Fachbereich Automatisierung und Informatik	▲ Hochschule Harz Hochschule für angewandte Wissenschaften Wernigerode

Handbuch

im Fach

„Mobile Programmierung mit Android“

SoSe 2020

Stand 08.07.2020

Inhaltsverzeichnis

1 Abbildungsverzeichnis 9

2 Android 13

2.1 Installation 13

2.1.1 Erste Installation 13

2.1.2 Zweite Installation 16

2.2 Erstellen eines neuen Projektes 21

2.3 SDK-Installation 23

2.3.1 • SDK durch das Android-Studio runterladen 23

2.3.2 • SDK durch einen Link runterladen 26

2.4 Virtuelle Maschine (AVD-Manager) 27

3 Erste einfache Beispiele 34

3.1 Wichtige Grundeigenschaften von UI-Elementen 34

3.2 Erstes Beispiel 36

3.2.1 UI-Prinzip 36

3.2.2 Erstellen eines Projektes 36

3.2.3 UI-View 38

3.2.4 Umbau des Projektes 40

3.2.5 Schalter einfügen 42

3.2.6 Zweiten PlainText eintragen 44

3.2.7 Java-Code 46

3.3 Zweites Beispiel mit UI-Elementen 50

3.3.1 Anlegen eines neuen Projektes 50

3.3.2 Anlegen der UI-Elemente 50

3.3.3 Anlegen von einer RadioGroup: 51

3.3.4 Anlegen von zwei RadioButtons: 51

3.3.5 Anlegen eines Switches: 52

3.3.6 Anlegen eines Spinner (Haupteintrag: Containers): 52

3.3.7 Anlegen eines 2. Spinner (Haupteintrag: Containers): 52

3.3.8 Anlegen eines ToggleButtons 53

3.3.9 Anlegen eines Buttons 53

3.3.10 Anlegen eines MultiLineTextes 53

3.3.11 onClickAction 54

3.3.12 Java-Code 55

3.4 Drittes Beispiel mit UI-Elementen: Add/Sub-Rechner 59

3.4.1 Anlegen eines neuen Projektes 59

3.4.2 Anlegen der UI-Elemente 59

3.4.3 Java_Code 66

3.5 Beispiel Spritverbrauch 69

3.5.1 String-Konstanten 71

3.5.2 Farb-Konstanten 72

3.5.3 Spezifikation der UI-Elemente 72

3.5.4 Java-Code 74

3.5.5 Speichern der Eingabedaten 76

3.5.6 Laden der Eingabedaten 76

4 Einfache UI-Elemente 77

4.1 TextView (Label) 77

4.2 EditText 78

4.2.1 Abfrage: 79

4.2.2 ChangeListener 80

4.3 Button 81
4.3.1 Abfrage: 82
4.3.2 ChangeListener 82
4.3.3 Button mit Rand (Border) 82

4.4 CheckBox 84
4.4.1 Abfrage: 85
4.4.2 ChangeListener 85

4.5 RadioButton 85
4.5.1 Abfrage: 86
4.5.2 ChangeListener 87

4.6 Switch 88
4.6.1 Abfrage 88
4.6.2 ChangeListener 89

4.7 Spinner 90
4.7.1 Abfrage: 91
4.7.2 ChangeListener 91

5 Menüs (drei Punkte) 93

6 Constraint-Layout 96

6.1 Attribute 96
6.1.1 layout_width 96
6.1.2 layout_height 96

6.2 Konstanten 96
6.2.1 Positionskonstanten 96
6.2.2 Relative positioning 97
6.2.3 Margin 97
6.2.4 Bias (Zentriert) 97
6.2.5 Winkel Positionen 98

6.3 Weitere Eigenschaften 98
6.3.1 Ratio 98

6.4 Groups 99
6.4.1 Barirers 99
6.4.2 Prozent 99
6.4.3 fillparent 100
6.4.4 Tipps 100
6.4.5 Unterschiedliche Länge der Beschriftungen 102

7 LinearLayout 105

7.1 vertikal 105

7.2 horizontal 105

7.3 Gravity-Konstanten 108

7.4 RelativeLayout 108
7.4.1 Beispiele: 111

8 TableLayout 112

8.1 Beispiel TableLayout 112

9 GridLayout 116

10 ListView 120

10.1 Einfaches ListView-Beispiel 121
10.1.1 activity_main.xml 121
10.1.2 Darstellung der Zeile 122

10.1.3 Java Code 123

10.2 ListView-Beispiel mit Studenten-Adapter 125
10.2.1 activity_main.xml 125
10.2.2 Strings Konstanten 127
10.2.3 Darstellung der Zeile 128
10.2.4 Semesterbilder 130
10.2.5 Klasse Student 130
10.2.6 Quellcode der Klasse Student 131
10.2.7 StudentenListe 132
10.2.8 StudentenAdapter 134
10.2.9 Java Code in MainActivity 136

10.3 ListView Studenten Beispiel Weitere Verfeinerung 137
10.3.1 Klasse Basis einfügen 138
10.3.2 Menü 138
10.3.3 Klick-Events 141
10.3.4 TextViews zum Testen in die neue Activity einfügen 143
10.3.5 Aufruf des neuen Views 146
10.3.6 Parameterübergabe 146
10.3.7 EditDeleteNew.java 148
10.3.8 Beenden 151
10.3.9 Schalter Cancel 152
10.3.10 Schalter Ok für Edit-Action 152
10.3.11 Abfangen der Rückantwort 152
10.3.12 Komplettes Layouts der zweiten Acitivty 153
10.3.13 Prüfen der Eingaben 158
10.3.14 Schalter Ok in EditDeleteNew.java 159

10.4 Änderungemethoden in der Studentenliste 160
10.4.1 Ändern eines Studenten (Studentenliste) 160
10.4.2 Löschen eines Studenten (Studentenliste) 161
10.4.3 Einfügen eines neuen Studenten (Studentenliste) 161

10.5 Laden und Speichern (Studentenliste) 161
10.5.1 Laden der Studenten: 161
10.5.2 Speichern der Studenten: 162

10.6 Laden und Speichern eines Studenten (Klasse Student.java) 162
10.6.1 Speichern eines Studenten 163
10.6.2 Laden eines Studenten 163

10.7 Umbau des Ladens der Liste 164

10.8 Abfrage des Ergebnisses des zweiten Views 164

10.9 Symbole im Optionsmenu 165

10.10 ListView mit Checkboxen 165
10.10.1 row_layout.xml 166
10.10.2 Student.java 167
10.10.3 StudentenAdapter 167
10.10.4 Optionsmenü 169

11 MultiPages 173

11.1 Bottom Navigation Activity 173
11.1.1 Neues Projekt erstellen 175
11.1.2 Struktur des Projektes 175
11.1.3 Plätze der Fragmente 176
11.1.4 bottom_nav_menu.xml 177
11.1.5 Anlegen eines neuen Fragments 180
11.1.6 Testlauf 186
11.1.7 Add-Fragment als Start-Fragemnt setzen. 186
11.1.8 Aktueller Stand 187

11.1.9 Weitere Symbole 187
11.1.10 Icon aus der Androidliste einfügen 188
11.1.11 Texte für die Fragmente 191
11.1.12 fragment_add aufbauen 191
11.1.13 Ändern des activity_main-Layout: 193
11.1.14 Quellcode von fragment_add.xml 194
11.1.15 Add-Fragment-Javacode 197
11.1.16 Klasse Basis.java 199
11.1.17 Tastatur entfernen 199
11.1.18 MainActivity.java 199
11.1.19 AddFragment.java 200
11.1.20 Musterlösung 201

11.2 Tabbed Activity 202
11.2.1 Neues Projekt erstellen 204
11.2.2 Struktur des Vorlagen-Projektes 204
11.2.3 Plätze der Fragmente 206
11.2.4 strings.xml 206
11.2.5 Anlegen des neuen AddFragment 206
11.2.6 strings.xml 207
11.2.7 Einfügen des AddFragmentes mit dem Menü. 207
11.2.8 SectionsPagerAdapter 209
11.2.9 Testlauf 210
11.2.10 Eintragen der XML-Daten in fragment_add.xml 210
11.2.11 Eintragen des Quellcodes in Addfragment.java 213
11.2.12 Klasse Basis.java 216
11.2.13 Aktueller Stand 216
11.2.14 Löschen der vorgefertigten Fragmente 216
11.2.15 Ändern in SectionsPageAdapter 217
11.2.16 Erstellen der restlichen Fragmente 218
11.2.17 fragment_sub.xml 218
11.2.18 fragment_mult.xml 221
11.2.19 Quellcode von SubFragment.java 223
11.2.20 Quellcode von MultFragment.java 225
11.2.21 MainActivity.java 227
11.2.22 AddFragment.java 228
11.2.23 Musterlösung 229

11.3 Tabbed Activity mit Optionsmenu 230
11.3.1 Neues Projekt 230
11.3.2 Anlegen und Kopieren der Klasse Basis.java 230
11.3.3 strings.xml 231
11.3.4 Anlegen der drei Fragmente Add, Sub, Mult 231
11.3.5 Ändern der Texte in den Fragmenten: 234
11.3.6 activity_main.xml 235
11.3.7 Navigation 235
11.3.8 Testlauf 236
11.3.9 Optionsmenu erstellen 236
11.3.10 Optionsmenu einfügen in die mainActivity 238
11.3.11 Optionsmenu-Start-Event abfangen 239
11.3.12 Optionsmenu-Klick-Event abfangen 239
11.3.13 Navigationscontroller 239
11.3.14 Fragment Add 240
11.3.15 AddFragment.java 243
11.3.16 Fragment Sub 245
11.3.17 SubFragment.java 248
11.3.18 Fragment Mult 250
11.3.19 MultFragment.java 253
11.3.20 Abbildung der Musterlösung 255

11.4 Swipe-Activity 256
11.4.1 Anlegen des Projektes 256

11.4.2 activity_main.xml 257
11.4.3 Kopieren der Bilder 257
11.4.4 Anlegen des Fragments 257
11.4.5 Ändern der Factory 258
11.4.6 SectionsPageAdapter 260
11.4.7 fragment_picture.xml 261
11.4.8 PictureFragment.java 261

11.5 Master Detail Activity 267

11.6 Eigene Navigation 268
11.6.1 Musterlösung 268
11.6.2 Projekt anlegen 269
11.6.3 Vorarbeiten 270
11.6.4 Startbildschirm 270
11.6.5 Neue Packages 273
11.6.6 Bottom Navigation Activity 275
11.6.7 Tabbed Bar Activity 276
11.6.8 ListView 280

12 Activity aufrufen 285

12.1 Neue Activity erstellen, mittels Menü) 285

13 Optionsmenü 288

13.1 Menü erstellen 288

13.2 Datei menu.xml anlegen 288

13.3 Menüeinträge erstellen 289

13.4 Optionsmenu einfügen in die mainActivity 290

13.5 Optionsmenu-Start-Event abfangen 291

13.6 Optionsmenu-Klick-Event abfangen 291

14 Speichern mittels DataOutputStream und DataInputStream 292

14.1 DataOutputStream 292
14.1.1 Pures Java 292
14.1.2 DataOutputStream in Android 292
14.1.3 Methoden der Klasse DataOutputStream 293

14.2 DataInputStream 293
14.2.1 Pures Java 294
14.2.2 DataInputStream in Android 294
14.2.3 Methoden der Klasse DataInputStream 295

14.3 Beispiele 295
14.3.1 Speichern und laden einer Instanz einer Klasse 295
14.3.2 Speichern und laden von mehreren Instanzen einer Klasse 297

15 Die Datenbank SQLite 301

15.1 SQLite 301
15.1.1 Eigenschaften 301
15.1.2 Datentypen 301
15.1.3 Gemappte Datentypen 301
15.1.4 Aufbau in Android 302
15.1.5 Datenbank erzeugen 303
15.1.6 Insert 304
15.1.7 getQuery 305
15.1.8 Löschen der Datenbank 306

15.2 Beispiel einer einfachen SQLite-App 306
15.2.1 Anlegen eines Projektes 307

15.2.2 colors.xml 307
15.2.3 strings.xml 307
15.2.4 activity_main.xml 308
15.2.5 Student.java 311
15.2.6 DBSOpenHandler.java 311
15.2.7 DBSQueries 313
15.2.8 Basis.java 316
15.2.9 MainActivity 316
15.2.10 ShowStudentsActivity1 318
15.2.11 ShowStudentsActivity2 322

16 ListView und die Studentendatenbank 327

16.1 Einfügen der Klasse DBSOpenHandler 327

16.2 Klasse Student 328

16.3 DBSQueries 329

16.4 MyCursorAdapter 331

16.5 MainActivity 332

16.6 EditDeleteNewActivity 335

17 Sensoren 340

17.1 Sensor-Typen 340

17.2 Allgemeiner Aufbau 340

17.3 Beispiele der Sensoren 341
17.3.1 Accelerator (Beschleunigung) 341
17.3.2 Temperatur 342
17.3.3 Gravity (Schwerkraft) 342
17.3.4 Gyroskope (Lade, zum Beispiel für Spiele) 343
17.3.5 Magnetfeld (für Kompass) 344
17.3.6 Luftdruck (Pressure) 345
17.3.7 Luftfeuchtigkeit (Humidity) 346
17.3.8 Licht (Light) 347

17.4 Alle Sensoren 349
17.4.1 String.xml 349
17.4.2 Color.xml 349
17.4.3 activity_main.xml 350
17.4.4 Klasse SensorItem.java 352
17.4.5 MainActivity.java 352

18 Chronometer 359

18.1 Projekt-Erstellung 359

18.2 colors.xml 359

18.3 strings.xml 360

18.4 Layout: 360

18.5 MainActivity.java 363

19 Anhang 365

19.1 toast 365

19.2 decimalFormat 365

19.3 Umwandlung eines Textes in eine Integer-Zahl 365

19.4 Umwandlung eines Textes in eine Float-Zahl 366

19.5 Random-Zahl erzeugen 366
19.6 ListView, Update 367
19.7 Constraint-Layout, zwei Editoren untereinander 367
19.8 Klasse Basis.java 368
19.9 onClick, Code Behind 370
19.10 Erstellen eines „Plus“-Icons 370
19.11 Beispiel Symbole 373
20 Stichwortverzeichnis 378

1 Abbildungsverzeichnis

Abbildung 1 Begrüßungsdialog 13
Abbildung 2 Bitte mit der Virtualisierung 14
Abbildung 3 Installationsordner (Meine Wahl: c:\Android Studio) 14
Abbildung 4 Eintrag im Startmenü 14
Abbildung 5 Entzippen des Installationsprogramms 15
Abbildung 6 Die Installation wurde entzippt 15
Abbildung 7 Der erste Teile, der schnellere, ist beendet. 15
Abbildung 8 Import von vorherigen Versionen)optional) 16
Abbildung 9 Download des SDK 16
Abbildung 10 Warten vor denm langen Download 16
Abbildung 11 Hier Custom auswählen 17
Abbildung 12 Bessere Wahl 17
Abbildung 13 Java Runtime Edition Ordner (ja nach Pfad anders) 17
Abbildung 14 Farbwahl 18
Abbildung 15 Bitte unbedingt beide Haken setzen 18
Abbildung 16 Wer nicht mindestens 16 Hauptspeicher hat, unbedingt nichts ändern 18
Abbildung 17 Liste der Downloads 19
Abbildung 18 Jetzt geht der zweite Teil richtig los! 19
Abbildung 19 Laden der Zusatzdateien 19
Abbildung 20 Immer noch runterladen 20
Abbildung 21 Immer noch runterladen 20
Abbildung 22 Startdialog 21
Abbildung 23 Android Startfenster 21
Abbildung 24 Auswahl "Empty Activity" 22
Abbildung 25 Eigenschaften des neuen Projektes 22
Abbildung 26 Neues Projekt im Android Studio 23
Abbildung 27 Aufruf SDK Manager 24
Abbildung 28 SDK erstellen (Auswahl der Komponenten) 24
Abbildung 29 SDK Verzeichnis erstellen 25
Abbildung 30 Runterladen von Komponenten 25
Abbildung 31 SDK ist nun installiert 26
Abbildung 32 Aufruf SDK Manager 27
Abbildung 33 Erstellen einer virtuellen Maschine 27
Abbildung 34 Auswahl des Smartphones 28
Abbildung 35 Nun ein weiterer Download 29
Abbildung 36 Lizenz Bedingungen 30
Abbildung 37 Endlich fertig 30
Abbildung 38 Anzeige aller virtuellen Maschinen 31
Abbildung 39 Eigenschaften der virtuellen Madschine 31
Abbildung 40 Abschlussdialog 32
Abbildung 41 Die eventuelle Sicherheitsabfrage bejahen 33
Abbildung 42 Neue Projekt 36
Abbildung 43 Daten des ersten Projektes 37
Abbildung 44 Das erste Rumpfprojekt 37
Abbildung 45 UI-View im Textmodus 38
Abbildung 46 UI-View im Grafikmodus 39
Abbildung 47 UI-View im Textmodus / Grafikmodus 39
Abbildung 48 Constrainst des ersten TextViews 40
Abbildung 49 Anzeige des ersten Beispiels 41
Abbildung 50 Constrainst des ersten PlainText-Elemenmtes 42

Abbildung 51 Constrainst des Schalters 43
Abbildung 52 Constrainst des ersten PlainText-Elemenmtes....45
Abbildung 53 Textcolor ändern....45
Abbildung 54 Textsize ändern....45
Abbildung 55 Das User-Interface des ersten Beispiels 46
Abbildung 56 Anlegen des zweiten Beispiels Bsp02 50
Abbildung 57 Constrainst der CheckBox 51
Abbildung 58 Constrainst der RadioGroup 52
Abbildung 59 Constrainst des Switch....52
Abbildung 60 Constrainst der Überschrift....59
Abbildung 61 Constrainst derersten Beschriftung....60
Abbildung 62 Constrainst der 1. Zahl....60
Abbildung 63 Constrainst zweiten Beschriftung 60
Abbildung 64 Constrainst der 2. Zahl....61
Abbildung 65 Erster Stand des dritten Beispiels 61
Abbildung 66 Constrainst der dritten Beschriftung....62
Abbildung 67 Musterlösung 69
Abbildung 68 Berechnung nach der Berechnung Europa 70
Abbildung 69 Berechnung nach der Berechnung USA 71
Abbildung 70 Neue XML-Date für Border anlegen....83
Abbildung 71 Button mit rotem Rand 84
Abbildung 72 Alternatives Aussehen eines Buttons....84
Abbildung 73 Anlegen einer Menüdatei....93
Abbildung 74 Aktueller Strukturbaum des Projektes 94
Abbildung 75 Anzeige eines Menüs....94
Abbildung 76 Menü New und Delete 95
Abbildung 77 Menü for das Anzeigen der Baseline....101
Abbildung 78 Anzeige der Baseline 101
Abbildung 79 Verknüpfen der beiden Baseline....101
Abbildung 80 Gleiche Länge von Beschriftungen 102
Abbildung 81 Unterschiedliche Länge von Beschriftungen....103
Abbildung 82 Überblick eines Adapters....120
Abbildung 83 Anlegen der Datei simplerow.xml....122
Abbildung 84 Einfache ListView mit einem ArrayAdapter....124
Abbildung 85 ListView StudentenListe 126
Abbildung 86 ListView, View zum Bearbeiten....126
Abbildung 87 Struktur des Beispiels ListView mit Studenten....127
Abbildung 88 Anlegen der Datei row_layout.xml....128
Abbildung 89 Aktuelle Struktur des Projektes 130
Abbildung 90 Klasse Student 131
Abbildung 91 Klasse Studentenliste....133
Abbildung 92 Neue Klasse StudnetenAdapter 134
Abbildung 93 Erste Fassung der ListView mit Studenten....137
Abbildung 94 Klasse Basis.java 138
Abbildung 95 Anlegen einer Menüdatei....139
Abbildung 96 Aktueller Strukturbaum des Projektes 139
Abbildung 97 Anzeige eines Menüs....140
Abbildung 98 Menü New 140
Abbildung 99 Neue Activity erstellen 142
Abbildung 100 Neue Activity EditDeleteNew 142
Abbildung 101 Erster Entwurf des Layouts 143
Abbildung 102 Aktuelle Fassung der zweiten Activity....144
Abbildung 103 Startanzeige mit Anklicken eines Eintrags....151
Abbildung 104 Anzeige der Details....151

Abbildung 105 Aktueller Stand der Studenten-ListView 157
Abbildung 106 Auswahl des Semesters 158
Abbildung 107 Auswahl der Note 158
Abbildung 108 ListView im Selected-Modus 166
Abbildung 109 Ansicht des Vorlagen-Projektes (BottomNavigation) 173
Abbildung 110 Fertiges Projekt 173
Abbildung 111 Komplette Struktur der Musterlösung 174
Abbildung 112 Bottom Navigation Activity 175
Abbildung 113 Eintrag "ui" aktivieren 180
Abbildung 114 Neues Package erstellen 180
Abbildung 115 Neues Package "add" erstellen 181
Abbildung 116 Das neue Package "add" wird eingetragen 181
Abbildung 117 Einfügen eines neuen Fragmentes 182
Abbildung 118 Dialog zum Eintragen des Namens des neuen Fragments 182
Abbildung 119 Eintragen des Namens des neuen Fragments 183
Abbildung 120 ViewModel Auswahl für ein neues Fragment 183
Abbildung 121 Aktueller Stand des Bottom-Navigation-Beispiels (mit Add-Fragment) 187
Abbildung 122 Auswahl des Menueintrags „drawable Symbol“ 189
Abbildung 123 Auswahldes neuen Vektore-Symbols 190
Abbildung 124 Auswahl des Plus-Symbols 190
Abbildung 125 Framelayout umwandeln in ein Constraint-Layout 192
Abbildung 126 Popupmenu zum Umwandeln ins Constraint-Layout 192
Abbildung 127 Änderungsmeldung bezüglich des Constraint-Layout 192
Abbildung 128 actionBarSize 193
Abbildung 129 Bottom-Navigation ohne ActionBar-Padding 194
Abbildung 130 Struktur der fragment-add-xml-Datei 197
Abbildung 131 Anzeige der Meldung bei korrekter Lösung 201
Abbildung 132 Anzeige der Meldung bei falscher Lösung 201
Abbildung 133 Ansicht des Vorlagen-Projektes (leicht verändert) 202
Abbildung 134 Fertiges Projekt 202
Abbildung 135 Komplette Struktur der Musterlösung 203
Abbildung 136 Bottom Navigation Activity 204
Abbildung 137 Vorlagenstruktur des Tabbed Projektes 205
Abbildung 138 Einfügen eines neuen Fragments 207
Abbildung 139 Dialog zum Eintragen des Namens des neuen Fragments 208
Abbildung 140 Eintragen des Namens des neuen Fragments 208
Abbildung 141 ViewModel Auswahl für ein neues Fragment 208
Abbildung 142 Aktueller Stand des TabbedBar-Beispiels (mit Add-Fragment) 216
Abbildung 143 Anzeige der Meldung bei korrekter Lösung 229
Abbildung 144 Anzeige der Meldung bei falscher Lösung 229
Abbildung 145 Anlegen des Ordners navigation im Eintrag "res" 235
Abbildung 146 Anlegen eines Ordners "menu" 237
Abbildung 147 Erstellen der xml-Datei "menu.xml" 237
Abbildung 148 AddSubMult in Optionsmenu 255
Abbildung 149 AddSubMult in Optionsmenu 255
Abbildung 150 Auswahl einer Tabbed Activity 256
Abbildung 151 Name der Tabbed Activity: SwipeTabbedActivity 256
Abbildung 152 Anlegen eines Pictures-Fragment 258
Abbildung 153 Fragment mit Factory 258
Abbildung 154 Anzeige der Swipe-Musterlösung 266
Abbildung 155 Startbildschirm 268
Abbildung 156 Vorspeisen 268
Abbildung 157 Hauptspeise 268
Abbildung 158 Nachspeise 268

Abbildung 159 Projekttyp "Empty Activity" 269
Abbildung 160 Projektname BspMyNavigation 269
Abbildung 161 Anzeige der drei neuen Packages 273
Abbildung 162 Neues Package 274
Abbildung 163 Eingabe des Namens des neuen Package 274
Abbildung 164 Eingabe des Namens "vorspeisen" 274
Abbildung 165 Eintragen der Bottom-Navigation 275
Abbildung 166 Einfügen der Activity VorspeisenActivity 275
Abbildung 167 Eintragen der Tabbed Activity 277
Abbildung 168 Einfügen der Activity VorspeisenActivity 277
Abbildung 169 PizzaFragment erstellen 278
Abbildung 170 ZucchiniFragment erstellen 279
Abbildung 171 Eintragen der Empty Activity 281
Abbildung 172 Einfügen der Activity NachspeisenActivity 281
Abbildung 173 PizzaFragment erstellen 282
Abbildung 174 Erstellen einer neuen Activity 285
Abbildung 175 Eingabe des Namens für die neue Activity 286
Abbildung 176 Anlegen eines Ordners "menu" 288
Abbildung 177 Erstellen der xml-Datei "menu.xml" 288
Abbildung 178 Beispiel für Symbole des Optionsmenus 289
Abbildung 179 Neues Symbol im Optionsmenu 290
Abbildung 180 Anzeige der Einträge im Optionsmenu (mit Symbolen) 290
Abbildung 181 Übersicht über SQLIte in Android 303
Abbildung 182 Löschen der Daten eines Simulators 306
Abbildung 183 Struktur des ersten SQLite-Beispiels 307
Abbildung 184 Musterlösung des ersten SQLIte-Beispiels 307
Abbildung 185 row_layout für die Datenbank-Anzeige 319
Abbildung 186 LiustView mit einem SimpleCursorAdapter 321
Abbildung 187 row_layout für die Datenbank-Anzeige 322
Abbildung 188 Start des Projektes "Alle Sensoren" 349
Abbildung 189 Sonsor Magnetfeld + Accelerator 349
Abbildung 190 Musterlösung einer Stoppuhr und eines Counddown-Zählers 359
Abbildung 191 Layout des Chronometers 360
Abbildung 192 Auswahl des Menueintrags „drawable Symbol“ 371
Abbildung 193 Auswahldes neuen Vektore-Symbols 372
Abbildung 194 Auswahl des Plus-Symbols 372

2 Android

Dieses Kaptels zeigt die beiden Installationsprozeduren und die Prozedure des „Nachladens“. Android wird während des Arbeitens immer mal wieder neue Dateien laden.

Hinweis:

- **Der Rechner sollte ein 64-Bit-System sein und mindestens 8 GB Hauptspeicher haben.**

2.1 Installation

Die Installation verläuft in **fünf** Schritten:

- Download der Exe-Datei bzw. dgm-Datei
- Erste Installation
- Aufruf und laden weiterer Dateien
- Runterladen des SDK
- Runterladen der virtuellen Maschine

2.1.1 Erste Installation



Abbildung 1 Begrüßungsdialog

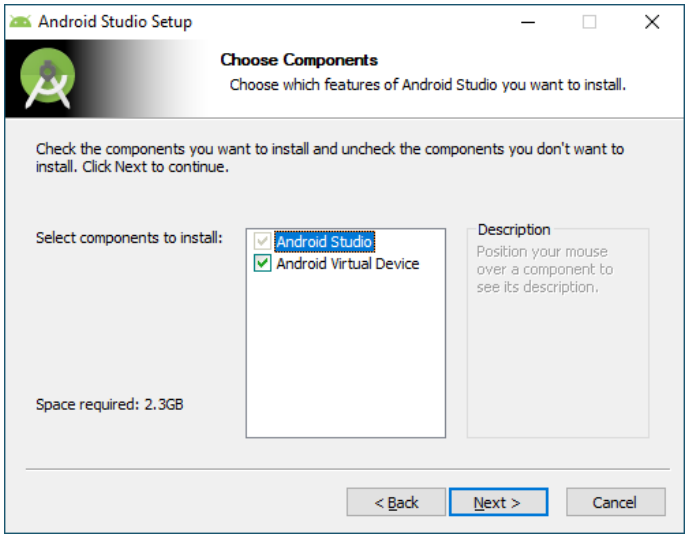


Abbildung 2 Bitte mit der Virtualisierung

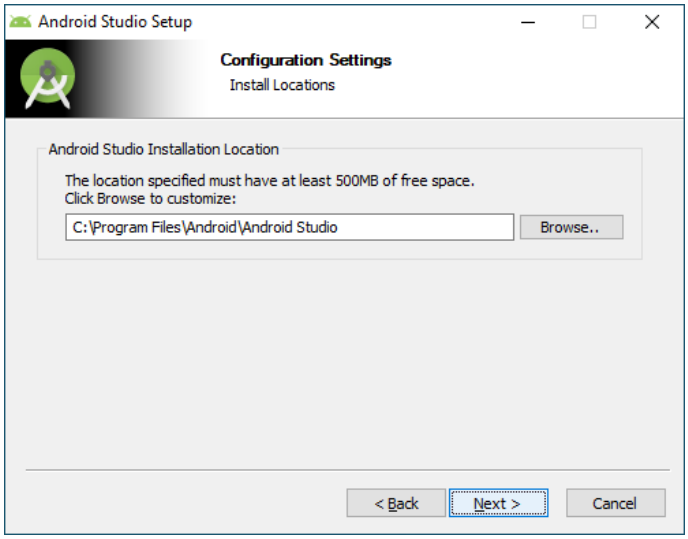


Abbildung 3 Installationsordner (Meine Wahl: c:\Android Studio)

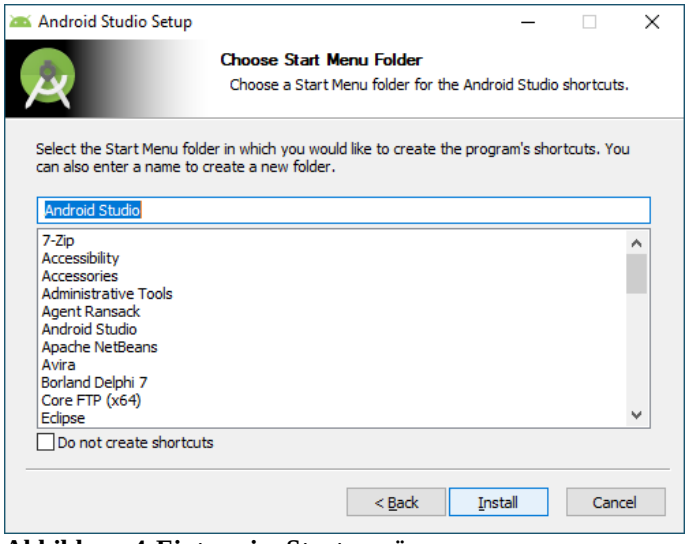


Abbildung 4 Eintrag im Startmenü

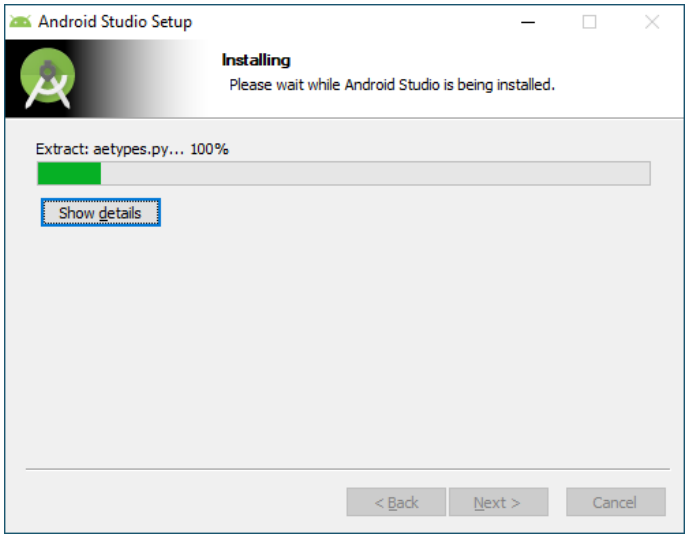


Abbildung 5 Entzippen des Installationsprogramms

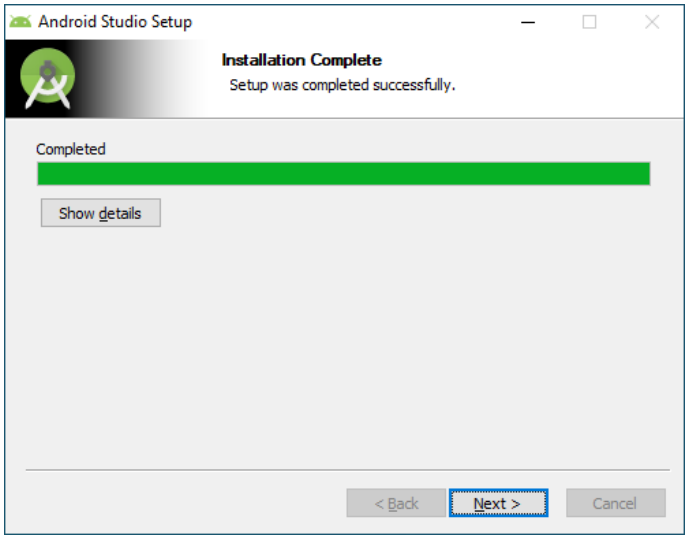


Abbildung 6 Die Installation wurde entzippt

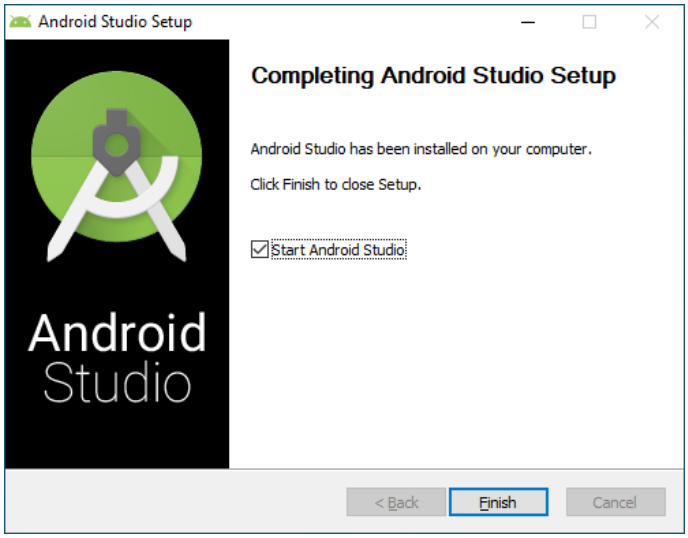


Abbildung 7 Der erste Teile, der schnellere, ist beendet.

2.1.2 Zweite Installation

Nun starten des Android Studios:

Im zweiten Teile werden die SDK, Emulatoren etc. geladen.

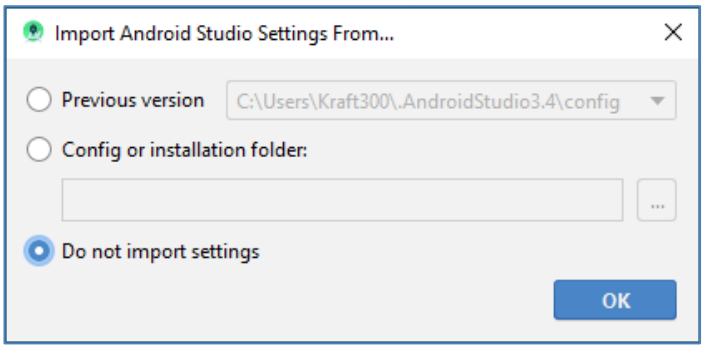


Abbildung 8 Import von vorherigen Versionen)optional)



Abbildung 9 Download des SDK

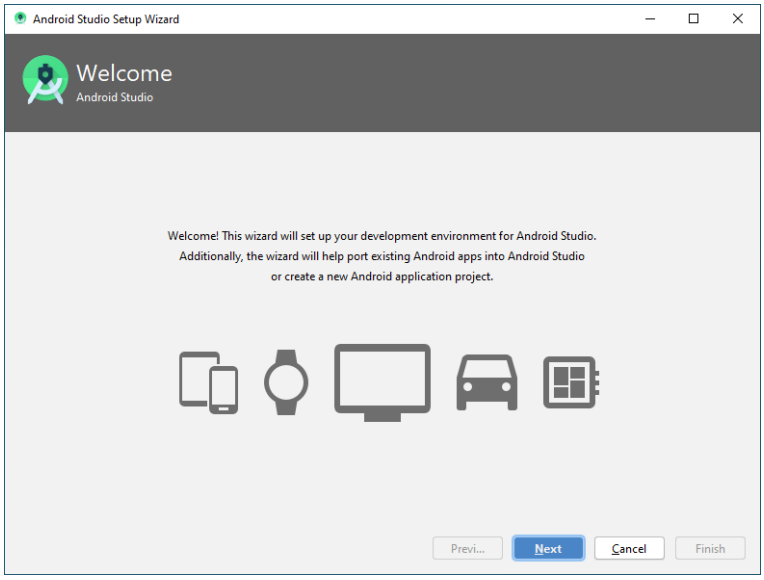


Abbildung 10 Warten vor denm langen Download

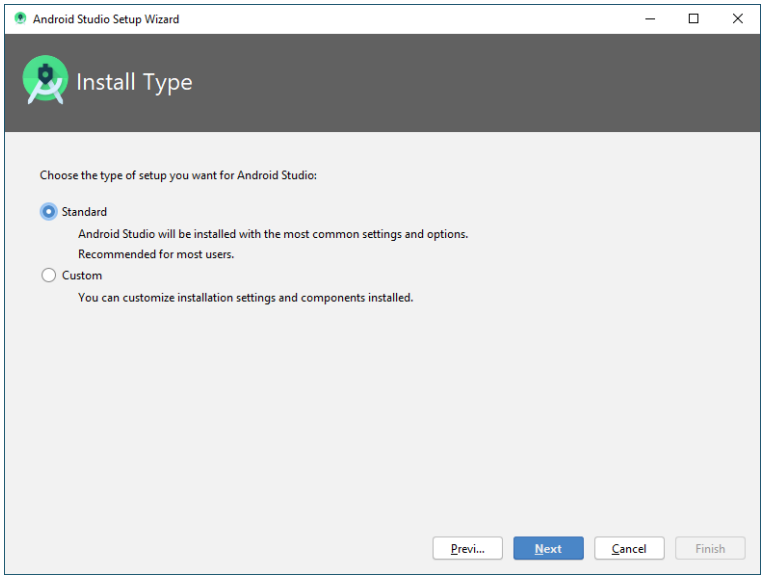


Abbildung 11 **Hier Custom auswählen**

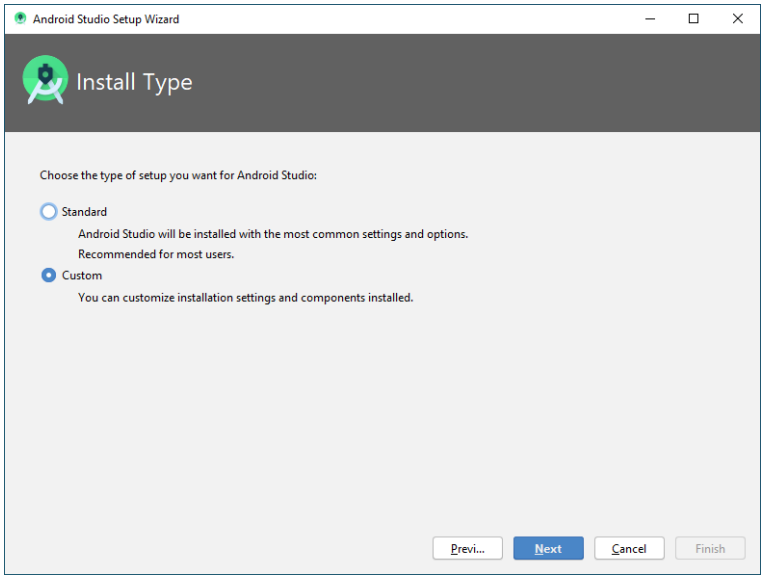


Abbildung 12 **Bessere Wahl**

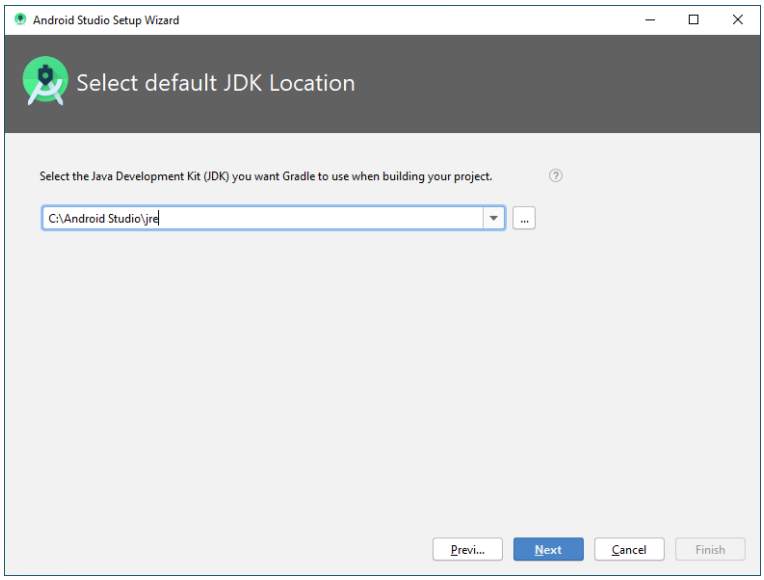


Abbildung 13 **Java Runtime Edition Ordner (ja nach Pfad anders)**

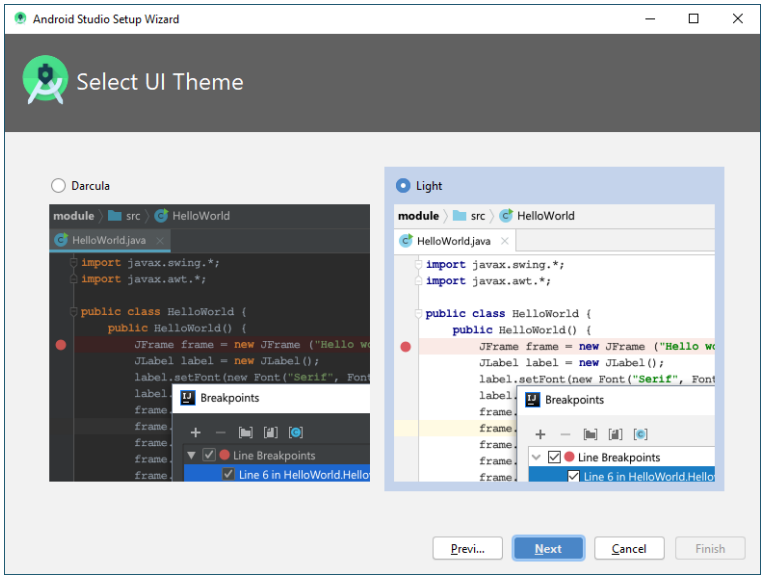


Abbildung 14 Farbwahl

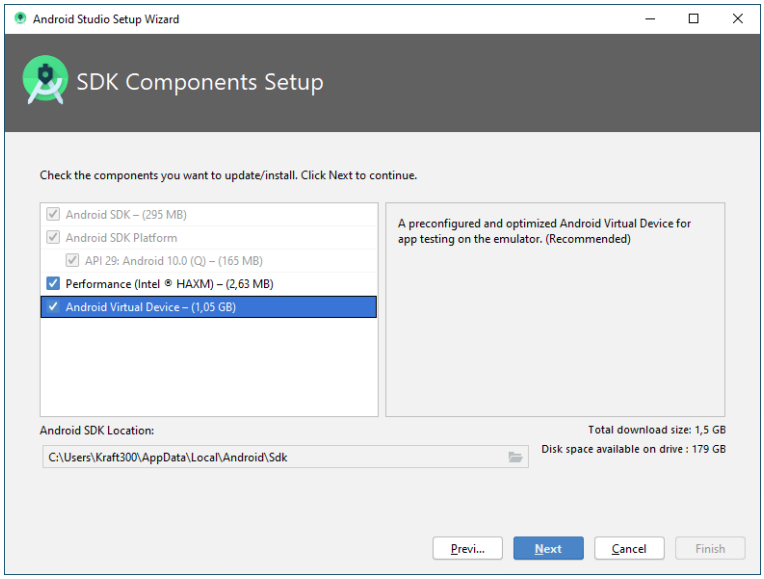


Abbildung 15 Bitte unbedingt beide Haken setzen

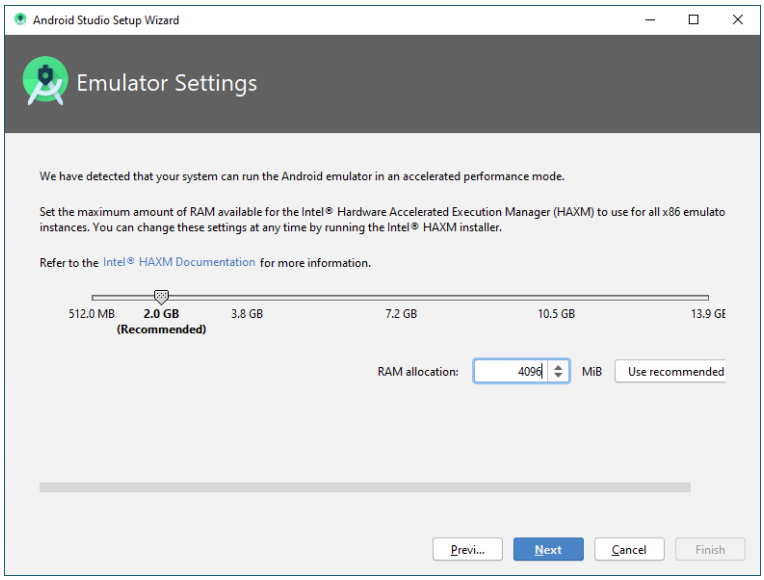


Abbildung 16 Wer nicht mindestens 16 Hauptspeicher hat, unbedingt nichts ändern

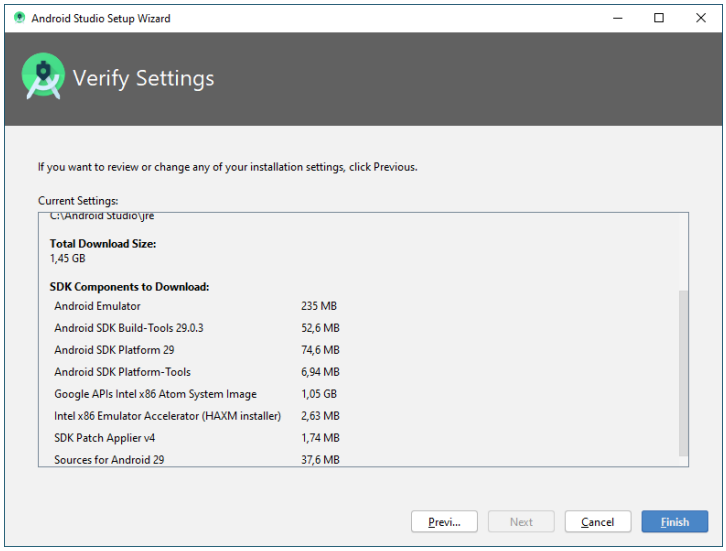


Abbildung 17 Liste der Downloads

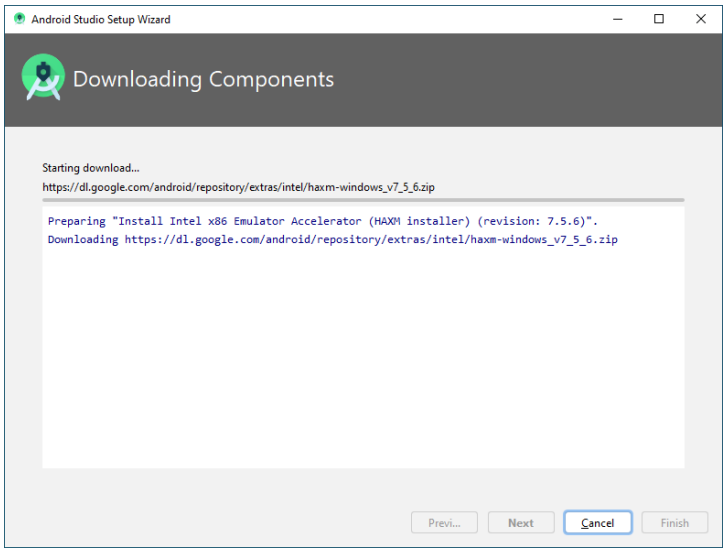


Abbildung 18 Jetzt geht der zweite Teil richtig los!

