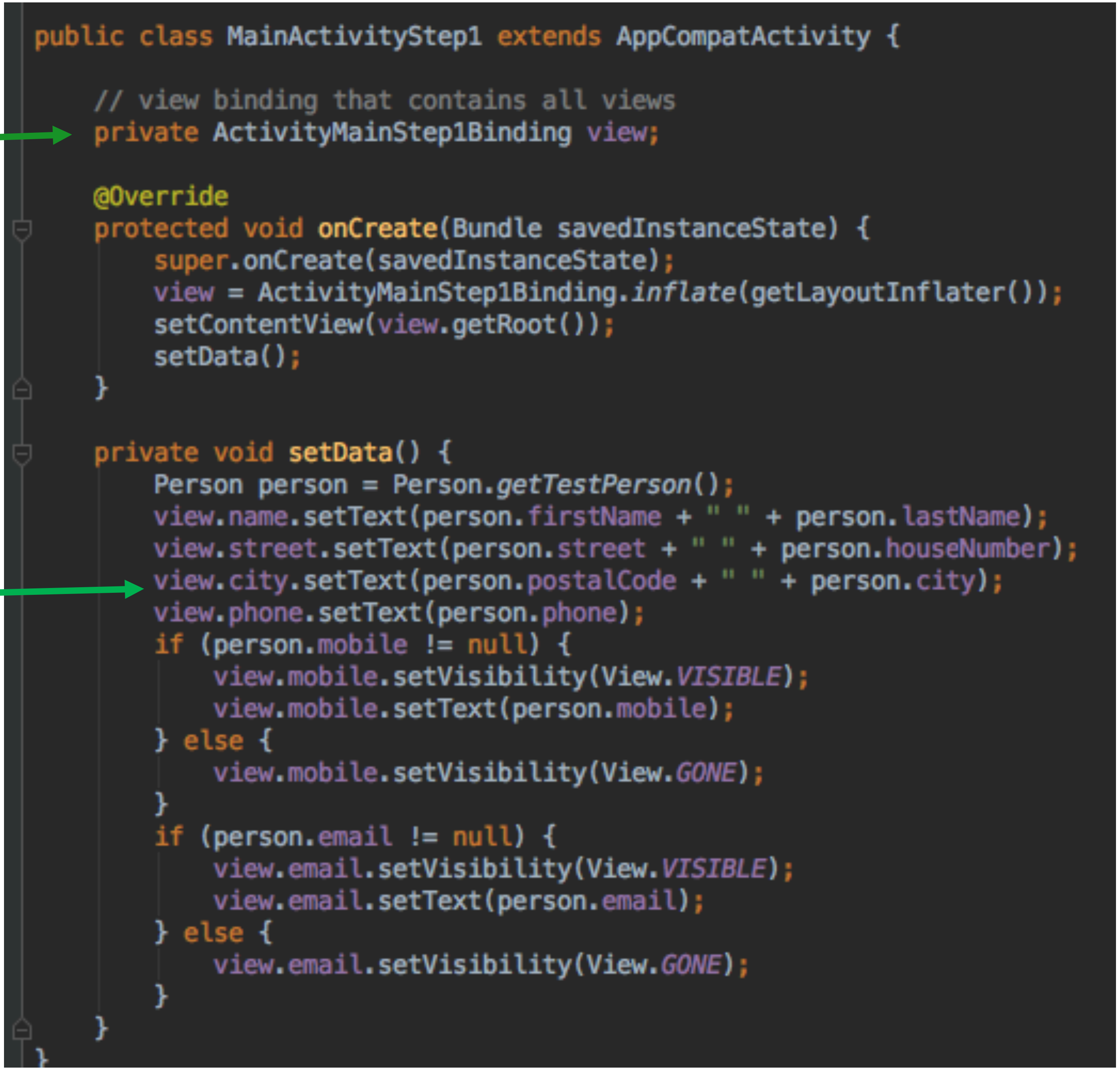
Autogeneriertes Binding Object

- Ein statt mehrerer View Member
- Kein lästiges findViewById() mehr

Spezifische Views in Binding autogeneriert

- Kein Cast mehr

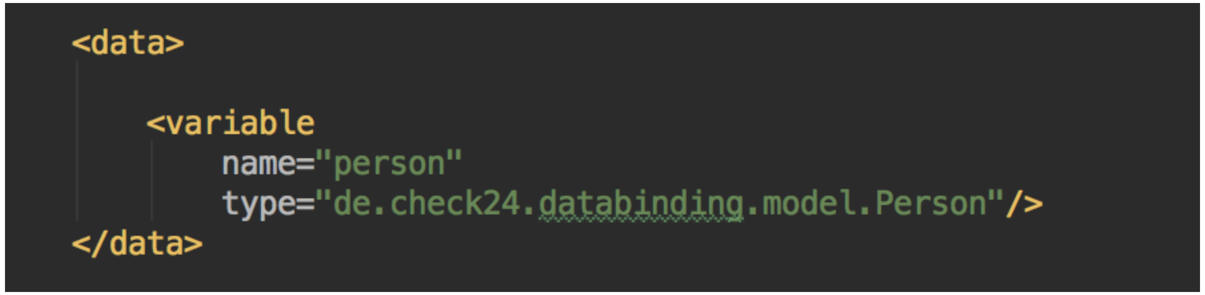
Daten setzen noch sehr unschön



```
public class MainActivityStep1 extends AppCompatActivity {

    // view binding that contains all views
    private ActivityMainStep1Binding view;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        view = ActivityMainStep1Binding.inflate(getLayoutInflater());
        setContentView(view.getRoot());
        setData();
    }

    private void setData() {
        Person person = Person.getTestPerson();
        view.name.setText(person.firstName + " " + person.lastName);
        view.street.setText(person.street + " " + person.houseNumber);
