



Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

Android-Zusammenfassung

Felix Krautschuk

(Matrikelnummer: 34230)

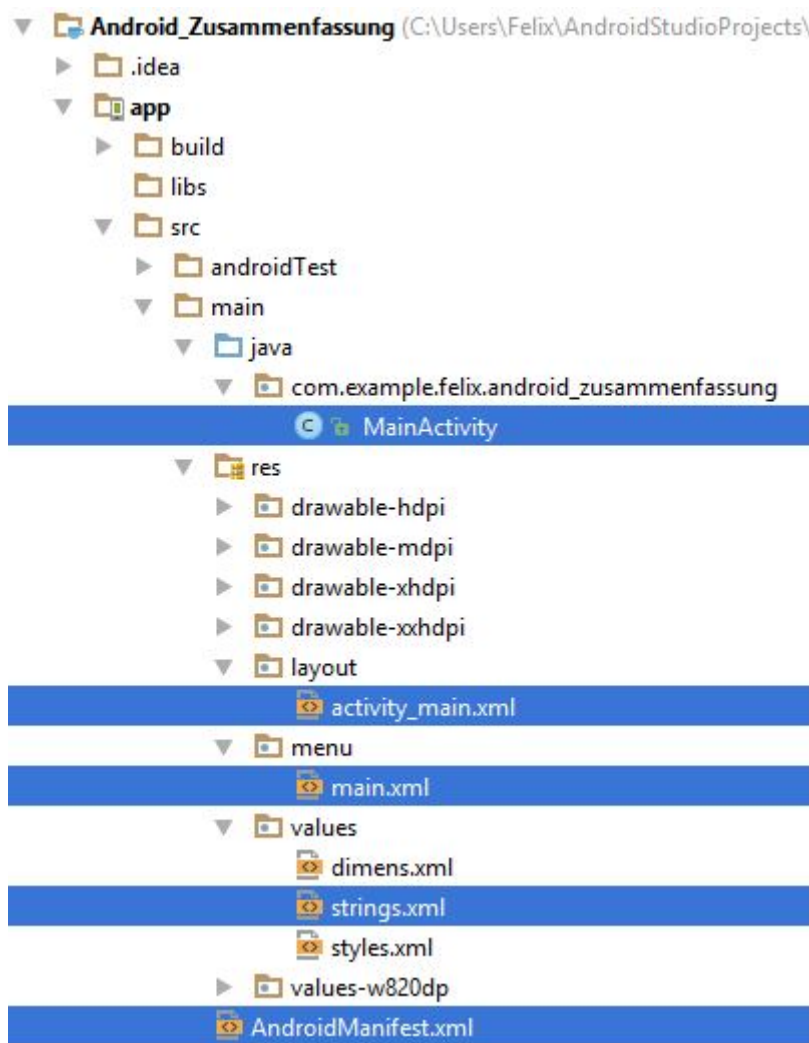
Studiengang Informatik

5. Semester

Contents

1	Programmstruktur - die wichtigsten Verzeichnisse und Dateien	1
1.1	AndroidManifest.xml	1
1.2	strings.xml	3
1.3	main.xml	4
1.4	activity_main.xml	5
2	Layouts, Views und Komponenten	6
2.1	Layouts	6
2.2	Views und Widgets	6
2.3	Basiskomponenten einer App	6
2.3.1	Activity	6
2.3.2	Intents	6
2.3.3	Fragments	6
2.3.4	Services	6
2.3.5	Menüs	6
3	Allgemeiner Ablauf bei einer einfachen Beispiel-App	6
3.1	Anlegen eines Projektes und Erstellung einer MainActivity	6
3.2	Layout festlegen und Views hinzufügen	6
3.3	ID und Namen eines jeden Views festlegen	6
3.4	Einbinden der View-Elemente in die Activity-Klasse	6
3.5	Actionbar mit Image-Buttons erstellen	6
3.6	Intents zum Aufruf einer Activity aus der aktuellen Activity	6
3.7	Anmelden der Activities im Manifest	6

1 Programmstruktur - die wichtigsten Verzeichnisse und Dateien



1.1 AndroidManifest.xml

Zu jeder App gehört eine zentrale Beschreibungsdatei. Sie enthält eine Liste der Komponenten, aus denen das Programm besteht und befindet sich in der obersten Ebene des Projektverzeichnis. Außerdem werden in ihr die benötigten Berechtigungen sowie etwaige zusätzlich verwendete Bibliotheken vermerkt. Auch Angaben zur mindestens nötigen oder gewünschten Android-Version werden hier eingetragen.