Dauer der letzten Teile der Installation: ca. 30 bis 60 Minuten

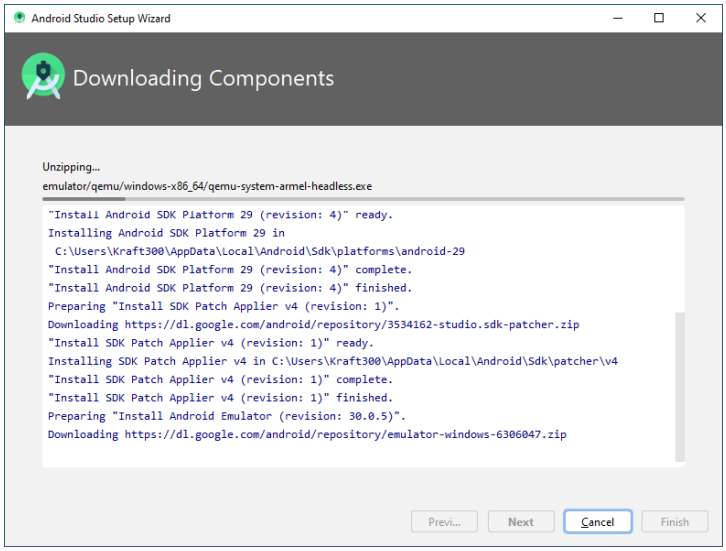


Abbildung 19 Laden der Zusatzdateien

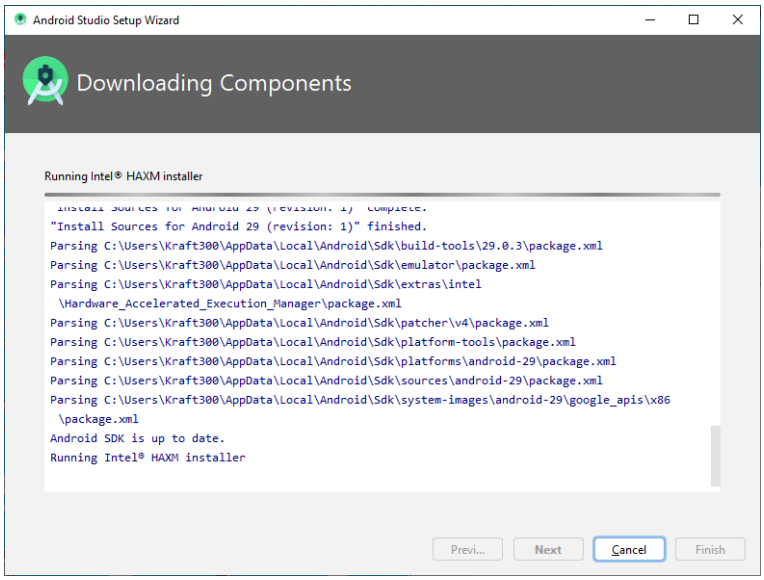


Abbildung 20 Immer noch runterladen

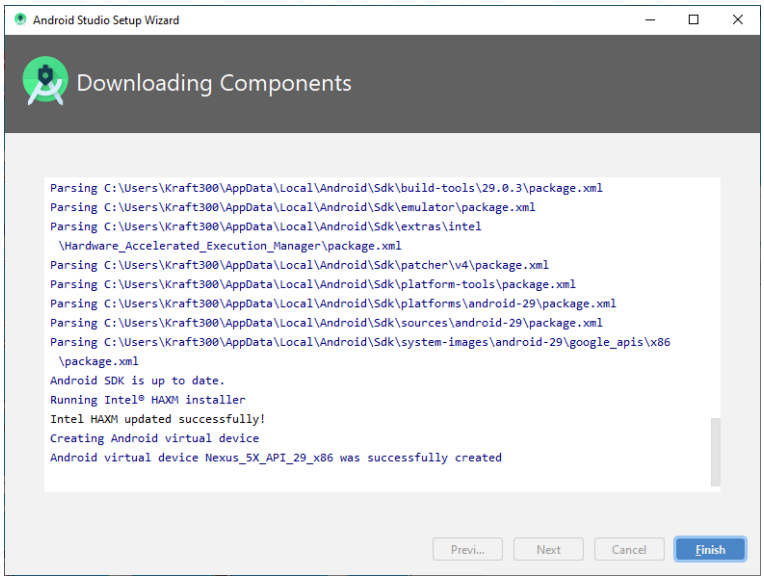


Abbildung 21 Immer noch runterladen

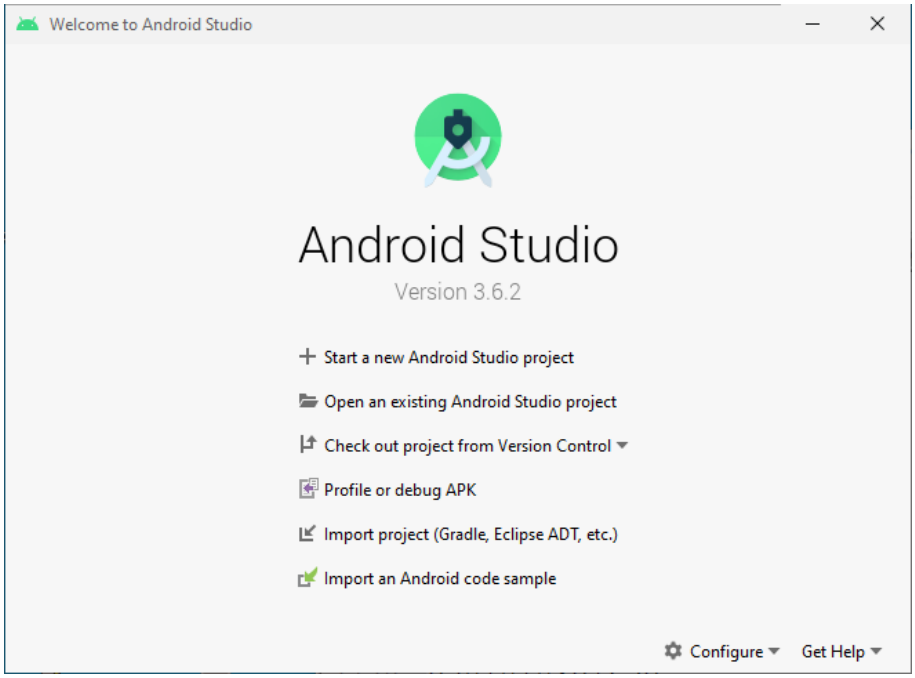


Abbildung 22 Startdialog

Nun ein neues Projekt anlegen. Danach muss das SDK und mindestens eine virtuelle Maschine angelegt werden.

2.2 Erstellen eines neuen Projektes

a) Aufruf des Androis Studios

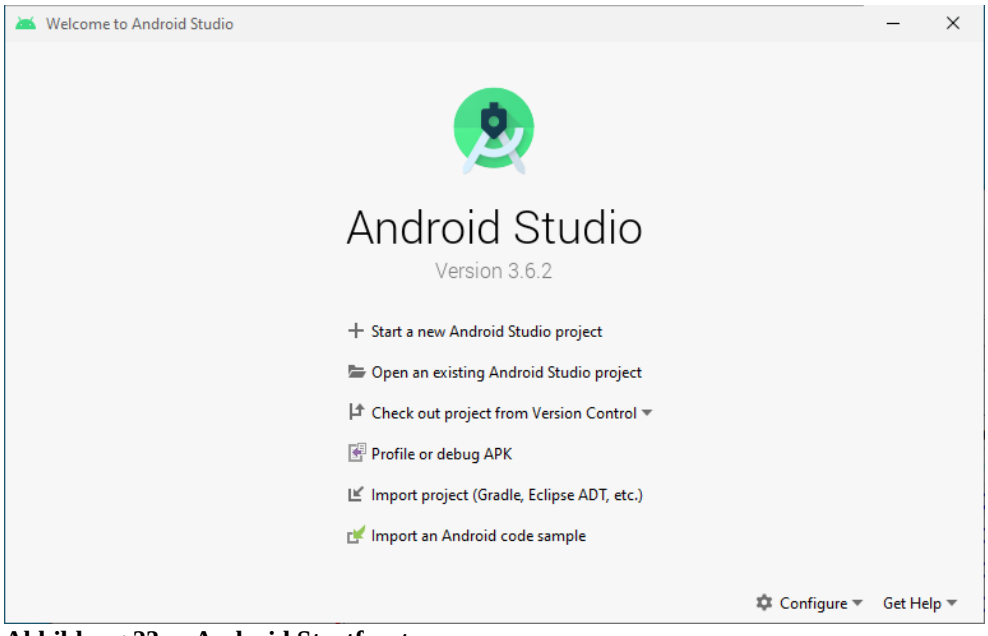


Abbildung 23 Android Startfenster

b) Schalter „Start anew Android Studio project“ anklicken

c) Auswahl des Projektes

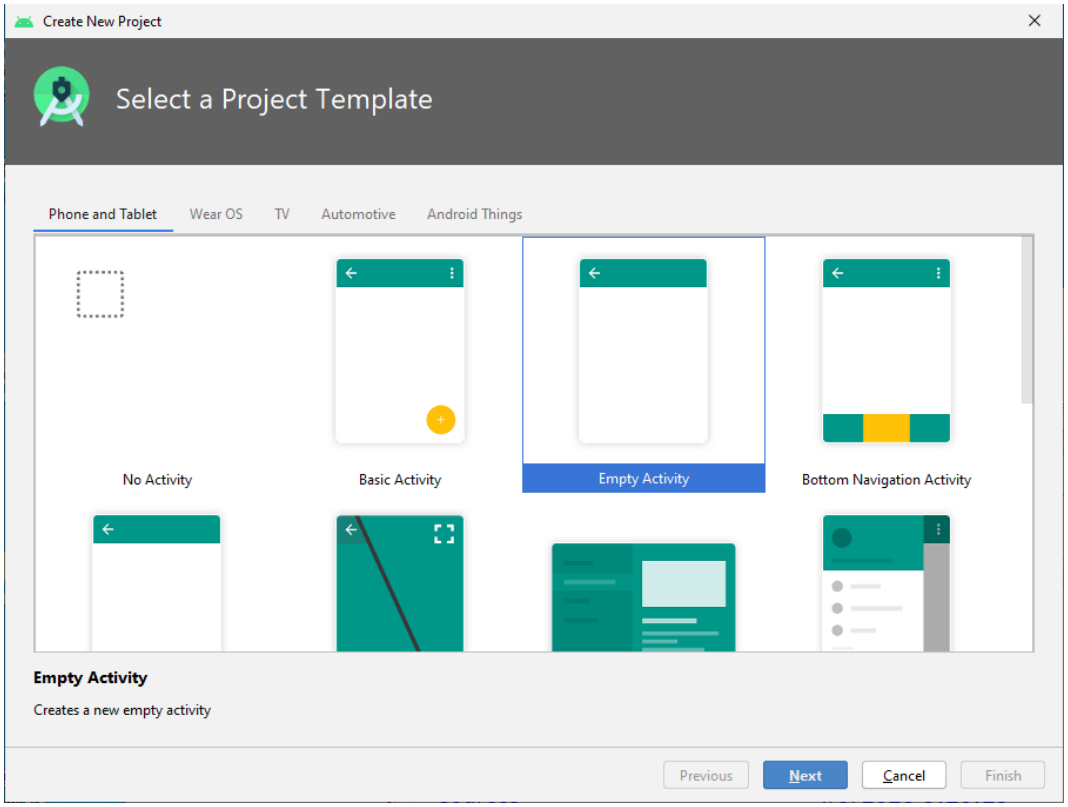


Abbildung 24 Auswahl "Empty Activity"

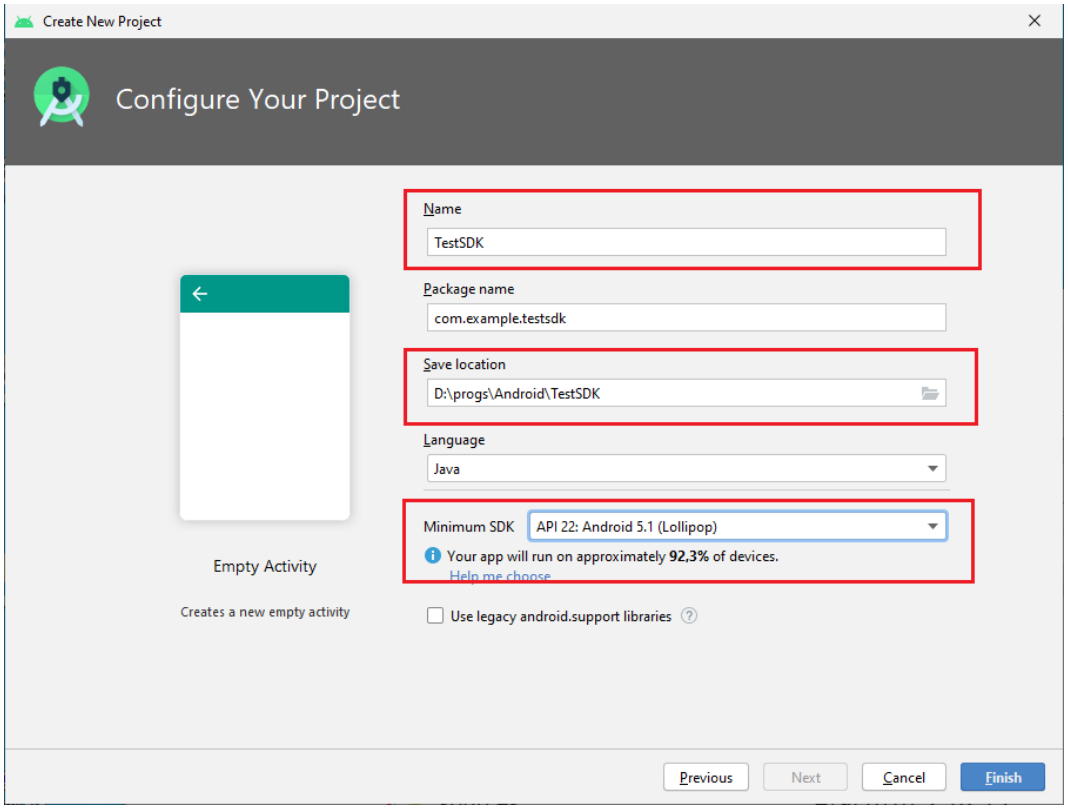


Abbildung 25 Eigenschaften des neuen Projektes

Einträge:

- Name TestSDK
- Save location D:\progs\Android\TestSDK
  - **Bitte erstellen einen Ordner für Ihre Projekte**
- Language Java
- Minimum SDK API 22 Android 5.1

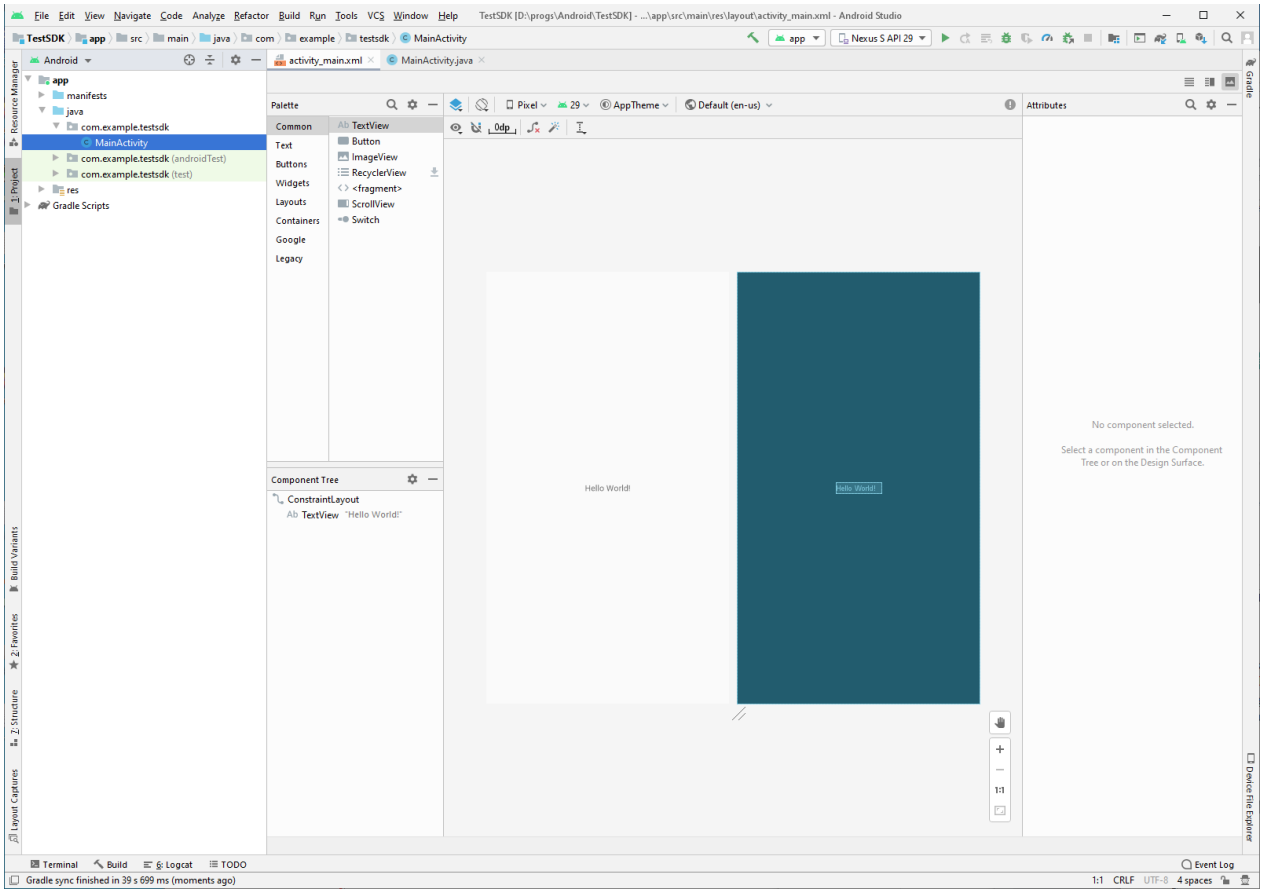


Abbildung 26 Neues Projekt im Android Studio

2.3 SDK-Installation

Zwei Varianten:

- SDK durch das Android-Studio runterladen
- SDK durch einen Link runterladen

2.3.1 •SDK durch das Android-Studio runterladen

Menü Tools
Eintrag SDK Manager

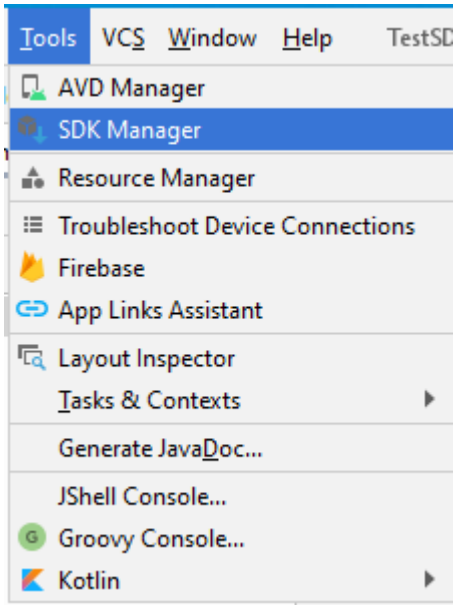


Abbildung 27 Aufruf SDK Manager

Nun Anklicken des Android SDK 10.0, so dass die Teilkomponenten sichtbar werden. Das komplette SDK ist mit dem Quellcode zu groß.

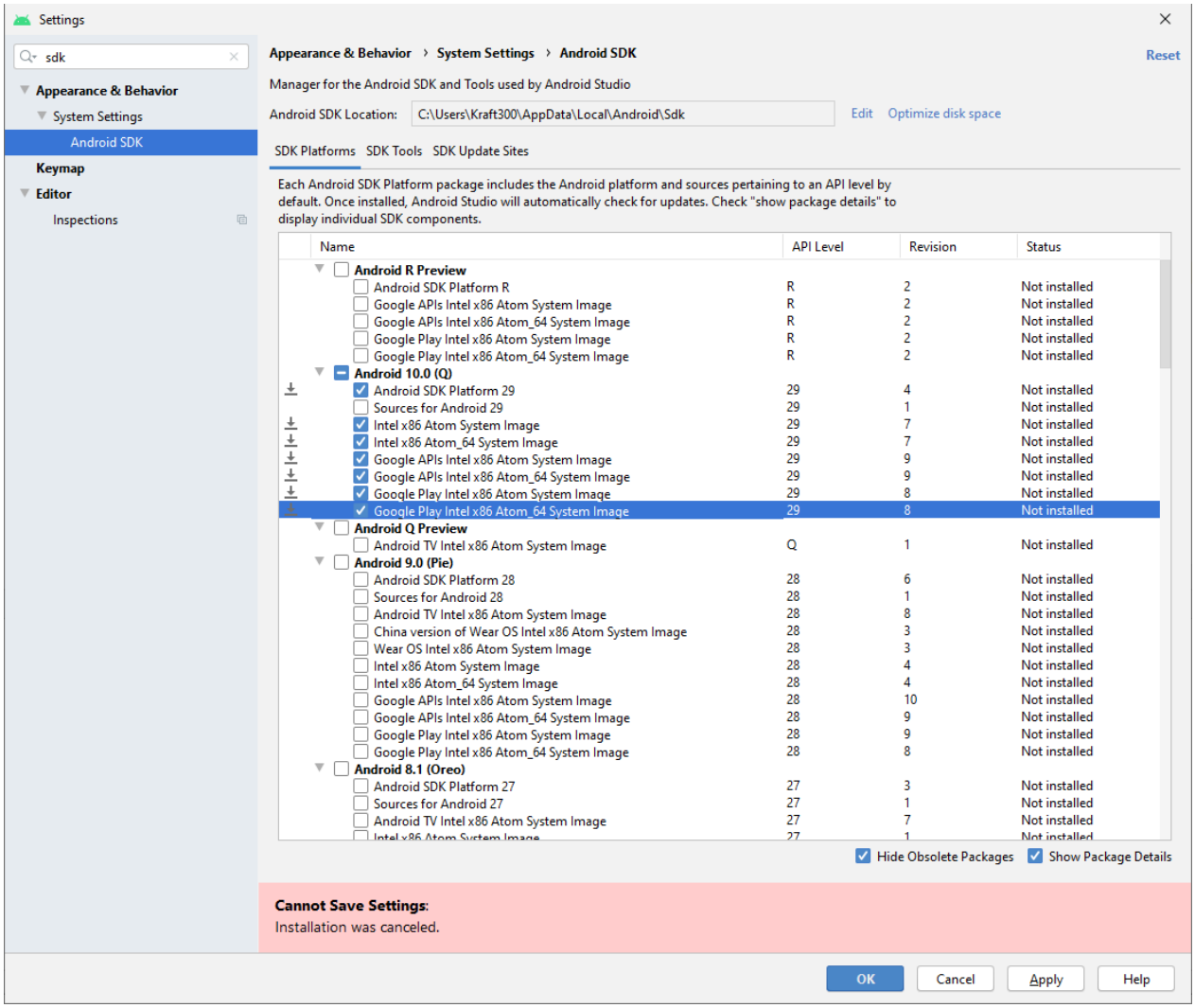


Abbildung 28 SDK erstellen (Auswahl der Komponenten)

Nun den Schalter „Edit“ für den Ordner des SDK betätigen.

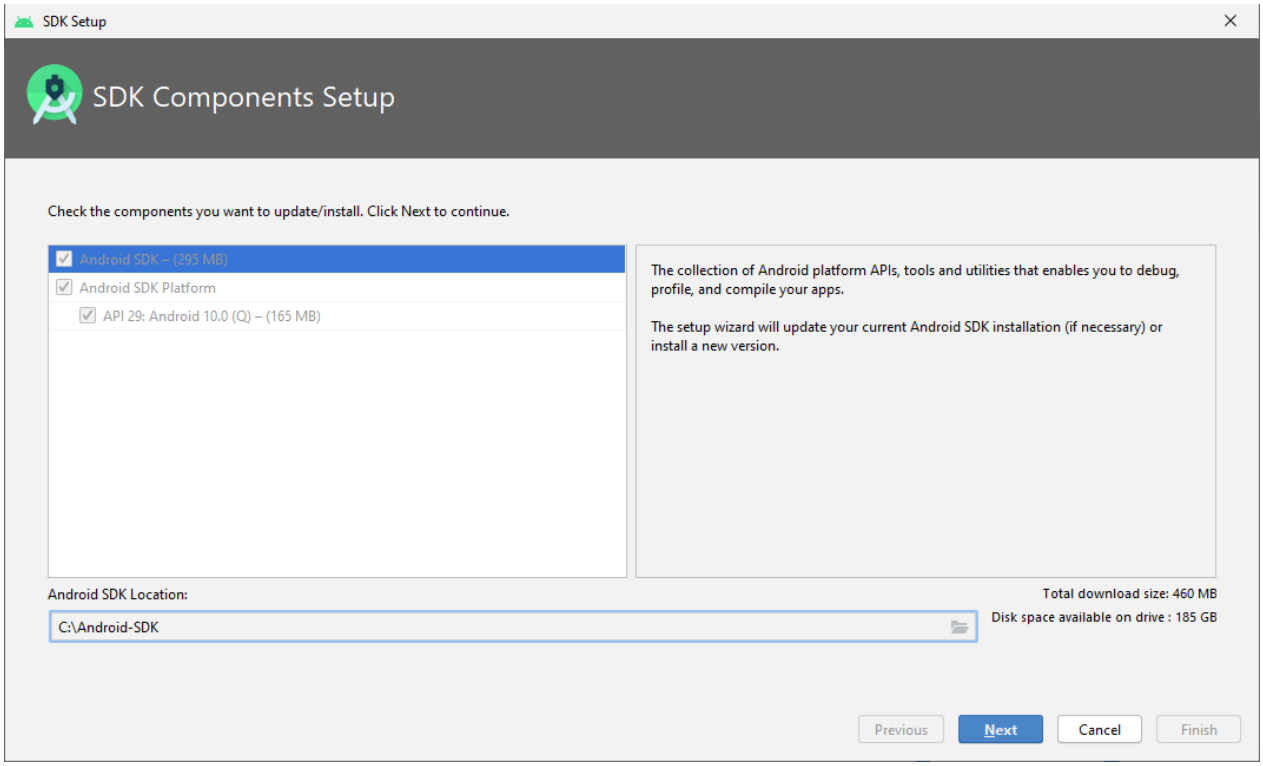


Abbildung 29 SDK Verzeichnis erstellen

Mit dem Schalter „Next“ wird das SDK runtergeladen. Das dauert ca. 30 Minuten.

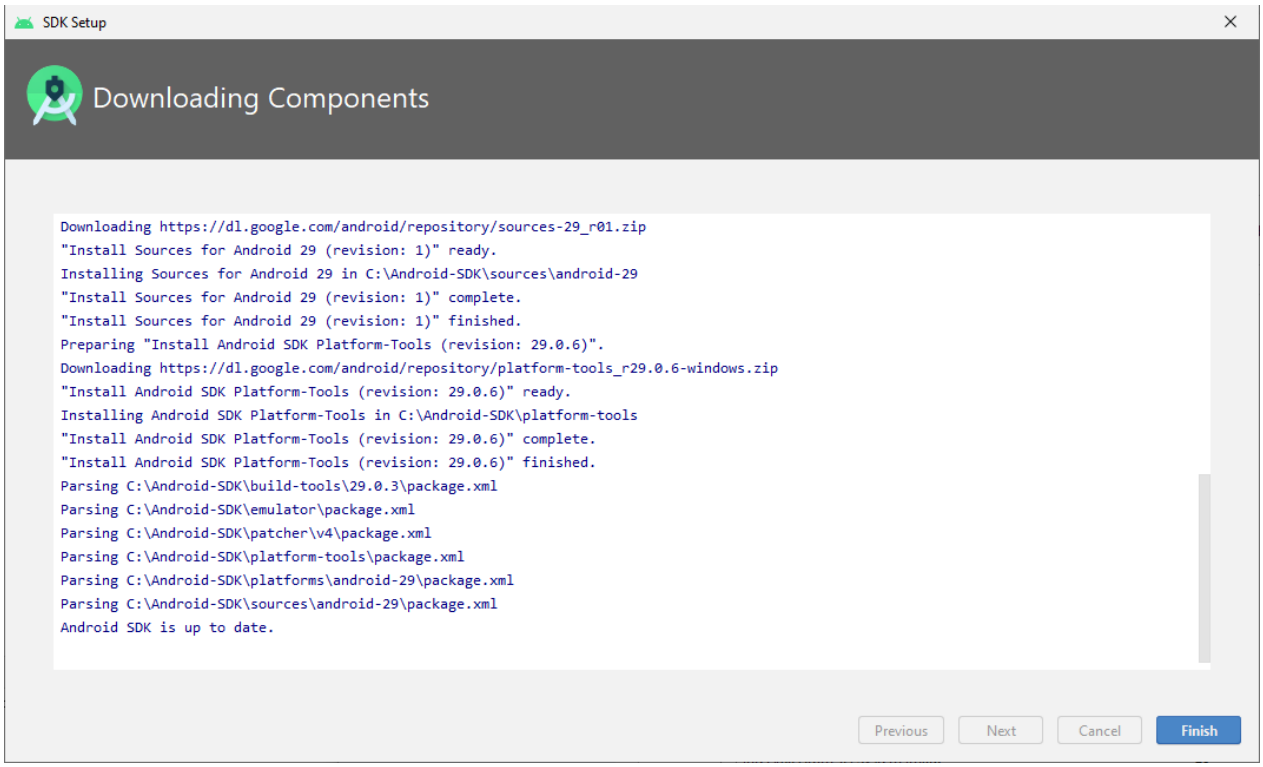


Abbildung 30 Runterladen von Komponenten

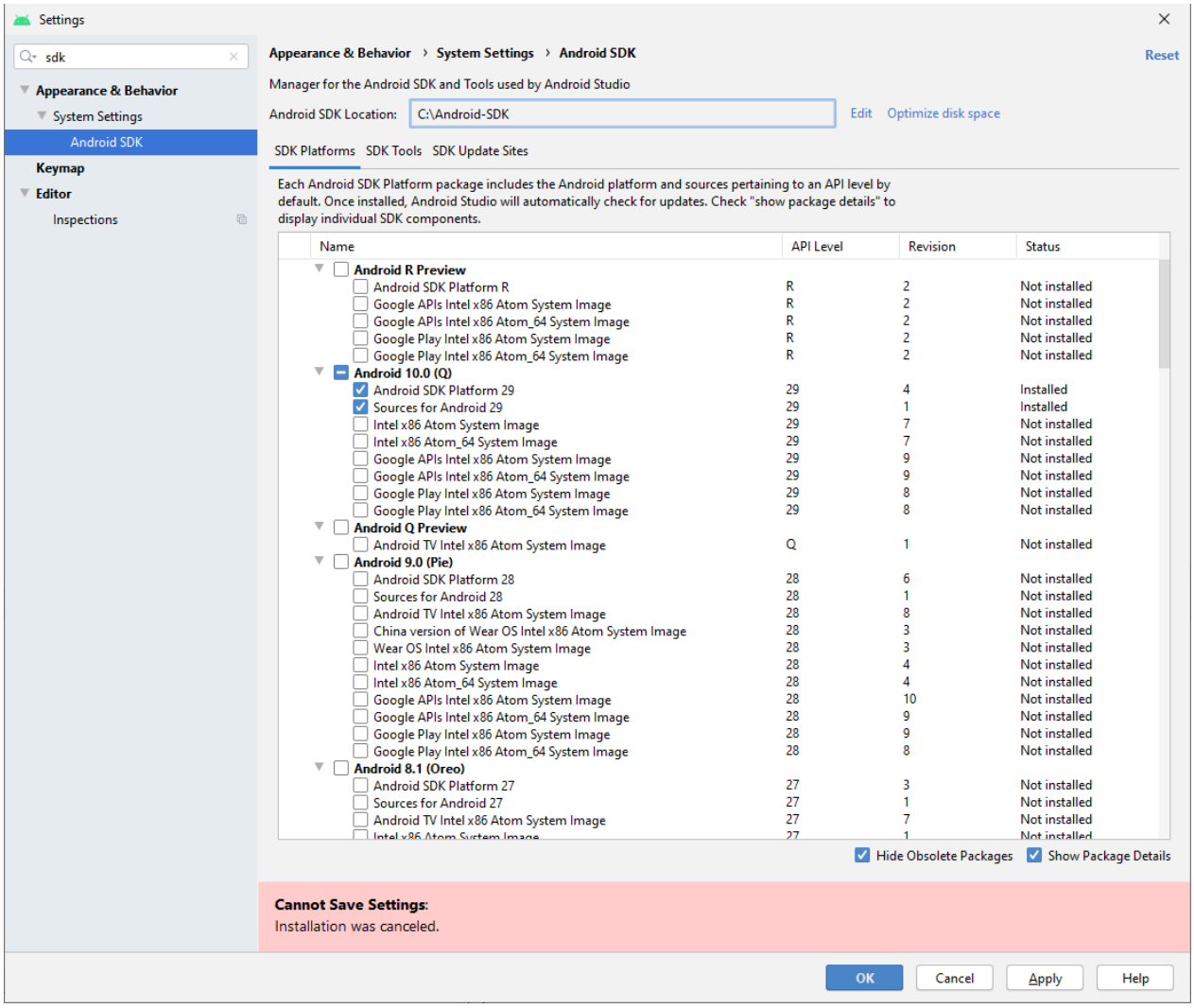


Abbildung 31 SDK ist nun installiert

Mit dem Schalter „ok“ den Dialog beenden.

2.3.2 •SDK durch einen Link runterladen

Link: https://www.miwilhelm.de/scripte/android/Android-SDK.zip

Menü Tools
Eintrag SDK Manager

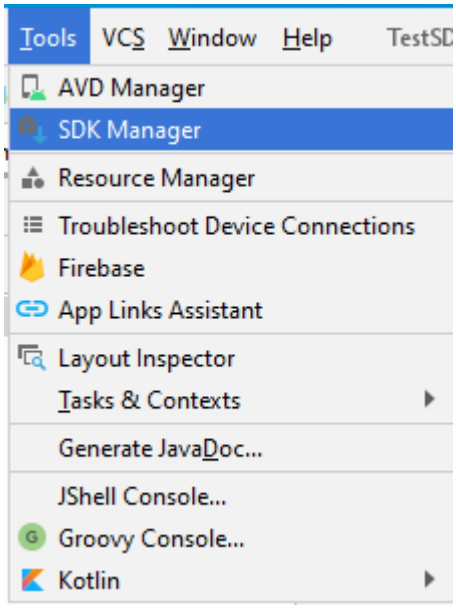


Abbildung 32 Aufruf SDK Manager

Nun den Schalter „edit“ anklicken und den ordner auswählen.

2.4 Virtuelle Maschine (AVD-Manager)

Menü Tools,
Eintrag AVD-Manager

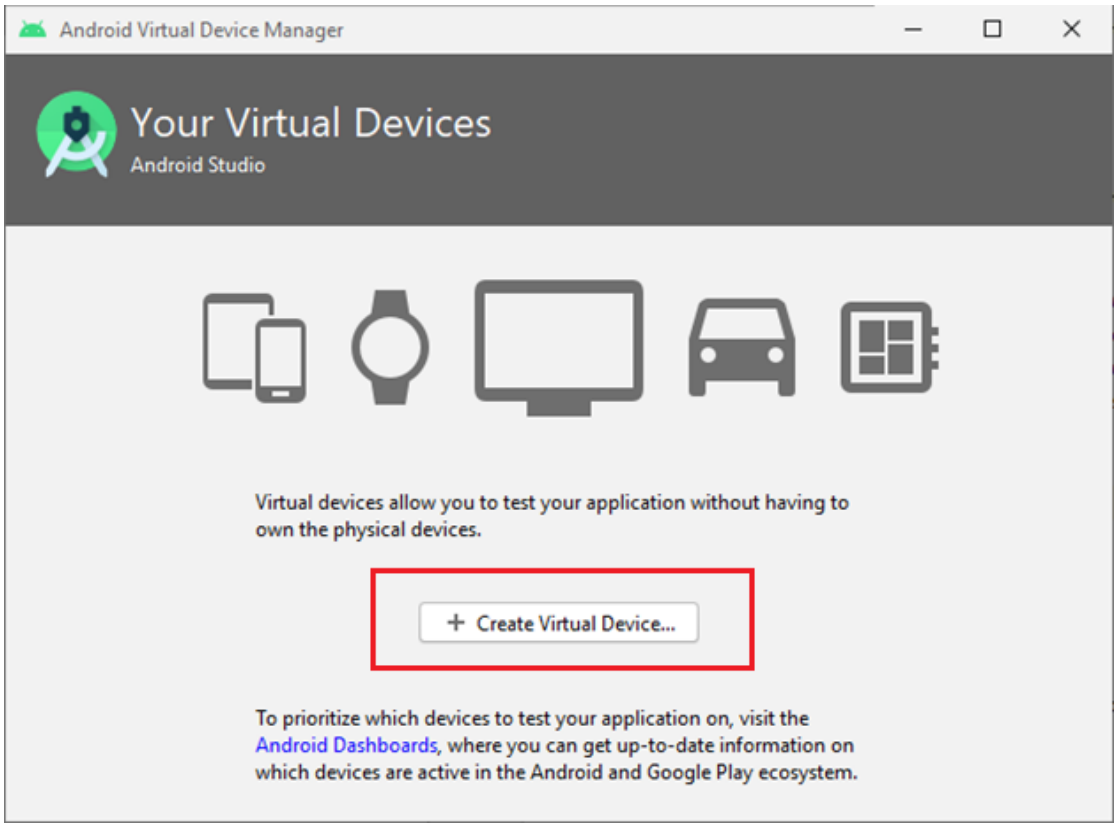


Abbildung 33 Erstellen einer virtuellen Maschine

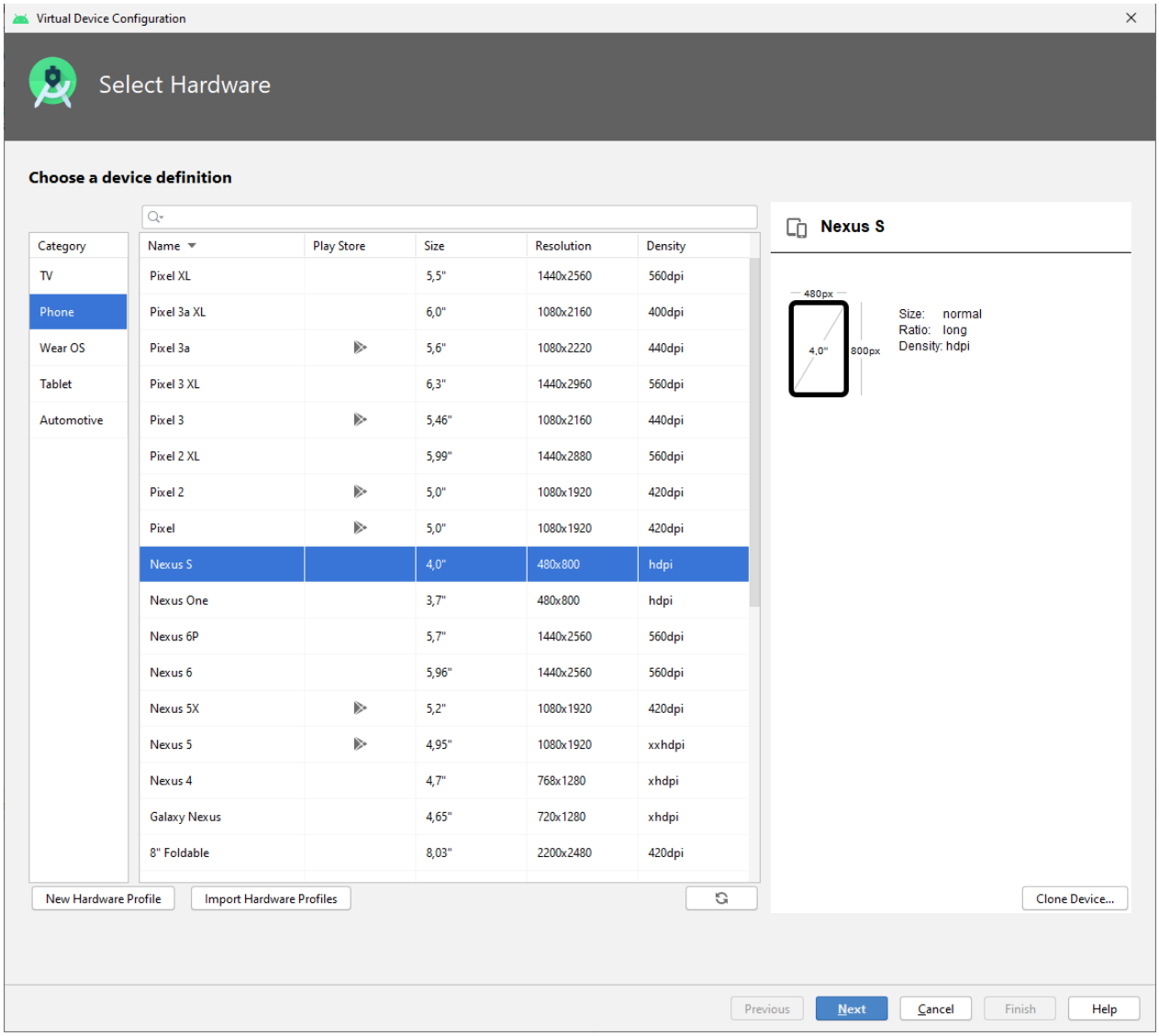


Abbildung 34 Auswahl des Smartphones

Bitte wählen Sie nicht ein Smartphone aus, welches eine zu hohe Auflösung hat. Die Anzeige des „Smartphone“ muss ja auch auf Ihrem Bildschirm passen.