        view.city.setText(person.postalCode + " " + person.city);
        view.phone.setText(person.phone);
        if (person.mobile != null) {
            view.mobile.setVisibility(View.VISIBLE);
            view.mobile.setText(person.mobile);
        } else {
            view.mobile.setVisibility(View.GONE);
        }
        if (person.email != null) {
            view.email.setVisibility(View.VISIBLE);
            view.email.setText(person.email);
        } else {
            view.email.setVisibility(View.GONE);
        }
    }
}
```

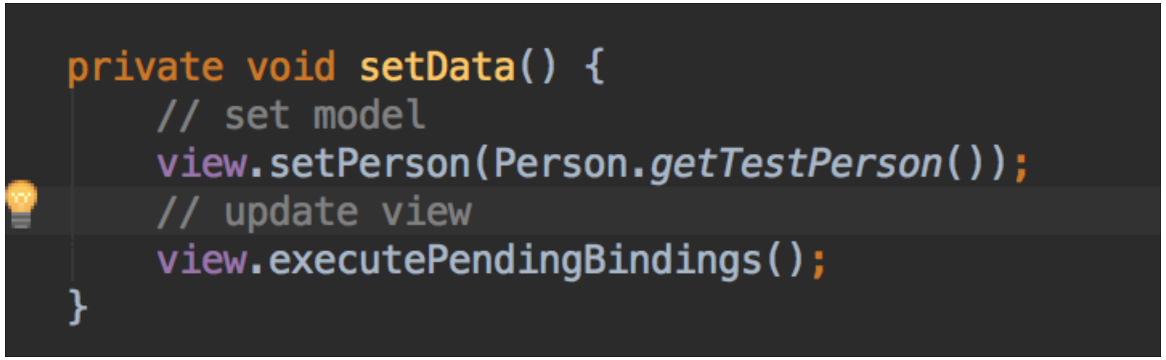
Databinding in Android - Schritt 2

- <data> tag im Layout setzen



```
<data>

    <variable
        name="person"
        type="de.check24.databinding.model.Person"/>
</data>
```