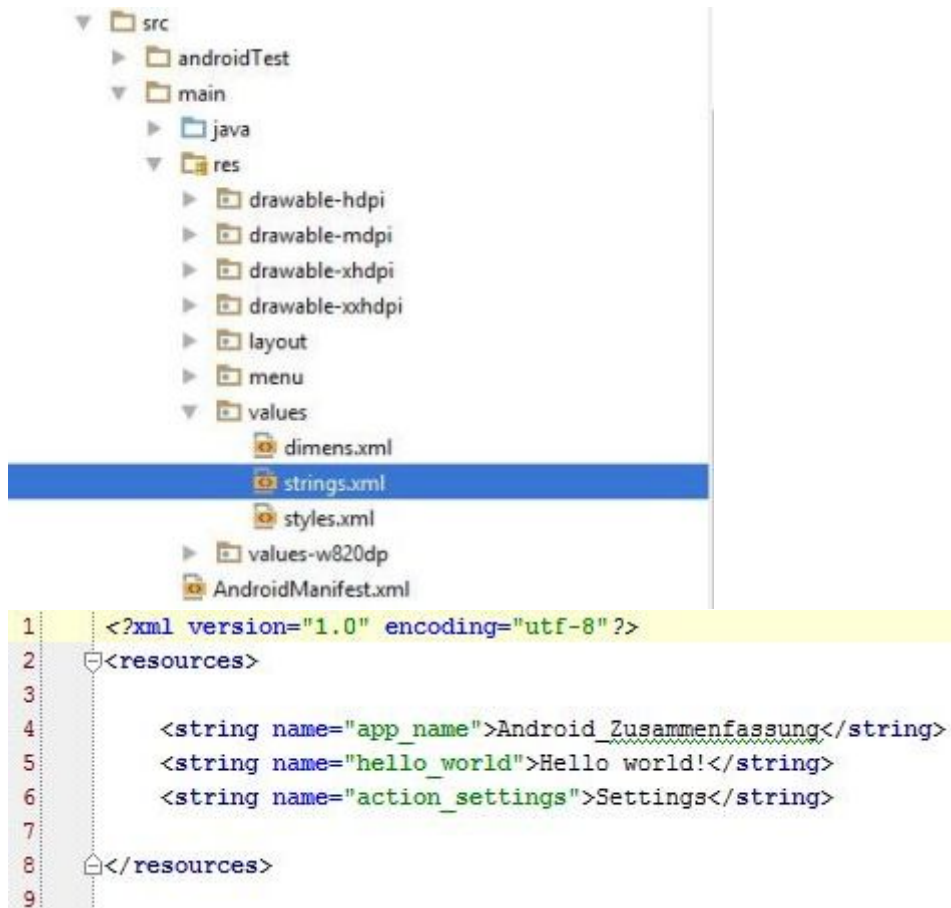
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.felix.android_zusammenfassung" >
4
5      <application
6          android:allowBackup="true"
7          android:icon="@drawable/ic_launcher"
8          android:label="Android_Zusammenfassung"
9          android:theme="@style/AppTheme" >
10         <activity
11             android:name=".MainActivity"
12             android:label="Android_Zusammenfassung" >
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>

```

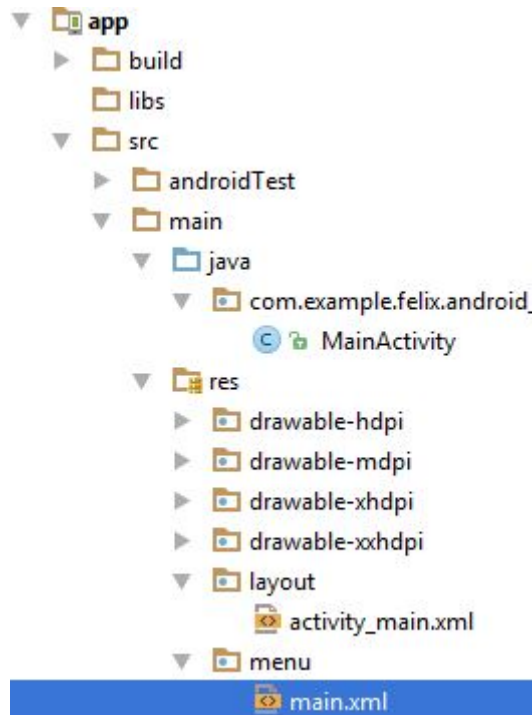
Die Komponenten einer Anwendung sind Kinder des Elements `<application/>`. Wenn man im Assistenten zum Anlegen neuer Projekte Create Activity mit einem Häkchen versehen und einen Namen einträgt, enthält das Manifest ein Element `<activity />`. Dessen Attribut `android:name` beinhaltet den im Assistenten eingegebenen Activity-Namen. Wenn man manuell eine Activity-Klasse anlegt (eine Klasse anlegt und mit *extends Activity* versehen), muss man nachträglich die erzeugte Activity im Manifest bekannt machen. Mithilfe des Elementes `<intent-filter />` kann man eine Activity zur Haupt-Activity machen. Dessen Kindelement `<action />` kennzeichnet die Activity als Haupteinstiegspunkt in die Anwendung. `<category />` sorgt dafür, dass sie im Programmstarter angezeigt wird.

1.2 strings.xml

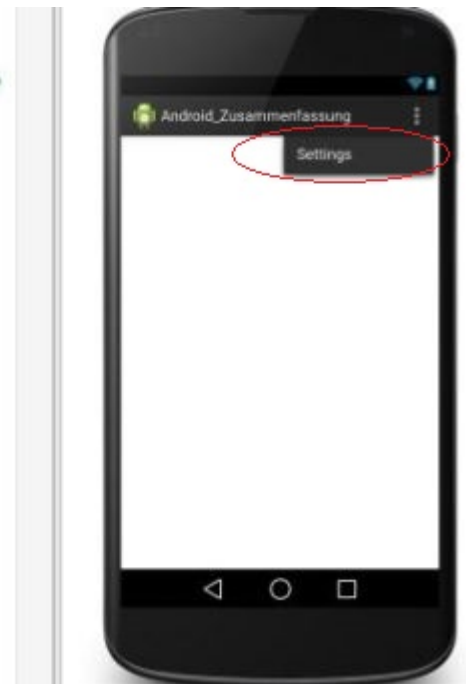


Elemente haben einen Titel. Üblicherweise werden sämtliche Titel in die String-Resource Datei *string.xml* eingetragen und über *@string/...* referenziert. Die Speicherung von Texten an einem zentralen Ort hat zahlreiche Vorteile. Beispielsweise werden identische Textteile leichter entdeckt, als wenn diese in den Quelltexten der Klassen verborgen sind und man erst jede Klasse oder Layout-Datei durchsuchen muss. Damit lässt sich, wenn auch in eher bescheidenem Umfang, Speicherplatz sparen. Außerdem macht die Trennung von Daten und Programmlogik die Internationalisierung, also die Übersetzung einer App in verschiedene Sprachen, viel einfacher. Hierzu wird für jede zu unterstützende Sprache im Ordner *res* ein Verzeichnis angelegt. Dessen Name beginnt mit *values-* und endet mit dem ISO-Sprachschlüssel. Für Deutsch ist dies *de*. Das Verzeichnis muss also *values-de* heißen. Jeder dieser Ordner erhält eine eigene Version von *strings.xml*. Deren Bezeichner sind stets gleich, die Texte liegen hingegen in den jeweiligen Sprachen vor. Texte in der Standardsprache verbleiben in *values*.

1.3 main.xml

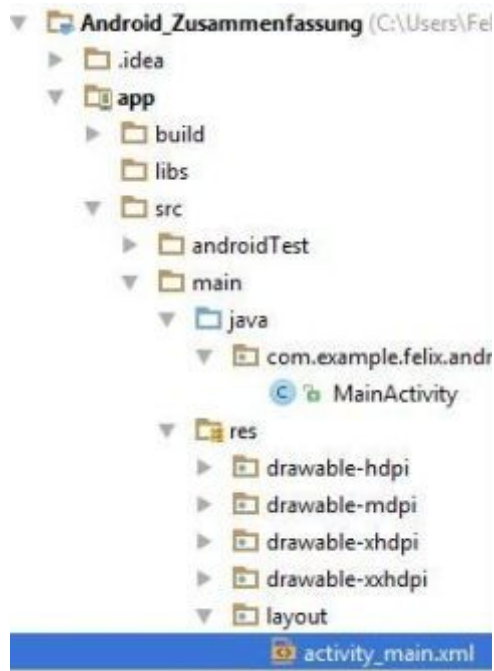