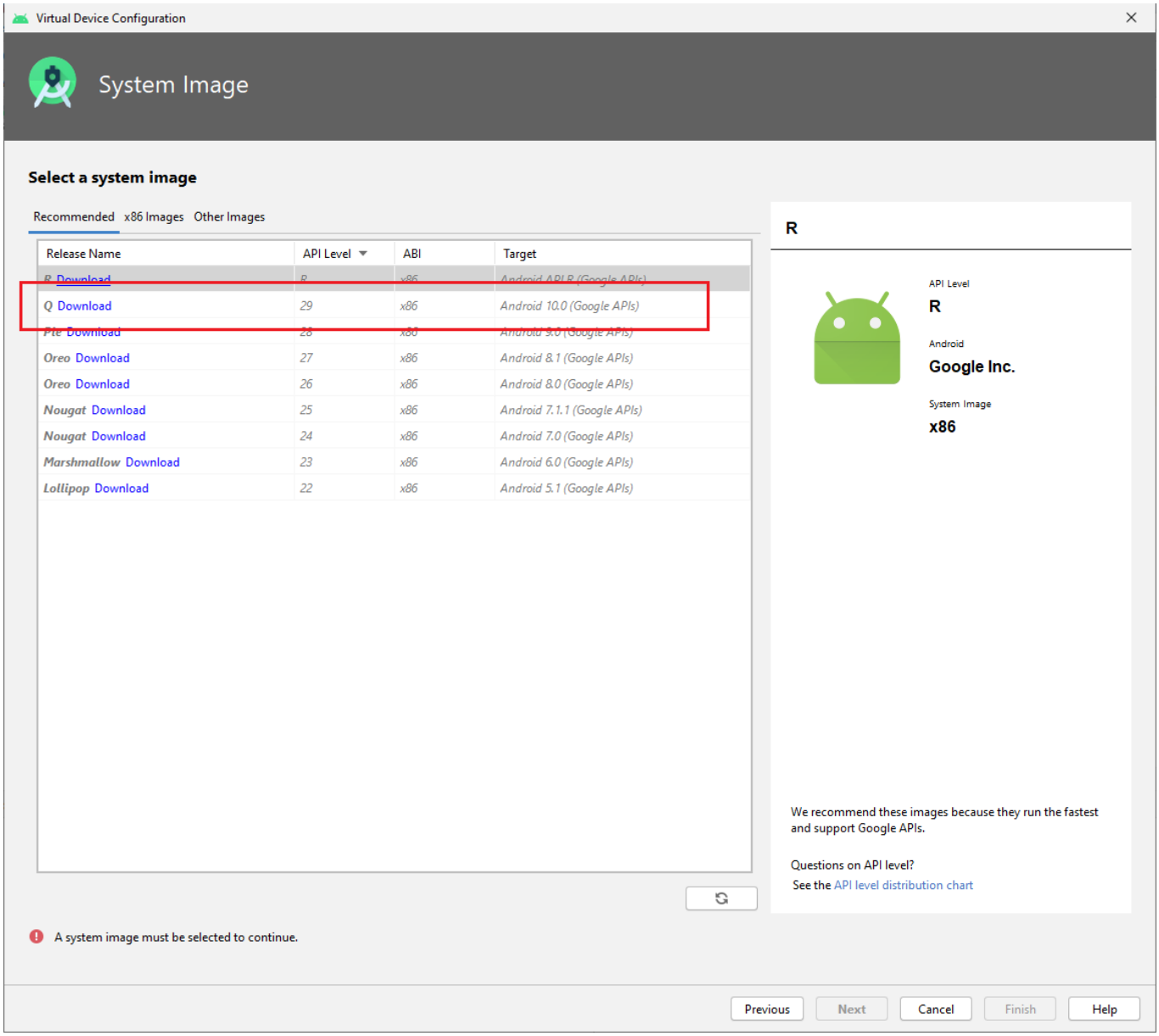


Abbildung 35 Nun ein weiterer Download

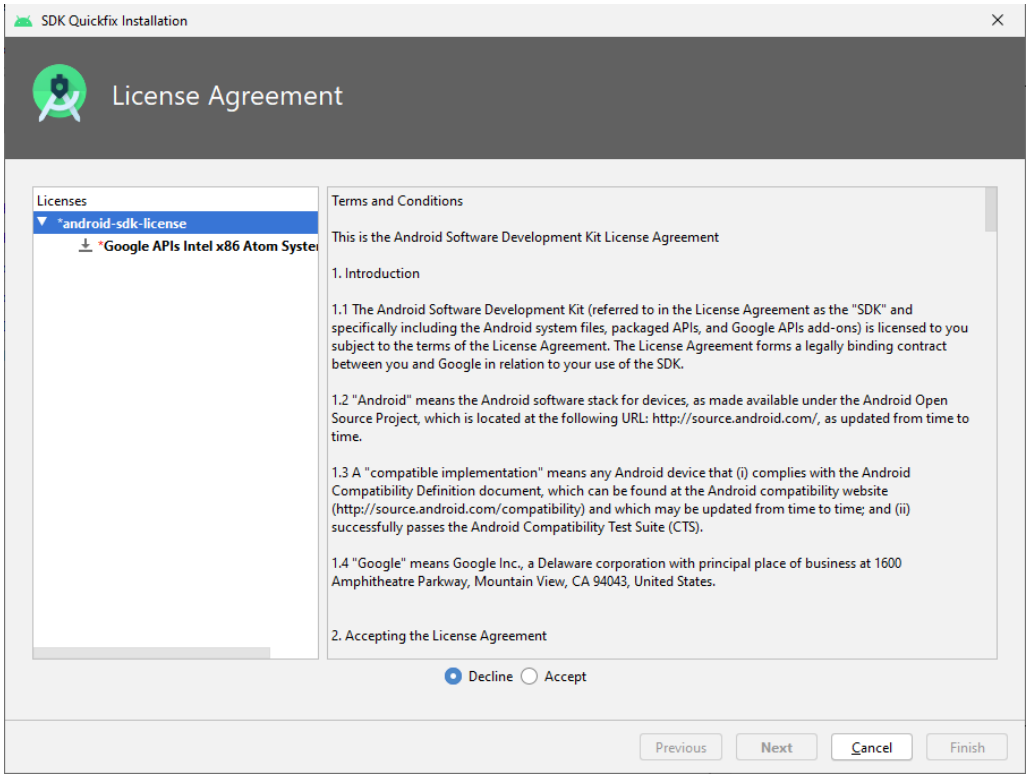


Abbildung 36 Lizenz Bedingungen

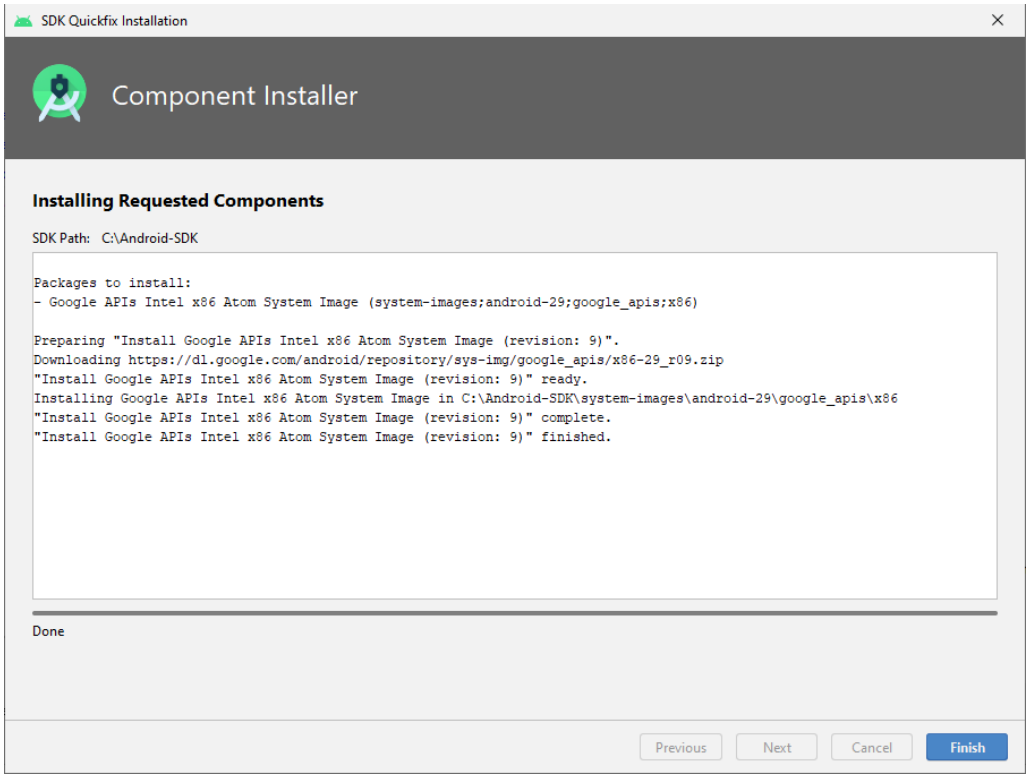


Abbildung 37 Endlich fertig

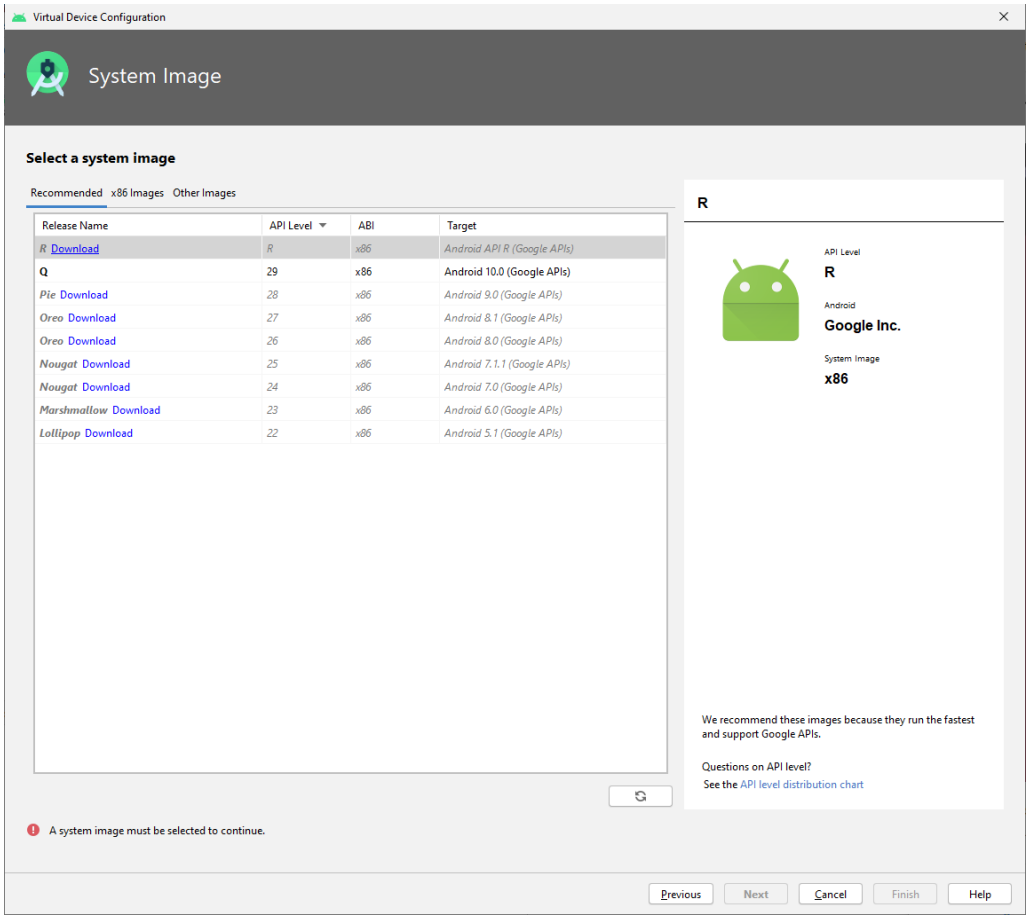


Abbildung 38 Anzeige aller virtuellen Maschinen

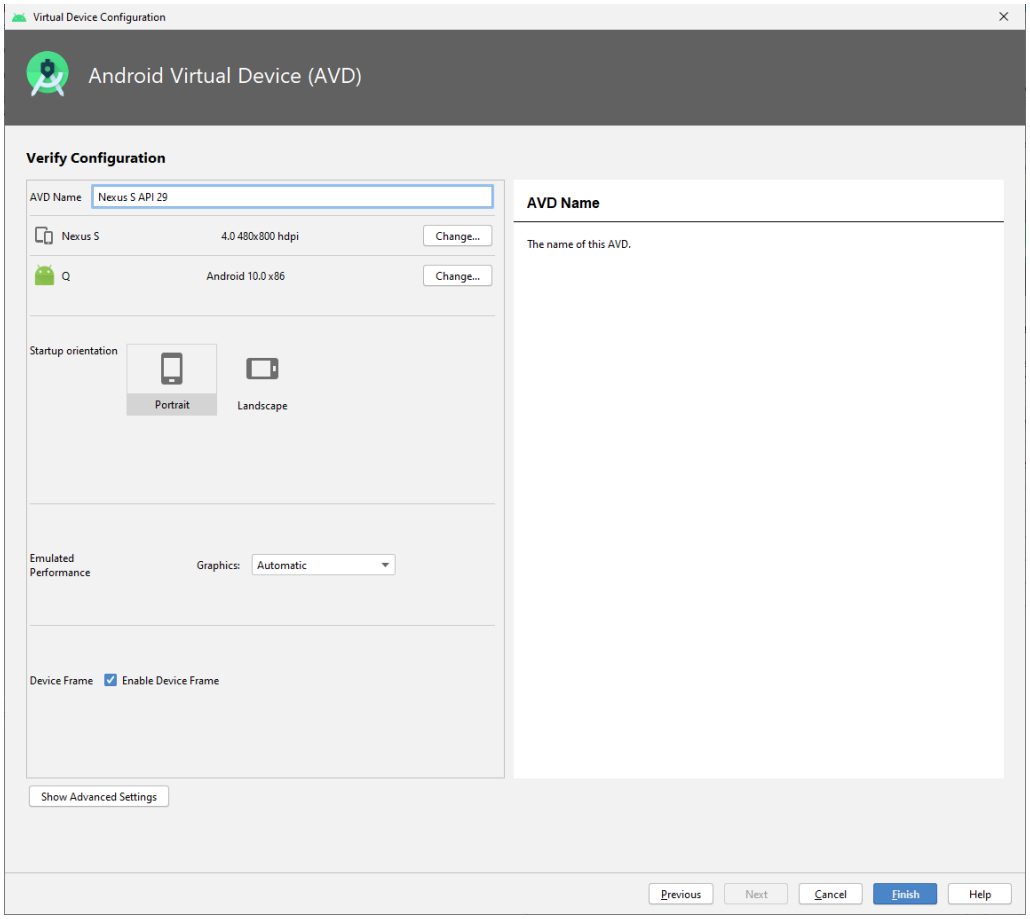


Abbildung 39 Eigenschaften der virtuellen Madschine

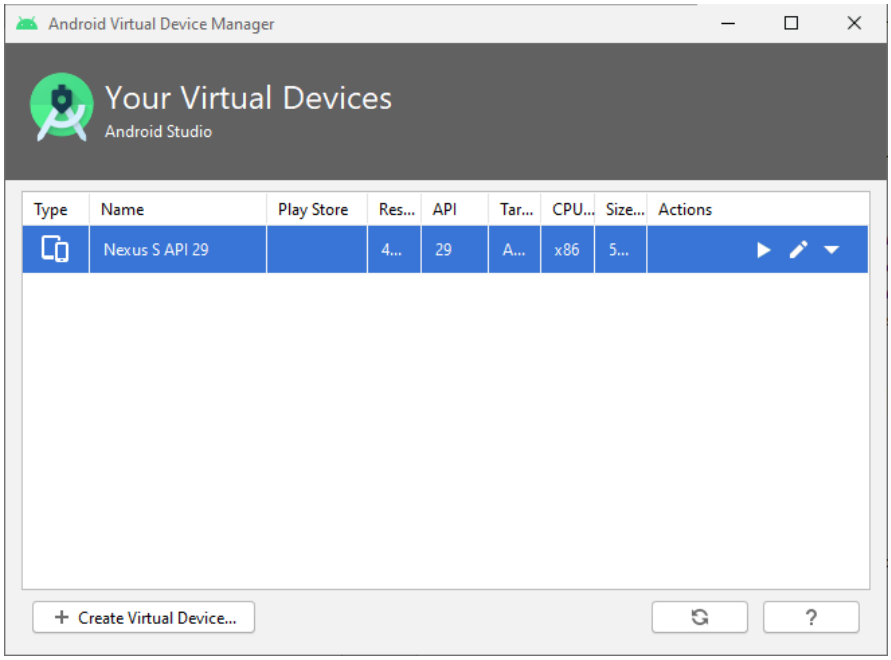
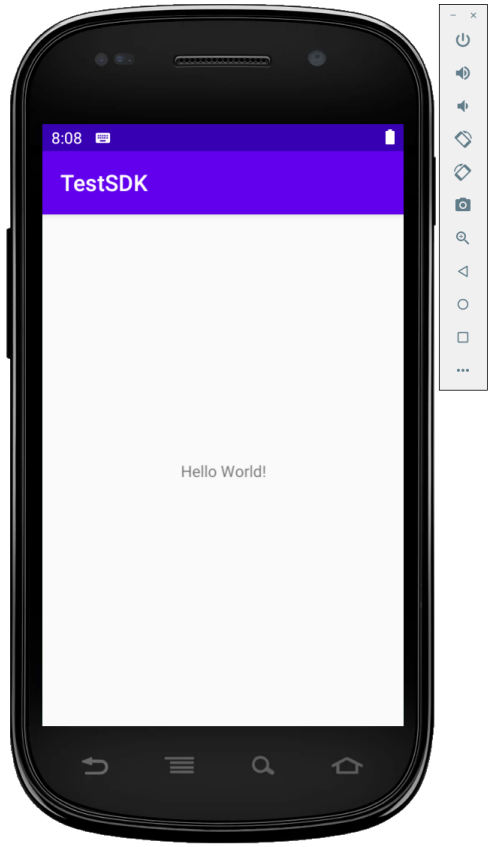


Abbildung 40 Abschlussdialog

Nun starten des Projektes mit

- Menu Run, Eintrag Run 'App'
- oder
- Shift F10

Ergebnis:



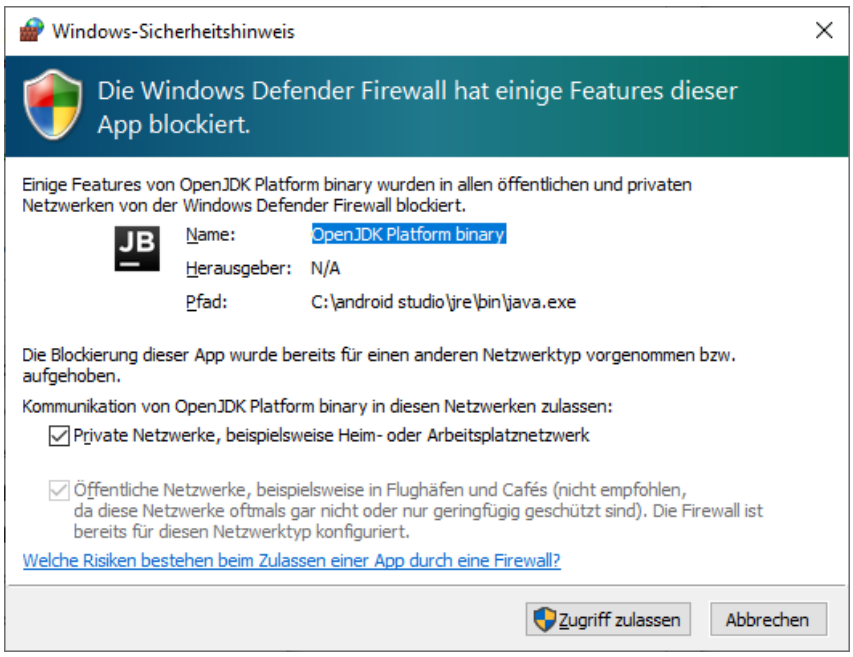


Abbildung 41 Die eventuelle Sicherheitsabfrage bejahen

3 Erste einfache Beispiele

3.1 Wichtige Grundeigenschaften von UI-Elementen

Grundeigenschaften

- **id**
  - Die id wird zur Verknüpfung mit dem Java-Code und in den XML-Dateien benötigt.
  - Beispiel:
    - @+id/textview1
    - @+id/nachname
    - @+id/bnOkay
- **layout_width**
  - Die Breite kann entweder einen festen Wert haben, wrap_content oder match_parent beinhalten.
  - Fester Wert: 120sp, 12dp, 12px
  - Breite über eine id: @+id/tinput
  - wrap_content zeigt an, dass die Breite so gesetzt wird, dass der komplette Inhalt zu sehen ist.
  - match_parent zeigt an, dass nur die Breite komplett über die „Zeile“ geht (abzüglich der beiden Ränder). Die Breite richtet sich auch nach der Schriftgröße.
- layout_height match_parent / wrap_content / @id/
  - Die Höhe kann entweder einen festen Wert haben, wrap_content oder match_parent beinhalten.
  - Fester Wert: 24sp
  - Höhe über eineid: @+id/tinput
  - wrap_content zeigt an, dass die Höhe so gesetzt wird, dass der komplette Inhalt zu sehen ist.
  - match_parent zeigt an, dass nur die Höhe komplett über die „Zeile“ geht (abzüglich der beiden Ränder). Die Höhe richtet sich auch nach der Schriftgröße.
- layout_marginStart
  - Linker Rand
- layout_marginEnd
  - Rechter Rand
- layout_marginTop
  - Oberer Rand
- layout_marginBottom
  - Unterer Rand (sinvoll nur wenn die Höhe „match_parent“ ist).
- gravity

TextView (zusätzliche Eigenschaften)

- text
- textAlignment ="viewStart"
- textSize 36sp
- textColor

EditText (zusätzliche Eigenschaften):

- text
- textAlignment viewStart
- textSize 36sp
- textColor

- inputType
- lines
- minlines
- maxlines
- lineSpaceExtra none, 8sp

Wichtige inputType:

- text
- textMultiLine
- textPersonName
- number
- numberSigned
- numberDecimal

Button:

- id
- layout_width match_parent / wrap_content / @+id/tinput
- layout_height match_parent / wrap_content / @+id/tinput
- text
- textAlignment ="viewStart"
- textSize 36sp
- textColor @android:color/holo_green_light

Einheit	Beschreibung
dp	Die Einheit dp (ehemals auch als dip bezeichnet) repräsentiert eine abstrakte Einheit, die je nach Pixeldichte (dpi) variiert und eine Basispixeldichte von 160dpi besitzt. Dies bedeutet, dass 1dp auf einem 160dpi Bildschirm letztlich einen Pixel darstellt, dagegen auf einem 320dpi Bildschirm 2px Platz beansprucht. Die Einheit skaliert somit gemäß der Pixeldichte und sorgt dafür, dass beispielsweise GUI-Komponenten wie einen Button, optisch für den Benutzer immer gleich groß sind(=Smartphones wie Tablets) Eine Linie, die mit einer Breite von 2dp angegeben wird, beansprucht auf einem 160dpi Bildschirm 2px in der Breite, dagegen auf einem 240dpi Bildschirm drei (pysikalische" Pixel), sowie auf einem 320dpi Bildschirm 4 Pixel.
sp	Die Einheit sp verhält sich analog der Einheit dp mit nur einem Unterschied: Die Einheit berücksichtigt zudem die vom Benutzer gewünschte und systemweite Schriftgröße. Demzufolge wird diese Einheit ausschließlich im Umgang mit Schriftgrößen eingesetzt um somit sowohl die Pixeldichte des Displays zu berücksichtigen, als auch die Benutzereinstellung hinsichtlich gewünschter Schriftgröße zu berücksichtigen.
px	Diese Einheit repräsentiert einen physikalischen Pixel auf dem Display des Benutzers. Eine Linie, die mit einer Breite von 2px angegeben wird, beansprucht somit unabhängig vom Geräte-Typ, sowohl auf einem Smartphone 2px in der Breite, als auch auf einem Tablet.

3.2 Erstes Beispiel

Dieses Beispiel zeigt die Gestaltung der UI-Elemente im View-Editor und im Texteditor und die Verknüpfung mit dem Java-Code inklusive eines ActionsEvents.

3.2.1 UI-Prinzip

Die UI-Elemente werden nacheinander in den UI-View eingetragen. Die Constrainst definieren, wie die Elemente untereinander sich positionieren.

3.2.2 Erstellen eines Projektes

Erstellen eines neuen Projektes:

- Menü File
- Eintrag New
- Eintrag New Projekt
- Auswahl eines „Empty Activity“

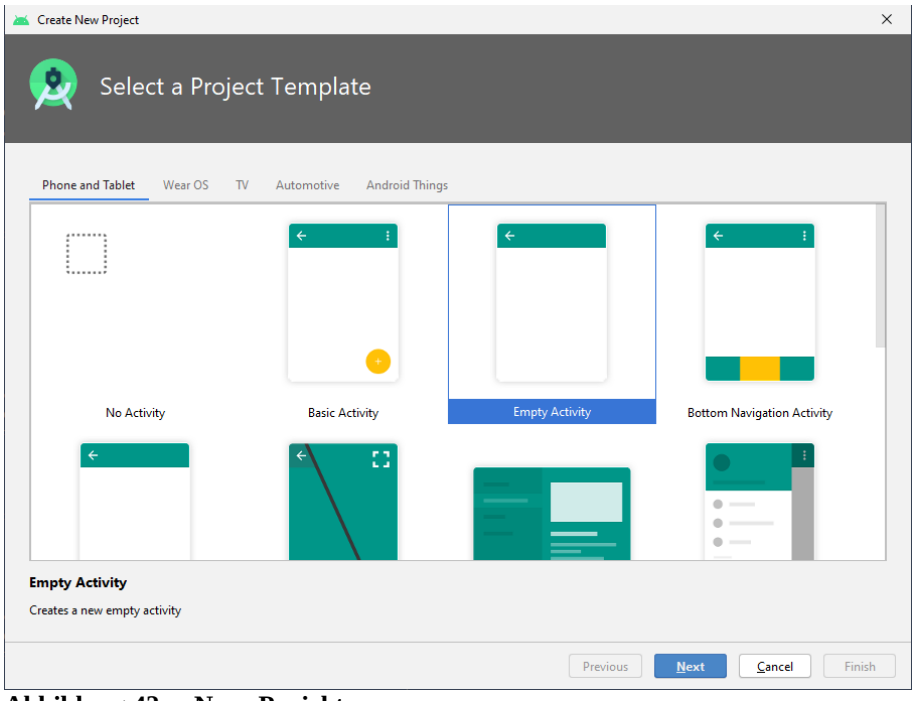


Abbildung 42 Neue Projekt

Nun eintragen:

- Name bsp01
- Automatisach sollte der Ordner gesetzt: D:\progs\Android\2020\Bsp01
- Language Java
- Minimum SDK API 25

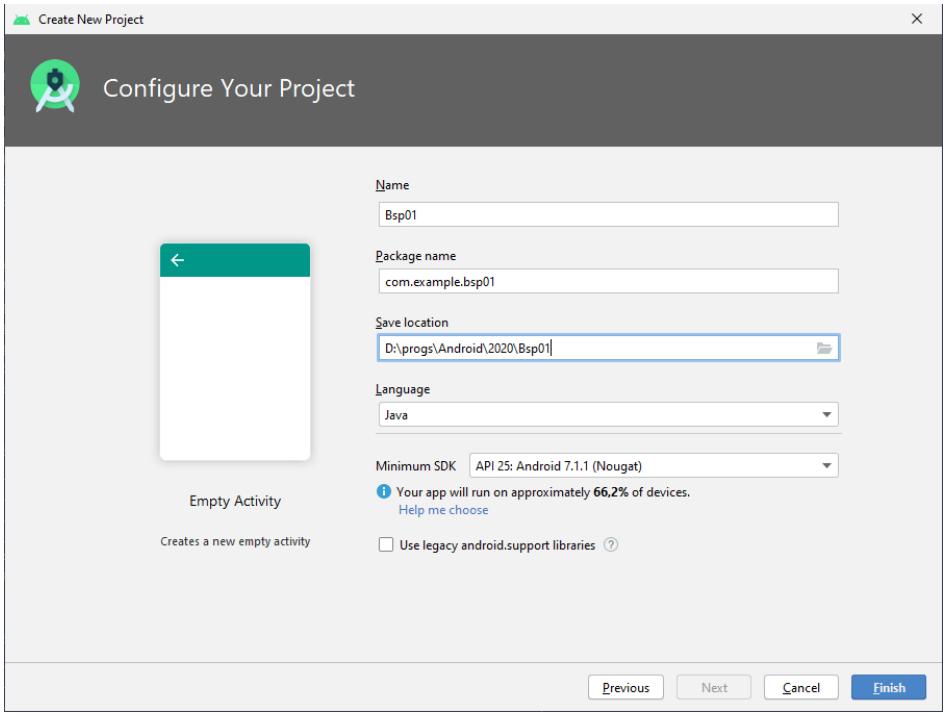


Abbildung 43 Daten des ersten Projektes

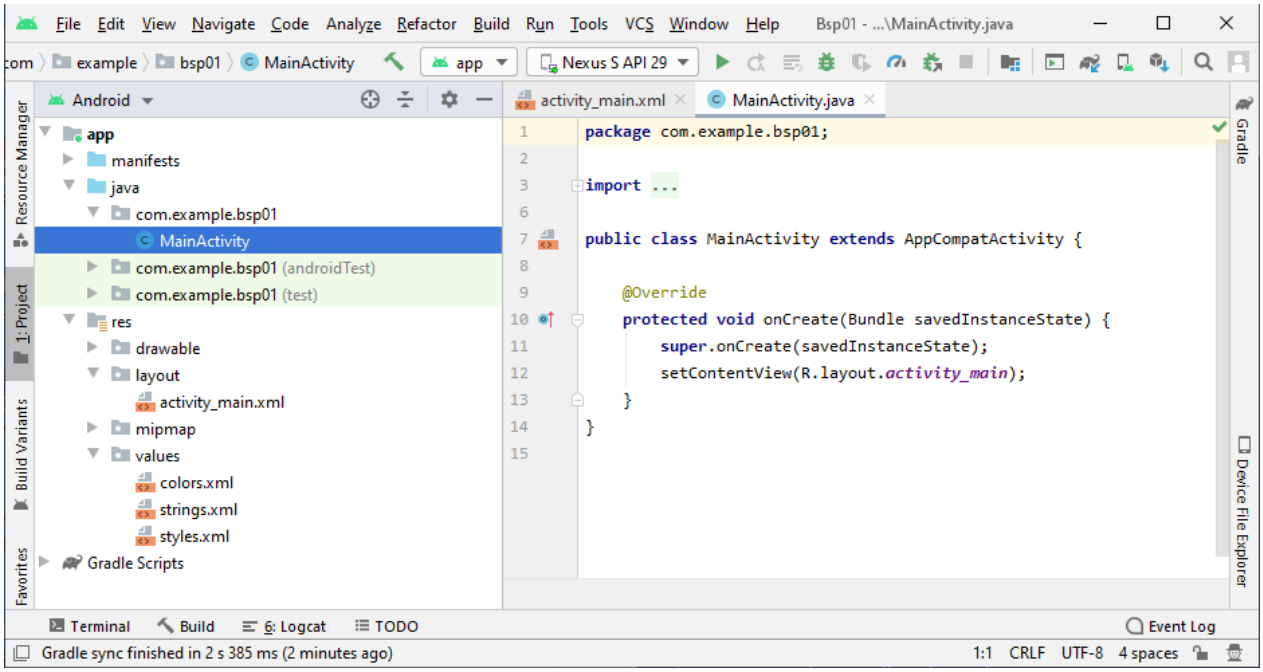


Abbildung 44 Das erste Rumpfprojekt

Auf der linken Seite ist der Projektbaum mit den wichtigsten Elementen zu sehen:

- java
  - MainActivity
    - Hier ist der Java-Quellcode gespeichert.

- res Speichert die Resourcen (XML, Strings, Farbe, Bilder)
  - layout
    - activity_main.xml
      - UI-XML
  - values Speichert die Farben und Strings

Auf der rechten Seite werden alle XML-Dateien und Javatexte angezeigt.

3.2.3 UI-View

Der UI-View hat drei Anzeigevariante:

- Text
- Grafik
- Text und Grafik



Abbildung 45 UI-View im Textmodus

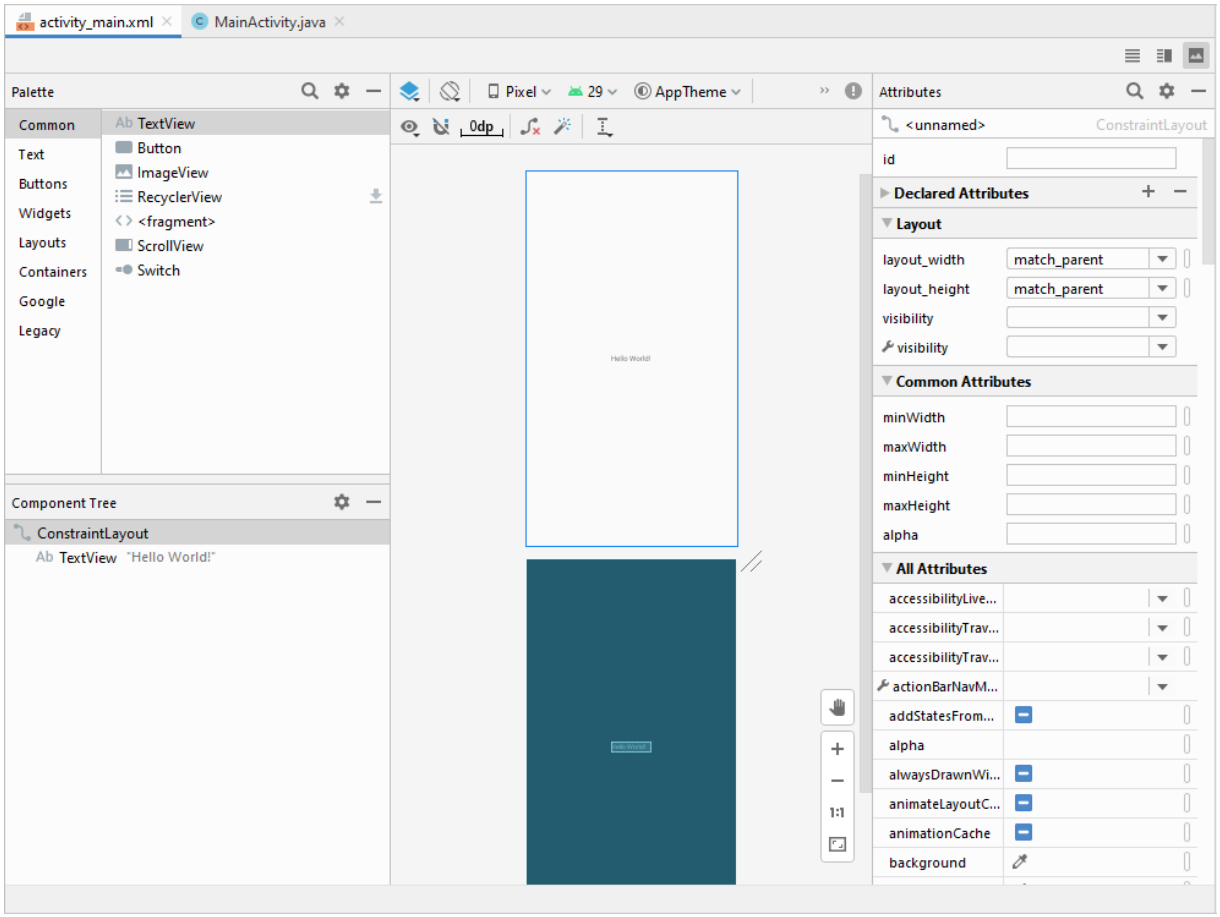


Abbildung 46 UI-View im Grafikmodus

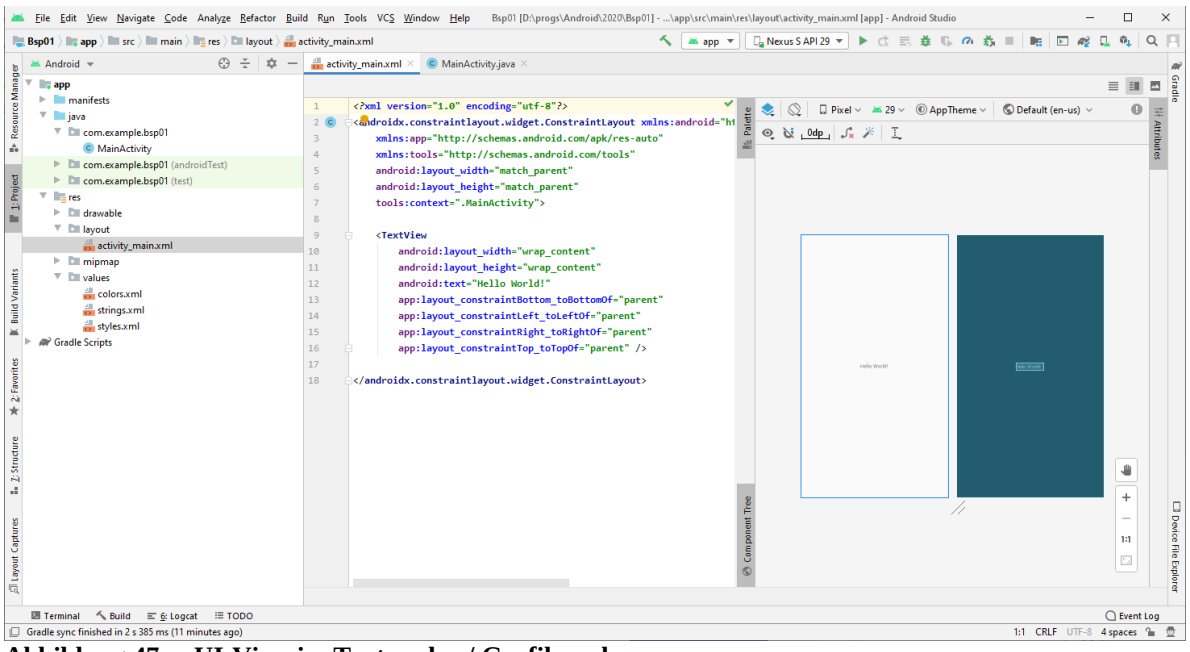


Abbildung 47 UI-View im Textmodus / Grafikmodus

Schalter:


AutoConnect der Constrainst


Löschen aller Constrainst


Infer Contraints

3.2.4 Umbau des Projektes

- Anklicken des TextViews und die entf-Taste drücken (löschen)
- Umschalten in den Grafikmodus
- In der Palette „Text“ auswählen
- Den „TextView“ Eintrag anklicken und mit Drag&Drop auf den Screen ziehen.
  - Bitte links oben positionieren
- Eintragen im Property-Fenster
  - id textview
  - text Eingabe
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

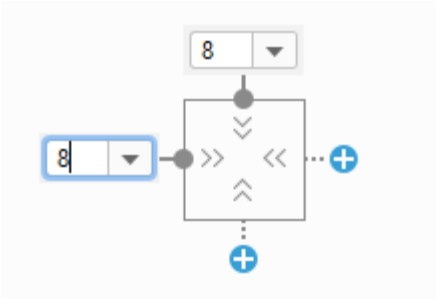


Abbildung 48 Constrainst des ersten TextViews

Mit Shift-F10 starten:

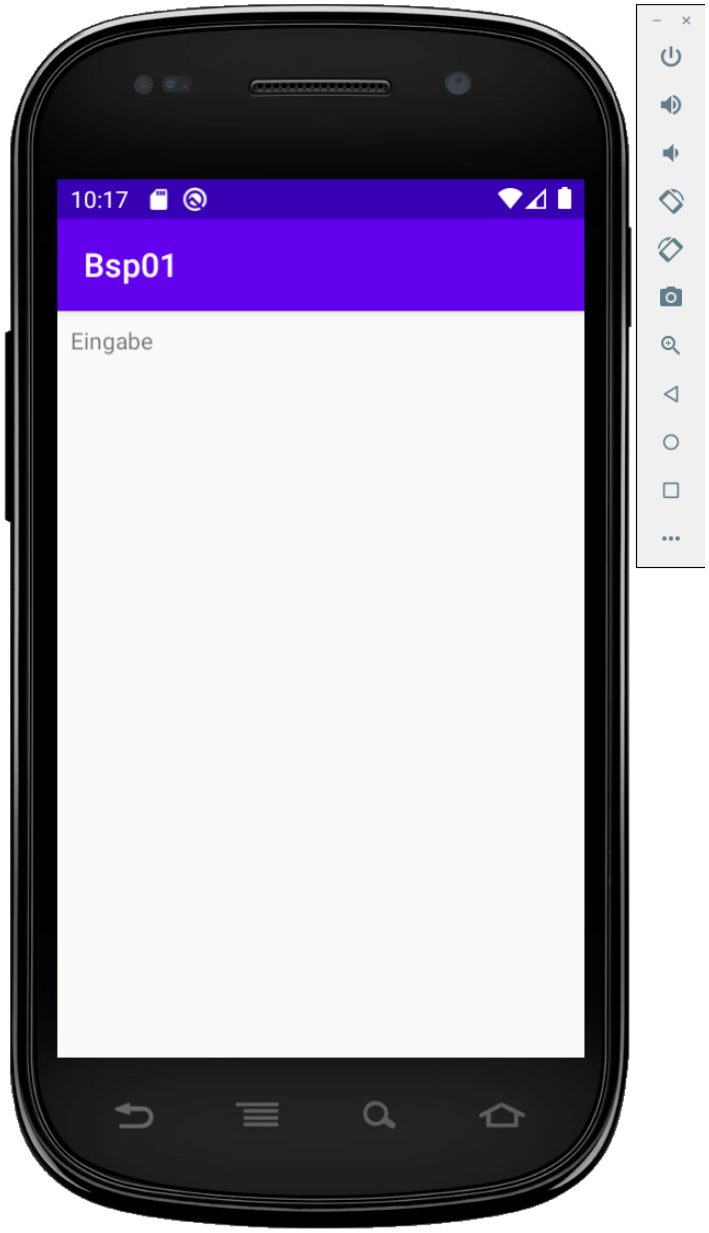


Abbildung 49 Anzeige des ersten Beispiels

- In der Palette „Text“ auswählen
- Den „PlainText“ Eintrag anklicken und mit Drag&Drop unterhalb des TextView auf den Screen ziehen.
- Eintragen im Property-Fenster
  - id tInput
  - text Meier
  - layout_width: match_parent
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Dreimal 8 eingeben (Top, left, right)

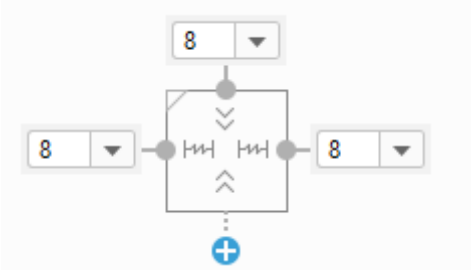
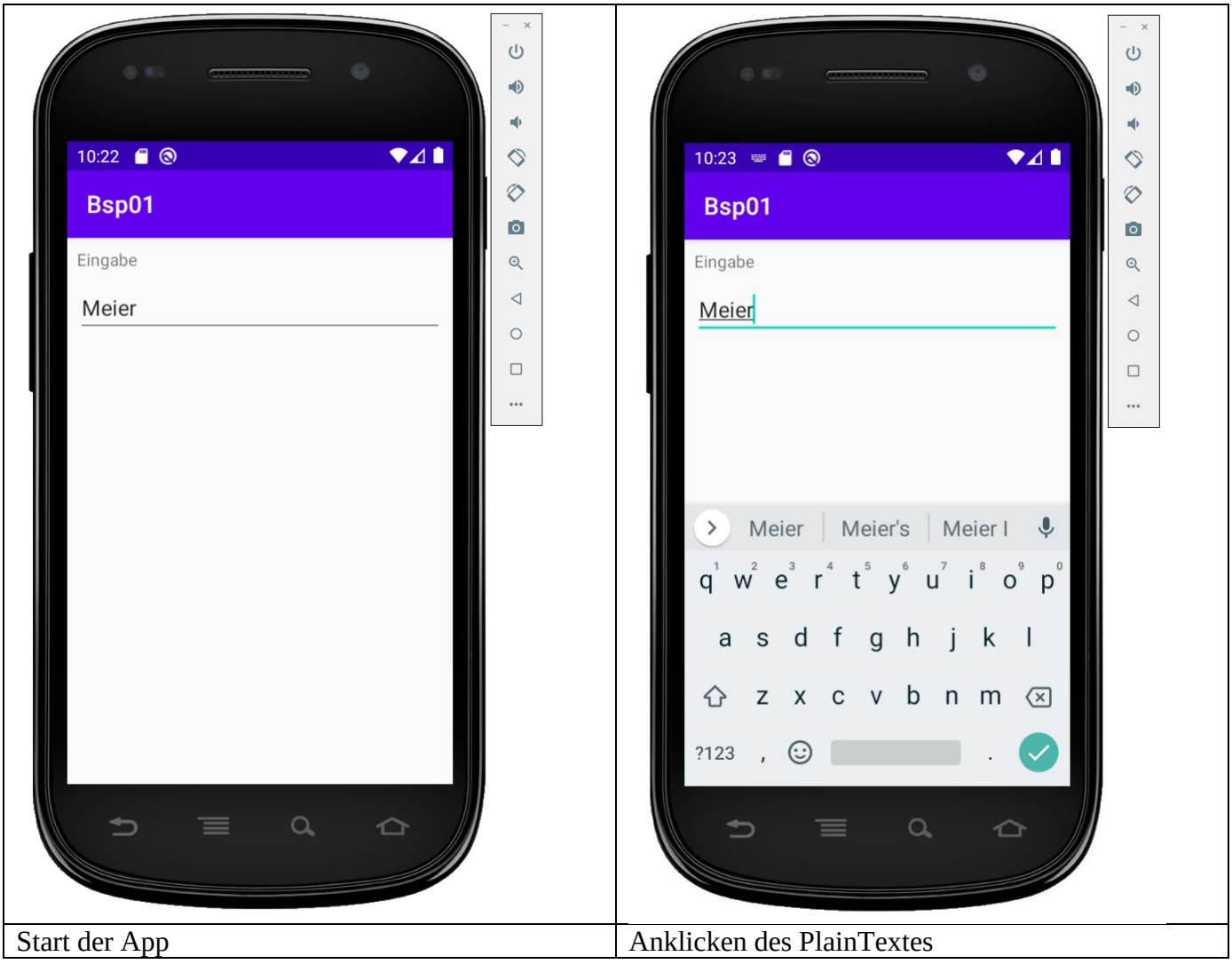


Abbildung 50 Constrainst des ersten PlainText-Elemenmtes

Mit Shift-F10 starten:



Start der App	Anklicken des PlainTextes

Nachdem Anklicken des PlainText erscheint die mit dem inputType ausgewählte Tastatur.

3.2.5 Schalter einfügen

- In der Palette „Buttons“ auswählen
- Den „Button“ Eintrag anklicken und mit Drag&Drop auf den Screen ziehen.
  - Bitte unter dem Textfeld positionieren
- Eintragen im Property-Fenster
  - id bnAction
  - text Klick mich

- Im Constrainst-Quadrat den oberen, linken und rechten plus-Schalter anklicken
- Dreimal 8 eingeben (Top, Left)

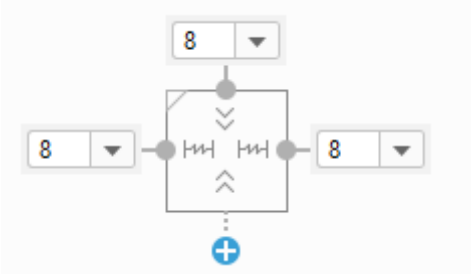


Abbildung 51 Constrainst des Schalters

Da ein Schalter auch ein Events auslöst, muss dieser Event auch mit dem Java-Code verknüpft werden. Dazu gibt es drei Methoden:

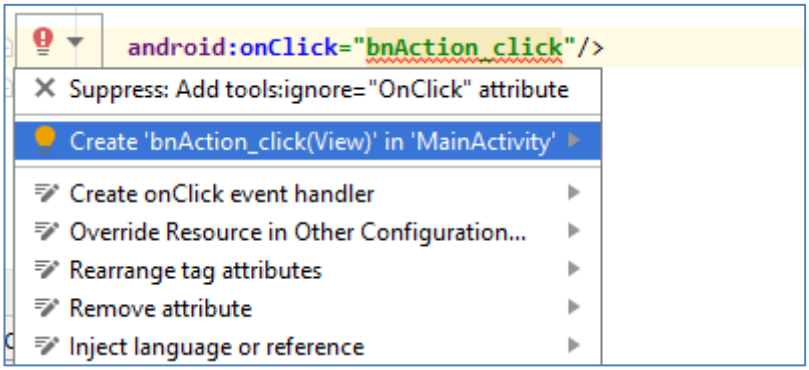
a) Java Code, ähnlich der Swing oder Java FX Notation

```
bn1.setOnClickListener(new View.OnClickListener() {
  @Override
    public void onClick(View v) {
      bn1_click();    // private Methode
    }
});
```

b) per XML, direktes Eingeben

Man trägt im XML-Code folgende Code ein:
android:onClick="bnAction_click"
Damit ist aber noch nicht die Event-Methode erzeugt.
Ablauf:

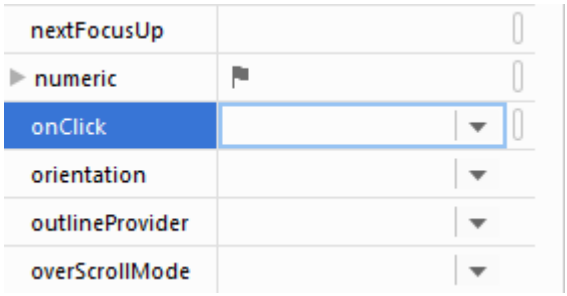
- Umschalten in den Textmodus
- Anklicken des Textes „bnAction_click“
- Nun muss rechts das rote oder gelbe Ausrufezeichen erscheinen.
- Die ComboBox öffnen und den Eintrag „Create…“ anklicken.



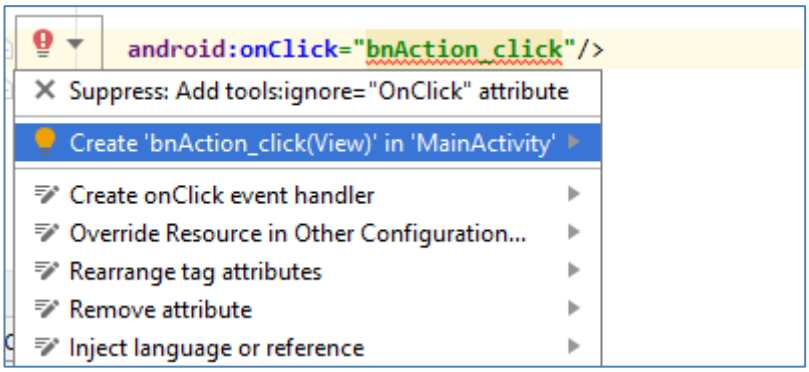

```
public void bnAction_click(View view) {
}
```

c) Im Property-Dialog

- Man schaltet in den Grafikmodus
- Anklicken des Schalters
- Runterscrollen zum Eintrag „All Attributes“
- Weiterscrollen bis zum Eintrag „onClick“
- Im Textfeld den Text „bnAction_click“ eintragen
- Dann die Returntaste drücken
- Nun in den Textmodus wechseln



- Anklicken des Textes „bnAction_click“
- Nun muss rechts das rote oder gelbe Ausrufezeichen erscheinen.
- Die ComboBox öffnen und den Eintrag „Create…“ anklicken.




```
public void bnAction_click(View view) {
}
```

3.2.6 Zweiten PlainText eintragen

- In der Palette „Text“ auswählen
- Den „PlainText“ Eintrag anklicken und mit Drag&Drop unterhalb des Schalters auf den Screen ziehen.
- Eintragen im Property-Fenster
  - id tOutput
  - text -
  - layout_width: match_parent
  - textcolor ändern (siehe Abbidlung unten)
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Dreimal 8 eingeben (Top, left, right)

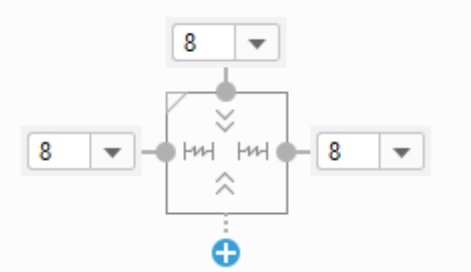


Abbildung 52 Constrainst des ersten PlainText-Elemenmtes



Abbildung 53 Textcolor ändern

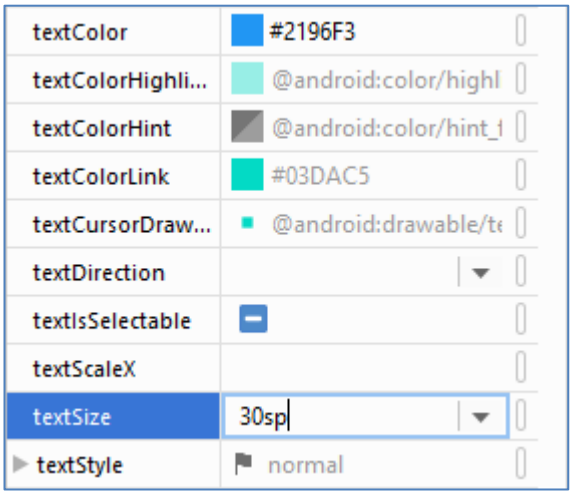


Abbildung 54 Textsize ändern

Besser ist es aber, wenn man die folgenden Konstanten benutzt:

- android:textAppearance="?android:attr/textAppearanceLarge"
- android:textAppearance="?android:attr/textAppearanceLargeInverse"
- android:textAppearance="?android:attr/textAppearanceLargePopupMenu"
- android:textAppearance="?android:attr/textAppearanceMedium"
- android:textAppearance="?android:attr/textAppearanceMediumInverse"

- android:textAppearance="?android:attr/textAppearanceMediumPopupMenu"
- android:textAppearance="?android:attr/textAppearanceSmall"
- android:textAppearance="?android:attr/textAppearanceSmallInverse"
- android:textAppearance="?android:attr/textAppearanceSmallPopupMenu"

Nun ist das UserInterface fertig.