- Daten setzen auf Binding Objekt



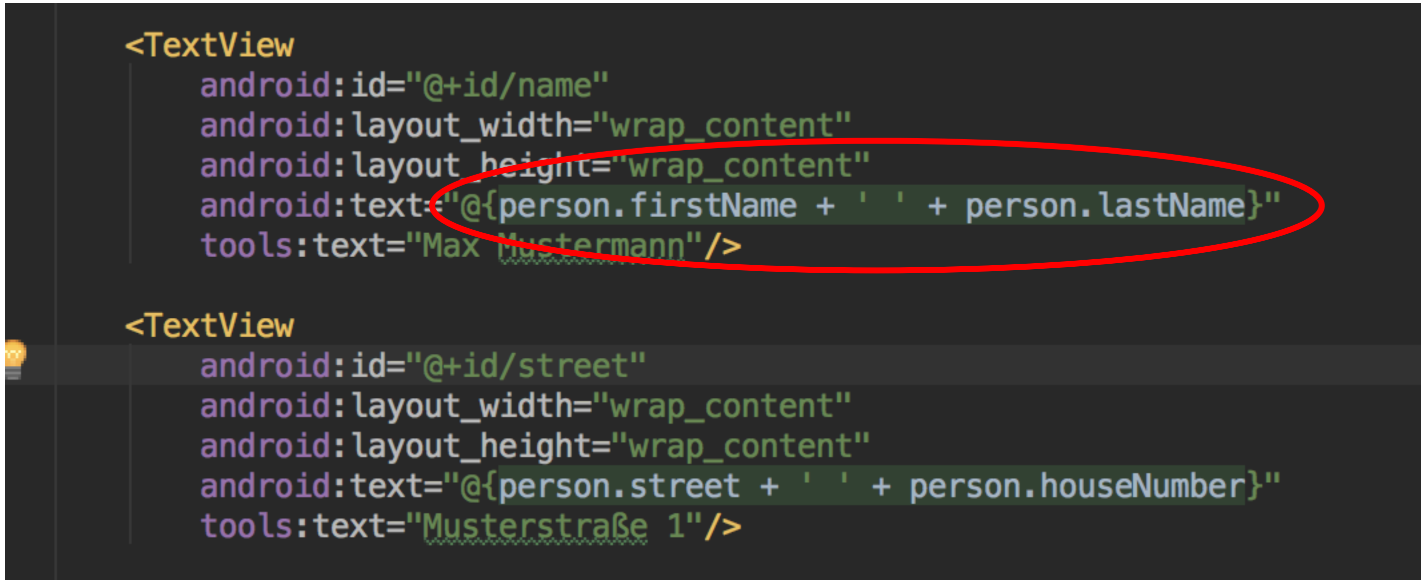
```
private void setData() {
    // set model
    view.setPerson(Person.getTestPerson());
    // update view
    view.executePendingBindings();
}
```

Databinding in Android - Schritt 2

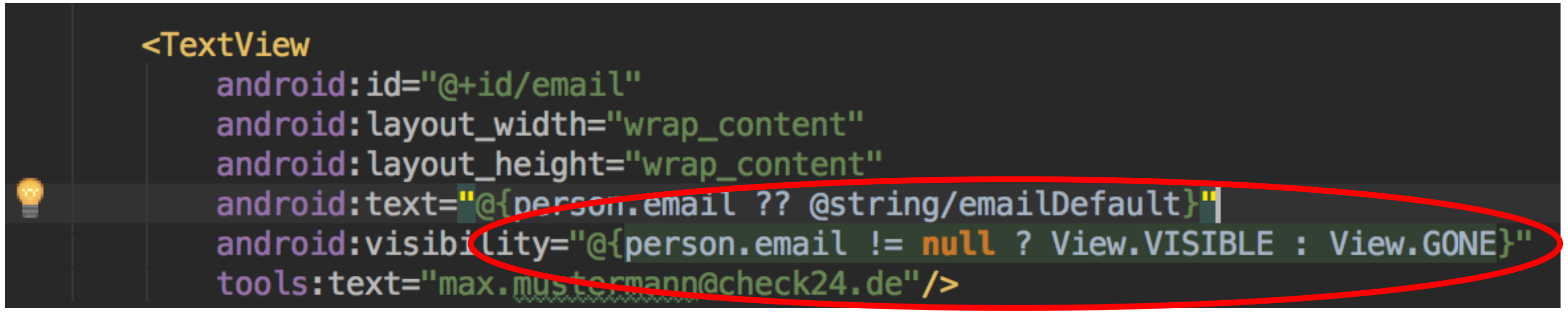
Zugriff auf das Model im Layout mit Binding Expression @{...}

Zu viel Logik im Layout!