```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity" >
    <item android:id="@+id/action_settings"
          android:title="Settings"
          android:orderInCategory="100"
          android:showAsAction="never"
        />
</menu>
```



Die *main.xml* ist eine sogenannte Menu-Resource-Datei. Menu-Resource-Dateien sind im Verzeichnis 'src/menu' hinterlegt und dienen dazu Options-Menüs oder Kontextmenüs zu erstellen (Erläuterungen dazu im späteren Kapitel Menüs), die man z.B. mit der Menütaste des Smartphones betätigen aufrufen kann. Menü-Items sollten immer in solchen Menü-Dateien aufgebaut werden statt in den Activity-Dateien. Man kann die erstellten Menüs später in Activities oder Fragments einbinden.

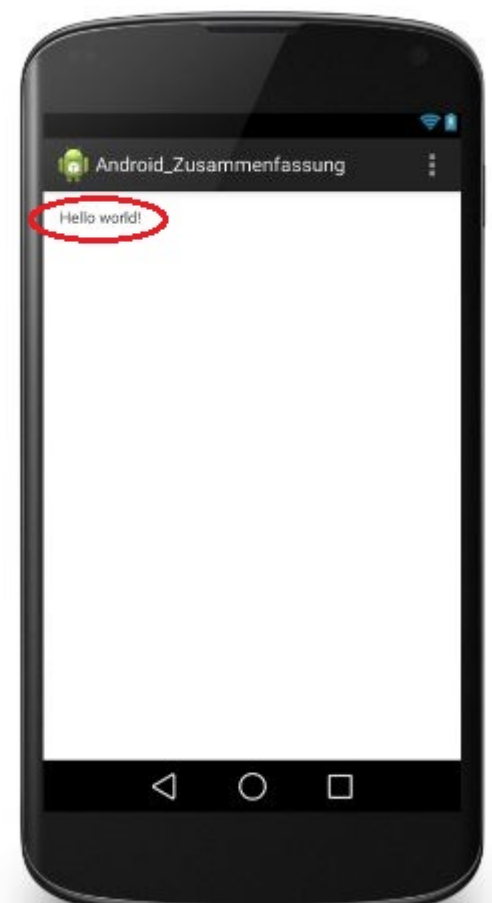
1.4 activity_main.xml