Mit Shift-F10 starten:

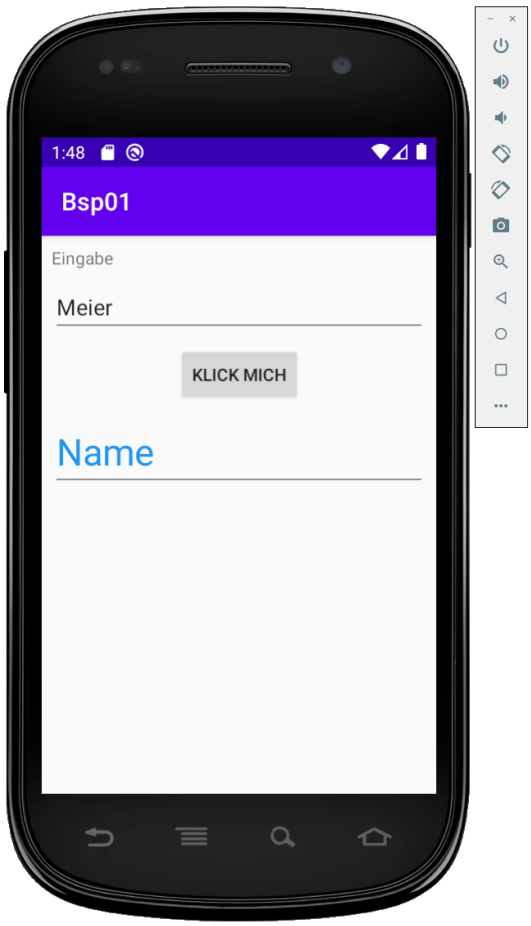


Abbildung 55 Das User-Interface des ersten Beispiels

3.2.7 Java-Code

Im Java-Code müssen vier Bereiche eingetragen werden:

- Import der UI-Klassen
- Globale Variablen der UI-Elemente
- Holen der Referenzen der UI-Elemente
- Methode „bnAction_click“ füllen

a) Import der UI-Klassen

Die ist eine Zeile:

```
import android.widget.*;
```

b) Eintragen der Referenzen

```
private TextView textview = null;
private EditText tInput =  null;
```

```
private EditText tOutput  =  null;
private Button bnAction = null;
```

Wichtig sind aber nur die beiden EditText-Zeilen!

c) Holen der Referenzen

```
textview1 = (TextView) findViewById(R.id.textview);
tInput = (EditText) findViewById(R.id.tInput);
tOutput = (EditText) findViewById(R.id.tOutput);
bnAction = (Button) findViewById(R.id.bnAction);
```

Wichtig sind aber nur die beiden EditText-Zeilen!

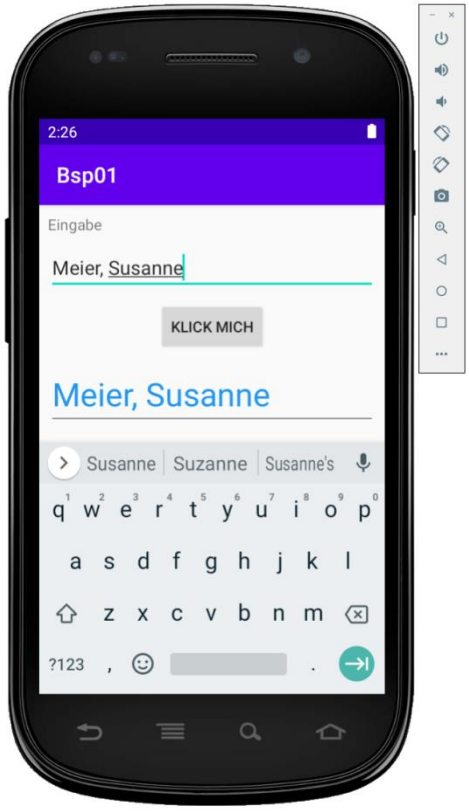
d) Methode „bnAction_click“ füllen

```
public void bnAction_click(View view) {
   String s = tInput.getText().toString();
   tOutput.setText(s);
}
```

Man beachte:

- Die Zeile „String s = tInput.getText();“ ergibt einen Fehler!

Ergebnis



Kompletter XML-Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Eingabe"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/tInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:inputType="textPersonName"
        android:text="Meier"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview" />

    <Button
        android:id="@+id/bnAction"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:onClick="bnAction_click"
        android:text="Klick mich"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tInput" />

    <EditText
        android:id="@+id/tOutput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:inputType="textPersonName"
        android:text="Name"
        android:textColor="#2196F3"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/bnAction" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java-Code:

```
package com.example.bsp01;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
```

```
import android.widget.*;

public class MainActivity extends AppCompatActivity {

    private TextView textview = null;
    private EditText tInput =  null;
    private EditText tOutput  =  null;
    private Button bnAction = null;



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

        textview = (TextView) findViewById(R.id.textview);
        tInput = (EditText) findViewById(R.id.tInput);
        tOutput = (EditText) findViewById(R.id.tOutput);
        bnAction = (Button) findViewById(R.id.bnAction);

    }


    public void bnAction_click(View view) {
        String s = tInput.getText().toString();
        tOutput.setText(s);
    }
}
```

3.3 Zweites Beispiel mit UI-Elementen

Dieses Beispiel zeigt die Verwendung von anderen UI-Elementen:

- CheckBox
- RadionButton
- Switch
- Spinner
- MultiLineText

3.3.1 Anlegen eines neuen Projektes

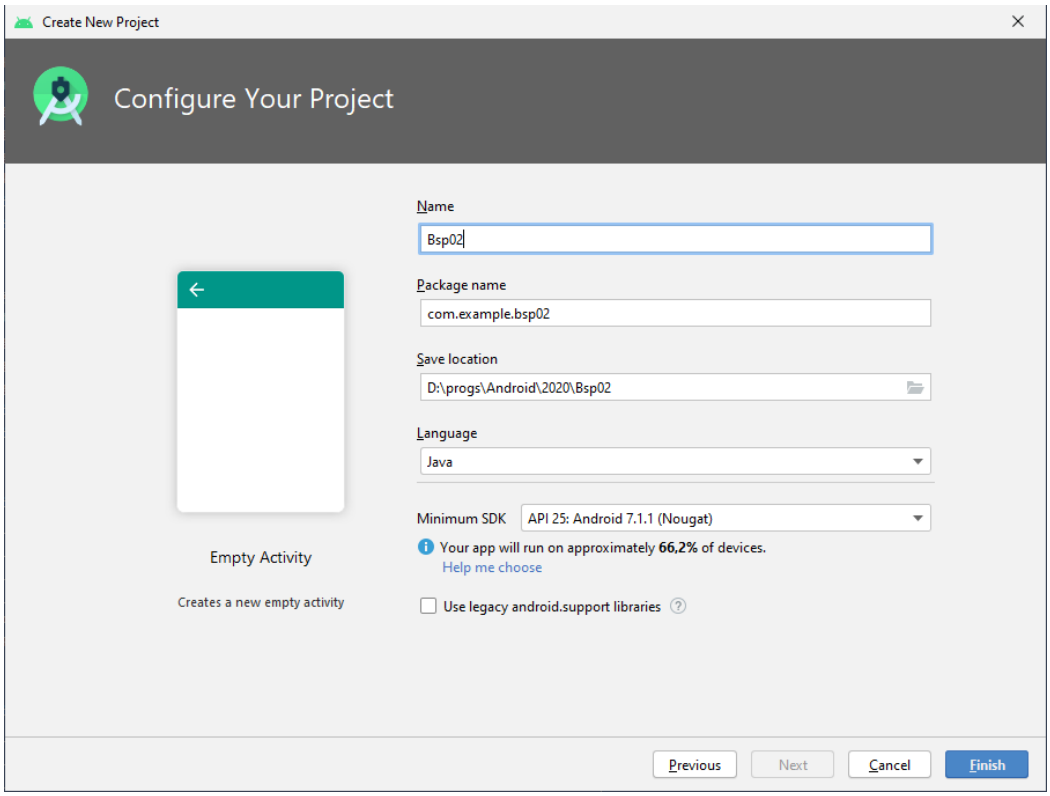


Abbildung 56 Anlegen des zweiten Beispiels Bsp02

3.3.2 Anlegen der UI-Elemente

Anlegen einer CheckBox:

- id checkbox1
- Text: CheckBox
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

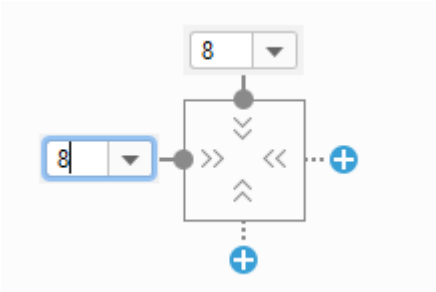


Abbildung 57 Constrainst der CheckBox

3.3.3 Anlegen von einer RadioGroup:

- id radiogroup1
- layout_width: wrap_content
- layout_height: wrap_content

3.3.4 Anlegen von zwei RadioButtons:

- Drag & Drrop in den Treeitem „radiogroup“ !!!
- 1. RadioButton
  - id radiobutton1
  - Text: RadioButton1
  - layout-marginLeft 10dp
  - layout-marginTop 10dp
  - checked true
  - Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
  - Zweimal 8 eingeben (Top, Left)

- 2. RadioButton
  - id radiobutton2
  - Text: RadioButton2
  - layout-marginLeft 10dp
  - layout-marginTop 10dp
  - Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
  - Zweimal 8 eingeben (Top, Left)

Nachdem die beiden RadioButtons in die RadioGroup eingetragen wurden, kann man die Constraints für die RadioGroup setzen:

- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

Abfrage:

```
if (rbadd.isChecked()) {
  Log.i(TAG, "----------------------  add is checked");
}
```

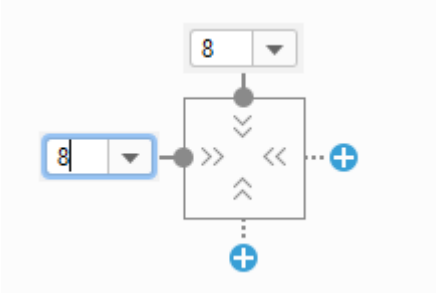


Abbildung 58 Constrainst der RadioGroup

3.3.5 Anlegen eines Switches:

- id switch1
- Text: Switch
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Top: 16
- Left: 8

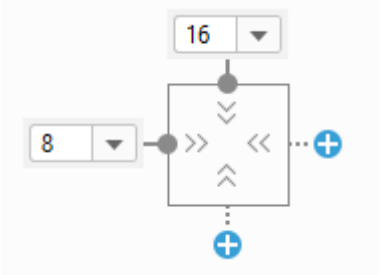


Abbildung 59 Constrainst des Switch

3.3.6 Anlegen eines Spinner (Haupteintrag: Containers):

- id spinner1
- layout_width match_parent
- prompt Hier ist der 1. Spinner
- spinnermode dialog à la JList, die auf dem Bildschirm erscheint.
- Im Constrainst-Quadrat den oberen, linke und rechten plus-Schalter anklicken
- Dreimal 8 eingeben (Top, Left)

3.3.7 Anlegen eines 2. Spinner (Haupteintrag: Containers):

- id spinner2
- layout_width match_parent
- prompt Hier ist der 2. Spinner
- spinnermode dropdown
- Im Constrainst-Quadrat den oberen, linke und rechten plus-Schalter anklicken
- Dreimal 8 eingeben (Top, Left)

3.3.8 Anlegen eines ToggleButtons

- id togglebutton1
- textOff: togglebutton1 Aus
- textOn: togglebutton1 An
- text: Togglebutton1
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

3.3.9 Anlegen eines Buttons

- id bnAction1
- text: Action
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

3.3.10 Anlegen eines MultiLineTextes

Dieser Editor muss sich flexibel positionieren. Dass komplizierter:

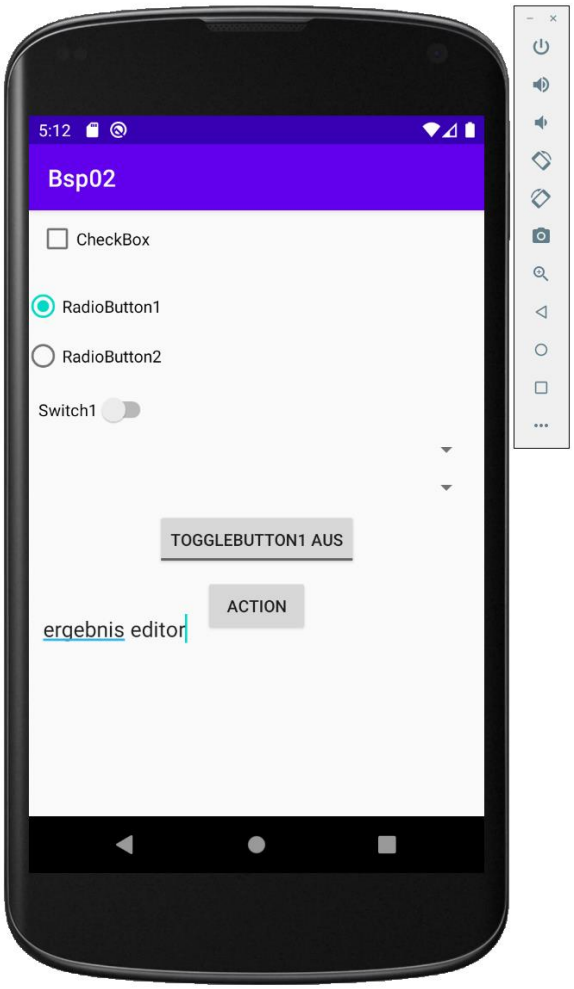
Ablauf:

- Fügen Sie einen MultiLine-Texteditor per Drag & Drop in den View.
- Positionieren und ändern Sie die Abmessungen für Ihren Bedarf.
- **Contraint-Layout** **Wichtig:**
  - Der Editor hat an jeder Seite einen Kreis für die Contraints.
  - Klicken Sie den oberen Kreis an und verschieben die „Linie“ zum Schalter „Action“.
  - Damit wurd der parent gesetzt.
  - Nun setzen Sie die vier Eingaben in den Plus-Elementen
    - Viermal 8 eintragen
- Sicherstellen, dass der „layout_constraintTop_toBottomOf“ korrekt gesetzt ist.
- Setzen von layout_width
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein
- Setzen von layout_height
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein

```
<EditText
  android:id="@+id/ergebnis"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:layout_marginStart="8dp"
  android:layout_marginTop="8dp"
  android:layout_marginEnd="8dp"
  android:layout_marginBottom="8dp"
  android:gravity="start|top"
  android:inputType="textMultiLine"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
```

```
   app:layout_constraintTop_toBottomOf="@+id/bnAction1"
/>
```

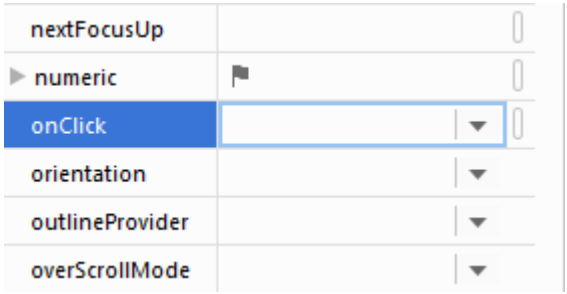
- id ergebnis
- Im Constrainst-Quadrat alle vier plus-Schalter anklicken
- Viermal 8 eingeben (Top, Left, Right, Bottom)



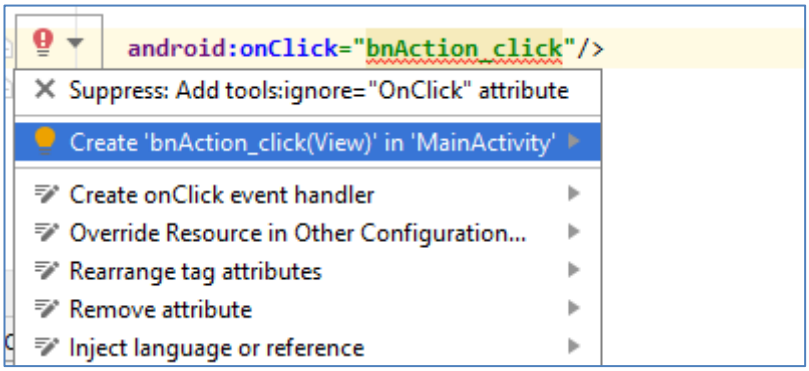
3.3.11 onClickAction

Im Property-Dialog

- Man schaltet in den Grafikmodus
- Anklicken des Schalters
- Runterscrollen zum Eintrag „All Attributes“
- Weiterscrollen bis zum Eintrag „onClick“
- Im Textfeld den Text „bnAction_click“ eintragen
- Dann die Returntaste drücken
- Nun in den Textmodus wechseln



- Anklicken des Textes „bnAction_click“
- Nun muss rechts das rote oder gelbe Ausrufezeichen erscheinen.
- Die ComboBox öffnen und den Eintrag „Create…“ anklicken.




```
public void bnAction_click(View view) {
}
```

3.3.12 Java-Code

Ablauf:

- Im Java-Code werden alle Referenzen geholt und in den globalen Variablen gespeichert.
- Der zweite Spinner wird mittels Java-Code gefüllt (à la Datenbank).
- Dann werden im onClickEvent die aktuellen Werte der UI-Elemente abgefragt und inden Editor ausgegeben.
- Desweiteren werden Change-Event für alle UI-Elemente definiert.

Referenzen:

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

    checkbox1  = (CheckBox) findViewById(R.id.checkBox1);
    radiogroup1 = (RadioGroup) findViewById(R.id.radiogroup1);
    radionbutton1  = (RadioButton) findViewById(R.id.radiobutton1);
    radionbutton2 = (RadioButton) findViewById(R.id.radiobutton2);
    switch1 = (Switch) findViewById(R.id.switch1);
    spinner1 = (Spinner) findViewById(R.id.spinner1);
    spinner2 = (Spinner) findViewById(R.id.spinner2);
    togglebutton1 = (ToggleButton) findViewById(R.id.togglebutton1);
    ergebnis  = (EditText) findViewById(R.id.ergebnis);

    // setzen der Werte des 2. Spinners mittels "Datenbank"
```

```
    String spinnerItems[] =   new String[]{
                    "Section 1",
                    "Section 2",
                    "Section 3",
                    "Section 4",
                    "Section 5"
            };
    ArrayAdapter aa = new
ArrayAdapter(this,android.R.layout.simple_spinner_item,spinnerItems);

aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    //Setting the ArrayAdapter data on the Spinner
    spinner2.setAdapter(aa);
}
```

onClick-Methode des Schalters „Action“:

```
public void bnAction_Click(View view) {
  String s="Ergebnis\n";

  if (checkbox1.isChecked()) {
    s+="CheckBox gesetzt\n";
  }
  else {
    s+="CheckBox nicht gesetzt\n";
  }

  switch (radiogroup1.getCheckedRadioButtonId()) {
    case R.id.radiobutton1:
      s+="RadioButton1 gesetzt\n";
      break;
    case R.id.radiobutton2:
      s+="RadioButton1 gesetzt\n";
      break;
    default:
      s+="keine RadioButton gesetzt\n";
      break;
    }

    if (switch1.isChecked()) {
      s+="switch1 gesetzt\n";
    }
    else {
      s+="switch1 nicht gesetzt\n";
    }

    if (togglebutton1.isChecked()) {
      s+="togglebutton1 gesetzt\n";
    }
    else {
      s+="togglebutton1 nicht gesetzt\n";
    }



  // spinner
  if (spinner1. getSelectedItemPosition()>=0) {
    Object obj1= spinner1.getSelectedItem();  // oder String
```

```
    s+="spinner1: "+obj1+"\n";
  }
  else {
    s+="spinner1 nicht gesetzt\n";
  }

  if (spinner2. getSelectedItemPosition()>=0) {
    Object obj2= spinner2.getSelectedItem();
    s+="spinner2: "+obj2+"\n";
  }
  else {
    s+="spinner1 nicht gesetzt\n";
  }

  ergebnis.setText(s);
}  // bnnAction_Click
```

ChangeListener

a) Definition im Konstruktor

```
//Setting the ArrayAdapter data on the Spinner
spinner2.setAdapter(aa);
setOnClick(checkbox1);
setOnClick(radionbutton1);
setOnClick(radionbutton2);
setOnClick(switch1);
setChangeListener(spinner1);
setChangeListener(spinner2);
setOnClick(togglebutton1);
// setChangeListener(ergebnis);  geht nicht wegen rekursiv !! oder toast
```

b) Quellcode der Methoden

Beachten dass, dass die Button-Methode mehrfach aufgerufen wird!

```
private int clicknr=0;

private void setOnClick(Button bn) {
  bn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      ergebnis.setText("Button wurde gedrueckt: "+(++clicknr));
    }
  });
}

  // import android.view.View;
private int spinnernr=0;
private void setChangeListener(Spinner spinner) {
  spinner.setOnItemSelectedListener(
    new AdapterView.OnItemSelectedListener() {
      @Override
      public void onItemSelected(AdapterView<?> parent, View view,
          int position, long id) {
        ergebnis.setText("Spinner-Eintrag wurde veraendert: "+
            (++spinnernr));
      }
```

```
      @Override
      public void onNothingSelected(AdapterView<?> parent) {
      }
    }
  );
}

    // import android.text.*;
  private void setChangeListener(EditText edit) {
    edit.addTextChangedListener(new TextWatcher() {

      public void afterTextChanged(Editable s) {
        ergebnis.setText("editor wurde geändert");
      }

      public void beforeTextChanged(CharSequence s, int start,
        int count, int after) {
      }

      public void onTextChanged(CharSequence s, int start, int before,
        int count) {
      }
    }
  );
}
```

3.4 Drittes Beispiel mit UI-Elementen: Add/Sub-Rechner

Dieses Beispiel zeigt die Entwicklung eines „Taschenrechners“:

- Zwei decimal-Textfelder
- Drei RadionButtons
- Ein Action-Button
- Ein Editor für die Ergebnisse

3.4.1 Anlegen eines neuen Projektes

Name: AddSubRechner

3.4.2 Anlegen der UI-Elemente

3.4.2.1 Anlegen einer Überschrift:

- id textview0
- Text: Taschenrechner
- textSize 18sp
- textcolor #2196F3"
- layout_width wrap_content
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen, linken und rechten plus-Schalter anklicken

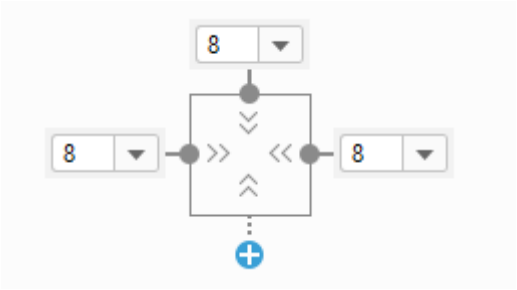


Abbildung 60 Constrainst der Überschrift

3.4.2.2 Anlegen einer Beschriftung (Textview):

- id textview1
- Text: 1. Zahl
- layout_width wrap_content
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen und linke plus-Schalter anklicken

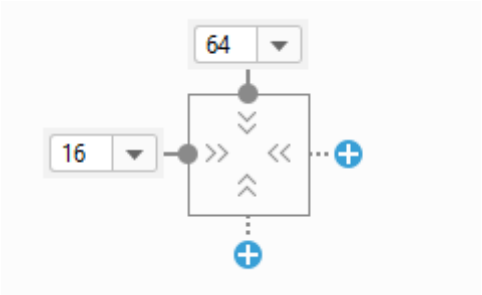


Abbildung 61 Constrainst dererstenBeschriftung

3.4.2.3 Anlegen eines EditText, decimal:

- id tzahl2
- Text: 0
- inputType numberDecimal
- layout_width 0dp
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen, linken und rechten plus-Schalter anklicken

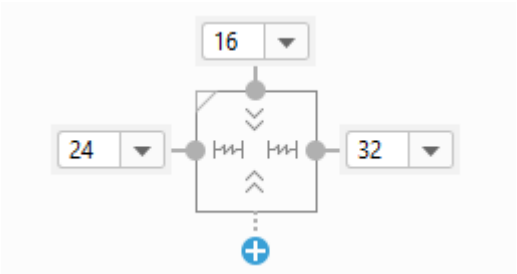


Abbildung 62 Constrainst der 1. Zahl

3.4.2.4 Anlegen einer Beschriftung (Textview):

- id textview2
- Text: 2. Zahl
- layout_width wrap_content
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen und linke plus-Schalter anklicken

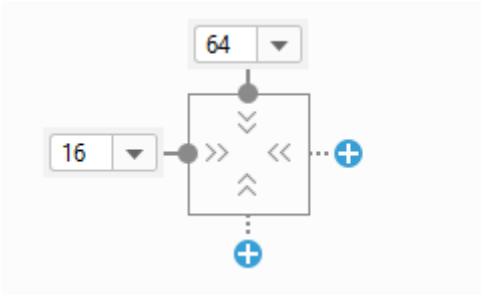


Abbildung 63 Constrainst zweiten Beschriftung

3.4.2.5 Anlegen eines EditText, decimal:

- id tzahl2
- Text: 0
- inputType numberDecimal
- layout_width 0dp
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen, linken und rechten plus-Schalter anklicken

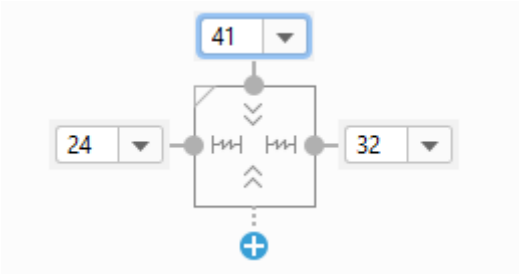


Abbildung 64 Constrainst der 2. Zahl

Aktueller Stand

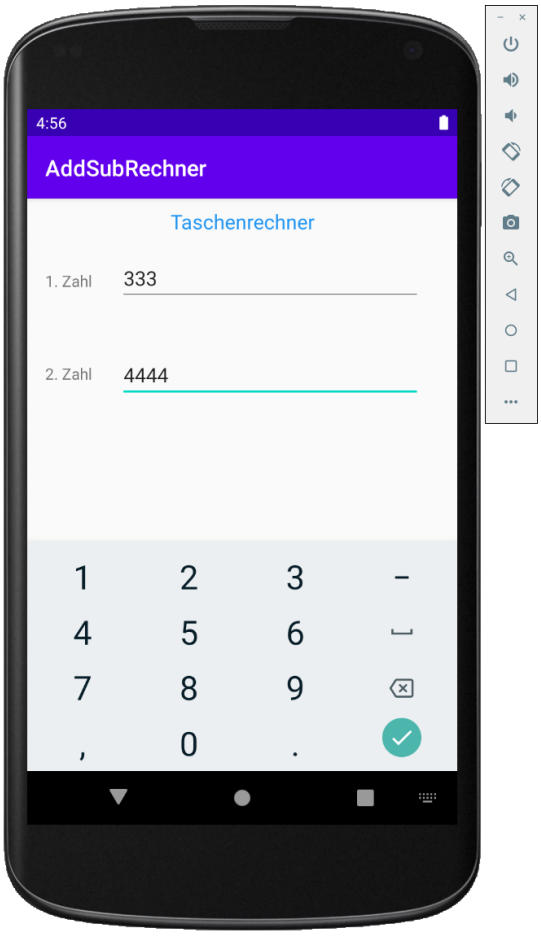


Abbildung 65 Erster Stand des dritten Beispiels

3.4.2.6 Funktion:

Eintragen eines TextViews:

- id textview3

- Text: Funktion
- layout_width wrap_content
- layout_height wrap_content
- Im Constrainst-Quadrat den oberen und linke plus-Schalter anklicken

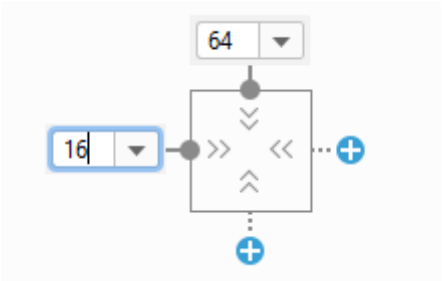
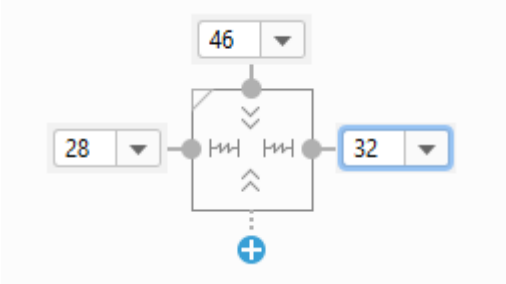


Abbildung 66 Constrainst der dritten Beschriftung

Nun werden die beiden RadioButton mit der RadioGroup eingefügt.



- 1. RadioButton
  - id rbadd
  - Text: rbsub
  - checked true

- 2. RadioButton
  - id radiobutton2
  - Text: RadioButton2
  - layout-marginLeft 10dp
  - layout-marginRight 10dp

3.4.2.7 Anlegen eines Action-Buttons

- id bnaction
- text: Berechnen
- Im Constrainst-Quadrat den oberen und linken plus-Schalter anklicken
- Zweimal 8 eingeben (Top, Left)

3.4.2.8 Anlegen eines MultiLineTextes

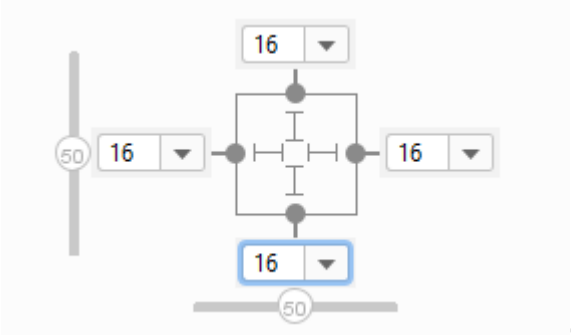
Dieser Editor muss sich flexibel positionieren. Dass komplizierter:

Ablauf:

- Fügen Sie einen MultiLine-Texteditor per Drag & Drop in den View.
- Positionieren und ändern Sie die Abmessungen für Ihren Bedarf.
- **Contraint-Layout** **Wichtig:**
  - Der Editor hat an jeder Seite einen Kreis für die Contraints.
  - Klicken Sie den oberen Kreis an und verschieben die „Linie“ zum Schalter „Action“.
  - Damit wurd der parent gesetzt.
  - Nun setzen Sie die vier Eingaben in den Plus-Elementen
    - Viermal 8 eintragen
- Sicherstellen, dass der „layout_constraintTop_toBottomOf“ korrekt gesetzt ist.
- Setzen von layout_width
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein
- Setzen von layout_height
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein

```
<EditText
   android:id="@+id/ergebnis"
   android:layout_width="0dp"
   android:layout_height="0dp"
   android:layout_marginStart="8dp"
   android:layout_marginTop="8dp"
   android:layout_marginEnd="8dp"
   android:layout_marginBottom="8dp"
   android:gravity="start|top"
   android:inputType="textMultiLine"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/bnAction1"
/>
```

- id ergebnis
- Im Constrainst-Quadrat alle vier plus-Schalter anklicken
- Viermal 8 eingeben (Top, Left, Right, Bottom)



zahl2

Kompletter Quellcode:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textview0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:text="Taschenrechner"
        android:textColor="#2196F3"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="64dp"
        android:text="1. Zahl"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/tzahl1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="32dp"
        android:inputType="numberDecimal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textview1"
        app:layout_constraintTop_toBottomOf="@+id/textview0" />

    <TextView
        android:id="@+id/textview2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="64dp"
        android:text="2. Zahl"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview1" />

    <EditText
        android:id="@+id/tzahl2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="41dp"
```

```
    android:layout_marginEnd="32dp"
    android:inputType="numberDecimal"
    app:layout_constraintEnd_toEndOf="@+id/tzahl1"
    app:layout_constraintStart_toEndOf="@+id/textview2"
    app:layout_constraintTop_toBottomOf="@+id/tzahl1" />

<TextView
    android:id="@+id/textview3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="64dp"
    android:text="Funktion"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textview2" />

<RadioGroup
    android:id="@+id/radiogroup1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="28dp"
    android:layout_marginTop="46dp"
    android:layout_marginEnd="32dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textview3"
    app:layout_constraintTop_toBottomOf="@+id/tzahl2">


    <RadioButton
        android:id="@+id/rbadd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:text="Addition" />

    <RadioButton
        android:id="@+id/rbsub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Subtraktion"
        app:layout_constraintLeft_toRightOf="@+id/rbadd"
        />
</RadioGroup>

<Button
    android:id="@+id/bnaction"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="16dp"
    android:onClick="bnAction_Click"
    android:text="Berechnen"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/radiogroup1" />

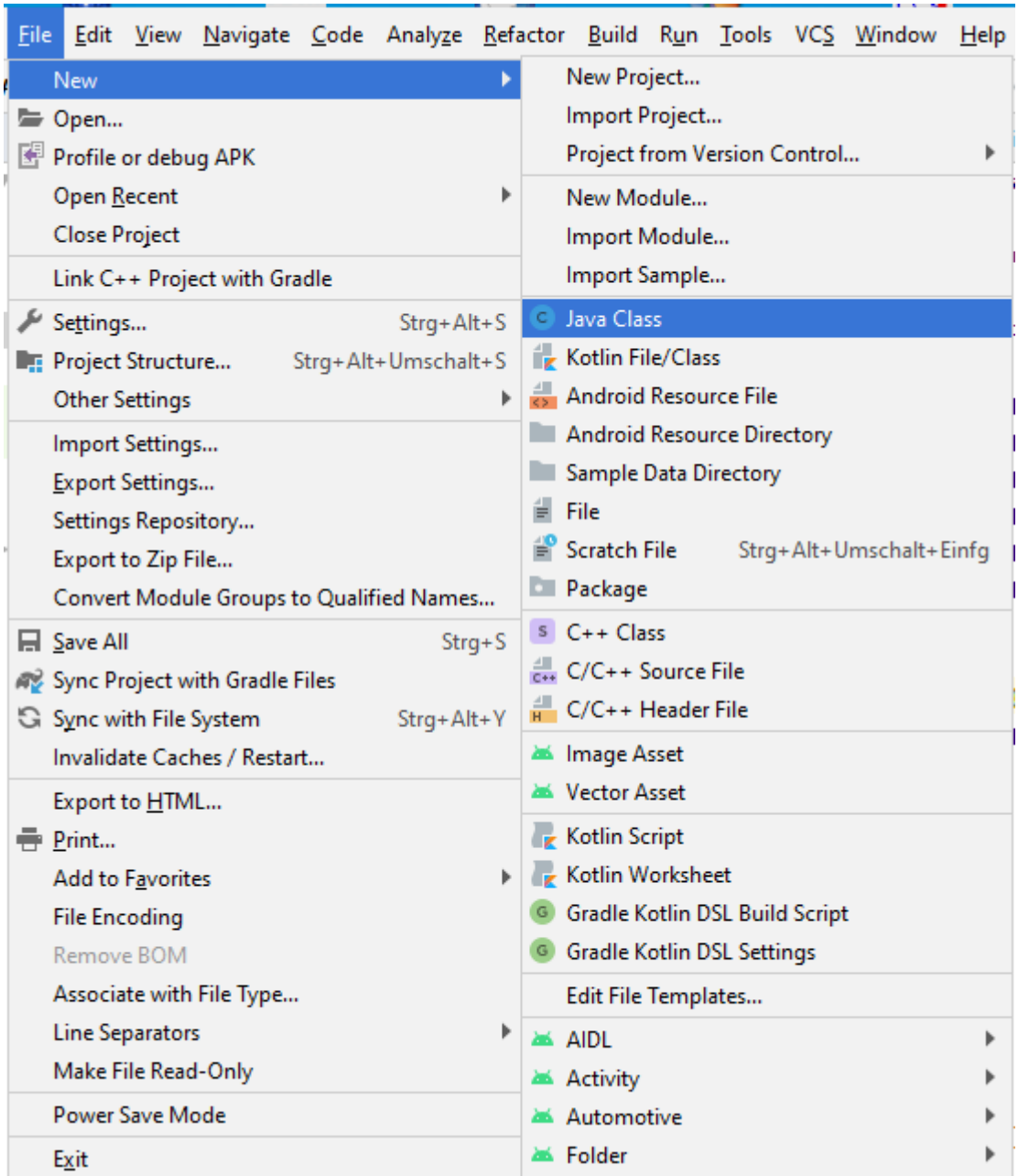
<EditText
android:id="@+id/tzahl2"
android:layout_width="0dp"
android:layout_height="wrap_content"
```

```
    android:layout_marginStart="24dp"
    android:layout_marginTop="41dp"
    android:inputType="numberDecimal"
    app:layout_constraintEnd_toEndOf="@+id/tzahl1"
    app:layout_constraintStart_toEndOf="@+id/textview2"
    app:layout_constraintTop_toBottomOf="@+id/tzahl1" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

3.4.3 Java_Code

3.4.3.1 Anlegen einer Klasse Basis



Name: Basis.java

Inhalt:

Bitte den Quellcode im Kapitel 19.8, Seite 368.

3.4.3.2 Main.java

```
package com.example.addsubrechner;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;

import android.widget.*;

public class MainActivity extends AppCompatActivity {

    private RadioGroup radiogroup1 = null;
    private RadioButton rbadd = null;
    private RadioButton rbsub = null;
    private EditText tzahl1 = null;
    private EditText tzahl2 = null;
    private EditText tergebnis = null;


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

        tzahl1  = (EditText) findViewById(R.id.tzahl1);
        tzahl2  = (EditText) findViewById(R.id.tzahl2);

        radiogroup1 = (RadioGroup) findViewById(R.id.radiogroup1);
        rbadd  = (RadioButton) findViewById(R.id.rbadd);
        rbsub = (RadioButton) findViewById(R.id.rbsub);
        tergebnis  = (EditText) findViewById(R.id.ergebnis);
    }

    public void bnAction_Click(View view) {
        Double zahl1 = Basis.getDoubleNumber(tzahl1, "Bitte eine gültige 1. Zahl eingeben");
        if (zahl1==Double.NaN) {
            return;
        }
        Double zahl2 = Basis.getDoubleNumber(tzahl2, "Bitte eine gültige 2. Zahl eingeben");
        if (zahl1==Double.NaN) {
            return;
        }
        double erg=0.0;
        switch (radiogroup1.getCheckedRadioButtonId()) {
            case R.id.rbadd:
                erg=zahl1+zahl2;
                break;
            case R.id.rbsub:
                erg=zahl1-zahl2;
                break;
            default:
                tergebnis.setText("weder add noch sub is selected");
                return;
```

```
        }
        tergebnis.setText( Double.toString(erg) );

    }
}
```

3.5 Beispiel Spritverbrauch

Entwickelt werden soll ein kleines Rechenprogramm, welches den Verbrauch eines Kraftfahrzeuges ermittelt.

Besonderheiten:

- Komplexes Constraint-layout
- Speicherung der Daten (Kilometer, Liter, Auswahl der Berechnung)
- Laden der Daten (Kilometer, Liter, Auswahl der Berechnung)
- Verwendung von definierten Strings
- Verwendung von definierten Farben

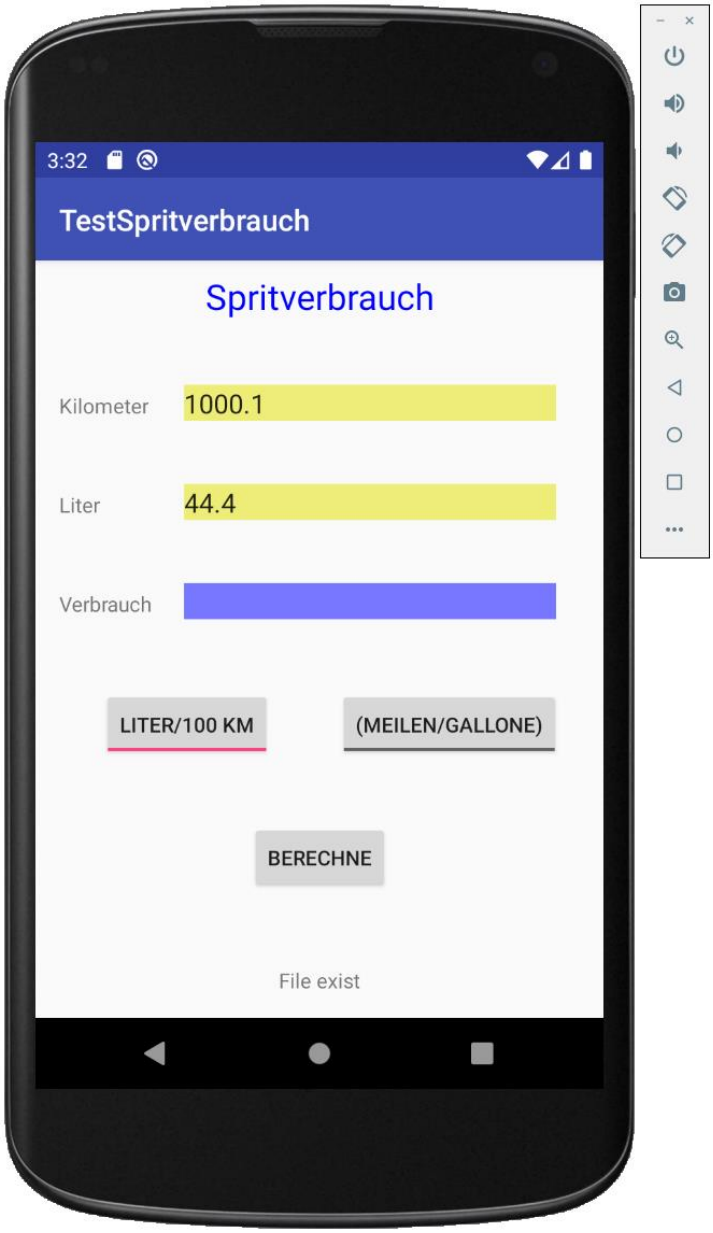


Abbildung 67 Musterlösung

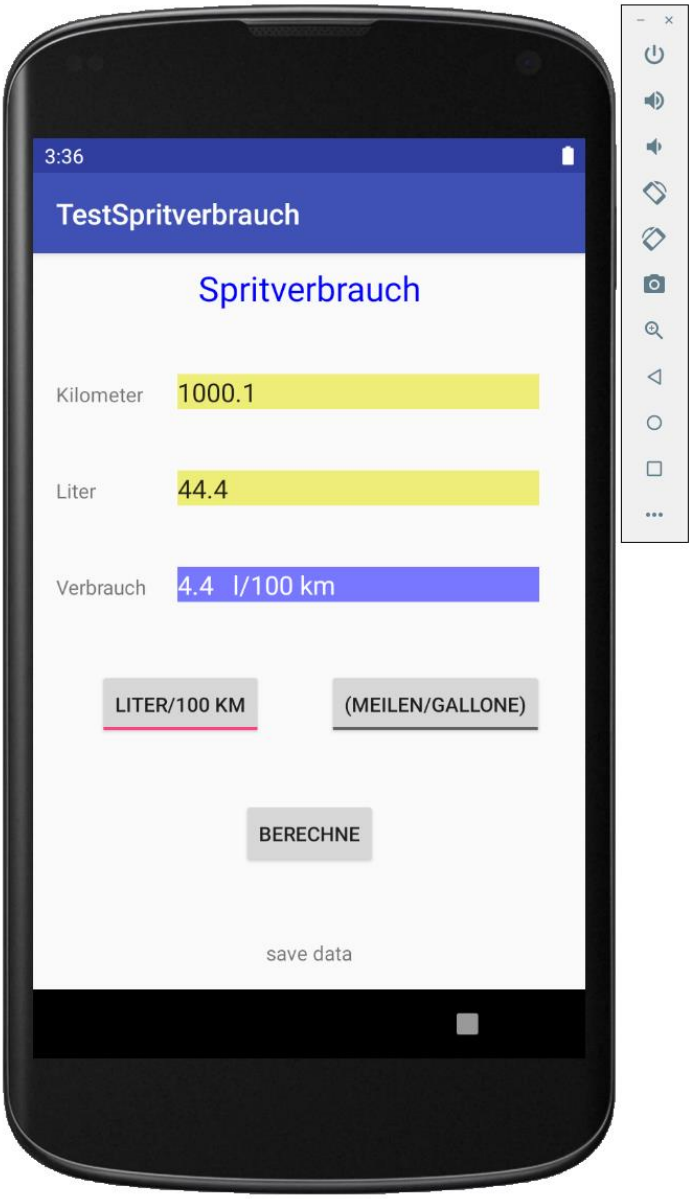


Abbildung 68 Berechnung nach der Berechnung Europa

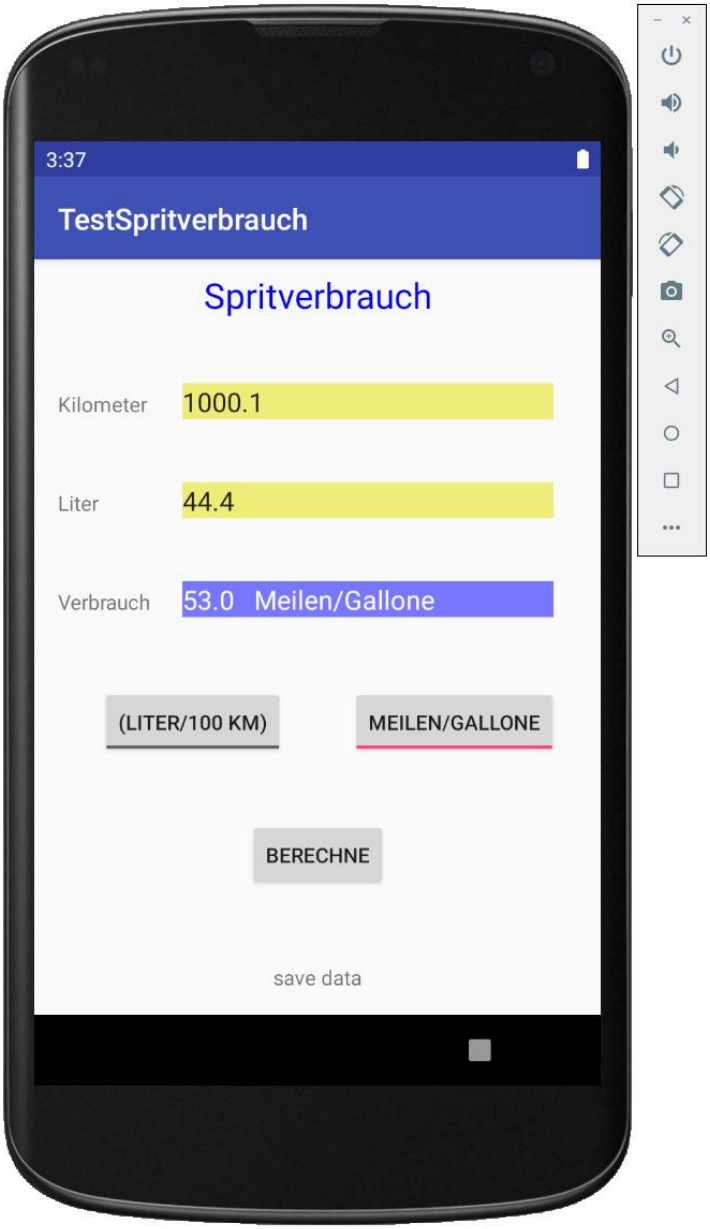


Abbildung 69 Berechnung nach der Berechnung USA

3.5.1 String-Konstanten

Im Baum-Eintrag „app/res/values/string.xml“ werden folgende String eingetragen. Dies hat den Vorteil dass man sie zentral verarbeiten kann. Damit kann man auch Internationalisierte App’s entwickeln.