- nicht zu debuggen
- Nicht testbar mit Unit-Test



```
<TextView
    android:id="@+id/name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{person.firstName + ' ' + person.lastName}"
    tools:text="Max Mustermann"/>

<TextView
    android:id="@+id/street"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{person.street + ' ' + person.houseNumber}"
    tools:text="Musterstraße 1"/>
```



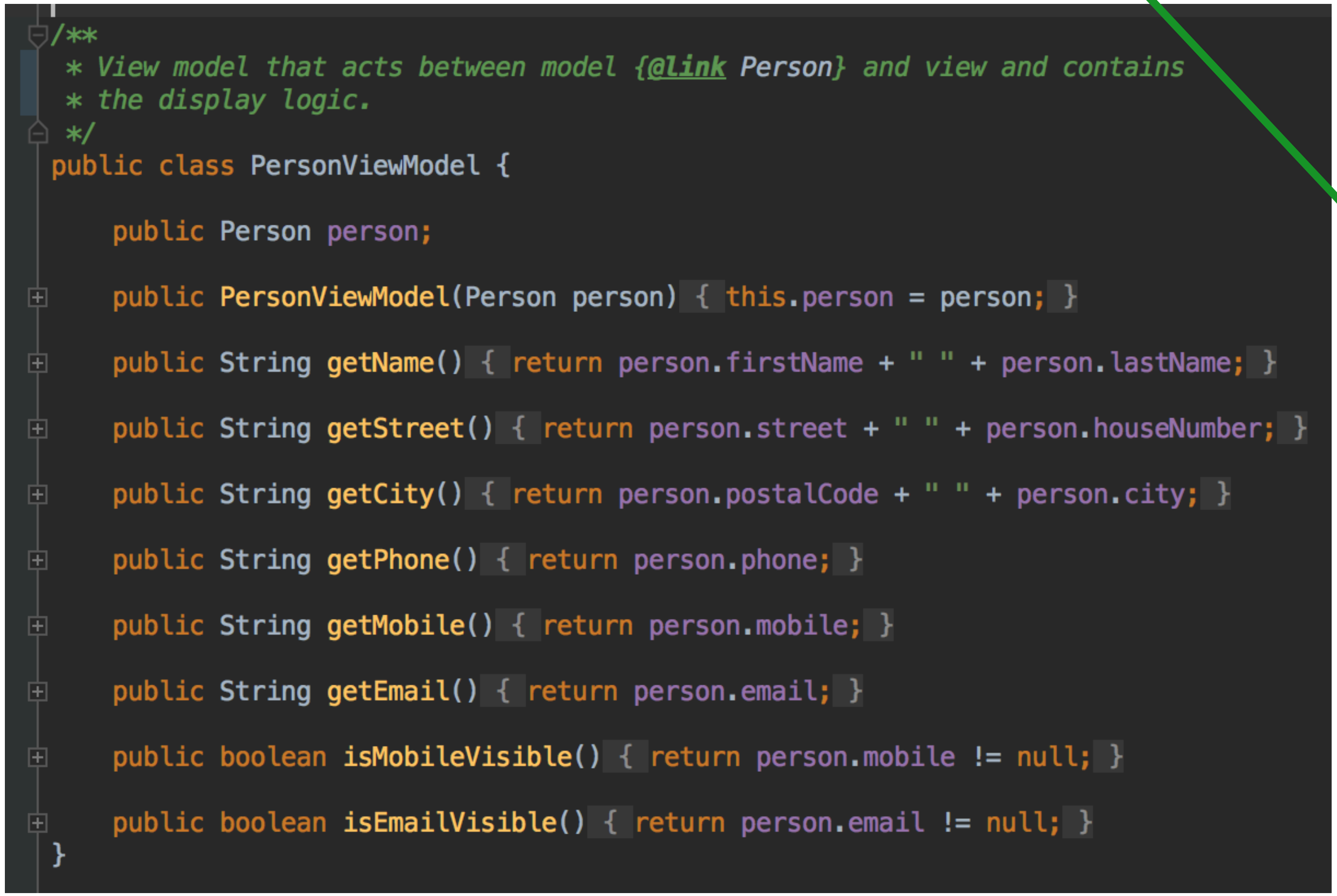
```
<TextView
    android:id="@+id/email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{person.email ?? @string/emailDefault}"
    android:visibility="@{person.email != null ? View.VISIBLE : View.GONE}"
    tools:text="max.mustermann@check24.de"/>
```