```
<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"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MainActivity">

    <TextView
        android:text="Hello world!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



2 Layouts, Views und Komponenten

2.1 Layouts

2.2 Views und Widgets

2.3 Basiskomponenten einer App

2.3.1 Activity

Normalerweise ist jeder Activity eine Benutzeroberfläche, also ein Baum bestehend aus Views und ViewGroups, zugeordnet. Activities bilden demnach die vom Anwender wahrgenommenen Bausteine einer App. Sie repräsentieren also meist die Benutzeroberfläche und Interaktionen. Jede Android- Anwendung besteht deshalb aus mindestens einer Activity. Activities können andere Activities aufrufen und mit ihnen Daten austauschen. Jede Activity besteht aus einer in XML definierten Layout-Datei und einer dazugehörigen Java-Klassendatei, welche bei Android Studio im Verzeichnis `'%Appname/src/main/java/%packagename/'` abgelegt ist. Die dazugehörige Layout-Datei (XML-Datei) ist im Verzeichnis `'%Appname/src/main/res/layout'` zu finden.

2.3.2 Intents

2.3.3 Fragments

2.3.4 Services

2.3.5 Menüs

3 Allgemeiner Ablauf bei einer einfachen Beispiel-App

3.1 Anlegen eines Projektes und Erstellung einer MainActivity

3.2 Layout festlegen und Views hinzufügen

3.3 ID und Namen eines jeden Views festlegen

3.4 Einbinden der View-Elemente in die Activity-Klasse

3.5 ActionBar mit Image-Buttons erstellen

3.6 Intents zum Aufruf einer Activity aus der aktuellen Activity

3.7 Anmelden der Activities im Manifest