```
<string name="title">Spritverbrauch</string>
<string name="kilometer">Kilometer</string>
<string name="liter">Liter</string>
<string name="comsuption">Verbrauch</string>
<string name="comsuption2">Verfahren</string>

<string name="comsuptionEuropa">Liter/100 km</string>
<string name="comsuptionEuropaOn">Liter/100 km</string>
<string name="comsuptionEuropaOff">(Liter/100 km)</string>

<string name="comsuptionUSA">Meilen/Gallone</string>
<string name="comsuptionUSAOn">Meilen/Gallone</string>
<string name="comsuptionUSAOff">(Meilen/Gallone)</string>
```

```
<string name="europa">Europa</string>
<string name="usa">USA</string>
<string name="Berechne">Starten</string>

<string name="wrongnumber">Fehlerhafte Zahl</string>
<string name="kilometergtnull">Die Kilometer müssen größer Null
sein</string>
<string name="litergtnull">Die Litermenge muss größer Null sein</string>
```

Anwendung:

```
android:text="@string/title"
```

3.5.2 Farb-Konstanten

Im Baum-Eintrag „app/res/values/colors“ werden folgende Farben eingetragen.

Neue Farbe:

```
<color name="colorTitle">#0000FF</color>
<color name="colorInput">#EEEE77</color>
<color name="colorOutput1">#7777FF</color>
<color name="colorOutput2">#FFFFFF</color>
```

Anwendung:

```
android:textColor="@color/colorTitle"
```

3.5.3 Spezifikation der UI-Elemente

TextView title:

- id @+id/title
- text string: title
- color colortitle
- textsize 24dp
- Position Zentriert

TextView Kilometer

- id @+id/tvkilometer
- text string: kilometer
- top parent title
- start 16dp linker Rand nur einmal definiert

EditText Kilometer

- id @+id/tkilometer
- text 1000.0
- start 24 dp
- end 32 dp rechter Rand nur einmal definiert
- color colorInput

- inputTyp decimal
- baseline tvkilometer
- start/end tvkilometer
- end/end parent

TextView Liter

- id @+id/tvliter
- text string: liter
- top parent tvkilometer
- start über tvkilometer

EditText Liter

- id @+id/tliter
- text 44,8
- color colorInput
- inputTyp decimal
- baseline tvliter
- start/start tkilometer
- end/end parent

TextView Verbrauch

- id @+id/ tvcomsuption
- text string: verbrauch
- top parent tvliter
- start über tvliter

EditText Verbrauch

- id @+id/ tcomsuption
- text -
- end 32 dp
- backgroundcolor colorInput1
- inputTyp decimal
- baseline tvcomsuption
- start/start tkilometer
- end/end parent

ToggleButton Europa:

- id @+id/ bneuropa
- start 16 dp
- end 32 dp
- text comsuptionEuropa
- textOn comsuptionEuropaOn
- textOff comsuptionEuropaOff
- start/start parent

- end/start bnusa
- end/end parent
- bias
- click bnEuropa_Click

ToggleButton Usa:

- id @+id/ bnusa
- margin start 16 dp
- text comsuptionUSA
- textOn comsuptionUSAOn
- textOff comsuptionUSAOff
- baseline bneuropa
-
- end/end parent
- bias
- click bnUSA_Click

ToggleButton Action:

- id @+id/ bnaction
- Position zentriert
- click bnAction_Click

TextView error

- id @+id/ tverror
- margin start 16 dp
- margin end 16 dp
- margin bottom 16 dp

3.5.4 Java-Code

Einbau der globalen Referenz-Variablen für:

- tKilometer
- tLiter
- tComsuption
- tverror
- bnEuropa
- bnUSA

Eintragen zusätzlicher globaler Variablen

- private boolean europa = true;
- private final String TAG = MainActivity.class.getSimpleName();
- private final String FILENAME = "patrol.bin";

- private DecimalFormat df = new DecimalFormat("##0.0");

Holen der Referenz-Variablen im Konstruktor:

- tKilometer
- tLiter
- tComsuption
- tverror
- bnEuropa
- bnUSA

Einbau des Click-Events für bnEuropa:

- Methode isChecked()
- Methode setChecked(…)
- Variable europa

Einbau des Click-Events für bnUSA:

- Methode isChecked()
- Methode setChecked(…)
- Variable europa

Einbau der Methode getNumber

```
private double getNumber(EditText tText) {
  try {
    String s = tText.getText().toString().replace(',','.');
    double zahl = Double.parseDouble(s);
    return zahl;
} catch (NumberFormatException e) {
    tComsuption.setText(getString(R.string.wrongnumber));
    return Double.NaN;
}
}
```

bnAction_Click programmieren:

- Holen der Kilometer (String)
  - Umwandeln in double
  - Fehlermeldung
- Holen der Liter (String)
  - Umwandeln in double
  - Fehlermeldung
- Ist ToggleButton Europa gesetzt?
  - Berechnen des Verbrauch
  - Ausgabe Verbrauch in Liter/100 km
  - tComsuption.setText(df.format(verbrauch)+" l/100 km");

- Ist ToggleButton USA gesetzt?
    - Berechnen des Verbrauch
    - 231 inch³ = 3,785411784 Liter
    - 1 km = 0.62136994937697 mi
    - 1 us-meile = 1.609347219 km
    - Ausgabe Verbrauch in Meilen/Gallone
    - tComsuption.setText(df.format(verbrauch)+" Meilen/Gallone");

3.5.5 Speichern der Eingabedaten

Folgende Daten werden in der bnAction-Click-Methode gespeichert:

- Kilometer
- Liter
- europa

3.5.6 Laden der Eingabedaten

Folgende Daten werden im Konstruktor geladen:

- Kilometer
- Liter
- europa

4 Einfache UI-Elemente

4.1 TextView (Label)

```
<TextView
  android:id="@+id/textview1"      ID
  android:layout_width="match_parent"    match_parent oder wrap_content
  android:layout_height="wrap_content"
  android:text="Eingabe"
  android:text="@string/button_finish"     aus Resource
  android:ellipsize="marquee"
/>
```

Eigenschaften:

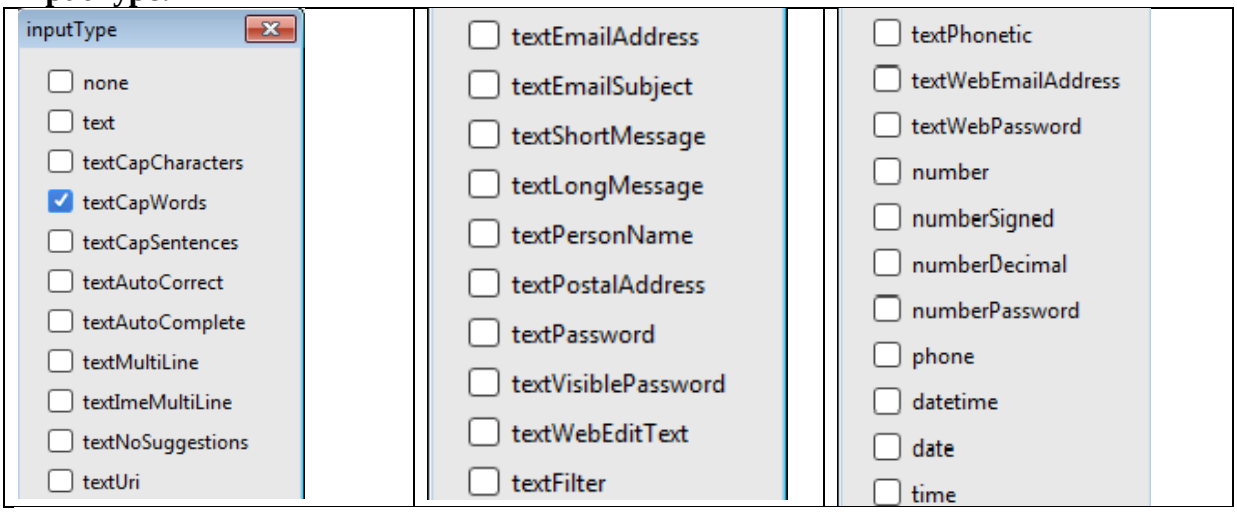
- fontFamily
- typeface normal, monospace
- **textSize** 14sp
- lineSpaceExtra none, 8sp
- textColor
- textAligment viewStart, textStart, center, textEnd, viewEnd
- Padding
- Margin
- background
- foreground
- lines
- minlines
- maxlines

4.2 EditText

```
<EditText
  android:id="@+id/edittext1"              ID
  android:layout_width="match_parent"       match_parent oder wrap_content
  android:layout_height="wrap_content"
  android:lines="1"
  android::textSize:"24sp"
  android:inputType="textCapWords"
  android:text="abc" />
```

In Android kann man die „Tastatur“ je nach Anwednungsfall bestimmen:

inputType:




```
android:inputType="textMultiLine"
```

Eigenschaften:

hint **selectAllOnFocus** fontFamily typeface normal, monospace textSize 14sp lineSpaceExtra none, 8sp textColor textAligment viewStart, textStart, center, textEnd, viewEnd Padding Margin	background foreground **singleLine** lines minlines maxlines numeric • integer • signed • decimal visibility

final EditText edittext1 = (EditText) findViewById(R.id.edittext1);

Eigenschaften:

- none
- text
- textCapCharacters
- textCapWords

- textCapSentences
    - textCapSentences | textAutoCorrect
- textAutoCorrect
- textAutoComplete
- textMultiLine
- textImeMultiLine
- textNoSuggestions
- textUri
- textEmailAddress
- textEmailSubject
- textShortMessage
- textLongMessage
- textPersonName
- textPostalAddress
- textPassword
- textVisiblePassword
- textWebEditText
- textFilter
- textPhonetic
- textWebEmailAddress
- textWebPassword
- number
    - Nur die Ziffern 0 bis 9 sind gültig
- numberSigned
    - Nur die Ziffern 0 bis 9 und das Minuszeichen sind gültig
- numberDecimal
    - Nur die Ziffern 0 bis 9 und der Punkt sind gültig
    - Kein Minuszeichen
- numberPassword
    - Zahlen 0 bis 9
- phone
    - Zahlen 0 bis 9
    - (/)-N*#+ space
    - Warten: ;
- datetime
    - Zahlen 0 bis 9
    - -/: space
- date
    - Zahlen 0 bis 9
    - -/ space
- time
    - Zahlen 0 bis 9:

4.2.1 Abfrage:

```
tInput.getText().toString()
```

4.2.2 ChangeListener

```
// import android.text.*;
private void setChangeListener(EditText edit) {
  edit.addTextChangedListener(new TextWatcher() {

    public void afterTextChanged(Editable s) {
      ergebnis.setText("editor wurde geändert");
    }

    public void beforeTextChanged(CharSequence s, int start,
        int count, int after) {
    }

    public void onTextChanged(CharSequence s, int start,
      int before, int count) {
    }
  });
}
```

Aufruf:

```
setChangeListener(ergebnis);
```

4.3 Button

```
<Button
  android:id="@+id/button1"
  android:text="Ok"
  android:layout_width="wrap_content"        match_parent oder
wrap_content
  android:layout_height="wrap_content"
/>
```

Attribut style:

Attribut	Aussehen
style="@style/Widget.AppCompat.Button.Borderless"	BUTTON
style="@style/Widget.AppCompat.Button.Borderless.Colored	BUTTON
style="@style/Widget.AppCompat.Button.ButtonBar.AlertDialog"	BUTTON
style="@style/Widget.AppCompat.Button.Small"	BUTTON

style

- Wigdet.Button
- Wigdet.Button.Inset
- Wigdet.Button.Small
- Wigdet.Button.Toggle
- Wigdet.Button.Transparent
- Wigdet.DeviceDefault.Button
- Wigdet.DeviceDefault.Button.Borderless

Eigenschaften:

background	lineSpaceExtra none, 8sp
foreground	elevation 8dp
text	visibility
fontFamily	**singleLine**
typeface normal, monospace	lines
textSize 14sp	minlines
textColor	maxlines
textAligment	Padding
viewStart, textStart, center, textEnd, viewEnd	Margin

```
final Button bn1 = (Button) findViewById(R.id.button1);
  bn1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      bn_Click();
   }
```

```
});
private void bn_click() {
  edittext1.setText( "abc"  );}
```

4.3.1 Abfrage:

```
if (button1.isChecked()) {
  // Action
}
```

4.3.2 ChangeListener

```
private void setOnClick(Button bn) {
  bn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      ergebnis.setText("Button wurde gedrueckt");
    }
  });
}
```

Aufruf:

```
setOnClick(bn);
```

4.3.3 Button mit Rand (Border)

Anklicken Eintrag „res/drawable“

Aufruf: Neue XML-Datei

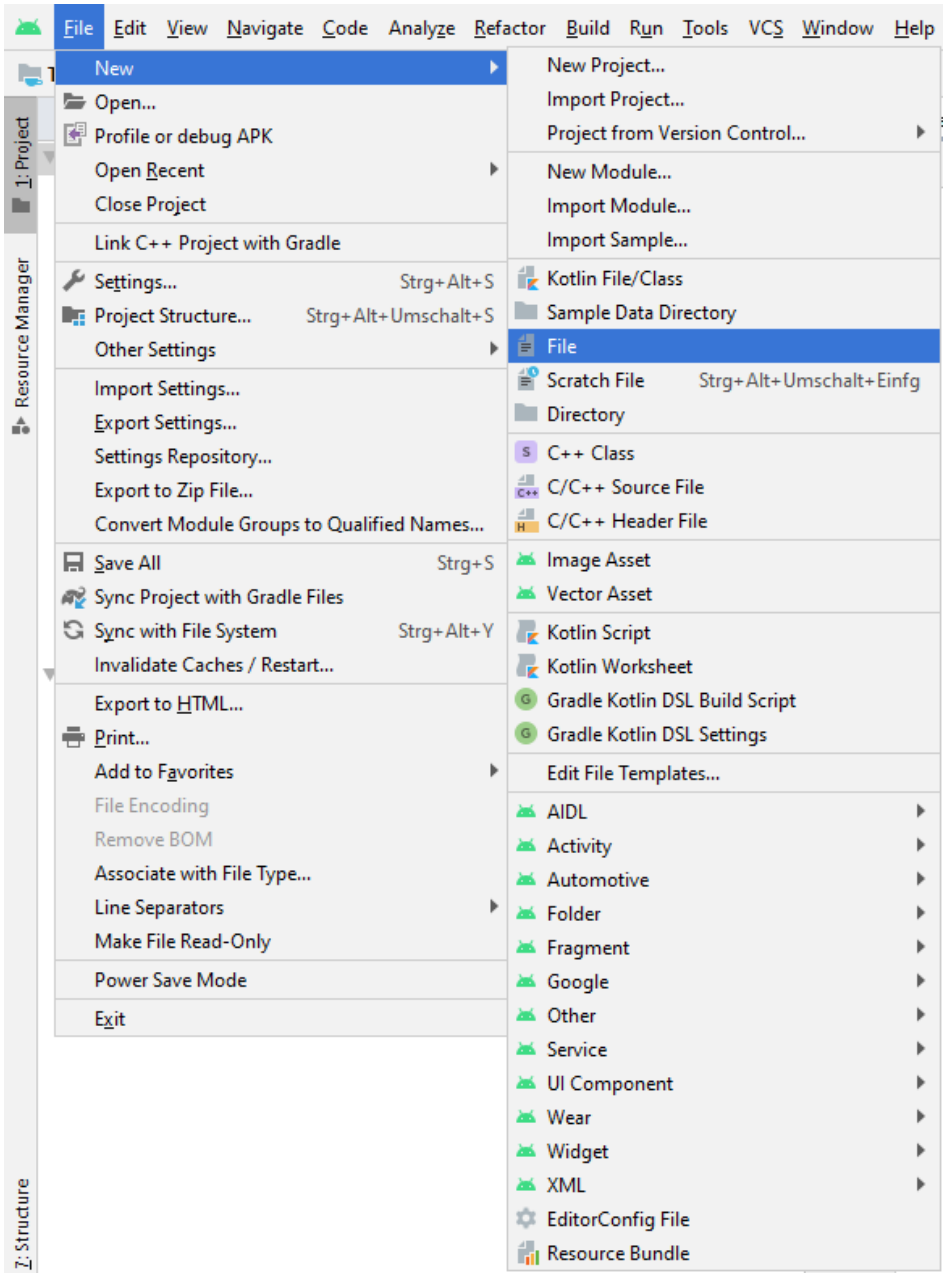
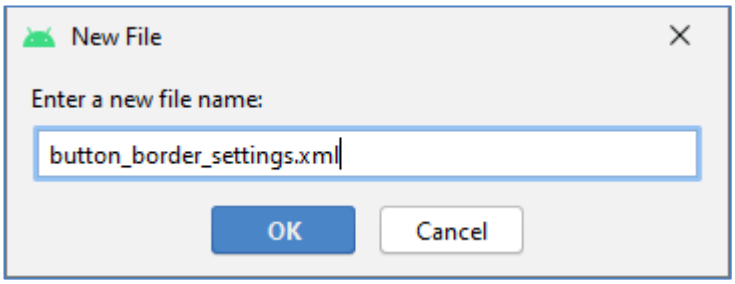


Abbildung 70 Neue XML-Date für Border anlegen



Dateiname: **button_border_settings.xml**

Inhalt:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item>
  <shape android:shape="rectangle">
    <solid android:color="#55dcf9"/>
    <stroke
```

```
       android:width="3dp"
       android:color="#fd0261" >
    </stroke>
  </shape>
</item>
</selector>
```

Die Datei wird im Ordner „layout“ eingetragen.

Eintragen in einem Button-Eintrag:

```
android:background="@drawable/button_border_settings"
```



Abbildung 71 Button mit rotem Rand

Alternative XM-Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <solid android:color="#55dcf9"/>
            <stroke
                android:dashGap="12dp"
                android:dashWidth="50dp"
                android:width="3dp"
                android:color="#fd0261" >
            </stroke>
        </shape>
    </item>
</selector>
```

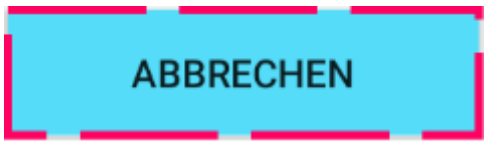


Abbildung 72 Alternatives Aussehen eines Buttons

4.4 CheckBox

```
 <CheckBox
  android:id="@+id/chk1"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:text="Ich bin eine CheckBox"
/>
```

Eigenschaften:

background foreground text	 visibility checked

fontFamily typeface normal, monospace textSize 14sp lineSpaceExtra none, 8sp textColor textAligment viewStart, textStart, center, textEnd, viewEnd Padding Margin	clickable onClick

```
final CheckBox chk1 = (CheckBox)findViewById(R.id.chk1);
```

4.4.1 Abfrage:

```
checkbox1.isChecked()
```

4.4.2 ChangeListener

```
private void setOnClick(Button bn) {
  bn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      ergebnis.setText("Button wurde gedrueckt");
    }
  });
}
```

Aufruf:

```
setOnClick(checkbox1);
```

4.5 RadioButton

```
<RadioButton
  android:id="@+id/rbAdd"
  android:layout_width="match_parent"      match_parent oder wrap_content
  android:layout_height="wrap_content"
  android:text="Add"
/>
```

Eigenschaften:

- background
- foreground
- text
- fontFamily
- typeface normal, monospace
- textSize 14sp
- lineSpaceExtra none, 8sp
- textColor

- textAligment
- viewStart, textStart, center, textEnd, viewEnd
- Padding
- Margin

- visibility
- checked
- clickable

- onClick

RadionButtons brauchen eine logische Gruppe, da es ja auch mehrere RadioButton-Gruppen in einem Fenster geben kann.

Layout-Code:

```
<RadioGroup
    android:id="@+id/rbgroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <RadioButton
        android:id="@+id/rbAdd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add"
        android:checked="true" />
    <RadioButton
        android:id="@+id/rbSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sub"
        />
</RadioGroup>
```

Eigenschaften:
- android:orientation="horizontal"
- android:orientation="vertical"

4.5.1 Abfrage:

```
final RadioGroup rbgroup = (RadioGroup)findViewById(R.id.rbgroup);
bn1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        switch (rbgroup.getCheckedRadioButtonId()) {
            case R.id.rbAdd:
                edittext1.setText("add is selected");
                break;
            case R.id.rbSub:
                edittext1.setText("sub is selected");
                break;
            default:
                edittext1.setText("weder add noch sub is selected");
```

```
                    break;
            }
        }
});
```

4.5.2 ChangeListener

```
private void setOnClick(Button bn) {
  bn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      ergebnis.setText("Button wurde gedrueckt");
    }
  });
}
```

Aufruf:

```
setOnClick(radiobutton1);
```

4.6 Switch

```
 <Switch
  android:id="@+id/switch1"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
/>
```

Eigenschaften:

- background
- foreground
- text
- fontFamily
- typeface normal, monospace
- textSize 14sp
- lineSpaceExtra none, 8sp
- textColor
- textAligment
- viewStart, textStart, center, textEnd, viewEnd

- visibility
- checked
- clickable
- Padding
- Margin

- onClick

Hinweis:
text wird links vom Switch geschrieben !

Java-Code:

```
final switch1 = (Switch)findViewById(R.id.switch1);
```

4.6.1 Abfrage

```
if (switch1.isChecked()) {
  // Action
}
```

4.6.2 ChangeListener

```
private void setOnClick(Button bn) {
  bn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
      ergebnis.setText("Button wurde gedrueckt");
    }
  });
}
```

Oder

Aufruf:

```
setOnClick(switch1);
```

4.7 Spinner

```
 <Spinner
  android:id="@+id/spinner1"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:prompt="hier ist ein Spinner"
/>
```

spinnermode:

- none
- dialog à la JList, die auf dem Bildschirm erscheint.
- dropdown wie normale ComboBox

Eigenschaften:

background foreground fontFamily typeface normal, monospace textColor textAligment viewStart, textStart, center, textEnd, viewEnd	visibility clickable Padding Margin onClick

Zu einem Spinner gehört eine Liste mit Einträgen. Diese kann man auf zwei verschiedene Arten definieren:

- Als Resource in der Datei „string.xml“
- Per Java-Code, sinnvoll, wenn man es aus einer Datenbak holt.

Liste in string.xml

```
<resources>
    <string-array name="semester_arrays">
        <item>1. Semester</item>
        <item>2. Semester</item>
        <item>3. Semester</item>
        <item>4. Semester</item>
        <item>5. Semester</item>
        <item>6. Semester</item>
        <item>Bachelorarbeit</item>
    </string-array>
</resources>
```

activity.xml

```
<Spinner
    android:id="@+id/sp_semester"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/textview3"
    android:layout_alignBaseline="@id/textview3"
    android:spinnerMode="dropdown"
    android:prompt="@string/semester_student"
     android:entries="@array/semester_arrays"
    android:layout_alignLeft="@id/matrnr"
/>
```

Java-Variante:

```
// Spinner
import java.util.ArrayList;
import java.util.List;
import android.widget.ArrayAdapter;
```

```
spinner1 = (Spinner) findViewById(R.id.spinner1);
List<String> list = new ArrayList<String>();
  list.add("Android");
  list.add("IOS");
  list.add("Swing");
  list.add("WPF");
  list.add("Python TKinter");
ArrayAdapter<String> dataAdapter =
new ArrayAdapter<String> (this, android.R.layout.simple_spinner_item,list);

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdow
n_item);
spinner1.setAdapter(dataAdapter);
```

4.7.1 Abfrage:

```
Object obj = spinner1.getSelectedItem()
int index = spinner1. getSelectedItemPosition();
```

4.7.2 ChangeListener

```
// import android.view.View;
private void changeListener(Spinner spinner) {
  spinner.setOnItemSelectedListener(
    new AdapterView.OnItemSelectedListener() {
      @Override
      public void onItemSelected(AdapterView<?> parent, View view,
        int position, long id) {
          ergebnis.setText("Spinner-Eintrag wurde veraendert");
      }

      @Override
      public void onNothingSelected(AdapterView<?> parent) {
      }
  });
}
```

Oder

```
// import android.view.View;
private int spinnernr=0;
private void changeListener(Spinner spinner) {
  spinner.setOnItemSelectedListener(
    new AdapterView.OnItemSelectedListener() {
      @Override
      public void onItemSelected(AdapterView<?> parent, View view,
        int position, long id) {
```

```
            ergebnis.setText("Spinner-Eintrag wurde veraendert: "
              +(++spinnernr));
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
   });
}



private void changeListener(Spinner spinner) {
         spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
             @Override
             public void onItemSelected(AdapterView<?> parent, View view,
int position, long id) {
                 ergebnis.setText("Spinner-Eintrag wurde veraendert: "
                    +(++spinnernr));
             }

             @Override
             public void onNothingSelected(AdapterView<?> parent) {
             }
         });
    }
```

Aufruf:

```
changeListener(spinner1);
```

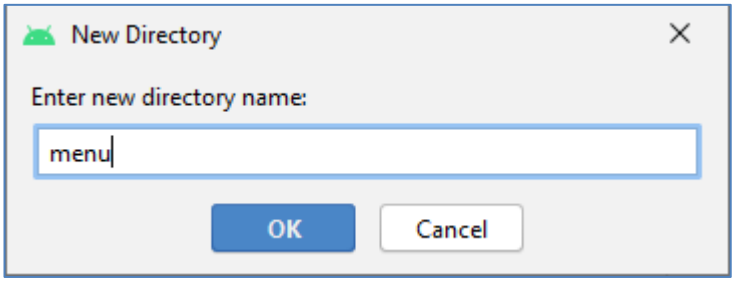
5 Menüs (drei Punkte)

Einfügen der Menüpunkte in string.xml:

```
<string name="new_item">Neu</string>
<string name="delete_item">Delete</string>
```

Anlegen eines Menüs

- Aktiviereen Sie den Eintrag „res“ im Projektbaum.
- Anlegen eines neuen Ordners „menu“.



Aktivieren des Eintrags „menu“
Anlegen einer neuen Datei

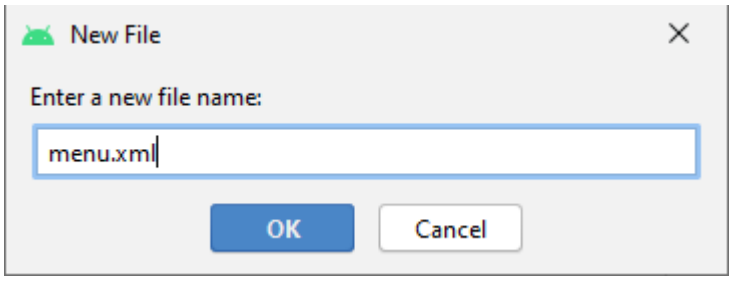


Abbildung 73 Anlegen einer Menüdatei

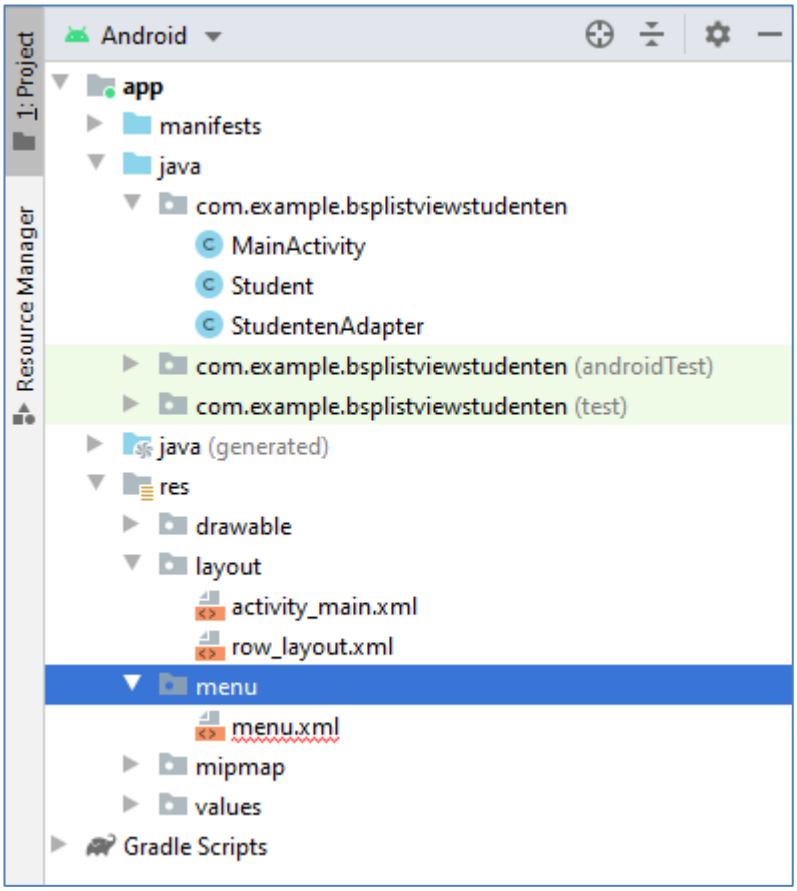


Abbildung 74 Aktueller Strukturbaum des Projektes

Eintragen des Menüs:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:id="@+id/menu_new_item"
    android:icon="@android:drawable/ic_menu_add"
    android:title="@string/new_item"
/>

  <item android:id="@+id/menu_delete_item"
    android:icon="@android:drawable/ic_menu_delete"
    android:title="@string/delete_item"
/>

</menu>
```

Als Icon wird ein internes Symbol „ic_menu_add“ verwendet.

Beim Testen erscheinen nun die bekannten Menüpunkte:

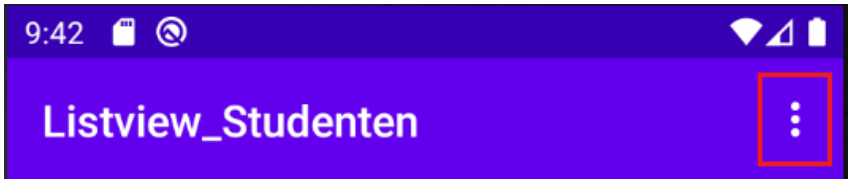


Abbildung 75 Anzeige eines Menüs

Beim Aklicken erscheinen deie Menüs (hier natürlich nur ein Menü):

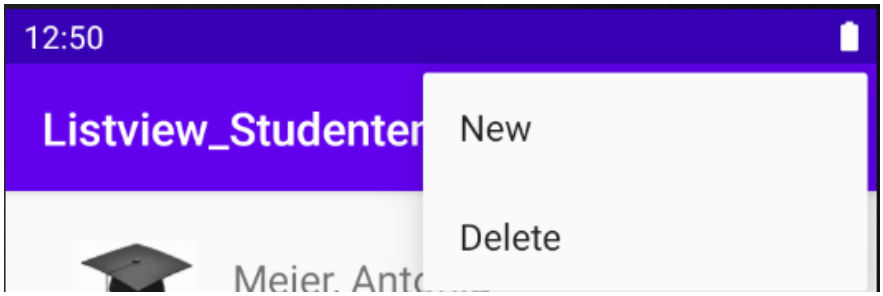


Abbildung 76 Menü New und Delete

Neuer Quellcode im Hauptprogramm:

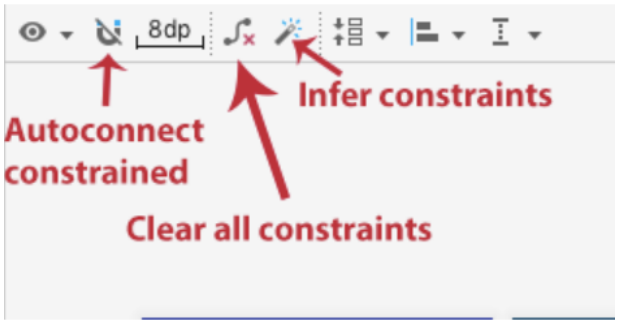
```
// import android.app.ActionBar;
// globale Variable
private Menu menu = null;

@Override
protected void onStart() {
  super.onStart();
  ActionBar ab = getActionBar();
  if (ab != null) {
    ab.setDisplayHomeAsUpEnabled(true);
  }
}

// Erzeugen der Menüs
@Override
  public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    this.menu = menu;  // um die id's zu überpüfen
    return super.onCreateOptionsMenu(menu);
}

  @Override
  public boolean onOptionsItemSelected(MenuItem mnItem) {
    if (mnItem.getItemId()==this.menu.getItem(0).getItemId()) {
      Toast.makeText(this, "Menü new angeklickt",
          Toast.LENGTH_SHORT).show();
    }
    else if  (mnItem.getItemId()==this.menu.getItem(1).getItemId()) {
      Toast.makeText(this, "Menü delete angeklickt",
          Toast.LENGTH_SHORT).show();
    }
    else {
      Toast.makeText(this, "keinMenue angeklickt: "+mnItem.getTitle(),
      Toast.LENGTH_SHORT).show();
    }
    return true;
}
```

6 Constraint-Layout



6.1 Attribute

6.1.1 layout_width

Werte:

- 0dp
- wrap_content — nur die Breite die notwendig ist
- match_parent — komplette Breite
- @+id/tinput — die Breite des angegebenen UI-Elementes

6.1.2 layout_height

Werte:

- wrap_content — nur die Höhe die notwendig ist
- match_parent — komplette Höhe
- @+id/tinput — die Höhe des angegebenen UI-Elementes

6.2 Konstanten

Jeweils OHNE Leerzeichen

6.2.1 Positionskonstanten

Attribut	Beschreibung
layout_constraint**Baseline_toBaselineOf**	Die Texte beider Elemente liegen auf einer Höhe. app: layout_constraint**Baseline_toBaselineOf** = "@+id/buttonA"
layout_constraint**Start_toEndOf**	Der Start des aktuellen Elementes ist bestimmt durch das Ende des „EndOf"-Elementes
layout_constraint**Start_toStartOf**	Die x-Koordinate ist beiden beiden gleich (linksbündig)

layout_constraint**End_toStartOf**	Das Ende des aktuellen Elementes ist bestimmt durch den Anfang des „StartOf“ –Elementes
layout_constraint**End_toEndOf**	Das Ende des aktuellen Elementes ist bestimmt durch das Ende des „EndOf“ (rechtsbündig) -Elementes app:layout_constraintEnd_toEndOf="parent"

6.2.2 Relative positioning

Attribut	**Beschreibung**
layout_constraint**Left_toLeftOf**	Beide Elemente haben eine gemeinsame linke Linie (linksbündig) app:layout_constraintLeft_toLeftOf = "@+id/buttonA"
layout_constraint**Left_toRightOf**	Das Element liegt rechts von angegeben Element app:layout_constraintLeft_toRightOf = "@+id/buttonA"
layout_constraint**Right**_toLeftOf	Das Element liegt links von angegeben Element app:layout_constraintRight_toLeftOf = "@+id/buttonA"
layout_constraint**Right**_toRightOf	Beide Elemente haben eine gemeinsame rechts Linie (rechtsbündig)
layout_constraint**Top_toTopOf**	Beide Elemente haben den gleichen Abstand zum oberen parent.
layout_constraint**Top_toBottomOf**	Die untere Line des Elements liegt an der oberen Linie des parent.
layout_constraint**Bottom_toTopOf**	Die obere Line des Elements liegt an der unteren Linie des parent.
layout_constraint**Bottom_toBottomOf**	Beide Elemente haben den gleichen Abstand zum unten parent.

6.2.3 Margin

Attribut	**Beschreibung**
android:layout_marginStart	
android:layout_marginEnd	
android:layout_marginLeft	
android:layout_marginTop	
android:layout_marginRight	
android:layout_marginBottom	

6.2.4 Bias (Zentriert)

Attribut	**Beschreibung**
layout_constraintHorizontal_bias	
layout_constraintVertical_bias	

Beispiel:

```
<android.support.constraint.ConstraintLayout ...>
  <Button android:id="@+id/button" ...
    app:layout_constraintHorizontal_bias="0.3"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent/>
</>
```

6.2.5 Winkel Positionen

Attribut	Beschreibung
layout_constraintCircle	references another widget id
layout_constraintCircleRadius	the distance to the other widget center
layout_constraintCircleAngle	which angle the widget should be at (in degrees, from 0 to 360)

Beispiel:

```
<Button android:id="@+id/buttonA" ... />
<Button android:id="@+id/buttonB" ...
  app:layout_constraintCircle="@+id/buttonA"
  app:layout_constraintCircleRadius="100dp"
  app:layout_constraintCircleAngle="45" />
```

6.3 Weitere Eigenschaften

6.3.1 Ratio

```
<Button android:layout_width="wrap_content"
  android:layout_height="0dp"
  app:layout_constraintDimensionRatio="1:1" />
```

Der Wert kann zwei Ausdrücke annehmen:

- ein float-Wert, der das Verhältnis von Breite und Höhe definiert.
  - 1.777777 (entspricht 16:9)
- Ein Verhältnisangabe in der Form "width:height"
  - 16:9

You can also use ratio if both dimensions are set to MATCH_CONSTRAINT (0dp). In this case the system sets the largest dimensions that satisfies all constraints and maintains the aspect ratio specified. To constrain one specific side based on the dimensions of another, you can pre append W," or H, to constrain the width or height respectively. For example, If one dimension is constrained by two targets (e.g. width is 0dp and centered on parent) you can indicate which side should be constrained, by adding the letter W (for constraining the width) or H (for constraining the height) in front of the ratio, separated by a comma:

```
<Button android:layout_width="0dp"
```

```
  android:layout_height="0dp"
  app:layout_constraintDimensionRatio="H,16:9"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintTop_toTopOf="parent"/>
```

6.4 Groups

Gruppen bilden einen virtuellen Container von beliebigen Elementen. Das gibt es auch in PhotoShop, MSPaint etc.

Beispiel:

```
<Button
  android:id="@+id/button15"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Button"
  tools:layout_editor_absoluteX="44dp"
  tools:layout_editor_absoluteY="28dp"
/>

<Button
  android:id="@+id/button16"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Button"
  tools:layout_editor_absoluteX="89dp"
  tools:layout_editor_absoluteY="118dp"
/>

<android.support.constraint.Group
  android:id="@+id/group"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:visibility="invisible"
  app:constraint_referenced_ids="button16,button15"
/>
```

6.4.1 Barirers

6.4.2 Prozent

```
app:layout_constraintWidth_default="percent"
app:layout_constraintWidth_percent="0.8"
```

6.4.3 fillparent

Das Andocken nach unten bis zum unteren Rand ist beim Constraint-Layout etwas komplizierter:

- Sicherstellen, dass der „layout_constraintTop_toBottomOf“ korrekt gesetzt ist.
- Setzen von layout_width
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein
- Setzen von layout_height
  - match_parent darf man nicht benutzen
  - stattdessen trägt man **0dp** ein

```
<EditText
  android:id="@+id/editText2"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:layout_marginStart="8dp"
  android:layout_marginTop="16dp"
  android:layout_marginEnd="8dp"
  android:layout_marginBottom="8dp"
  android:textSize="24sp"
  android:gravity="start|top"
  android:inputType="textMultiLine"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.0"
  app:layout_constraintVertical_bias="0.0"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toBottomOf="@+id/button"
/>
```

6.4.4 Tipps

6.4.4.1 Baseline

Mit dem Constraint „baseline“ werden Elemnete, vorzugsweise TextView und EditText auf eine Höhe bezüglich der Schrift gebracht.

Text direkt eintragen:

Man kann direkt in den Text den Text eintragen:
app:layout_constraintBaseline_toBaselineOf="@+id/tvzahl"

Der Eintrag beschriebt, dass die Schrift des EditTexts auf der gleichen Höhe wir das Element mit der Id „tvzahl“ ist.

Grafisches Eintragen

- Beim EditText die rechte Maus anklicken
- Auswahl des Eintrags „show baseline“
  - Es erscheint ein länglicher Zylinder
- Mit der linken Maus den Zylinder anklicken

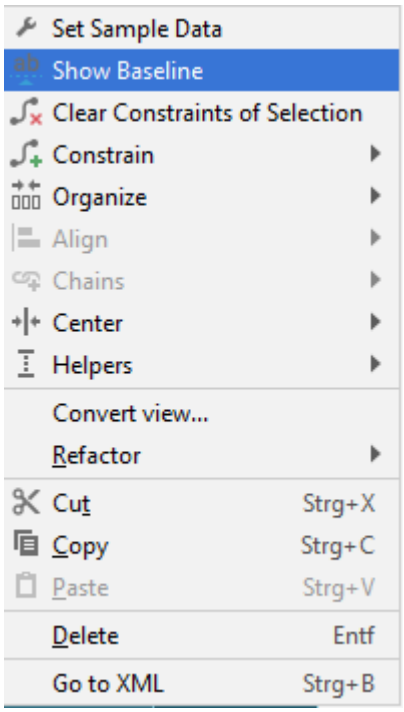


Abbildung 77 Menü for das Anzeigen der Baseline



Abbildung 78 Anzeige der Baseline



Abbildung 79 Verknüpfen der beiden Baseline

Hinweis:

- Bei einem Spinner wird die BaseLine erst dann angezeigt, wenn dem Spinner ein Array zugewiesen wurde.
- Tipp:
  - Einfach ein Dummy-Array in den String deklarieren.
  - `<string-array name="dummy_arrays">`
  - `<item>dummy</item>`
  - `</string-array>`

6.4.5 Unterschiedliche Länge der Beschriftungen

Der EditText von tzahl2 soll auf der gleichen x-Position sein wie tzahl1. Dazu muss man drei Eintragungen vornehmen:

- Der Start von tzahl2 ist identisch mit tzahl1
- Das Ende von tzahl2 ist identisch mit tzahl1
- **Die Breite von tzahl1 ist 0dp!**

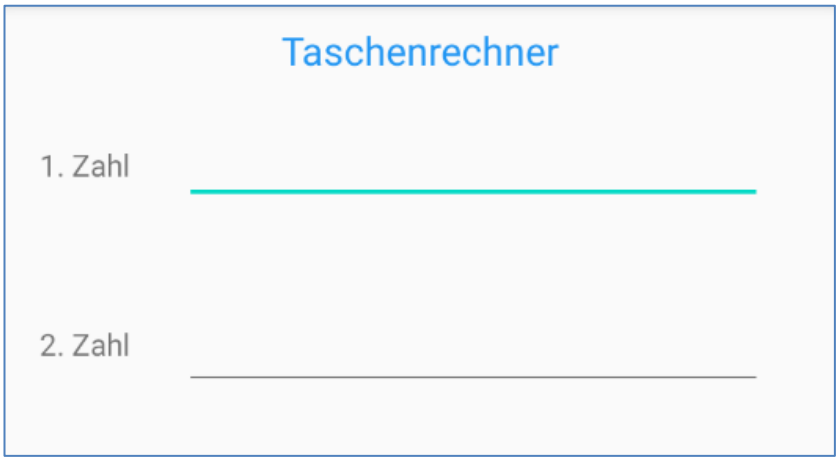


Abbildung 80 Gleiche Länge von Beschriftungen

Die beiden Grenzen, Start und Ende, kann man mit Drag&Drop vornehmen.
Die Breite, layout_width, trägt man manuell in den Eigenschaften ein.

EditText zahl1:

```
<EditText
   android:id="@+id/tzahl1"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginStart="24dp"
   android:layout_marginTop="16dp"
   android:layout_marginEnd="32dp"
   android:inputType="numberDecimal"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toEndOf="@+id/textview1"
   app:layout_constraintTop_toBottomOf="@+id/textview0"
/>
```

EditText zahl2:

```
<EditText
   android:id="@+id/tzahl2"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginStart="24dp"
   android:layout_marginTop="41dp"
   android:inputType="numberDecimal"
   app:layout_constraintEnd_toEndOf="@+id/tzahl1"
   app:layout_constraintStart_toEndOf="@+id/textview2"
   app:layout_constraintTop_toBottomOf="@+id/tzahl1"
/>
```

Wenn man nun eine Beschriftung verlängert, so kommt es zu einer Verschiebung:

```
android:text="1. Zahl wichtig"
```

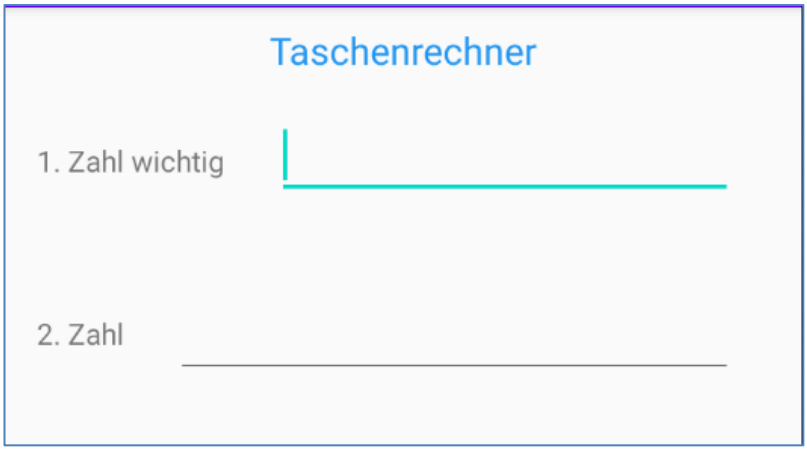


Abbildung 81 Unterschiedliche Länge von Beschriftungen

Abhilfe:

- In der unteren EditText muss man den linken Abstand zum TextView „2. Zahl“ entfernen, da sonst die Darstellung fehlerhaft ist.
- Dann muss man eine Regel einbauen, die den linken Rand von „tzahl2“ verknüpft mit „tzahl1“.

```
<EditText
   android:id="@+id/tzahl2"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginStart="24dp"                    löschen
   android:layout_marginTop="41dp"
   android:layout_marginEnd="32dp"
   android:inputType="numberDecimal"
   app:layout_constraintStart_toStartOf = "@+id/tzahl1"      neu
   app:layout_constraintEnd_toEndOf="@+id/tzahl1"
   app:layout_constraintStart_toEndOf="@+id/textview2"   löschen
   app:layout_constraintTop_toBottomOf="@+id/tzahl1"
/>
```

```
<EditText
   android:id="@+id/tzahl2"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginTop="41dp"
   android:inputType="numberDecimal"
   app:layout_constraintStart_toStartOf = "@+id/tzahl1"
   app:layout_constraintEnd_toEndOf="@+id/tzahl1"
   app:layout_constraintTop_toBottomOf= "@+id/tzahl1"
/>
```

6.4.5.1 Numberdecimal

Die EditText-Eigenschaft „numberDecimal“ definiert positive Dezimalzahl. Um auch negative Dezimalzahl zu bekommen, fügt man die binäre Konstante „numberSigned“ hinzu:

```
android:inputType="numberDecimal|numberSigned"
```

7 LinearLayout

7.1 vertikal

```
<?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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <TextView
        android:id="@+id/textview1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Eingabe"
        android:textAlignment="viewEnd" />

    <Button
        android:id="@+id/button1"
        android:text="Aktion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:elevation="0dp" />

</LinearLayout >
```

7.2 horizontal

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/activity_main"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textview2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Switch für das Ende der Vorlesung"
        android:textAlignment="viewEnd" />
    <Switch
        android:id="@+id/switch1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ssssssss" />

</LinearLayout>
```

- 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.

weightSum

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

    <Button
        android:text="left"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight=".70" />

    <Button
        android:text="right"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight=".30" />

</LinearLayout>
```

```
<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
    android:weightSum="30"
>
    <TextView    … />
     <EditText
        android:layout_weight="20"
        android:inputType="textMultiLine"  />
    <EditText
        android:layout_weight="10"
        android:inputType="textMultiLine"/>
```

```
</LinearLayout>
```

7.3 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.