Databinding in Android - Schritt 3

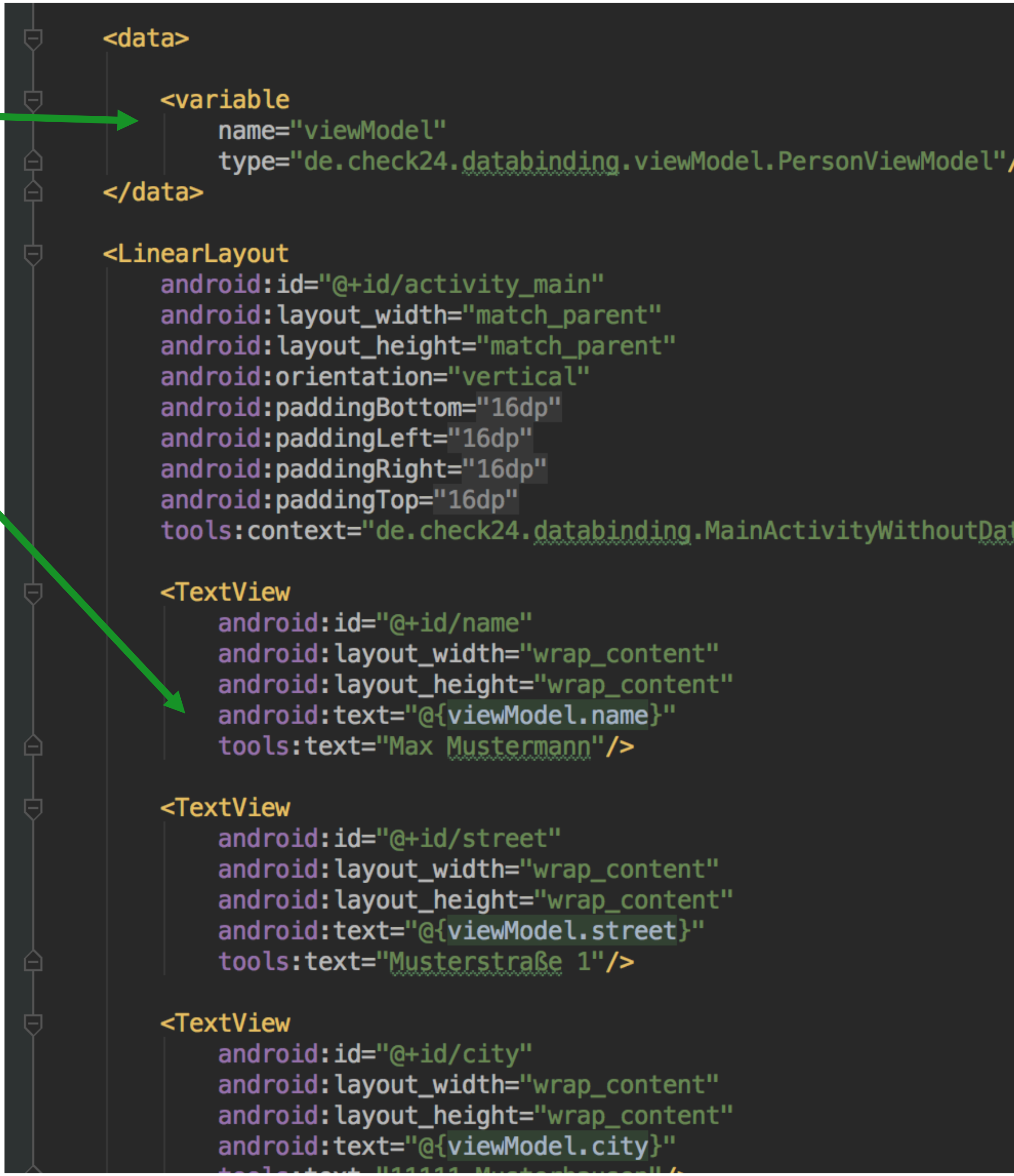
Einführung von ViewModel

Display Logik in ViewModel

Zugriff auf ViewModel in xml



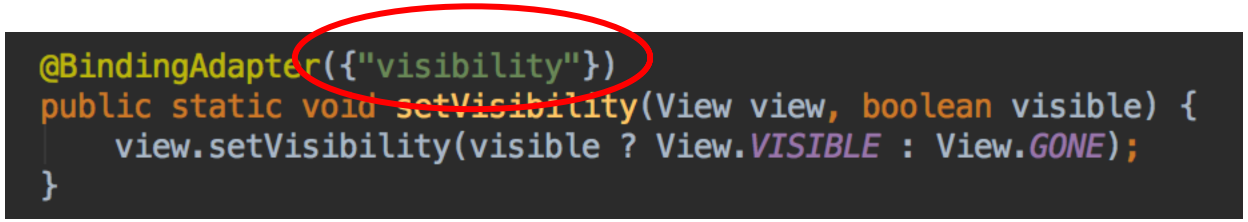
```java
/**
 * View model that acts between model {@link Person} and view and contains
 * the display logic.
 */
public class PersonViewModel {

    public Person person;

    public PersonViewModel(Person person) { this.person = person; }

    public String getName() { return person.firstName + " " + person.lastName; }

    public String getStreet() { return person.street + " " + person.houseNumber; }

    public String getCity() { return person.postalCode + " " + person.city; }

    public String getPhone() { return person.phone; }

    public String getMobile() { return person.mobile; }

    public String getEmail() { return person.email; }

    public boolean isMobileVisible() { return person.mobile != null; }

    public boolean isEmailVisible() { return person.email != null; }
}
```



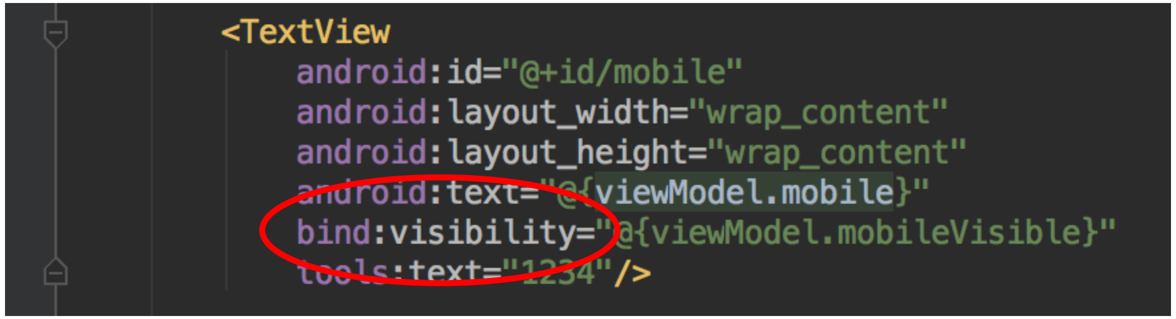
```xml
<data>

    <variable
        name="viewModel"
        type="de.check24.databinding.viewModel.PersonViewModel"/
</data>

<LinearLayout
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="de.check24.databinding.MainActivityWithoutDat

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@{viewModel.name}"
        tools:text="Max Mustermann"/>

    <TextView
        android:id="@+id/street"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@{viewModel.street}"
        tools:text="Musterstraße 1"/>

    <TextView
        android:id="@+id/city"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@{viewModel.city}"
        tools:text="11111 Musterhausen"/>
```

Databinding in Android - Schritt 3

- Custom Binding Adapter



```
@BindingAdapter({"visibility"})
public static void setVisibility(View view, boolean visible) {
    view.setVisibility(visible ? View.VISIBLE : View.GONE);
}
```



```
        <TextView
            android:id="@+id/mobile"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{viewModel.mobile}"
            bind:visibility="@{viewModel.mobileVisible}"
            tools:text="1234"/>
```

Weitere Features

- Null Safety
- Null Coalascing Operator
- Custom Binding Conversions
- Observable Objects
- Observable Fields
- Two-Way-Binding
- …

Fazit

- Weniger Boilerplate Code (View member, findViewById)
- ‘Bequemere’ Verknüpfung zwischen Model und View

- Vorsicht, nicht zuviel Logik ins Layout (nicht testbar, nicht debuggable)
- Display Logik ins ViewModel

Links

Github Sample Project
https://github.com/AndreRoss/databinding_sample

Android Developer Documentation
https://developer.android.com/topic/libraries/data-binding/index.html

Fragmented Podcast Episode *057: Data Binding with GDE Lisa Wray*
http://fragmentedpodcast.com/2016/09/

Vielen Dank für die Aufmerksamkeit!

Interesse an weiteren Android-Themen?

Databinding Advanced
Android Grundlagen
Material Design
Layouts
Dependency Injection mit Dagger 2
RxJava
Unit-Testing
UI Testing mit Espresso