7.4 RelativeLayout

Ähnlich dem Layout in iOS
Alle UI-Elemente orientieren sich am

- parent
- sibling

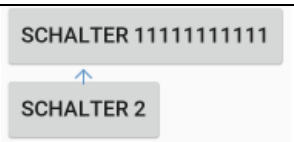
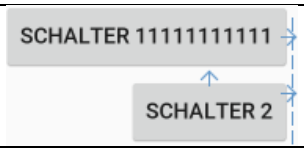
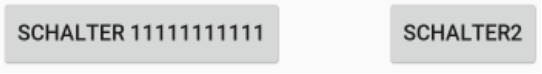
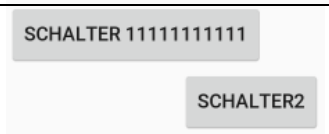
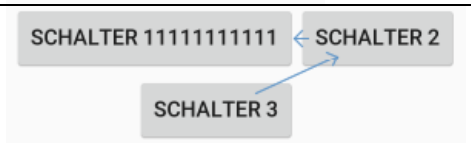
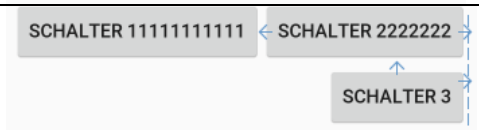
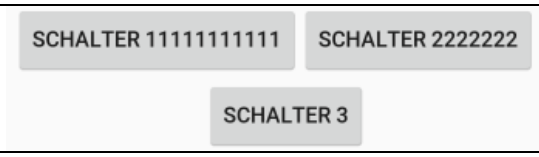
Die Verknüpfung geschieht durch die Id's

Attribut	T/F	ID/View	Beschreibung
layout_above		ID	Element ist oberhalb des Parents
layout_alignBaseline		ID	Element liegt auf der Höhe der Baseline des Parents.
layout_alignBottom		ID	Element positioniert sich auf der Baseline des Parents (ev. Unterschiedliche Höhen).
layout_alignEnd		ID	Element positioniert sich am Ende des Parents (ev. Unterschiedliche Längen).
layout_toEndOf		ID	Element liegt unterhalb vom Parent. Obere Ecke liegt an der unteren Ecke des Parents.
layout_toLeftOf		ID	Element liegt links vom Parent. Rechte Ecke liegt an der linken Ecke des Parents.
layout_toRightOf		ID	Element liegt rechts vom Parent. Linke Ecke liegt an der rechten Ecke des

Attribut	T/F	ID/View	Beschreibung
			Parents.
layout_toStartOf		ID	Element liegt oberhalb vom Parent. Untere Ecke liegt an der oberen Ecke des Parents.
layout_alignLeft		ID	Element positioniert sich auf der am Anfang des Parents (ev. Unterschiedliche Längen).
layout_alignRight		ID	Element positioniert sich anhand der Right-Koordinate des Parents.
layout_alignStart		ID	Element positioniert sich anhand der Startecke des Parents.
layout_alignTop		ID	Element positioniert sich anhand der Top-Koordinate des Parents.
layout_below		ID	Element ist unterhalb des Parents
layout_alignParentBottom	**T/F**	View	Element positioniert sich auf am Bottom-egde des Parents (ev. Unterschiedliche Höhen).
layout_alignParentEnd	**T/F**	View	Element positioniert sich auf der am Ende des Parents (ev. Unterschiedliche Längen).
layout_alignParentLeft	**T/F**	View	Element positioniert sich auf der am Anfang des Parents (ev. Unterschiedliche Längen).
layout_alignParentRight	**T/F**	View	Element positioniert sich auf der am rechten Rand des Parents (ev. Unterschiedliche Längen).
layout_alignParentStart	**T/F**	View	Element positioniert sich anhand der Startecke am Parent.
layout_alignParentTop	**T/F**	View	Element positioniert sich anhand der Top-Koordinate des Parents.
layout_alignWithParentIfMissing	**T/F**	View	Wenn der Anker für layout_toLeftOf, layout_toRightOf, etc. nicht gefunden werden kann, dient der Parent als Anker.
layout_centerHorizontal	**T/F**	View	Element ist horizontal zentriert bezüglich des Parents
layout_centerInParent	**T/F**	View	Element ist x/y zentriert bezüglich des Parent
layout_centerVertical	**T/F**	View	Element ist vertikal zentriert bezüglich des Parents

Attribut	T/F	Beschreibung
layout_above		Positions the bottom edge of this view above the given anchor view ID.
layout_alignBaseline		Positions the baseline of this view on the baseline of the given anchor view ID.
layout_alignBottom		Makes the bottom edge of this view match the bottom edge of the given anchor view ID.
layout_alignEnd		Makes the end edge of this view match the end edge of the given anchor view ID.
layout_alignLeft		Makes the left edge of this view match the left edge of the given anchor view ID.
layout_alignParentBottom	**T/F**	If true, makes the bottom edge of this view match the bottom edge of the parent.
layout_alignParentEnd	**T/F**	If true, makes the end edge of this view match the end edge of the parent.
layout_alignParentLeft	**T/F**	If true, makes the left edge of this view match the left edge of the parent.
layout_alignParentRight	**T/F**	If true, makes the right edge of this view match the right edge of the parent.
layout_alignParentStart	**T/F**	If true, makes the start edge of this view match the start edge of the parent.
layout_alignParentTop	**T/F**	If true, makes the top edge of this view match the top edge of the parent.
layout_alignRight		Makes the right edge of this view match the right edge of the given anchor view ID.
layout_alignStart		Makes the start edge of this view match the start edge of the given anchor view ID.
layout_alignTop		Makes the top edge of this view match the top edge of the given anchor view ID.
layout_alignWithParentIfMissing	**T/F**	If set to true, the parent will be used as the anchor when the anchor cannot be be found for layout_toLeftOf, layout_toRightOf, etc.
layout_below		Positions the top edge of this view below the given anchor view ID.
layout_centerHorizontal	**T/F**	If true, centers this child **T/F** horizontally within its parent.
layout_centerInParent	**T/F**	If true, centers this child horizontally and vertically within its parent.
layout_centerVertical	**T/F**	If true, centers this child vertically within its parent.
layout_toEndOf		Positions the start edge of this view to the end of the given anchor view ID.
layout_toLeftOf		Positions the right edge of this view to the left of the given anchor view ID.
layout_toRightOf		Positions the left edge of this view to the right of the given anchor view ID.
layout_toStartOf		Positions the end edge of this view to the start of the given anchor view ID.

7.4.1 Beispiele:

bn2: android:layout_below="@id/bn1"	
bn2: android:layout_above="@id/bn1"	Schalter bn2 liegt oberhalb der Viewgrenze
bn2: android:layout_below="@id/bn1" android:layout_alignRight="@id/bn1"	
android:layout_toRightOf="@id/bn1" android:layout_below="@id/bn1"	
bn2: android:layout_alignParentRight="true" rechtsbündig am VIEW	
bn2: android:layout_toRightOf="@id/bn1"	
bn2: android:layout_below="@id/bn1" android:layout_centerHorizontal="true"	
bn3: android:layout_toLeftOf="@id/bn2" android:layout_below="@id/bn2"	
bn3: android:layout_alignEnd="@id/bn2" android:layout_below="@id/bn2" Unterschied zu alignRight?	
bn2: android:layout_centerInParent="true"	Zentriert im View (vertikal und horizontal)
bn2: android:layout_centerVertical="true"	Zentriert vertikal im View Ausrichtung am linken Rand
bn2: android:layout_centerHorizontal="true"	Zentriert horizontal im View Ausrichtung am oberen Rand
bn3: android:layout_centerHorizontal="true" android:layout_below="@id/bn2" Zentriert horizontal im View	

8 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"

8.1 Beispiel TableLayout

```
<?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" >

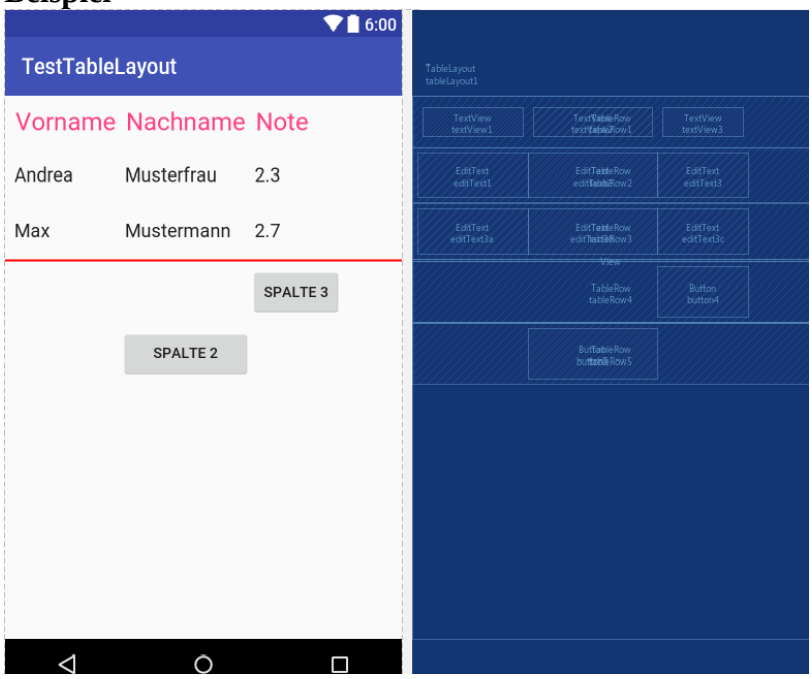
 <TableRow
   android:id="@+id/tableRow1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:padding="5dip" >

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

 </TableRow>

</TableLayout>
```

Beispiel




```
<?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
   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="5dp" >

  <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>
```

Gewichtungen mit android:layout_weight="1"

```
<TableRow
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <Button
    android:id="@+id/button3"
    android:layout_column="0"
    android:layout_span="1"
    android:layout_weight="1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />

  <Button
    android:id="@+id/button4"
    android:layout_column="1"
    android:layout_span="2"
    android:layout_weight="3"
    android:layout_width="155dp"
    android:layout_height="wrap_content"
    android:text="Button" />

</TableRow>
```

Aussehen:



9 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

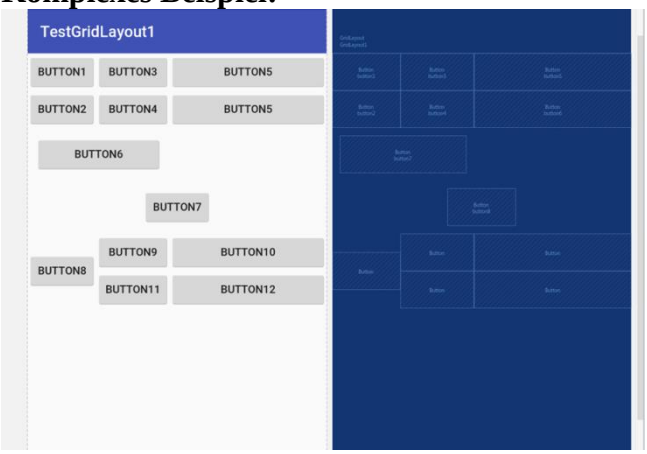
Aufbau:

```
<?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>
```

Komplexes Beispiel:



Quellcode:

```
<?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/button6"
        android:layout_column="2"
        android:layout_row="1"
        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>
```

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"

10 ListView

Eine Listview kapselt eine Menge von Daten in einer Tabelle. Die Zeilen können eine beliebige Struktur annehmen, da diese als „View“ deklariert werden können.

Eine ListView arbeitet nicht direkt mit den Daten. Man benötigt einen Adapter, der den Zugriff standardisiert organisiert.

Folgende Adapter existieren:

- SimpleAdapter
- CustomSimpleAdapter
- CustomArrayAdapter
- **BaseAdapter**
- ArrayAdapter
- **Cursor Adapter**

Intereressant sind nur zwei Adapter. Der Cursor-Adapter ist für Datenbanken zuständig. Der BaseAdapater reicht als Grundlage für eine ListView völlig aus. Damit ist man am flexibelsten.

Für ListView ist die einfachste Klasse der ArrayAdapter. Dieser erwartet eine beliebige Klasse und zeigt die Daten mittels der toString-Methode an.

Erstellen eines ArrayAdapters:
ArrayAdapter<MeineKlasse> arrayAdapter = new ArrayAdapter<MeineKlasse>(this, listview, row, array)

Der Adapter erhält per Parameter den Context der App, eine Referenz der ListView, die Deklaration der Zeile (eine XML-Datei) und die ArrayList oder Array.

Für komplexere Darstellungen ist der BaseAdapater besser geeignet. Die zu implementierenden Methoden sind als Interface definiert:

- int **getCount**()
- Object **getItem**(int position)
- long getItemId(int position) // nicht so wichtig
- View **getView**(int position, w(int position, View convertView, ViewGroup parent)

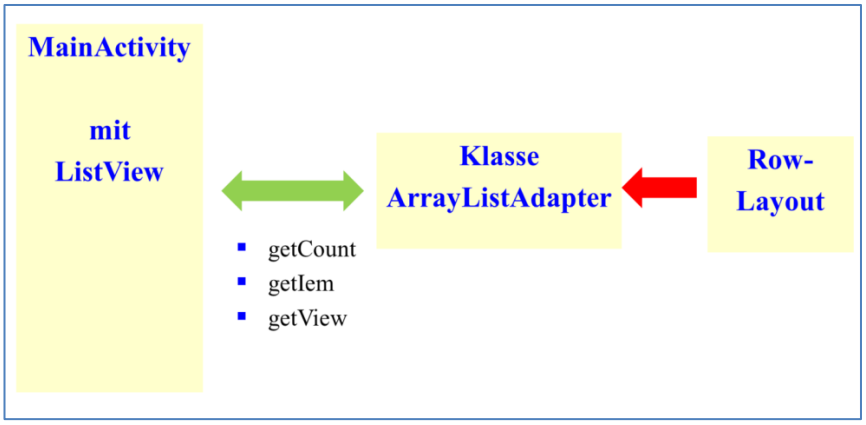


Abbildung 82 Überblick eines Adapters

Die drei wesentlichen Methoden werden über ein Interface im Adapter aufgerufen. Für die Darstelleung wird noch eine xml-Datei benötigt.

Der untere Quellcode zeigt die Deklaration einer Zeile für eine ListView. Für komplexere Darstellungen (Bilder, Symbole, mehrere Texte) benötigt man dementsprechend komplexere xml-Dateien.

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rowTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:textSize="16sp" >
</TextView>
```

10.1 Einfaches ListView-Beispiel

Dieses Beispiel verwendet eine ListView, eine einfache Zeile und ein TextArray.

Anlegen eines neuen Projektes:
Project Template Empty Activity
Name: BspListView01

10.1.1 activity_main.xml

- Die TextView löschen
- Mit Drag & Drop eine ListView komplett auf den View ziehen.
  - Register Legacy, Eintrag ListView
- Ränder: 4×16 dp
- id listview

Musterlösung:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Wichtige Eigenschaften:

- choiceMode singleChoice,multipeChoiceModal, None, multipeChoice
- clickable
- listfooter
- listheader
- longClickable
- listitem
- onClick
- orientation

10.1.2 Darstellung der Zeile

- Anklicken des Eintrags
  - res
  - layout
- Anlegen einer Datei

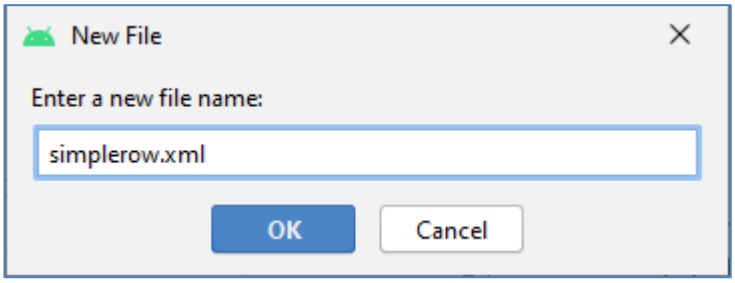


Abbildung 83 Anlegen der Datei simplerow.xml

Leeres row_layout mit Constriant-Layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="?android:attr/listPreferredItemHeight"
    android:padding="1dp"
    >

</androidx.constraintlayout.widget.ConstraintLayout>
```

Einfügen folgenden Textes:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/rowTextView"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:padding="10dp"
  android:textSize="16sp" >
</TextView>
```

10.1.3 Java Code

Import Anweisungen:

```
import android.widget.*;
import java.util.*;
```

Globale Variablen:

```
private ListView listview = null;
private String[] favoriteLanguages = new String[] {
     "Java", "Scala", "Kotlin", "Swift",
     "C", "C++", "C#", "Assembler",
     "Cobol",  "Fortran", "Java Script",
      "PHP",  "Avenue", "Delphi"
};

private ArrayAdapter<String> listAdapter;
```

Konstruktor:

```
   // holen der Referenz
listview = (ListView) findViewById(R.id.listview);

     // erstellen eines Adapters
listAdapter = new ArrayAdapter<String>( this,
                      R.layout.simplerow, favoriteLanguages);
listview.setAdapter(listAdapter);
```

Aufruf mit Shift F10:

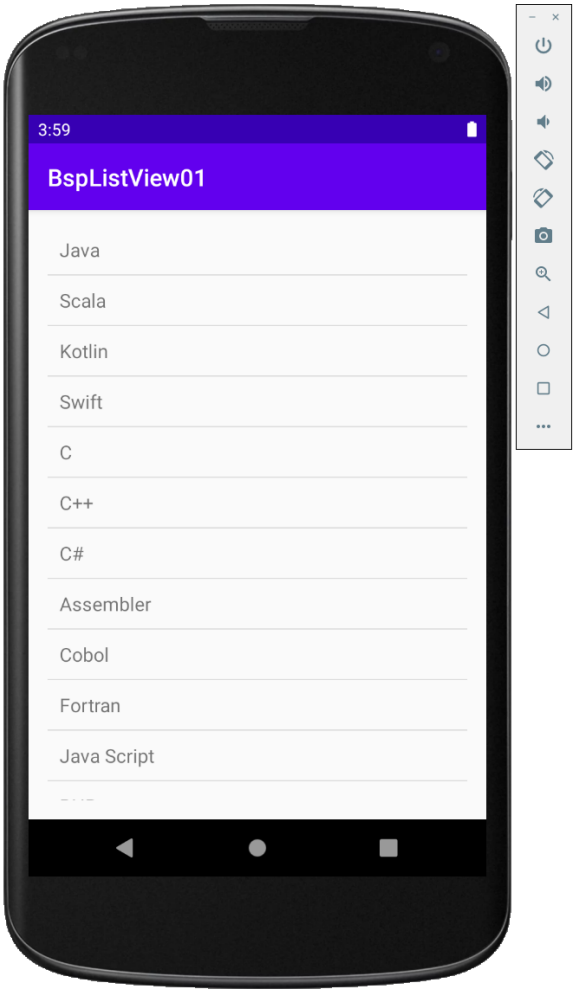


Abbildung 84 Einfache ListView mit einem ArrayAdapter

10.2 ListView-Beispiel mit Studenten-Adapter

Dieses Kapitel zeigt eine umfangreiche Implementierung einer ListView mit Hilfe eines BaseAdapters und einer eigenen Klasse.

Eigenschaften:

- View mit einer ListView
- Klasse Student
- Klasse Studentenliste
- StudentenAdapter aus BaseAdapter
- Umfangreiches row-Layout
- Menu für neue Einträge
- Speichern und Laden der Liste
- Aufrufen einer neuen „Seite“
- Parameterübergabe
    - Mit putExtraString etc.
    - Mit serialize

Anlegen eines neuen Projektes:
Project Template Empty Activity
Name: BspListViewStudentenSimple

10.2.1 activity_main.xml

- Die TextView löschen
- Mit Drag & Drop eine ListView komplett auf den View ziehen.
    - Register Legacy, Eintrag ListView
- Ränder: 4×16 dp
- id listview

Musterlösung:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Wichtige Eigenschaften:

- choiceMode singleChoice,multipeChoiceModal, None, multipeChoice
- clickable
- listfooter
- listheader
- longClickable
- listitem
- onClick
- orientation
- listSelector android:listSelector="@drawable/list_colors"

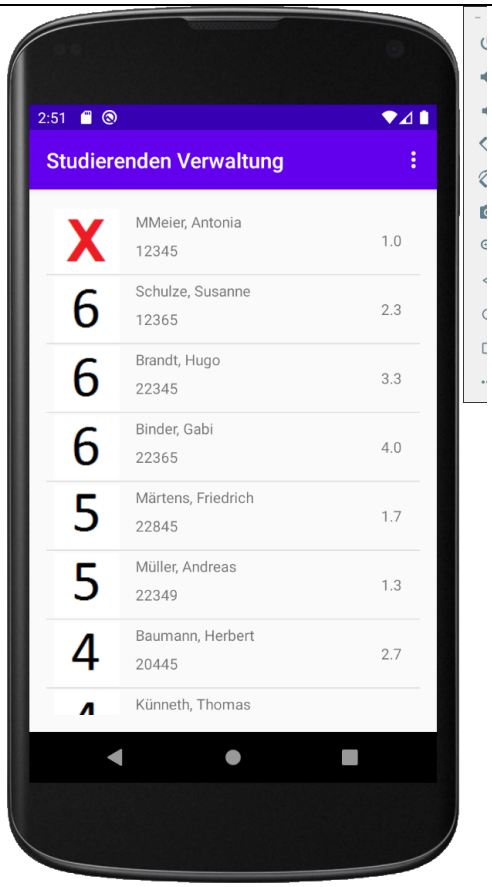


Abbildung 85 ListView StudentenListe

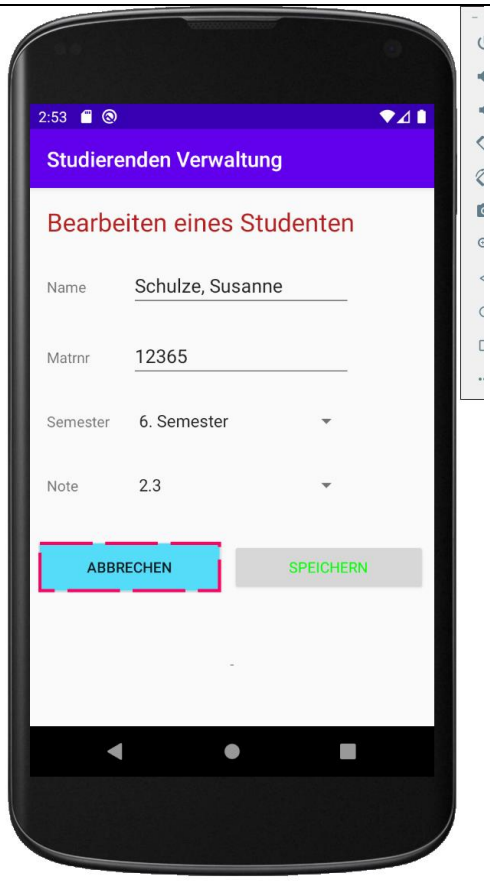


Abbildung 86 ListView, View zum Bearbeiten

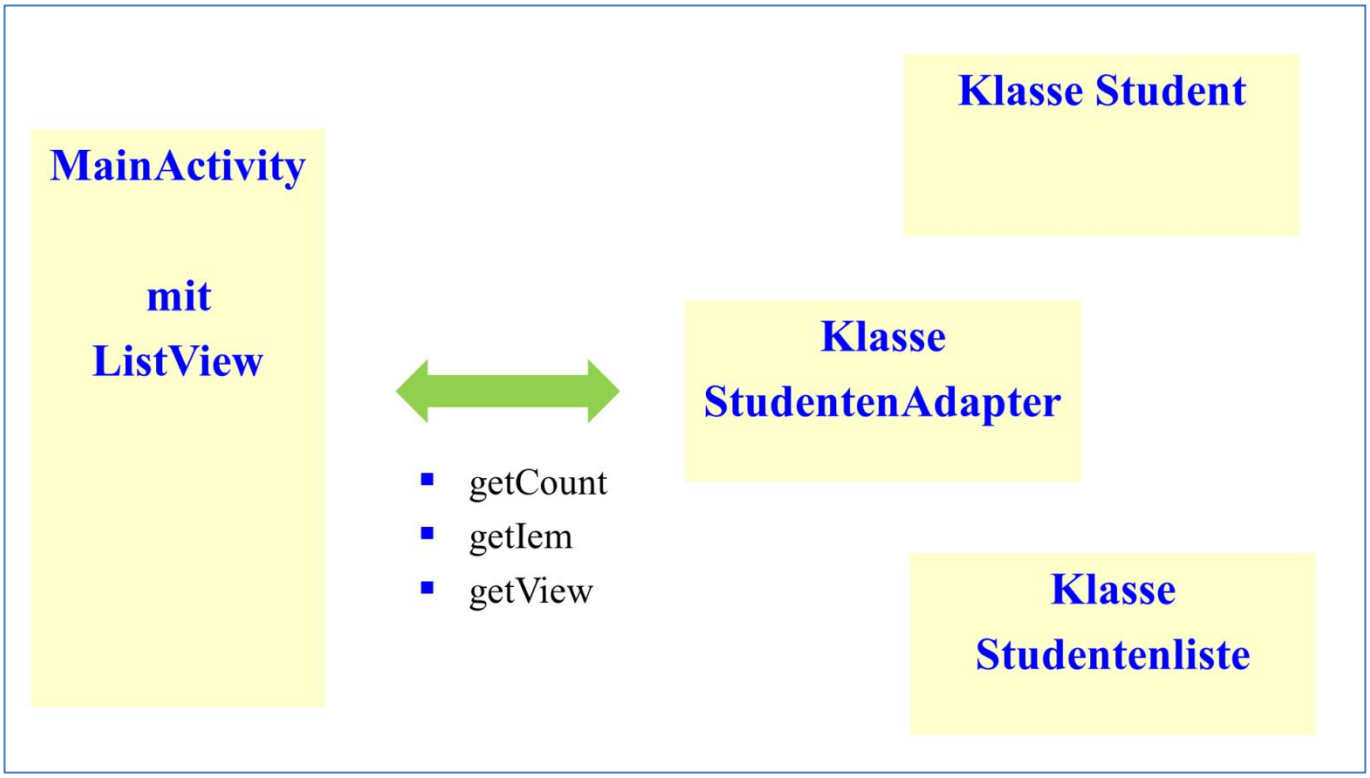


Abbildung 87 Struktur des Beispiels ListView mit Studenten

10.2.2 Strings Konstanten

Unten sind alle String für das Projekt eingetragen. Das Array für die Semester-Spinner ist auch eingetragen.

```
<resources>
    <string name="app_name">Studierenden Verwaltung</string>
    <string name="sem1">1. Semester</string>
    <string name="sem2">2. Semester</string>
    <string name="sem3">3. Semester</string>
    <string name="sem4">4. Semester</string>
    <string name="sem5">5. Semester</string>
    <string name="sem6">6. Semester</string>
    <string name="semx">>6. Semester</string>
    <string name="bachelor">Bachelorarbeit</string>

    <string name="hint">Status-Grafik</string>

    <string name="new_item">Neu</string>
    <string name="delete_item">Delete</string>
    <string name="lightstyle_item">Light Style</string>
    <string name="darkstyle_item">Dark Style</string>


    <string name="actiontitle_edit">Bearbeiten eines Studenten</string>
    <string name="actiontitle_delete">Löschen eines Studenten</string>
    <string name="actiontitle_new">Neuer Studenten</string>
    <string name="actiontitle_error">Fehlerhafte Aktion</string>

    <string name="spsemesterprompt">Bitte ein Semester auswählen</string>
    <string name="spnoteprompt">Bitte eine Note auswählen</string>
```

```
    <string name="tvstudent">Name</string>
    <string name="tvmatrnr">Matrnr</string>
    <string name="tvsemester">Semester</string>
    <string name="tvnote">Note</string>

    <string name="bndelete">Löschen</string>
    <string name="bnnew">Neu</string>
    <string name="bnsavechanges">Speichern</string>
    <string name="bncancel">Abbrechen</string>

    <string-array name="semester_arrays">
        <item>1. Semester</item>
        <item>2. Semester</item>
        <item>3. Semester</item>
        <item>4. Semester</item>
        <item>5. Semester</item>
        <item>6. Semester</item>
        <item>Bachelorarbeit</item>
        <item>Dauerstudent</item>
    </string-array>

</resources>
```

10.2.3 Darstellung der Zeile

- Anklicken des Eintrags
  - res
  - layout
- Anlegen einer Datei

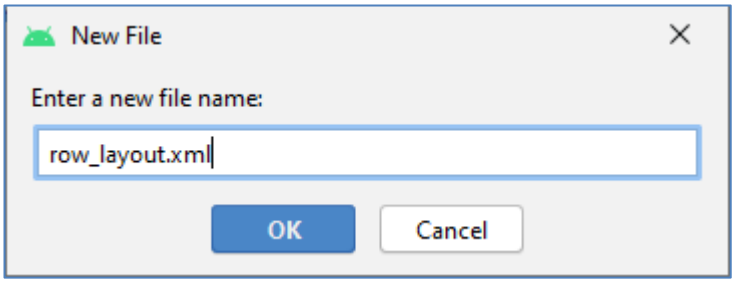


Abbildung 88 Anlegen der Datei row_layout.xml

Die Definition einer Zeiule wird meistens mit einem LinearLayout oder einem Relativ-Layout definiert.

Beispiel eines LinearLayouts:

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

  <TextView

    android:id="@+id/tvName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Name" />
```

```
  <TextView
    android:id="@+id/tvMnr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Matrikelnummer" />
</LinearLayout>
```

Das obere LinearLayoutstellet die beiden String nebeneinander dar. Für das Studentenbeispiel muss es komplizierter sein:

- Am linken Rand ein Semestersymbol
- In der Mitte zqei übereinanderliegende Strings (Name, Matrikelnr)
- Am rechten Rand eine Note

Einfügen folgenden Textes (**Constraint-Layout**):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="?android:attr/listPreferredItemHeight"
    android:padding="1dp"
    >

    <ImageView
        android:id="@+id/ivsemester"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="1dp"
        android:layout_marginTop="1dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/sem1" />

    <TextView
        android:id="@+id/tvname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:layout_marginTop="4dp"
        android:text="@string/tvstudent"

        app:layout_constraintStart_toEndOf="@+id/ivsemester"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvmatrnr"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:layout_marginTop="8dp"
        android:text="TextView"
        app:layout_constraintStart_toEndOf="@+id/ivsemester"
        app:layout_constraintTop_toBottomOf="@+id/tvname" />

    <TextView
```

```
        android:id="@+id/tvnote"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="16dp"

        android:layout_marginBottom="5dp"
        android:text="TextView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

10.2.4 Semesterbilder

Kopieren Sie die Bilder der Semester in das Projekt-Verzeichnis:
d:\BspListViewStudenten**app\src\main\res\drawable**

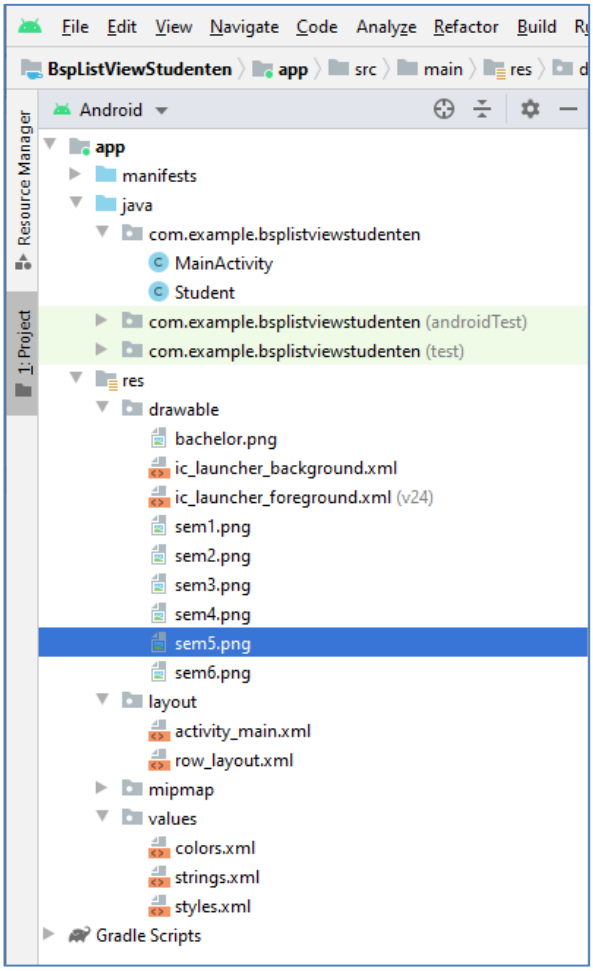


Abbildung 89 Aktuelle Struktur des Projektes

10.2.5 Klasse Student

- Eintrag MainActivity anklicken
- Neue Java Klasse

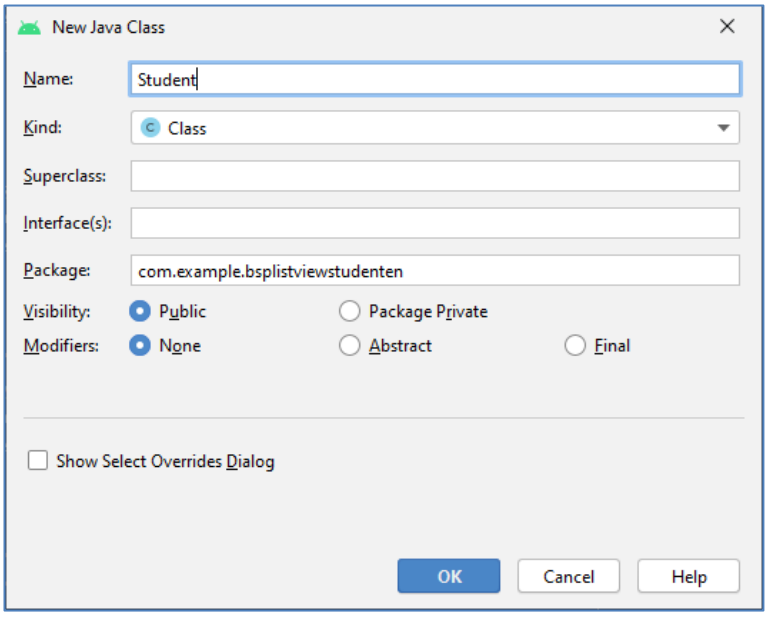


Abbildung 90 Klasse Student

10.2.6 Quellcode der Klasse Student

Eingetragen:

- Attribute
- Konstruktor
- Auswahl des Bildes (getIdForDrawable)

```
package com.example.bsplistviewstudenten;

import android.content.Context;
import android.content.Intent;
import android.util.Log;
import java.io.*;

public class Student implements Serializable{

    private static final String TAG =  MainActivity.class.getSimpleName();

    public String name="-";
    public int matrnr=1;
    public int semester=1;
    public double note=1.0;

    public Student(String name, int matrnr, int semester, double note) {
        this.name = name;
        this.matrnr = matrnr;
        this.semester=semester;
        this.note = note;
    }
    public Student() {
    }


    // je nach Semester muss das korrekte Symbol erscheinen
    public int getIdForDrawable() {
        switch (semester) {
```

```
            case 1:
                return R.drawable.sem1;
            case 2:
                return R.drawable.sem2;
            case 3:
                return R.drawable.sem3;
            case 4:
                return R.drawable.sem4;
            case 5:
                return R.drawable.sem5;
            case 6:
                return R.drawable.sem6;
            case 7:
                return R.drawable.bachelor;
            case 8:
                return R.drawable.semx;
            default:
                return R.drawable.sem1;
        }
    }  // getIdForDrawable

}
```

Es fehlen:

- Konstanten für die zweite View
- Methoden für die zweite View
- Methopden für das Laden und Speichern

10.2.7 StudentenListe

Neue Klasse „Studentenliste“

- Eintrag „MainActivity“ anklicken
- Menü File, Eintrag „New“, Eintrag „Java Class“

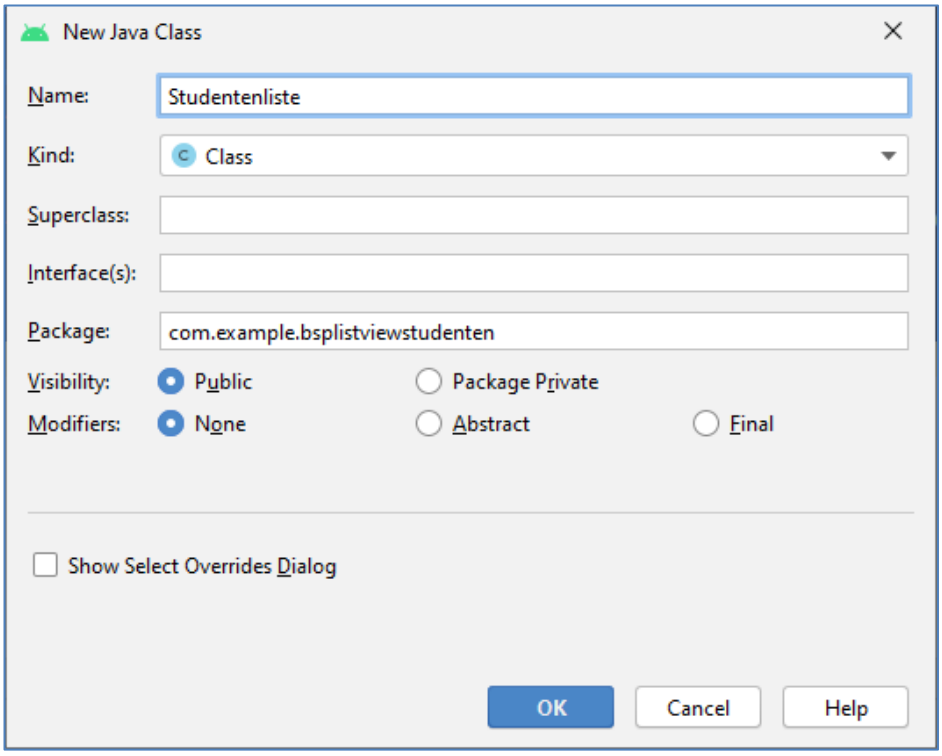


Abbildung 91 Klasse Studentenliste

Anlegen einer neuen Java-Klasse: Studentenliste
Diese Klasse wird als Singleton angelegt!

Die Studentenliste verwaltet die Daten der Studierenden wie eine Datenbank. Der ArrayAdapter darf nur die Liste als Referenz haben (eigentlich readonly). Am Besten wären also zwei Methoden:

- getCount()
- getStudent(int index)

Imports:

```
import android.content.Context;
import android.util.Log;

import java.io.*;
import java.util.ArrayList;
```

Globale Variablen / Singleton:

```
private static final String TAG = MainActivity.class.getSimpleName();

private final String FILENAME = "studenten.bin";
private static Studentenliste instance = new Studentenliste();

public static Studentenliste getInstance() {
   return instance;
}

// nicht sichtbar
private Studentenliste() {
}
```

```
    public ArrayList<Student> liste = new ArrayList<>();
    public Context context = null;  // für Laden und Speichern
```

Füllen der Liste der Studierenden:

```
private void initListe() {
   Log.i(TAG, "##############################  in initListe ");
   liste.clear();
   liste.add( new Student("Meier, Antonia", 12345, 8, 1.0)  );
   liste.add( new Student("Schulze, Susanne", 12365, 6, 2.3)  );
   liste.add( new Student("Brandt, Hugo", 22345, 6, 3.3)  );
   liste.add( new Student("Binder, Gabi", 22365, 6, 4.0)  );
   liste.add( new Student("Märtens, Friedrich", 22845, 5, 1.7)  );
   liste.add( new Student("Müller, Andreas", 22349, 5, 1.3)  );
   liste.add( new Student("Baumann, Herbert", 20445, 4, 2.7)  );
   liste.add( new Student("Künneth, Thomas", 324455, 4, 3.3)  );
   liste.add( new Student("Unbereit, Peter", 324355, 4, 2.7)  );
   liste.add( new Student("Meyer, Hubert ", 324355, 3, 2.0)  );
   liste.add( new Student("Altmeier, Jutta", 324355, 3, 3.3)  );
   liste.add( new Student("Gisin, Nicoletta", 33445, 2, 2.7)  );
   liste.add( new Student("Telemann, Dieter", 34645, 2, 5.0)  );
   liste.add( new Student("Wagner, Hans", 38345, 2, 4.0)  );
   liste.add( new Student("Wolff, Bernd", 39345, 1, 3.7)  );
   liste.add( new Student("Prandtl, Daria", 40345, 1, 1.3)  );
   liste.add( new Student("Burton, Michael", 41445, 1, 3.0)  );
   liste.add( new Student("Fehr, Hauke", 42445, 1, 2.0)  );
   liste.add( new Student("Klenke, Mario", 44445, 1, 3.7)  );
   Log.i(TAG,"************************* Modul Studenteliste, add");
}
```

10.2.8 StudentenAdapter

Erstellen einer neuen Java Klasse

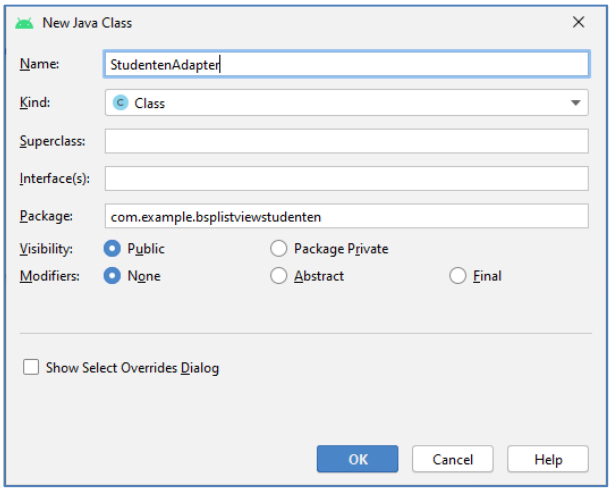


Abbildung 92 Neue Klasse StudnetenAdapter

Der Adpater ist von BaseAdpater abgeleitet und implementiert folgende Methoden:

- public int getCount()

- public Object getItem(int position)
- public long getItemId(int position)
- public View getView(int position, View convertView, ViewGroup parent)

Imports:

```
import android.content.Context;
import android.util.Log;
import android.view.*;
import android.widget.*;

import java.util.ArrayList;
```

Globale Variablen:

```
private static final String TAG = MainActivity.class.getSimpleName();

static class ViewHolder {
  ImageView ivsemester;
  TextView tvname;
  TextView tvmatrnr;
  TextView tvnote;
}

private ArrayList<Student> liste = new ArrayList<Student>();
private final LayoutInflater inflator;
```

Die Klasse **ViewHolder** dient dazu, Zeilen wieder zuverwenden und nicht beim Scrollen immer wieder eine neue Instanz der UI-Elemente zu erzeugen.

Konstruktor:

```
public StudentenAdapter(Context context) {
  inflator = LayoutInflater.from(context);
  Log.i(TAG, "Konstructor in StudentenAdapter: 2");

  Studentenliste studentenliste = Studentenliste.getInstance();
  //studentenliste.loadInput(context); // noch nicht aktiv

  liste = Studentenliste.getInstance().liste;
  Log.i(TAG, "Konstructor in StudentenAdapter: count: "+liste.size());

}
```

Methoden:

```
@Override
public int getCount() {
  return liste.size();
}

@Override
public Object getItem(int position) {
  return liste.get(position);
}

@Override
public long getItemId(int position) {
```

```
  return position;
}
```

Nun fehlt nur noch das Aufbauen des View pro Zeile

getView:

```
// hier baut man die Zeile auf
@Override
public View getView(int position, View convertView, ViewGroup parent) {
  ViewHolder holder;
// ist convertView schon vorhanden, wenn nicht erzeugen
  if (convertView == null) {
    convertView = inflator.inflate(R.layout.row_layout, parent, false);
    holder = new ViewHolder();
    holder.ivsemester = (ImageView) convertView.findViewById(
                                                    R.id.ivsemester);
    holder.tvname = (TextView) convertView.findViewById(R.id.tvname);
    holder.tvmatrnr = (TextView) convertView.findViewById(R.id.tvmatrnr);
    holder.tvnote = (TextView) convertView.findViewById(R.id.tvnote);
    convertView.setTag(holder);
  }
  else {
    // holen der UI-Elemente für diese Zeile
    holder = (ViewHolder) convertView.getTag();
  }

  Student std = liste.get(position);
  holder.ivsemester.setImageResource(std.getIdForDrawable());
  holder.tvname.setText(std.name);
  holder.tvmatrnr.setText(Integer.toString(std.matrnr));
  holder.tvnote.setText(Double.toString(std.note));

  return convertView;
} // gewtView
```

Bemerkungen:
Die Anweisung „inflator.inflate(R.layout.row_layout, parent, false);“ dient dazu, aus dem row-Layout und der Referenz auf die Haupt-Routine eine Zeile mit den UI-Elementen zu machen.

10.2.9 Java Code in MainActivity

Import Anweisungen:

```
Import android.app.ActionBar;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

import android.util.Log;
import android.view.*;
import android.widget.*;
```

Globale Variablen:

```
private static final String TAG = MainActivity.class.getSimpleName();

private StudentenAdapter adapter=null;
private ListView listView = null;
```

Konstruktor:

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

   listView = (ListView) findViewById(R.id.listview);
   adapter = new StudentenAdapter(this);
   listView.setAdapter(adapter);
   Log.i(TAG, "onCreate");
}
```

Testlauf:

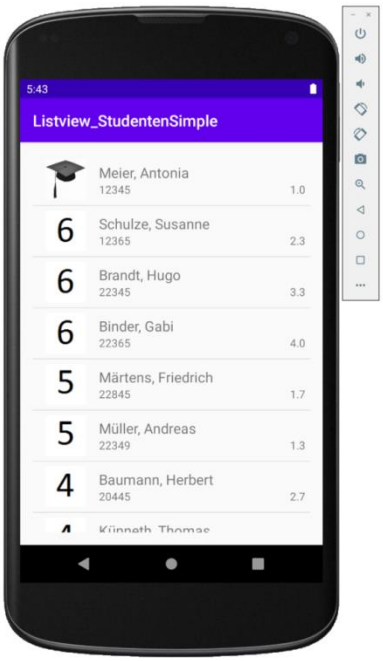


Abbildung 93 Erste Fassung der ListView mit Studenten

Es fehlen:

- Menü für neue Einträge
- Bearbeiten und löschen eines Eintrags
- longClick Event
- Click Event
- Speichern´und Laden der Daten à DataOutputStream

10.3 ListView Studenten Beispiel Weitere Verfeinerung

Dieses Kapitel basiert auf das Projekt „ListView-Beispiel mit StudentenAdapter“. Hier werden nun weitere Eigenschaften eingebaut.

Weitere Eigenschaften:

- Klick-Events für Bearbeiten und Löschen eines Studenten
- Klick-Events für neuen Studenten (Menü)
- Speichern und Laden der Liste

10.3.1 Klasse Basis einfügen

Neue Klasse „Basis.java“

- Eintrag „MainActivity“ anklicken
- Menü File, Eintrag „New“, Eintrag „Java Class“

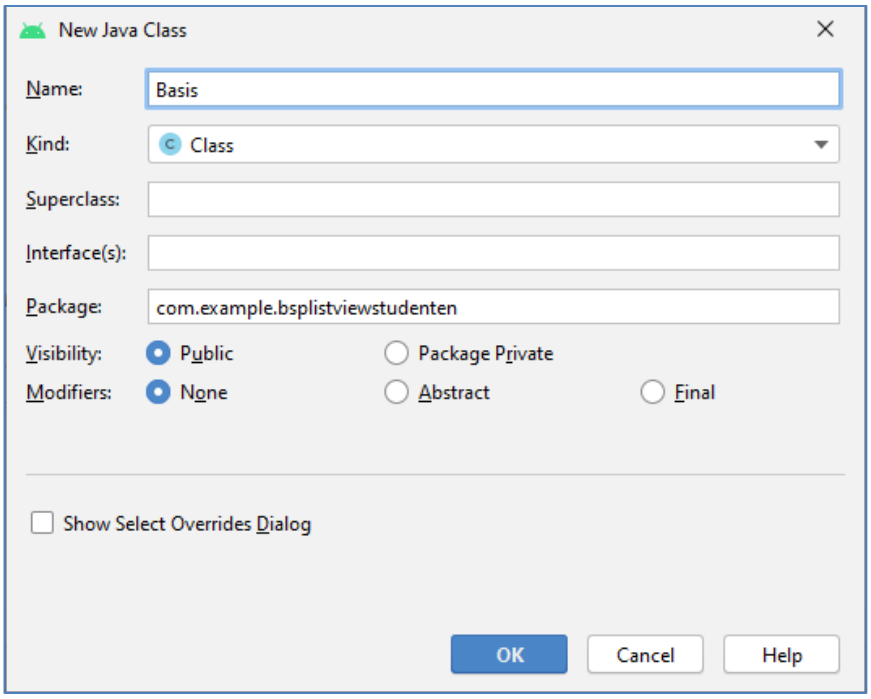


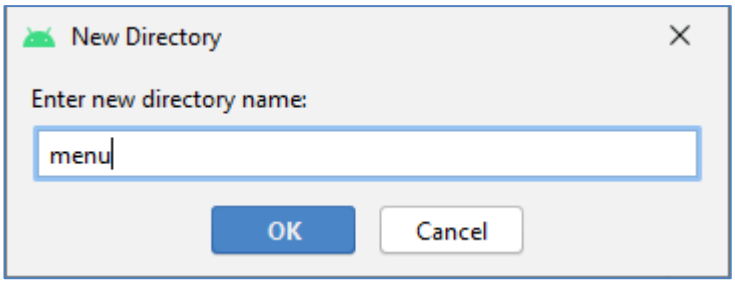
Abbildung 94 Klasse Basis.java

Bitte den Quellcode im Kapitel 19.8, Seite 368.

10.3.2 Menü

Anlegen eines Menüs

- Aktiviereen Sie den Eintrag „res“ im Projektbaum.
- Anlegen eines neuen Ordners „menu“.



Aktivieren des Eintrags „menu“
Anlegen einer neuen Datei

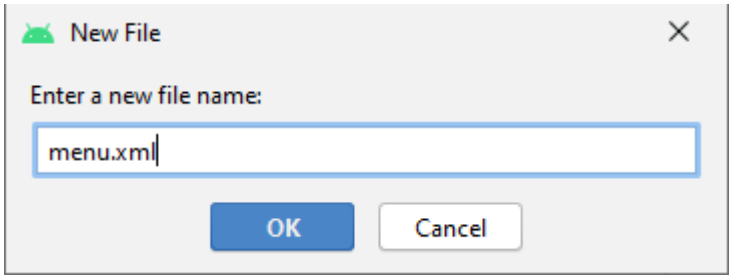


Abbildung 95 Anlegen einer Menüdatei

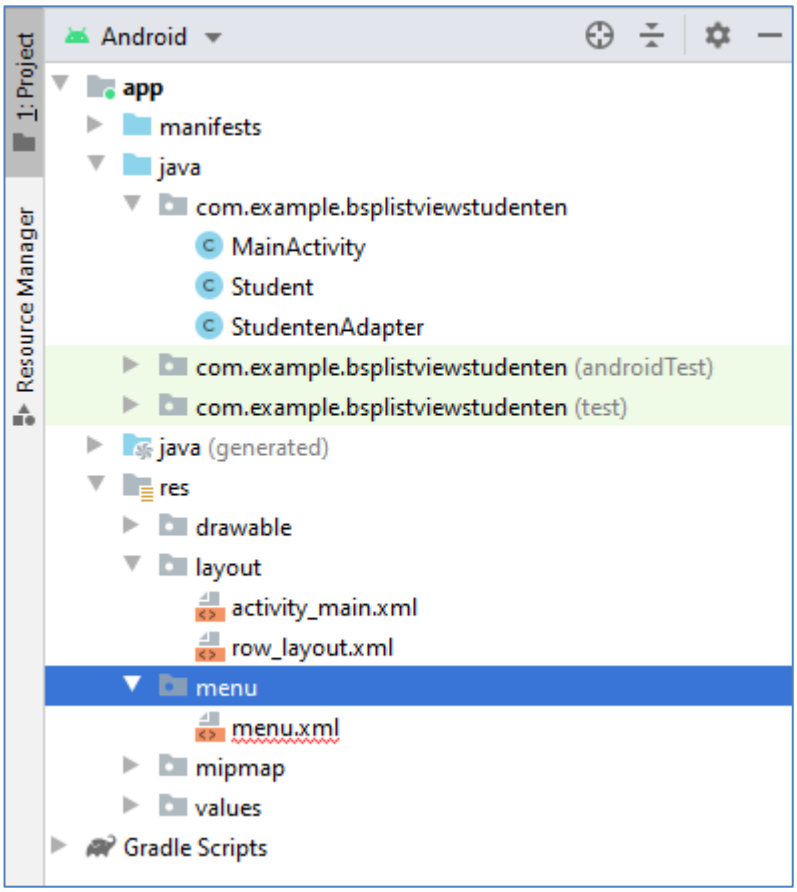


Abbildung 96 Aktueller Strukturbaum des Projektes

Eintragen des Menüs:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:id="@+id/menu_new_item"
    android:icon="@android:drawable/ic_menu_add"
    android:title="New"
  />

</menu>
```

Als Icon wird ein internes Symbol „ic_menu_add“ verwendet.

Beim Testen erscheinen nun die bekannten Menüpunkte:

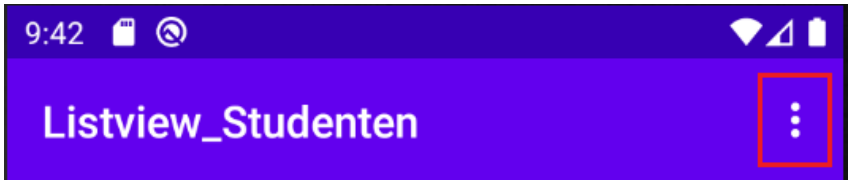


Abbildung 97 Anzeige eines Menüs

Beim Anklicken erscheinen des Menüs (hier natürlich nur ein Menü):

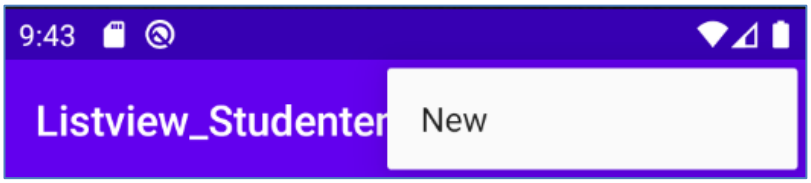


Abbildung 98 Menü New

Neuer Quellcode im Hauptprogramm:

```
// import android.app.ActionBar;


@Override
protected void onStart() {
   super.onStart();
   ActionBar ab = getActionBar();
   if (ab != null) {
     ab.setDisplayHomeAsUpEnabled(true);
   }
}


// diese Methode wird nach dem Anklicken der drei Punkte aktiviert
@Override
public boolean onCreateOptionsMenu(Menu menu) {
   MenuInflater inflater = getMenuInflater();
   inflater.inflate(R.menu.menu,menu);
   return super.onCreateOptionsMenu(menu);
}


@Override
public boolean onOptionsItemSelected(@NonNull MenuItem mnItem) {
   switch (mnItem.getItemId()) {
     case R.id.menu_new_item:
       Basis.showToast(this,"neuer Student");
       Student stdNew = new Student();
       showEditDeleteNew(EditDeleteNew.ACTION_NEW, stdNew,-1);
       break;
   }
   return true;
}
```

Nun noch die Aufruf-Methode einfügen:

```
private void showEditDeleteNew(int action, Student std, int position) {
   Log.i("********** in showEditDeleteNew","position: "+position);
```

```
}
```

10.3.3 Klick-Events

Dieses Kapitel zeigt die Implementierung des kurzen und langen Klicks. Dabei ist zweierlei zu beachten:

- Man sollte die beiden Klicks immer mit „Code-Behind“ deklarieren.
- Man muss erst den long-Klick und dann die Kurzklick definieren.

Quellcode (/Eingetragen in onCreate):

```
listView.setLongClickable(true);

listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
  @Override
  public boolean onItemLongClick(AdapterView<?> parent, View view,
          int position, long id) {
    Log.v("**********************long clicked","position: " + position);
    return true;
  }
});

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
  @Override
  public void onItemClick(AdapterView<?> parent, View view,
          int position, long id) {
    Log.i("********************* clicked","position: " + position);
  }
});
```

Mit demn Menü und den zwei Klick-Events hat man nun alle Möglichkeiten, die ListView zu bearbeiten.

Bevor man den View aufrufen kann, muss man die Activity erstellen.

Wichtig:
Man muss eine Activity immer über die Menüs erstellen. Nur so ist die Verbindung der XML-Datei mit dem Java-Code gegeben.

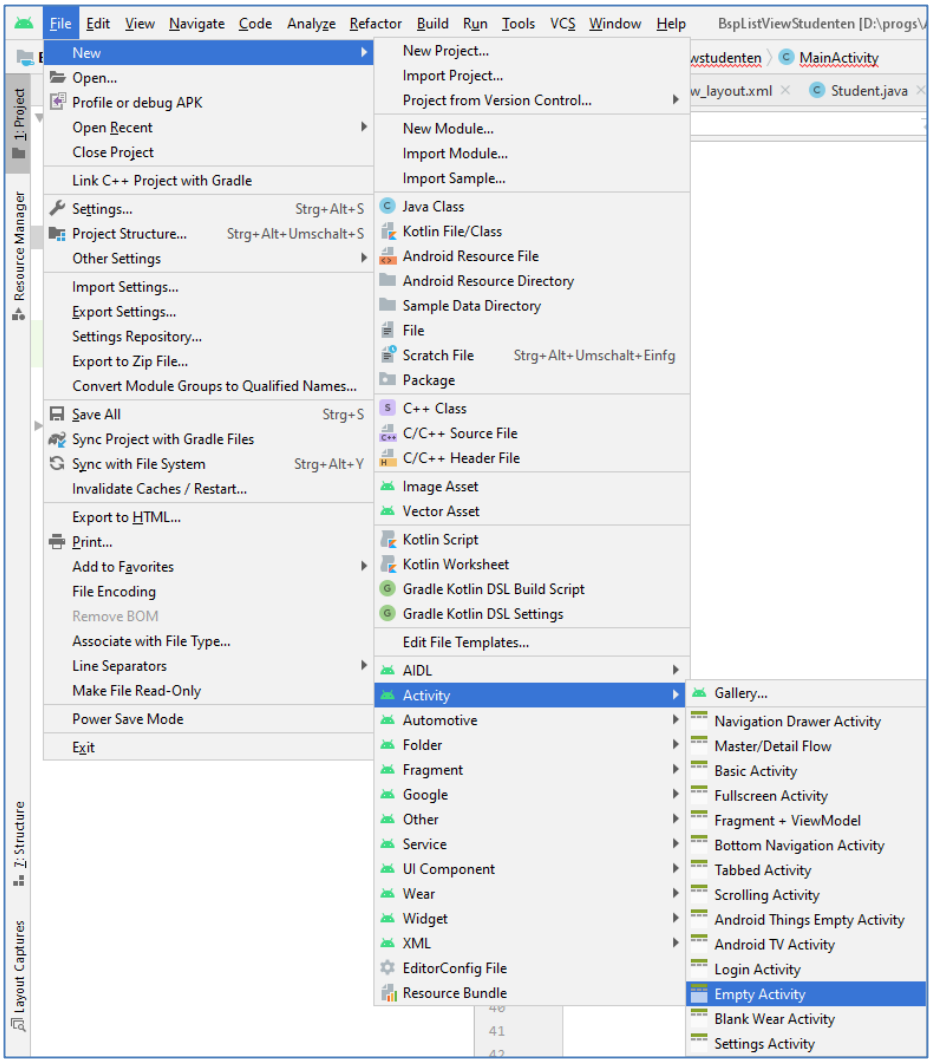


Abbildung 99 Neue Activity erstellen

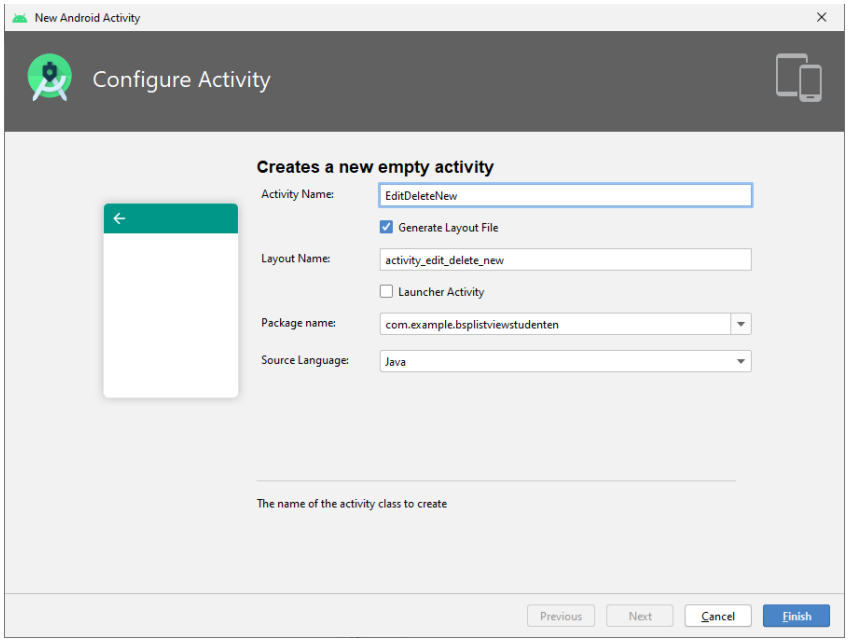


Abbildung 100 Neue Activity EditDeleteNew

Activityname: EditDeleteNew

10.3.4 TextViews zum Testen in die neue Activity einfügen

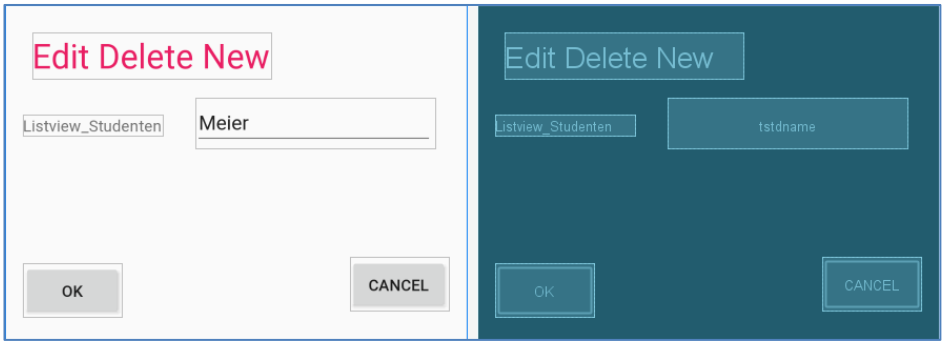


Abbildung 101 Erster Entwurf des Layouts

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".EditDeleteNew">

    <TextView
        android:id="@+id/tvtitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="Edit Delete New"
        android:textColor="#E91E63"
        android:textSize="30sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvstdname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/app_name"

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitle" />

    <EditText
        android:id="@+id/tstdname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="17dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:singleLine="true"
```

```
        android:text="Meier"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/tvstdname"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvstdname"
         />

    <Button
        android:id="@+id/bnok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="113dp"
        android:onClick="bnok_click"
        android:text="@android:string/ok"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvstdname" />

    <Button
        android:id="@+id/bnesc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="97dp"
        android:layout_marginEnd="16dp"
        android:onClick="bnesc_click"
        android:text="@android:string/cancel"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tstdname" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Die Activity hat nun einen einen EditText und zwei Schalter, mit der man die Anzeige testen kann. Später kommen die weiteren UI-Elemente für die Klasse Student hinzu.

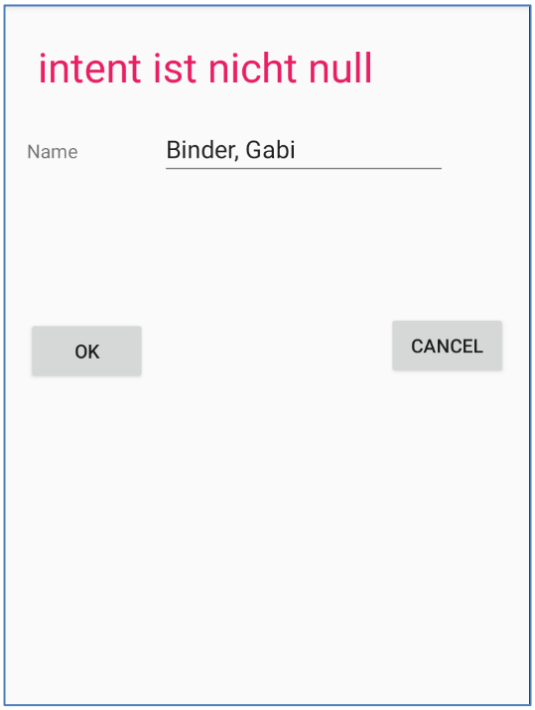


Abbildung 102 Aktuelle Fassung der zweiten Activity

Import in EditDeleteNew.java eintragen:

```
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import java.text.DecimalFormat;
import java.util.*;
```

Konstanten in EditDeleteNew.java eintragen:

Damit man unterscheiden kann, welche der drei Aktionen definiert man in der neuen Klasse „EditDeleteNew.java“ folgende Konstanten:

```
private static final String TAG = MainActivity.class.getSimpleName();

public static final String ACTION = "ACTION";

public static  final  int ACTION_EDIT = 1;
public static  final  int ACTION_DELETE = 2;
public static  final  int ACTION_NEW = 3;

private int action = ACTION_EDIT; // hier wird die aktuelle Action gesp.
```

Für die Übertragung der einzelnen Attrubute sollte man dementsprechend auch Konstanten definmieren:

Eintragen in Klasse: Student.java

```
// die Datenübertragung zu einem neuen View geschieht mit einer Hashtable
   // hier unten werden die "keys" einmalig definiert
public static final String STDNAME = "STDNAME";
public static final String STDMATRNR = "STDMATRNR";
public static final String STDSEMESTER = "STDSEMESTER";
public static final String STDNOTE = "STDNOTE";
public static final String  STDINDEX = "STDINDEX";

public static final String  STD = "STUDENT";
```

10.3.5 Aufruf des neuen Views

Methode onCreate:
Im onClick-Event wird der aktuelle Student geholt und die interne Methode show aufgerufen:

```
listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
  @Override
  public void onItemClick(AdapterView<?> parent, View view, int position,
                                                              long id) {

    Log.i("****************** clicked","position: "+position);
    // nicht die StudentenListe nehmen! GeheimnisPrinzip !!
    Student std = (Student) adapter.getItem(position);

    showEditDeleteNew(EditDeleteNew.ACTION_EDIT, std, position);
  }
});
```

10.3.6 Parameterübergabe

Die Parameterübergabe an eine neue Activity geschieht mittels eines „Briefes“ und nicht mittels Konstruktor, da die neue Activity von Android erzeugt wird.

Der „Brief“ wird durch die Klasse „Intent“ realisert. Nach dem Erzeugen werden die Parameter mit der Methode „putExtra“ in den „Brief“ eingefügt. Alternativ kann man auch die Instanz komplett übertragen (serialize).

```
public void showEditDeleteNew(Student std, int position) {
  Intent intent = new Intent(this,EditDeleteNew.class);
        //viewIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        // startActivityForResult(viewIntent);

    // einfügen der Parameter
  intent.putExtra(EditDeleteNew.ACTION, EditDeleteNew.ACTION_EDIT);
  intent.putExtra(Student.STDNAME, std.name);
  intent.putExtra(Student.STDMATRNR, std.matrnr);
  intent.putExtra(Student.STDSEMESTER, std.semester);
  intent.putExtra(Student.STDNOTE, std.note);

  intent.putExtra(Student.STDINDEX, position);
  startActivityForResult(intent,1);
}
```

Wie man leicht sieht, ist die Parameterübergabe über eine HashTable organisiert. Der erste Parameter symbolisiert den Schlüssel. Die Methode „startActivityForResult“ wechselt zur neuen Activity. Da bei der Eingabe der daten auch ein Abbruch der Eingabe stattfinden kann, wird der Rücksprung mit der Methode „onActivityResult“ abgefangen.

```
private void showEditDeleteNew(int action, Student std, int position) {
  // Erzeugen der zweiten View, Parameter erstellen  und starten
  Intent intent = new Intent(this, EditDeleteNew.class);
  Log.i("************ in showEditDeleteNew","position: "+position);

  // füllen der Parameter, es gibt leider keinen Konstruktor
```

```
  intent.putExtra(EditDeleteNew.ACTION, action);    // hashKey+wert

  // jetzt die Lösung mit Serialize

    intent.putExtras(Student.STD, std);
    intent.putExtra(Student.STDINDEX, position);

  startActivityForResult(intent,1);
}
```

Einbauen der „Serialize“-Schnittstelle in Student.java:

```
public class Student implements Serializable {

}
```

10.3.6.1 Parameter eintragen

Die Methode ist immer „putExtra“:

- intent.putExtra(key, byte);
- intent.putExtra(key, String);
- intent.putExtra(key, char);
- intent.putExtra(key, int);
- intent.putExtra(key, float);
- intent.putExtra(key, double);
- intent.putExtra(key, long);
- und natürlich Arrays in allen Varianten

10.3.6.2 Parameter abfragen

Bei der Abfrage muss man die geeignete Methode auswählen:

- getBooleanArrayExtra(String name)
- getBooleanExtra(String name, boolean defaultValue)
- getByteArrayExtra(String name)
- getByteExtra(String name, byte defaultValue)
- getCharArrayExtra(String name)
- getCharExtra(String name, char defaultValue)
- getCharSequenceArrayExtra(String name)
- getCharSequenceArrayListExtra(String name)
- getCharSequenceExtra(String name)
- getDoubleArrayExtra(String name)
- getDoubleExtra(String name)
- getFloatExtra(String name, float defaultValue)
- getFloatArrayExtra(String name)
- getIntArrayExtra(String name)
- getIntegerArrayListExtra(String name)
- getLongArrayExtra(String name)
- getLongExtra(String name, long defaultValue)
- getSerializableExtra // Bundle

10.3.7 EditDeleteNew.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import java.text.DecimalFormat;
import java.util.*;


public class EditDeleteNew extends AppCompatActivity {


  private static final String TAG = MainActivity.class.getSimpleName();

  public static final String ACTION = "ACTION";

  public static  final  int ACTION_EDIT = 1;
  public static  final  int ACTION_DELETE = 2;
  public static  final  int ACTION_NEW = 3;

  private int action = ACTION_EDIT;

  private TextView tvtitle=null;
  private EditText tname = null;
  private EditText tmatrnr = null;
  private Spinner spsemester = null;
  private Spinner spnote = null;
  private Button bnok = null;

  private TextView tverror = null;

  private int indexStudent=-1;  // für den 2. View

  private DecimalFormat df = new DecimalFormat("0.0");

  private List<Double> notenliste = null;




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

    tvtitle = findViewById(R.id.tvtitle);
    tname = findViewById(R.id.tname);
```

```
    tmatrnr =  findViewById(R.id.tmatrnr);
    spsemester = findViewById(R.id.spsemester);
    spnote =    findViewById(R.id.spnote);
    bnok =  findViewById(R.id.bnok);

    tverror = findViewById(R.id.tverror);

        // füllen der Noteliste
    notenliste = new ArrayList<Double>();
    notenliste.add(1.0);
    notenliste.add(1.3);
    notenliste.add(1.7);
    notenliste.add(2.0);
    notenliste.add(2.3);
    notenliste.add(2.7);
    notenliste.add(3.0);
    notenliste.add(3.3);
    notenliste.add(3.7);
    notenliste.add(4.0);
    notenliste.add(5.0);

    // Daten aus dem „Brief“ holen
    Intent intent = getIntent();
    if (intent==null) {
      tverror.setText(("Die Parameterübergabe ist fehlgeschlagen"));
      bnok.setEnabled(false);
    }
    else {
      boolean ok=true;
      action = intent.getIntExtra(EditDeleteNew.ACTION,-1);
      switch (action) {
        case EditDeleteNew.ACTION_EDIT:
          tvtitle.setText( getString(R.string.actiontitle_edit)  );
          bnok.setText(getString(R.string.bnsavechanges)  );
          break;
        case EditDeleteNew.ACTION_DELETE:
          tvtitle.setText( getString(R.string.actiontitle_delete)  );
          bnok.setText(getString(R.string.bndelete)  );
          break;
        case EditDeleteNew.ACTION_NEW:
          tvtitle.setText( getString(R.string.actiontitle_new)  );
          bnok.setText(getString(R.string.bnnew)  );
          break;
      } // switch

      if (action==EditDeleteNew.ACTION_NEW) {
        this.indexStudent=-1;
      }
      else {
        this.indexStudent = intent.getIntExtra(Student.STDINDEX, -1);
        if (this.indexStudent == -1) {
          tverror.setText(("Der Index des Studenten in der Liste ist
               fehlerhaft: Index: " + this.indexStudent));
          bnok.setEnabled(false);
          ok = false;
        }
      }

      Student std = (Student) intent.getSerializableExtra(Student.STD);
```

```
      if (std.matrnr==-1) {
        tverror.setText(("Die Matrikelnummer wurde nicht übergeben."));
        bnok.setEnabled(false);
        ok=false;
      }
      if (std.matrnr==0 || std.matrnr > 50000) {
        tverror.setText(("Die Matrikelnummer ist fehlerhaft. Matrnr: "
                                                  +std.matrnr));
        bnok.setEnabled(false);
        ok=false;
      }

      if (std.note==-1) {
        tverror.setText(("Die Matriklenummer ist fehlerhaft"));
        bnok.setEnabled(false);
        ok=false;
      }

      if (std.note!=1.0 && std.note!=1.3 && std.note!=1.7 && std.note!=2.0
          && std.note!=2.3 && std.note!=2.7 && std.note!=3.0 &&
          std.note!=3.3 && std.note!=3.7 && std.note!=4.0 &&
        std.note!=5.0) {
        tverror.setText(("Die Note hat einen falschen Wert: "+std.note));
        bnok.setEnabled(false);
        ok=false;
      }
        // wenn alles okay dann eintragen !
      if (ok) {
        tname.setText(std.name);
        tmatrnr.setText(Integer.toString(std.matrnr));
        spsemester.setSelection(std.semester-1);

        ArrayAdapter<Double> notenAdapter = new ArrayAdapter<Double>(this,
                    android.R.layout.simple_spinner_item,notenliste);
        spnote.setAdapter(notenAdapter);

      // setzen der Note
      for (int i=0; i<notenliste.size();i++) {
        if (std.note==notenliste.get(i)) {
          spnote.setSelection(i);
          break;
        }
      } // for
    }
  } // intent = null,

}  // protected void onCreate(Bundle savedInstanceState) {


  public void bnesc_Click(View view) {
  }


  public void bnok_Click(View view) {
}
```

Beim Holen der Parameter muss man die korrekten Datentypen beachten (siehe unten). Die globale Variable „indexStudent“ speichert den Index in der ListView.

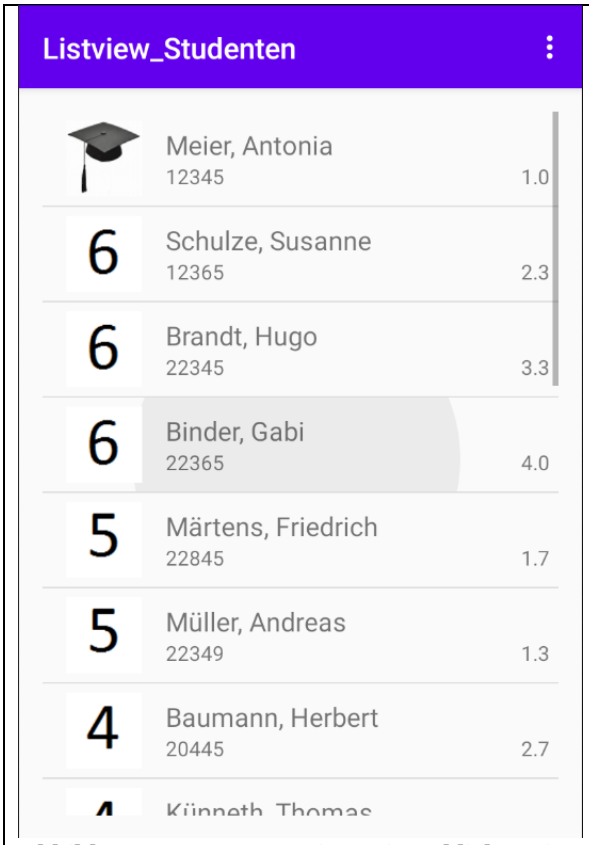


Abbildung 103 Startanzeige mit Anklicken eines Eintrags

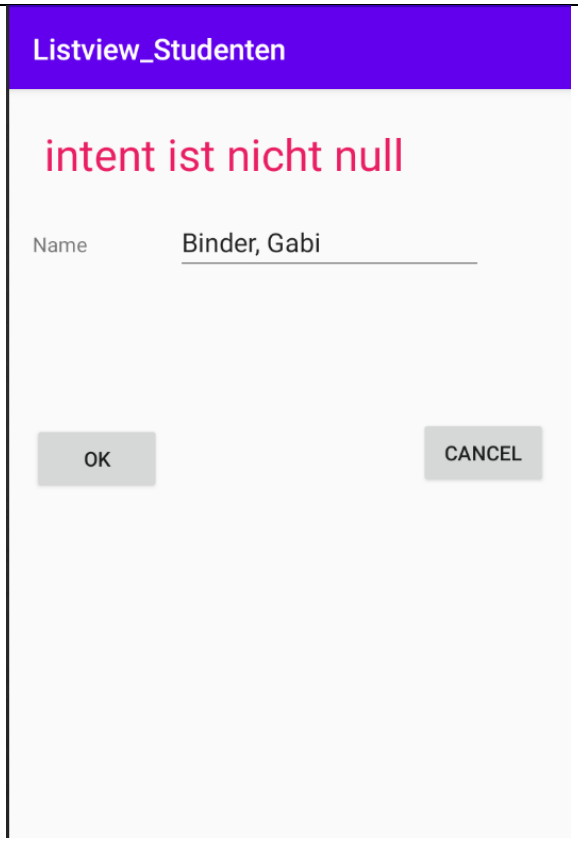


Abbildung 104 Anzeige der Details

10.3.8 Beenden

Das Beenden der zweiten Activity beinhaltet folgende Schritte:

- Schalter ok im Editmodus
  - Prüfen der Eingaben
  - Gegebenenfalls eine Fehlermeldung ausgeben
  - Die geänderten Daten in die Studentenliste speichern
  - Zurückspringen
  - Den ListView-Adapter anzeigen, dass die Daten sich erneuert haben.
- Schalter ok im Deletemodus
  - Prüfen der Eingaben
  - Gegebenenfalls eine Fehlermeldung ausgeben
  - Den Datensatz löschen
  - Zurückspringen
  - Den ListView-Adapter anzeigen, dass die Daten sich erneuert haben.
- Schalter ok im Newmodus
  - Prüfen der Eingaben
  - Gegebenenfalls eine Fehlermeldung ausgeben
  - Die neuen Daten in die Studentenliste speichern
  - Zurückspringen
  - Den ListView-Adapter anzeigen, dass die Daten sich erneuert haben.
- Schalter Cancel
  - Zurückspringen

Für diesen ersten Test reicht es aus, wenn man die Eingabe des Studentennamens ausliest und die Studentenliste aktualisiert.

10.3.9 Schalter Cancel

Die Variable „intent“ ist wieder der „Brief“, hier aber die Rückantwort. Danach wird der RückgabeResult gesetzt und mit „finisch“ wird die Activity beendet.

```
public void bnesc_click(View view) {
   Intent returnIntent = new Intent();
   setResult(Activity.RESULT_CANCELED, returnIntent);
   finish();
}
```

10.3.10 Schalter Ok für Edit-Action

Ablauf:

- Prüfen der Eingaben (entfällt)
- Die geänderten Daten in die Studentenliste speichern
    - Die Studentenliste holen
        - Studentenliste stdliste = Studentenliste.getInstance();
        - ArrayList<Student> liste = stdliste.liste;
    - Holen des aktuellen Studenten
        - Student std = liste.get(this.indexStudent);
    - Setzen der neuen Attribute
        - std.name = stdname;
- Zurückspringen

Kompletter Quellcode (noch vereinfacht):

```
public void bnok_click(View view) {
   String stdname = tstdname.getText().toString();
   Studentenliste stdliste = Studentenliste.getInstance();
   Student std = stdliste.liste.get(this.indexStudent);
   std.name = stdname;

   Intent returnIntent = new Intent();
   returnIntent.putExtra("result", "1111");
   setResult(Activity.RESULT_OK, returnIntent);
   finish();

}
```

10.3.11 Abfangen der Rückantwort

Wenn die zweite Activity beeendet wird, muss die ListView darauf reagieren, bzw. die Activity benötigt eine „callBack“-Methode:

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
         Intent data) {

   super.onActivityResult(requestCode, resultCode, data);
   if (requestCode == 1) {
```

```
    if (resultCode == Activity.RESULT_OK) {
      // neu laden der ListView
      adapter.notifyDataSetChanged();
      Basis.showToast(this,"ok gedrückt: „);

      //  Studentenliste.getInstance().saveInput(this);

    }  // if (resultCode == Activity.RESULT_OK) {
    else if (resultCode == Activity.RESULT_CANCELED) {
      Basis.showToast(this,"cancel gedrückt: ");
    }
  }
}//onActivityResult
```

10.3.12 Komplettes Layouts der zweiten Acitivty

Jedes Attribut der Klasse Student bekommt ein entsprechendes UI-Element. Das Semester und die Note werden als Spinner realisieren.

Details:

10.3.12.1 Studentenname

- Der Studentenname ist die Basis aller Eingabeelemente.
- „tvname“ ist unterhalb des titles.
- „tvtame“ hat eine Breite von 10ems und liegt auf der Baseline von „tvname“.
- „tvname“ hat als InputModus „textPersonName“

10.3.12.2 Matrikelnummer

- Die Matrikelnummer positioniert sich an tvname.
- „tvmatrnr“ ist unterhalb des „tvname“.
- „tmatrnr“ hat die gleiche Breite wie „tname“ und liegt auf der Baseline von „tvmatrnr“.

10.3.12.3 Semester

- Der Spinner „spsemester“ popsitioniert sich an tvname.
- „tvsemester“ ist unterhalb des „tvmatrnr“.
- „spsemester“ hat die gleiche Breite wie „tmatrnr“ und liegt auf der Baseline von „tvsemester“.
- Der spinnerMode ist "dialog".

10.3.12.4 Note

- Der Spinner „spnote“ positioniert sich an tvname.
- „tvnote“ ist unterhalb des „tvsemester“.

- „spnote“ hat die gleiche Breite wie „spsemester“ und liegt auf der Baseline von „tvnote“.
- Der spinnerMode ist " dropdown ".

10.3.12.5 Schalter

- Die Schalter liegen unterhalb von tvnote.
- „bnesc“ und „bnok“ liegen in einem horizontalen Layout.
- Bnok liegt auf der Baseline von „bncancel“.

Kompletter Quellcode von „activity_edit_delete_new.xml“

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
  tools:context=".EditDeleteNew">

  <TextView
    android:id="@+id/tvtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="EditDeleteNew"
    android:textColor="#B32121"
    android:textSize="24sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="36dp"
    android:text="@string/tvstudent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitle" />

  <EditText
    android:id="@+id/tname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="42dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvname"
    app:layout_constraintStart_toEndOf="@+id/tvname" />

  <TextView
```

```
    android:id="@+id/tvmatrnr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="48dp"
    android:text="@string/tvmatrnr"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvname" />

  <EditText
    android:id="@+id/tmatrnr"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

    android:inputType="number"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvmatrnr"
    app:layout_constraintEnd_toEndOf="@+id/tname"
    app:layout_constraintStart_toStartOf="@+id/tname" />

  <TextView
    android:id="@+id/tvsemester"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="42dp"
    android:text="@string/tvsemester"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvmatrnr" />

  <!--  Bitte erst  das Array füllen, dann Baseline -->
  <!--  Falls dynamisch, reinkopieren -->
  <Spinner
    android:id="@+id/spsemester"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:entries="@array/semester_arrays"
    android:prompt="@string/spsemesterprompt"
    android:spinnerMode="dialog"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvsemester"
    app:layout_constraintEnd_toEndOf="@+id/tname"
    app:layout_constraintStart_toStartOf="@+id/tname" />

  <TextView
    android:id="@+id/tvnote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="42dp"
    android:text="@string/tvnote"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvsemester" />

  <Spinner
    android:id="@+id/spnote"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

    android:prompt="@string/spnoteprompt"
    android:spinnerMode="dropdown"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvnote"

    app:layout_constraintEnd_toEndOf="@+id/tname"
    app:layout_constraintStart_toStartOf="@+id/tname" />
```

```
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="42dp"
    android:layout_marginEnd="8dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spnote">

    <Button
    android:id="@+id/bnesc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="5dp"
    android:layout_weight="1"
    android:background="@drawable/button_border_settings"
    android:onClick="bnesc_Click"
    android:text="@string/bncancel" />

    <Button
    android:id="@+id/bnok"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:layout_weight="1"

    android:onClick="bnok_Click"
    android:text="@string/bnsavechanges" />
  </LinearLayout>

  <TextView
    android:id="@+id/tverror"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="50dp"
    android:text="-"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

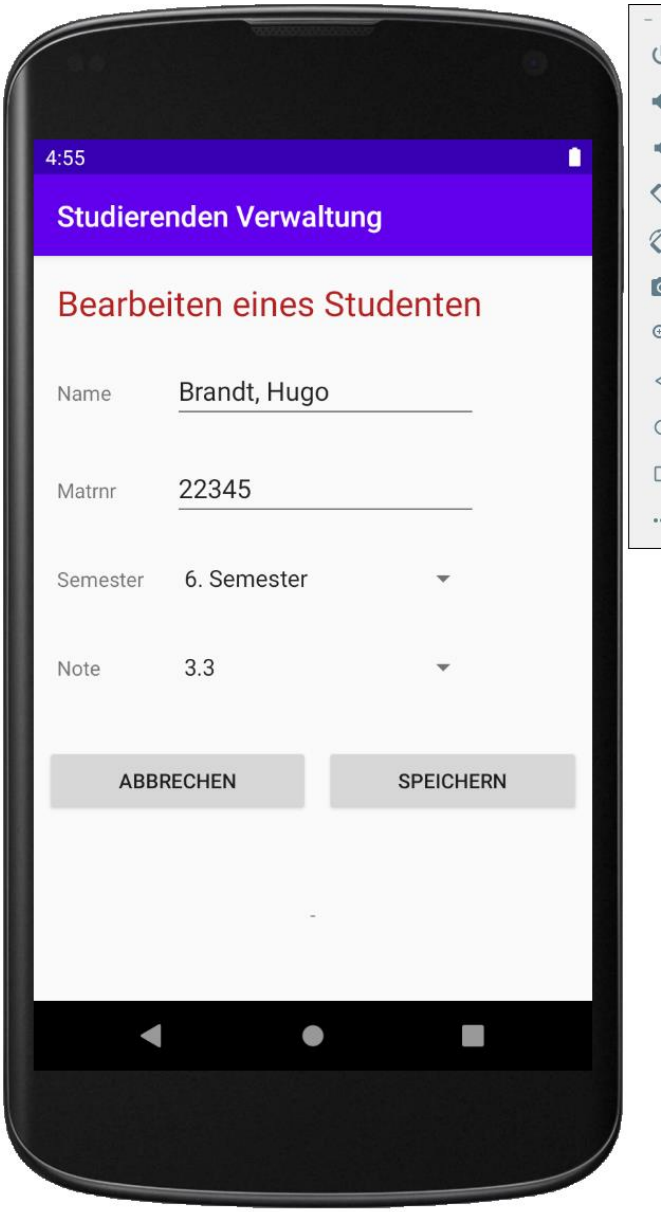


Abbildung 105 Aktueller Stand der Studenten-ListView

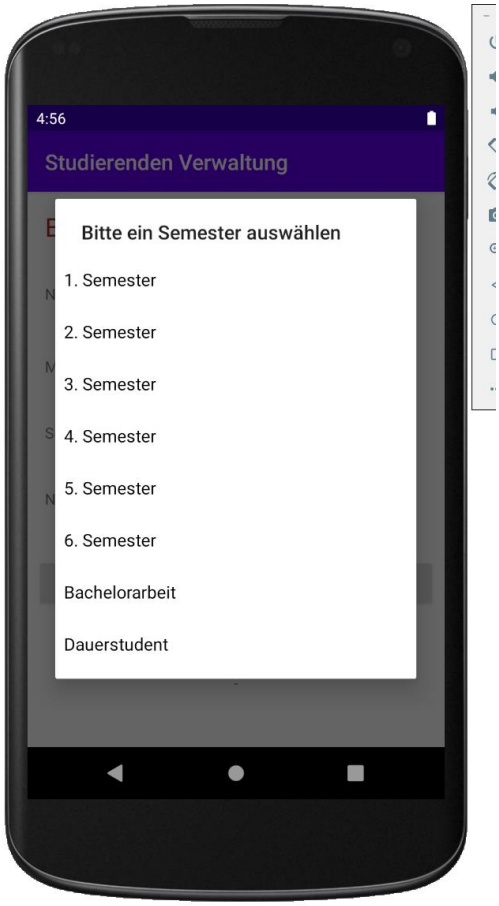


Abbildung 106 Auswahl des Semesters

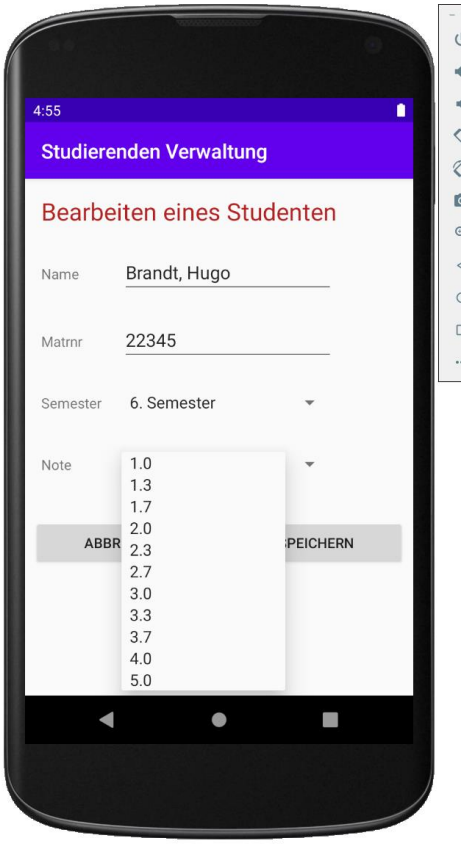


Abbildung 107 Auswahl der Note

In den beiden obigen Abbildungen sieht man die unterschiedlichen Darstellungen der beiden Spinner.

10.3.13 Prüfen der Eingaben

Dazu wird eine Methode „CheckInput entwickelt. Wenn die Eingaben alle korrekt sind, die Namen werden nicht mit regulären Ausdrücken geprüft wird eine Instanz der Klasse Student erzeugt und die neuen Daten darin eingetragen und zurückgegeben.
Im Fehlerfall wird „null“ zurückgegeben.

```
private Student checkInputs() {
  String dummy;
  String name = tname.getText().toString();

  Integer Int = Basis.getIntegerNumber(tmatrnr.getText().toString());
  if (Int==null) {
    Basis.showToast(this,getString(R.string.wrongmatrnr));
    return null;
  }
  int matrnr = Int.intValue();

  int index = spsemester.getSelectedItemPosition();
  if (index<0) {
    Basis.showToast(this,getString(R.string.spsemesterprompt));
```

```
      return null;
  }
  int semester = index+1;

  double note=0.0;
  int indexNote = (int) spnote.getSelectedItemPosition();
  if (indexNote>=0  && indexNote<notenliste.size()) {
    note = notenliste.get(indexNote);
  }
  else {
    Basis.showToast(this,getString(R.string.spnoteprompt));
    return null;
  }
  Student std = new Student((name, matrnr, semester, note );
  return std;
}
```

Alternative Lösung (ohne Prüfungen):

```
private Student checkInputs() {
  String dummy;
  int matrnr=0;
  double note=1.0;
  String name = tname.getText().toString();
  if (name.length()==0) {
    Basis.showToast(this,"Der Namen muss mindestens einen Buchstaben");
    return null;
  }
  Integer Int = Basis.getIntegerNumber(tmatrnr.getText().toString());
  if (Int == null) {
    Basis.showToast(this,
      "Die Matrikelnummer muss numerisch sein (ganzzahlig, >0)");
    return null;
  }
  matrnr = Int.intValue();

  int semester =  spsemester.getSelectedItemPosition()+1;
  int indexNote =  spnote.getSelectedItemPosition();
  note = notenliste.get(indexNote);
  return new Student(name, matrnr, semester, note );
}
```

10.3.14 Schalter Ok in EditDeleteNew.java

```
public void bnok_Click(View view) {
  // prüfe die Eingaben und ändere die Daten  change, delete insert
  Studentenliste stdliste = Studentenliste.getInstance();
  switch (action) {
    case EditDeleteNew.ACTION_EDIT:
      Student std = checkInputs();
      if (std==null) {
        return;  // Fehlermeldung ist in der Methode
      }
      if (!stdliste.changeStudent(std,indexStudent)) {
        Basis.showToast(this,     "Der Datensatz konnte nicht geändert
                werden, index:"+indexStudent);
```

```
        return;
      }
      stdliste.saveInput(this);
       break;

    case EditDeleteNew.ACTION_DELETE:
      if (!stdliste.deleteStudent(indexStudent)) {
        Basis.showToast(this, "Der Datensatz konnte nicht gelöscht werden,
              index:" + indexStudent);
        return;
      }
      stdliste.saveInput(this);
      break;

    case EditDeleteNew.ACTION_NEW:
      Student stdNew = checkInputs();
      if (stdNew==null) {
        return;  // Fehlermeldung ist in der Methode
      }
      if (!stdliste.insertNewStudent(stdNew)) {
        Basis.showToast(this,"Der Datensatz konnte nicht eingefügt werden,
              index:");
        return;
      }
      stdliste.saveInput(this);
      break;

    default:
      Basis.showToast(this,"Falscher switch-Fall in action: "+action);
      return;
  } // switch

  Intent returnIntent = new Intent();
    // returnIntent.putExtra("action",this.action);
  returnIntent.putExtra("returncode",1);
  setResult(Activity.RESULT_OK,returnIntent);
  finish();
}
```

10.4 Änderungemethoden in der Studentenliste

10.4.1 Ändern eines Studenten (Studentenliste)

Ändert man einen Studenten, so muss man die Liste aktualisieren:

```
public void changeStudent(Student stdChange, int index) {
  if (index>=0 && index<liste.size()) {
    Student std = liste.get(index);
    std.init(stdChange);
  }
  else {
    Log.i(TAG,"Modul Studenteliste, falscher Index: "+index);
  }
}
```

Mit folgender Methode in der Klasse Student:

```
public void init(Student std) {
  this.name = std.name;
  this.matrnr = std.matrnr;
  this.semester = std.semester;
  this.note = std.note;
}
```

10.4.2 Löschen eines Studenten (Studentenliste)

Beim Löschen eines Studenten, entfernt man ihn einfach aus der Liste:

```
public void deleteStudent(int index) {
  if (index>=0 && index<liste.size()) {
    liste.remove(index);
  }
}
```

10.4.3 Einfügen eines neuen Studenten (Studentenliste)

Beim Einfügen fügt man ihn am Ende in der Liste:

```
public void newStudent(Student stdnew) {
  liste.add(stdnew);
}
```

10.5 Laden und Speichern (Studentenliste)

Die Liste wird mittels DataOutputStream gespeichert und mittels DataInputStream geladen. Dabei lädt und speichert die Klasse Student ihre Datei eigenständig (Geheimnisprinzip).

10.5.1 Laden der Studenten:

```
public boolean loadInput(Context context) {
  FileInputStream fin = null;
  DataInputStream din = null;
  boolean result = true;
  liste.clear();
  try {
    fin = context.openFileInput(FILENAME);
    din = new DataInputStream(fin);
    int n = din.readInt();  // Anzehl der Datensätze
    for (int i=0; i<n; i++) {
      Student std = new Student();
      std.readFromFile(din);
```

```
      liste.add(std);
    }
    din.close();
    fin.close();
  }
  catch (IOException e) {
    Log.e(TAG, "save2File, Error in saveInput (Studentenliste) "+
                                                    e.toString());
    // e.printStackTrace();
    result = false;
  }

return result;

} // loadInput
```

10.5.2 Speichern der Studenten:

- Um die Liste zu lesen, benötigt man am Anfang die Anzahl in der Liste!

```
public boolean saveInput(Context context) {
  FileOutputStream fout = null;
  DataOutputStream dout = null;
  boolean result = true;

  try {
    fout = context.openFileOutput(FILENAME, Context.MODE_PRIVATE);
    dout = new DataOutputStream(fout);
    dout.writeInt(liste.size());  // Anzehl der Datensätze
    for (Student std: liste) {
      std.write2File(dout);
    }
    dout.close();
    fout.close();
  }
  catch (IOException e) {
    Log.e(TAG, "save2File, Error in saveInput (Studentenliste) "+
      e.toString());
        // e.printStackTrace();
    result = false;
  }

return result;

}  // saveInput
```

10.6 Laden und Speichern eines Studenten (Klasse Student.java)

Das Speichern der Liste funktioniert mittels DataOutputStream siehe Kapitel 14.3.2.2, Seite 298.

Import Anweisung:

```
import static android.content.Context.MODE_PRIVATE;
```

10.6.1 Speichern eines Studenten

Dafür muss man in der Klasse „Student“ eine Methode zum Speichern einfügt werden:

```
public boolean write2File(DataOutputStream dout) {
  try {
    dout.writeUTF(name);
    dout.writeInt(matrnr);
    dout.writeInt(semester);
    dout.writeDouble(note);
    return true;
  }
  catch (IOException e) {
    Log.e("Save","Error in write2File",e);
    return false;
  }
}
```

10.6.2 Laden eines Studenten

```
public boolean loadFromFile(DataInputStream din) {
  try {
    name = din.readUTF();
    matrnr = din.readInt();
    semester = din.readInt();
    note = din.readDouble();
    Log.i(TAG, "load "+name);
    return true;
  }
  catch (IOException e) {
    Log.e("Load","Error in loadFromFile",e);
    return false;
  }
}
```

10.7 Umbau des Ladens der Liste

Beim ersten Starten muss der Lioste initialisiert werden. Danach sollten die Daten aus der Datei geladen werden.

Das Problem dabei, dass man den Kontruktor in der Studentenliste nicht aufrufen kann. Deshalb wird eine neue Methode „loadListe“ eingefügt.

Neue Methode in der Studentenliste.java

```
public void loadListe(Context context) {
  // Abfragen, existiert die Datei
  Basis.showToast(context,"in loadInput");
  if (Basis.FileExists(context, FILENAME)) {
    if (!loadInput(context)) {
      Log.i(TAG, "################################ Fehler beim lesen ");
      liste.clear();  // es können einige Studenten eingetragen sein
      initListe();
    }
  }
  else {
    Log.i(TAG, "#####################################Init Data ");
    initListe();
  }
}
```

Neue Anweisung in der mainActivitye.java / onCreate

```
listView = (ListView) findViewById(R.id.listview);
adapter = new StudentenAdapter(this);

listView.setAdapter(adapter);

Studentenliste.getInstance().loadListe(this);  // neu
```

10.8 Abfrage des Ergebnisses des zweiten Views

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
                                                    Intent data) {

  super.onActivityResult(requestCode, resultCode, data);
  if (requestCode == 1) {
    if (resultCode == Activity.RESULT_OK) {
      // neu laden
      adapter.notifyDataSetChanged();
      Studentenliste stdliste = Studentenliste.getInstance();
      Basis.showToast(this,"ok gedrückt: "+std.name);
    }
    else if (resultCode == Activity.RESULT_CANCELED) {
      //Write your code if there's no result
      Basis.showToast(this,"cancel gedrückt: ");
    }
  }
```

```
}//onActivityResult
```

10.9 Symbole im Optionsmenu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:title="..." app:showAsAction="always"
    android:icon="@android:drawable/ic_menu_manage"
  >

    <menu>
      <item android:id="@+id/menu_new_item"
        android:icon="@android:drawable/ic_menu_add"
        android:title="@string/new_item"
      />

      <item android:id="@+id/menu_delete_item"
        android:icon="@android:drawable/ic_menu_delete"
        android:title="@string/delete_item"
      />

    </menu>

  </item>

</menu>
```

10.10 ListView mit Checkboxen

Um eine ListView mit CheckBoxen zu versehen, braucht man nur wenige Änderungen:

- Einbau einer CheckBox in der Datei „row_layout.xml“
- Einbau einer bool’schen Variable „selected“ in der Klasse Student.java
- Setzen der CheckBox im StudentenAdapter
- Einbau von neuen Menü im „Optionsmenü“
    - Selectmodus An
    - Selectmodus Aus
    - Alles markieren
    - Invertieren
    - Löschen aller selectierten Einträge

Hinweis:

- Wenn man die Checkbox anzeigt, also kein „gone“, gibt es keine onClick-Events. Dann kann man keinen Datensatz bearbeiten.

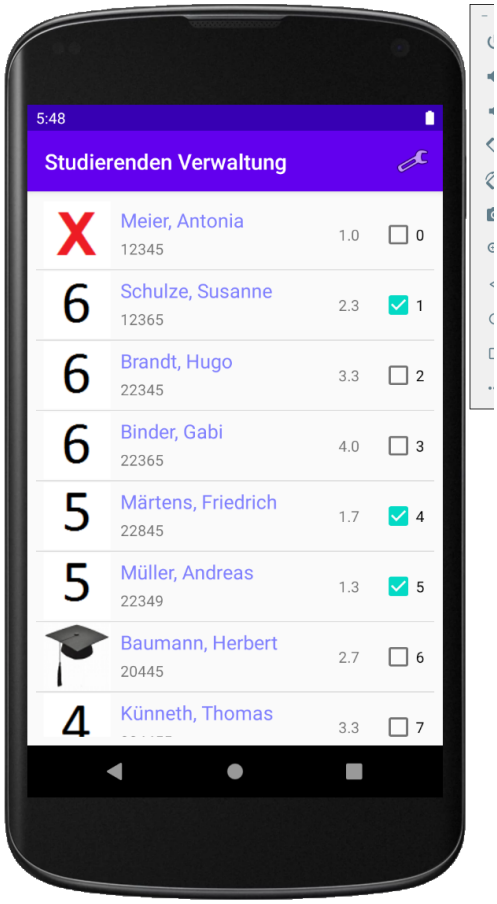


Abbildung 108 ListView im Selected-Modus

10.10.1 row_layout.xml

In der Definition der Zeile muss man am Ende eine CheckBox eintragen:

```
<TextView
        android:id="@+id/tvnote"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="5dp"
        android:text="@string/tvnote"
        android:textAppearance="?android:attr/textAppearanceSmall"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/chkselect"
        app:layout_constraintTop_toTopOf="parent" />

    <CheckBox
        android:id="@+id/chkselect"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="--"
        app:layout_constraintBottom_toBottomOf="parent"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

10.10.2 Student.java

In der Klasse „Student“ wird eine bool’sche variable eingefügt. Diese wird aber nicht in der „Datenbank“ gespeichert.

Neuer bzw. geänderter Quellcode:

```
public void init(Student std) {
  this.name = std.name;
  this.matrnr = std.matrnr;
  this.semester = std.semester;
  this.note = std.note;
  this.isSelected = isSelected;
}

public boolean getSelected() {
  return isSelected;
}

public void setSelected(boolean selected) {
  isSelected = selected;
}

public void invertSelected() {
  isSelected=!isSelected;
}
```

10.10.3 StudentenAdapter

Im Studentenadapter gibt es zwei Änderungen:

- Eine öffentliche Variable, die den Modus steuert. Dieser Variable wird aus MainActivity durch die Optionsmenü-Einträge gesteuert.
- Verstecken der CheckBox (kein Selected-Modes)
- Setzen der CheckBox (Selected-Modes)
- Setzen des onClick-Event für jede CheckBox (Selected-Modes)

öffentliche Variable

```
public boolean selectmodus=false;
```

Ändern der Klasse ViewHolder:

```
static class ViewHolder {
  ImageView ivsemester;
  TextView tvname;
  TextView tvmatrnr;
  TextView tvnote;
```

```
  CheckBox chkselect;
}
```

Ändern der Methode „getView“

```
@Override
public View getView(final int position,View convertView,ViewGroup parent){
  ViewHolder holder;
  if (convertView==null) {
    convertView = inflator.inflate(R.layout.row_layout,parent,false);
    holder = new ViewHolder();
    holder.ivsemester = convertView.findViewById(R.id.ivsemester);
    holder.tvname = convertView.findViewById(R.id.tvname);
    holder.tvmatrnr = convertView.findViewById(R.id.tvmatrnr);
    holder.tvnote = convertView.findViewById(R.id.tvnote);
    holder.chkselect=convertView.findViewById(R.id.chkselect);
    convertView.setTag(holder);
  }
  else {
    holder = (ViewHolder) convertView.getTag();
  }

  Context context = parent.getContext();
    // einfügen der Daten in die "Zeile"
  Student std = (Student) getItem(position);
  holder.ivsemester.setImageResource(std.getIdForDrawable());
  holder.tvname.setText(std.name);
  holder.tvmatrnr.setText(Integer.toString(std.matrnr));
  holder.tvnote.setText(df.format(std.note));

  if (selectmodus) {
    holder.chkselect.setChecked(std.getSelected());
    holder.chkselect.setText(Integer.toString(position));
    holder.chkselect.setTag( position);
    holder.chkselect.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        CheckBox chkbox = (CheckBox) v;
        Integer pos = (Integer)  chkbox.getTag();
        if(liste.get(pos).getSelected()){
          liste.get(pos).setSelected(false);
        }else {
          liste.get(pos).setSelected(true);
        }  // if
      }
    });  //  // onClick
  }  // if selectedModus
  else {
    holder.chkselect.setVisibility(View.GONE);
  }
  return convertView;
}
```

10.10.4 Optionsmenü

Im Optionsmenü müssen nun die Menüs definiert werden (menu.xml) und in MainActivity müssen die Events abgefangen werden.

strings.xml:

```
<string name="new_item">Neu</string>

<string name="selectmoduson_item">Select on</string>
<string name="selectmodusoff_item">Select off</string>
<string name="selectall_item">Select all</string>
<string name="invertselect_item">Invert selection</string>
<string name="show_items">Show items</string>
```

menu.xml:

Die untere Version zeigt auch die Symbole an!

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:title="Main" app:showAsAction="always"
    android:icon="@android:drawable/ic_menu_preferences"
  >

    <menu>
      <item android:id="@+id/menu_new_item"
        android:icon="@android:drawable/ic_menu_add"
      android:title="@string/new_item"
      />

    <item android:id="@+id/menu_selectmoduson_item"
        android:icon="@android:drawable/ic_menu_more"
        android:title="@string/selectmoduson_item"
      />
      <item android:id="@+id/menu_selectmodusoff_item"
        android:icon="@android:drawable/ic_menu_close_clear_cancel"
        android:title="@string/selectmodusoff_item"
      />

      <item android:id="@+id/menu_selectall_item"
        android:icon="@android:drawable/ic_menu_slideshow"
        android:title="@string/selectall_item"
      />

      <item android:id="@+id/menu_invertselect_item"
        android:icon="@android:drawable/ic_menu_revert"
        android:title="@string/invertselect_item"
      />

      <item android:id="@+id/menu_deleteselect_item"
        android:icon="@android:drawable/ic_menu_delete"
        android:title="@string/delete_item"
      />
      <item android:id="@+id/menu_show_items"
        android:icon="@android:drawable/ic_menu_report_image"
```

```
          android:title="@string/show_items"
        />

     </menu>
    </item>

</menu>
```

Bitte beachten:

- Es gibt zwei „Orte“ aus denen man Symbole holen kann.
  - @android:drawable/ic_menu_add
  - @drawable/ic_add_24dp
- Eigene Symbole werden immer mit der zweiten Varainte adressiert.

MainActivity.java:

- Hier muss man die Optionsmenüs als globale Variablen für enabled-Aktionen holen.
- Es müssen die Events abgefragt werden.

Globale Variablen:

```
MenuItem menu_selectmoduson = null;
MenuItem menu_selectmodusoff = null;
MenuItem menu_selectall = null;
MenuItem menu_invertselect = null;
MenuItem menu_deleteselect = null;
MenuItem nmenu_show_items = null;
```

onCreateOptionsMenu:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.menu,menu);

  menu_selectmoduson =  menu.findItem(R.id.menu_selectmoduson_item);
  menu_selectmodusoff =  menu.findItem(R.id.menu_selectmodusoff_item);
  menu_selectall =  menu.findItem(R.id.menu_selectall_item);
  menu_invertselect =  menu.findItem(R.id.menu_invertselect_item);
  menu_deleteselect =  menu.findItem(R.id.menu_deleteselect_item);
  nmenu_show_items =  menu.findItem(R.id.menu_show_items);

  menu_selectmodusoff.setEnabled(false);
  menu_selectall.setEnabled(false);
  menu_invertselect.setEnabled(false);
  menu_deleteselect.setEnabled(false);

  return super.onCreateOptionsMenu(menu);
}
```

Abfrage der Events:

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem mnItem){
  switch (mnItem.getItemId()) {
    case R.id.menu_new_item:
      Student stdNew =  new Student();
      showEditDeleteNew(EditDeleteNewActivity.ACTION_NEW, stdNew, -1);
      break;

    case R.id.menu_selectmoduson_item:
      adapter.selectmodus=true;
      listView.setAdapter(adapter);
      menu_selectmoduson.setEnabled(false);
      menu_selectmodusoff.setEnabled(true);
      menu_selectall.setEnabled(true);
      menu_invertselect.setEnabled(true);
      menu_deleteselect.setEnabled(true);
      break;

    case R.id.menu_selectmodusoff_item:
      adapter.selectmodus=false;
      listView.setAdapter(adapter);
      menu_selectmoduson.setEnabled(true);
      menu_selectmodusoff.setEnabled(false);
      menu_selectall.setEnabled(false);
      menu_invertselect.setEnabled(false);
      menu_deleteselect.setEnabled(false);
      break;

    case R.id.menu_selectall_item:
      for(int i=0; i<adapter.getCount(); i++) {
        Student std = (Student) adapter.getItem(i);
        std.setSelected(true);
      }
      listView.setAdapter(adapter);
      break;

    case R.id.menu_invertselect_item:
      for(int i=0; i<adapter.getCount(); i++) {
        Student std = (Student) adapter.getItem(i);
        std.invertSelected();
      }
      listView.setAdapter(adapter);
      break;

    case R.id.menu_deleteselect_item:
      break;

    case R.id.menu_show_items:
      Log.i(TAG,"******************");
      for(int i=0; i<adapter.getCount(); i++) {
        Student std = (Student) adapter.getItem(i);
        if (std.getSelected()) {
          Log.i(TAG,"   menu_show_items isSelected: "+i);
        }
      } // for
      break;

  }
  return true;
```

```
}
```

11 MultiPages

11.1 Bottom Navigation Activity

App's mit einer Bottom Navigation verfügen über mehrere Symbole in der unteren Leiste des Views. Das Prinzip dahinter ist das Erstellen von „Fragmenten", die jeweils in den oberen bereich eingetragen werden.



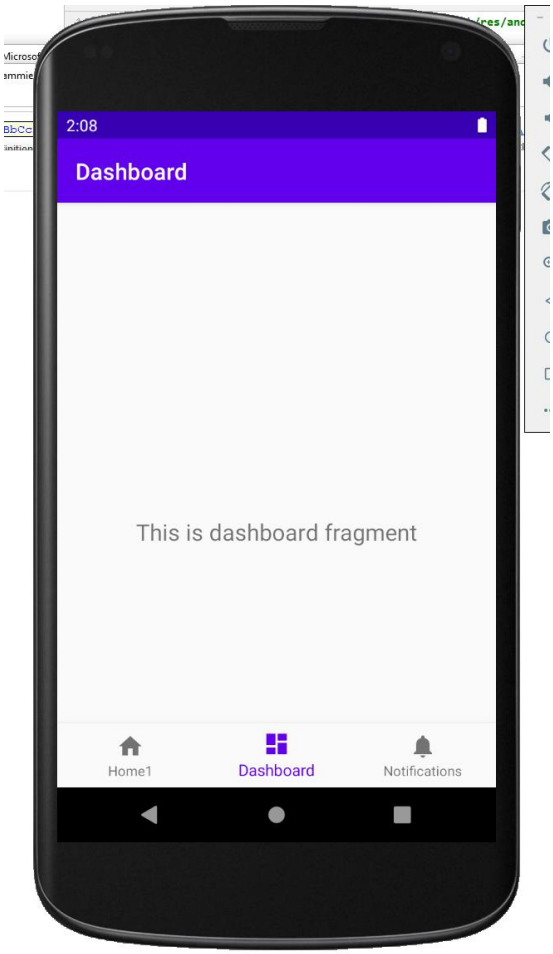


Abbildung 109 Ansicht des Vorlagen-Projektes (BottomNavigation)

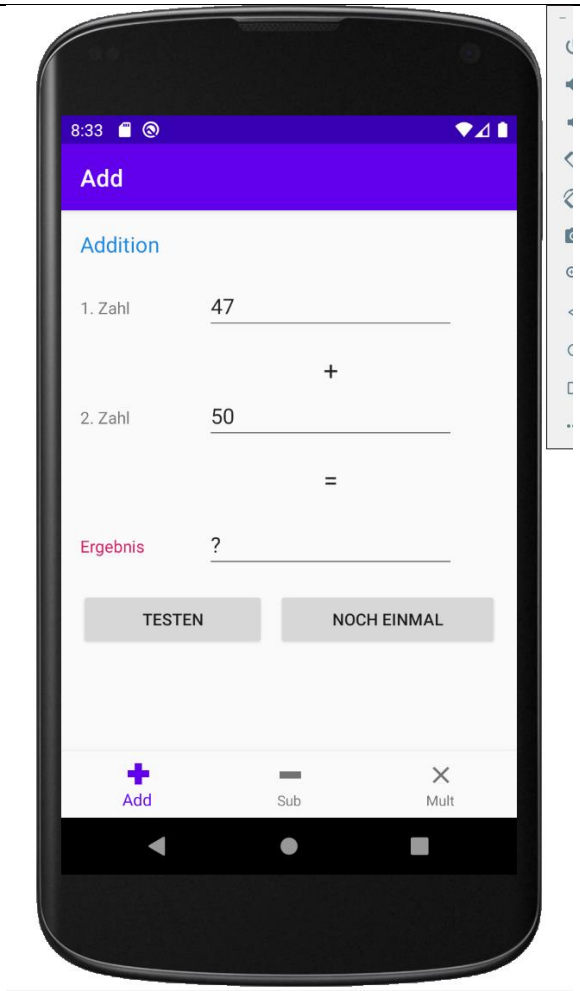


Abbildung 110 Fertiges Projekt

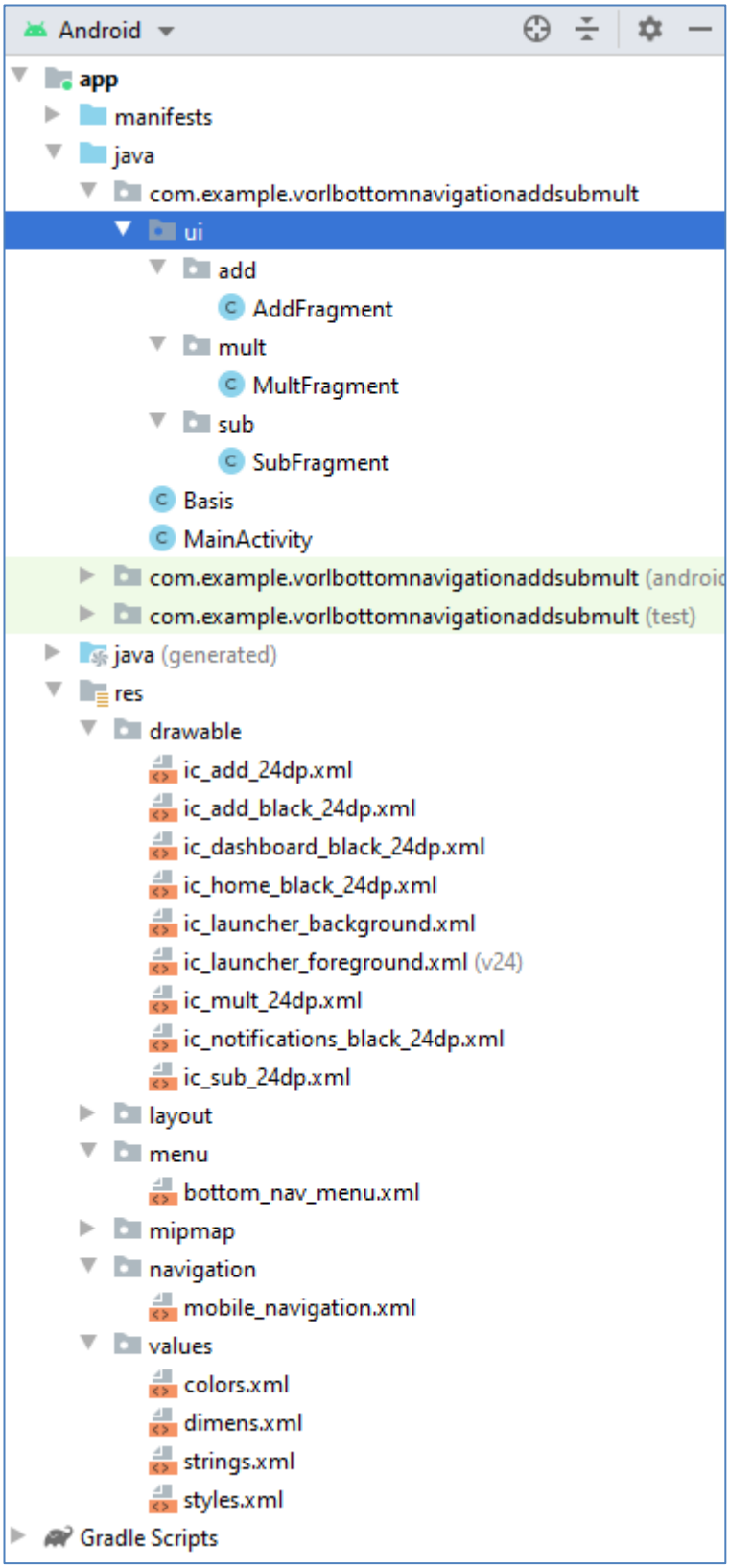


Abbildung 111 Komplette Struktur der Musterlösung

11.1.1 Neues Projekt erstellen

Als Vorlage wird die Bottom-Navigation ausgewählt.

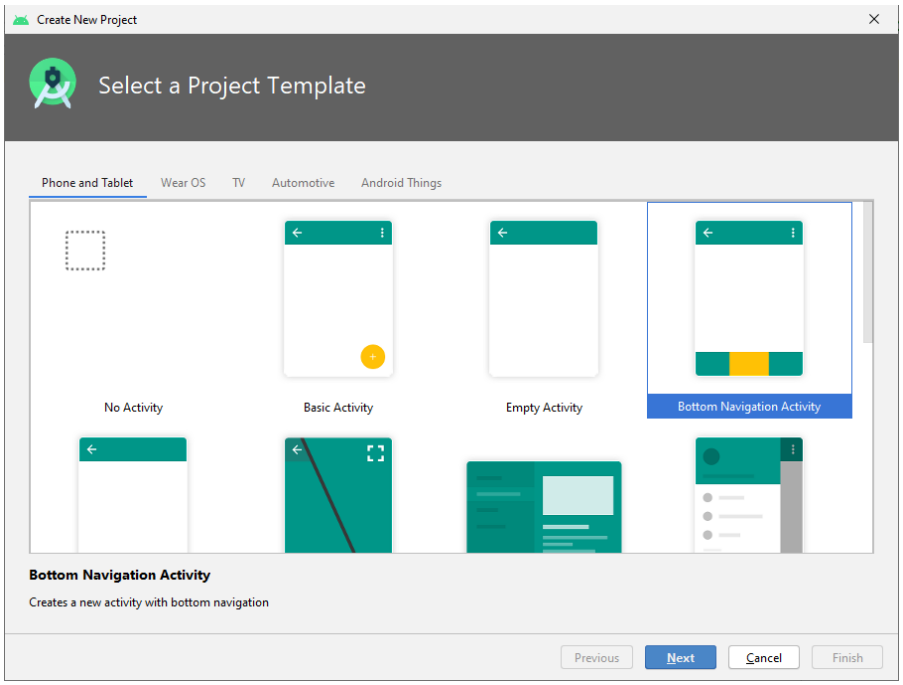
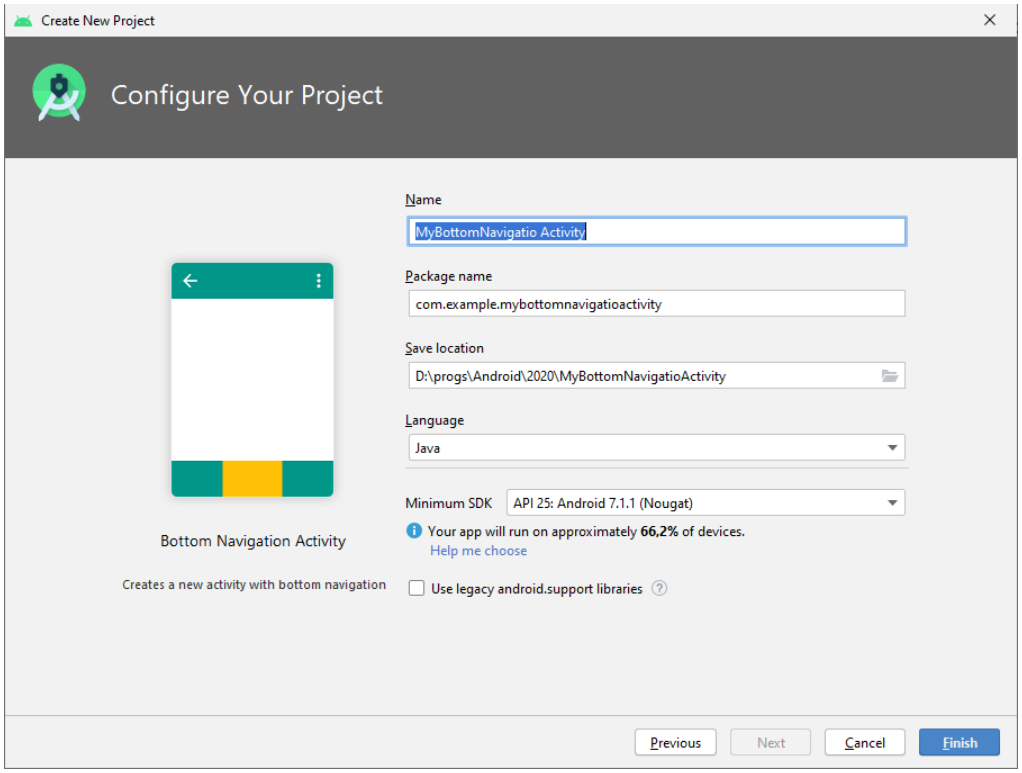


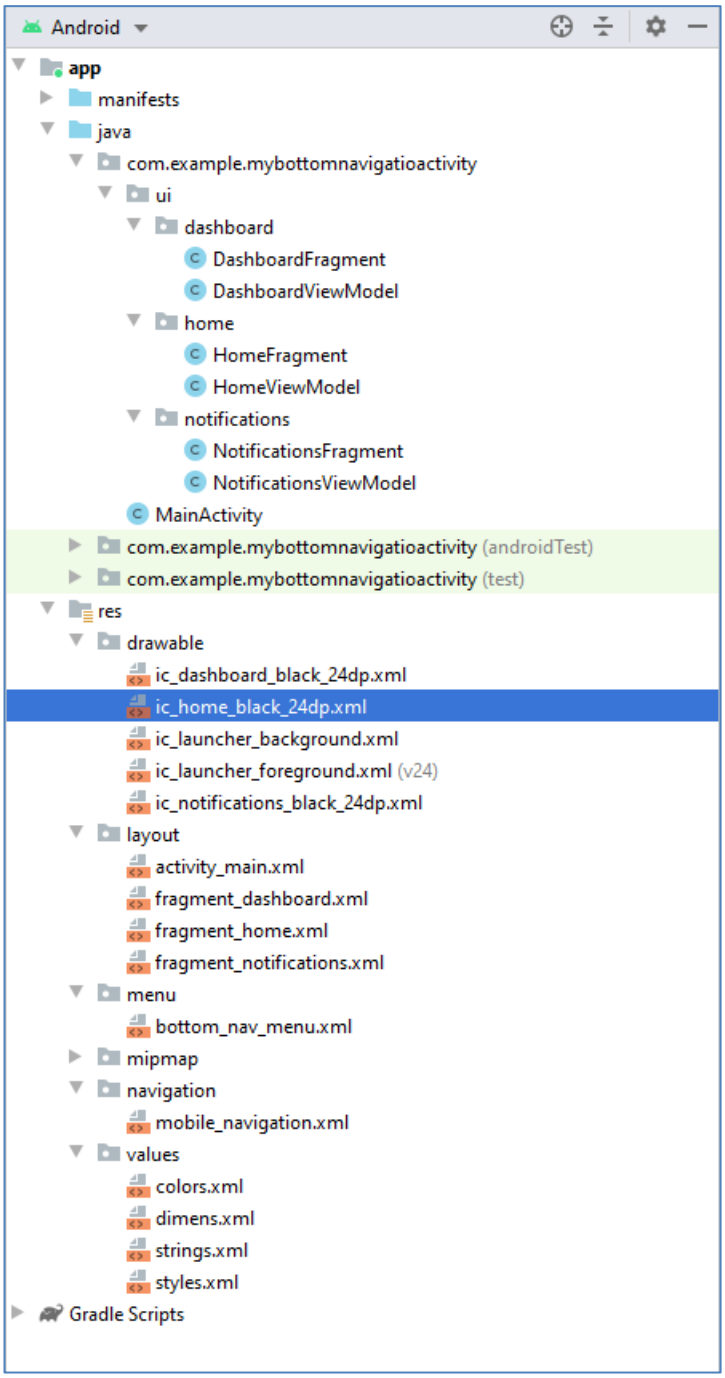
Abbildung 112 Bottom Navigation Activity

Der Projektname lautet: MyBottomNavigatio Activity



11.1.2 Struktur des Projektes

Die Struktur hat sich deutlich zum letzten Jahr geändert.



- In der Main-Activity ist ein Platzhalter eingebaut, in dem alle „Varianten“ eingefügt werden.
- Jede Variante ist ein Fragment und keine Activity!
- Man muss also pro Variante ein neues Fragment erstellen.
- Jedes Fragment besteht aus einer xml-Datei und einer Java-Datei.

11.1.3 Plätze der Fragmente

Im Eintrag „layout“ sind wie im letzten Jahr alle fragment-xml-Dateien eingetragen.

Die Daten der Fragmente sind auf sechs „Plätze“ verteilt:

11.1.3.1 strings.xml

```
<resources>
    <string name="app_name">MyBottomNavigatio Activity</string>
    <string name="title_home">Home</string>
    <string name="title_dashboard">Dashboard</string>
    <string name="title_notifications">Notifications</string>
</resources>
```

In der Datei string.xml werden alle Titel eingetragen. Diese Bezeichnungen werden unten in der Navigation-Bar angezeigt. Sinnvoll sind aber nur drei bis sieben Fragmente.

11.1.3.2 mobile_navigation.xml

In der Datei „mobile_navigation.xml“sind alle Verweise auf die Fragmente eingetragen:

```
<fragment
    android:id="@+id/navigation_home"
    android:name="com.example.mybottomnavigatioactivity.ui.home.HomeFragment"
    android:label="@string/title_home"
    tools:layout="@layout/fragment_home" />
```

Zum einen die id	@+id/navigation_home
Dann das Package mit der JavaDatei:	mybottomnavigatioactivity.ui.home.HomeFragment
Der Titel	title_home
Und die xml-Datei	fragment_home

11.1.4 bottom_nav_menu.xml

In dieser xml-Datei wird die untere Navigation-Bar definiert:

```
<item
    android:id="@+id/navigation_home"
    android:icon="@drawable/ic_home_black_24dp"
    android:title="@string/title_home" />
```

Zum einen die id	@+id/navigation_home
Dann das Symbol	ic_home_black_24dp
Und die Bezeichnung	title_home

11.1.4.1 layout/fragment_home.xml

Hier ist die xml-datei mit den ui-Elementen gespeichert.

11.1.4.2 drawable/ ic_home_black_24dp.xml

Im Baumeintrag/Ordner drawable werden die Symbole gespeichert.

ic_home_black_24dp.xml:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M10,20v-6h4v6h5v-8h3L12,3 2,12h3v8z" />
</vector>
```

Der letzte Punkt betrifft die Java-Dateien.
Hierzu muss immer ein neues Package erstellt werden.

11.1.4.3 ui/home

Hier werden die Java-Dateien gespeichert. Dazu erstellt man erst ein neues Package pro Fragment und erstellt danach das neue Fragment (siehe unten).

In der Vorlage sind die Fragment-Java-Dateien mit einem ViewModel verknüpft (Model View Controller und Klasse MutableLiveData).

```
public class HomeViewModel extends ViewModel {

    private MutableLiveData<String> mText;

    public HomeViewModel() {
        mText = new MutableLiveData<>();
        mText.setValue("This is home fragment");
    }

    public LiveData<String> getText() {
        return mText;
    }


    public void setText(String text) {
         mText.setValue(text);
    }

}
```

Bitte beachten, dass mText eine Wrapper-Klasse ist.

```
public class HomeFragment extends Fragment {

  private HomeViewModel homeViewModel;

  private TextView textView = null;
  private Button bnTestMVC=null;

  public View onCreateView(@NonNull LayoutInflater inflater,
```

```
                        ViewGroup container, Bundle savedInstanceState) {

    View root = inflater.inflate(R.layout.fragment_home, container,
false);

    textView = root.findViewById(R.id.text_home);

    homeViewModel =  ViewModelProviders.of(this).get(HomeViewModel.class);

    homeViewModel.getText().observe(getViewLifecycleOwner(),
                                            new Observer<String>() {
      @Override
      public void onChanged(@Nullable String s) {
        textView.setText(s);
      }
    });


    bnTestMVC =root.findViewById(R.id.bnTestMVC);
    bnTestMVC.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
          bnTestMVC_Click();
        }
      }
    );


    return root;
  }


  private void bnTestMVC_Click() {
    homeViewModel.setText("abc");
  }
}
```

Das ViewModell ändert durch das Change-Event „observe“ automatisch den Inhalt der TextView.

11.1.5 Anlegen eines neuen Fragments

11.1.5.1 strings.xml

```
<resources>
    <string name="title_add">Add</string>
    <string name="caption_add">Addition</string>
</resources>
```

In der Datei string.xml wird ein neuer Eintrag eingetragen:

- title_add wird angezeigt in der Bottom-NavigatioonBar
- caption_add wird angezeigt im Fragment als Überschrift

11.1.5.2 neues package erstellen

Anklicken des ui-Eintrags:

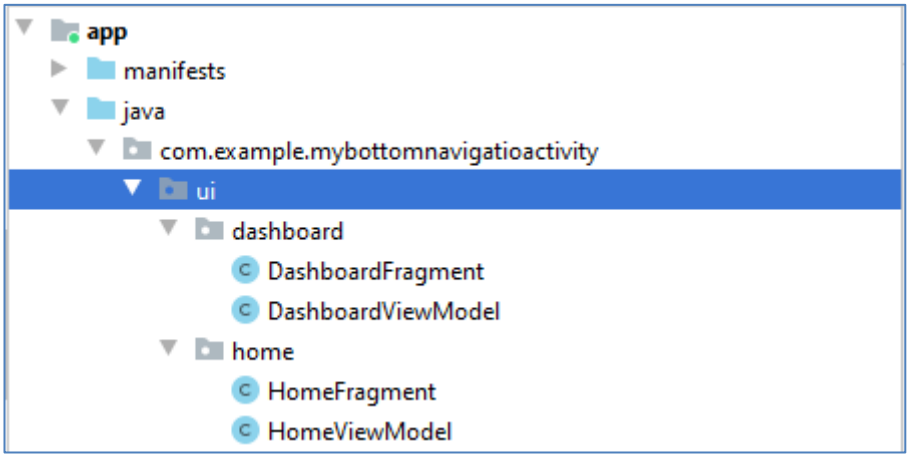


Abbildung 113 Eintrag "ui" aktivieren

- Nun rechte Maustaste anklicken
- Eintrag „New“
- Eintrag „Package“

Es erscheint folgendes Dialogfenster:

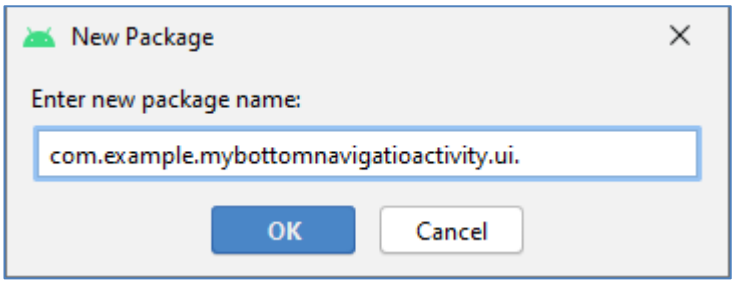


Abbildung 114 Neues Package erstellen

Nun eintragen: com.example.mybottomnavigatioactivity.ui.**add**

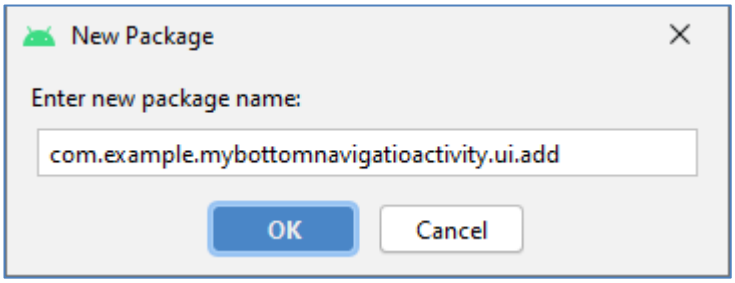


Abbildung 115 Neues Package "add" erstellen

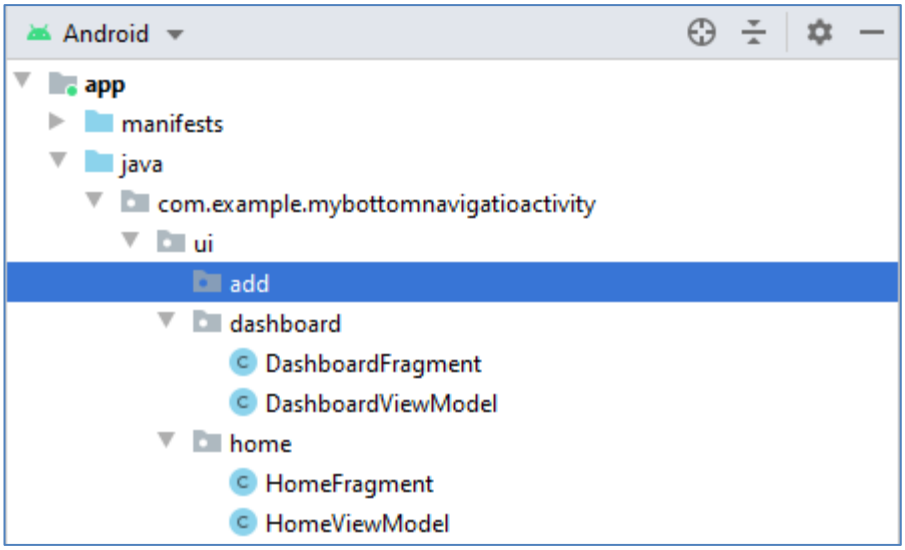


Abbildung 116 Das neue Package "add" wird eingetragen

Nun muss eine leeres Fragment erstellt werden.

11.1.5.3 layout/add_home.xml

- Anklicken des Eintrags „ui/add“
- Eintrag „New“
- Eintrag „Fragment“
- Eintrag „Fragment (Blank“

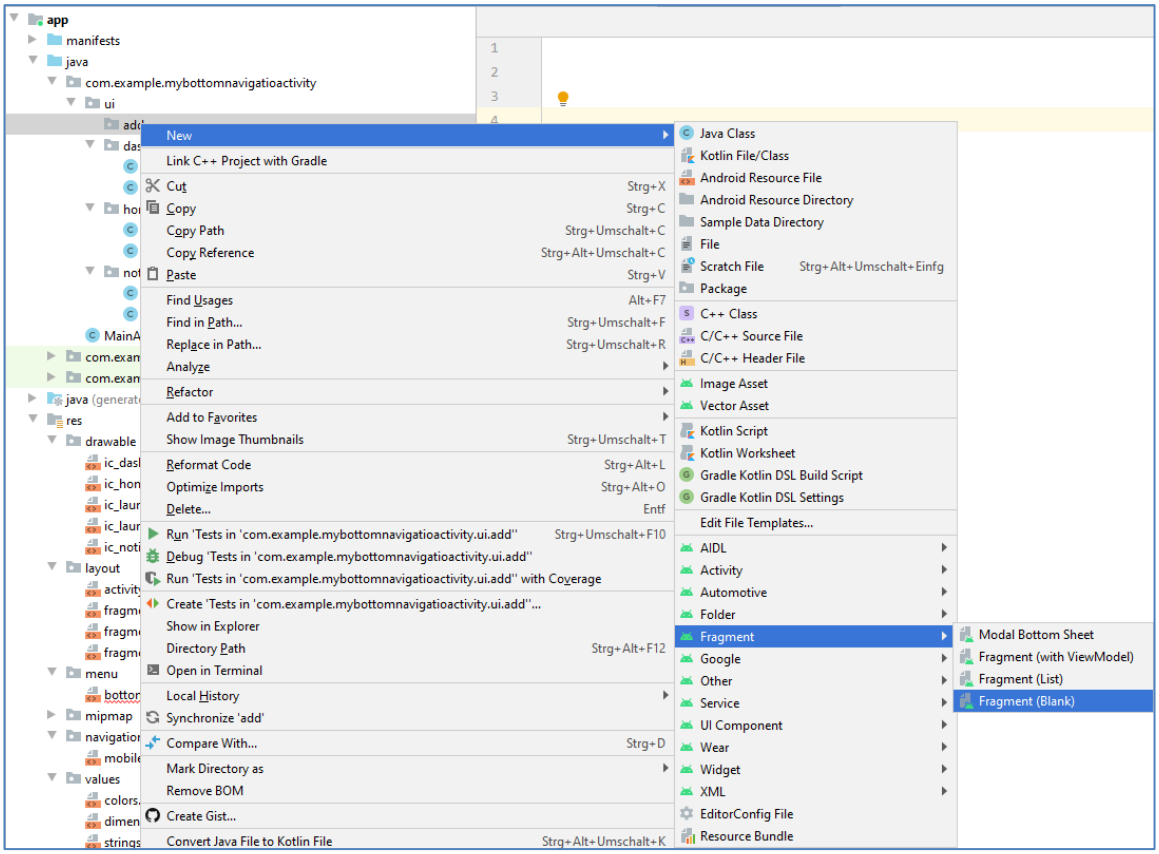


Abbildung 117 Einfügen eines neuen Fragmentes

Es erschenint folgendes Fenster:

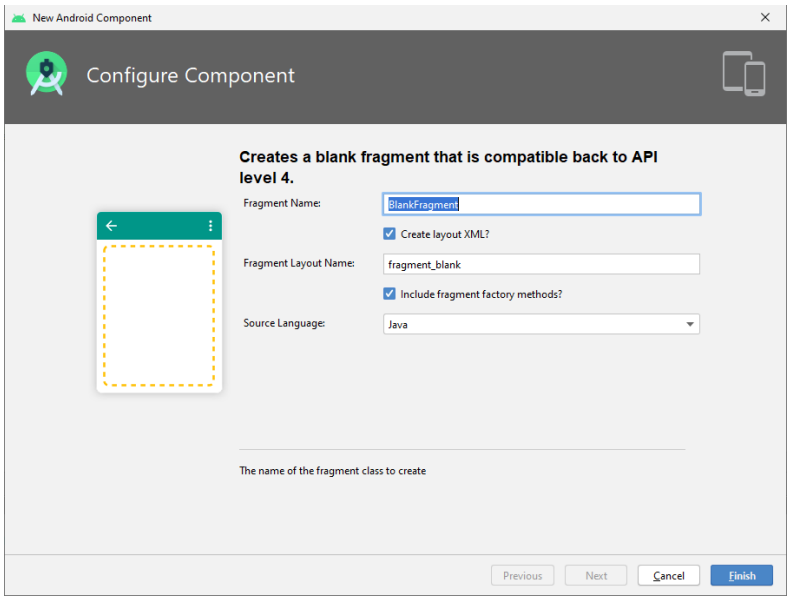


Abbildung 118 Dialog zum Eintragen des Namens des neuen Fragments

- Name: AddFragment
- Ohne „include fragment factory methods“

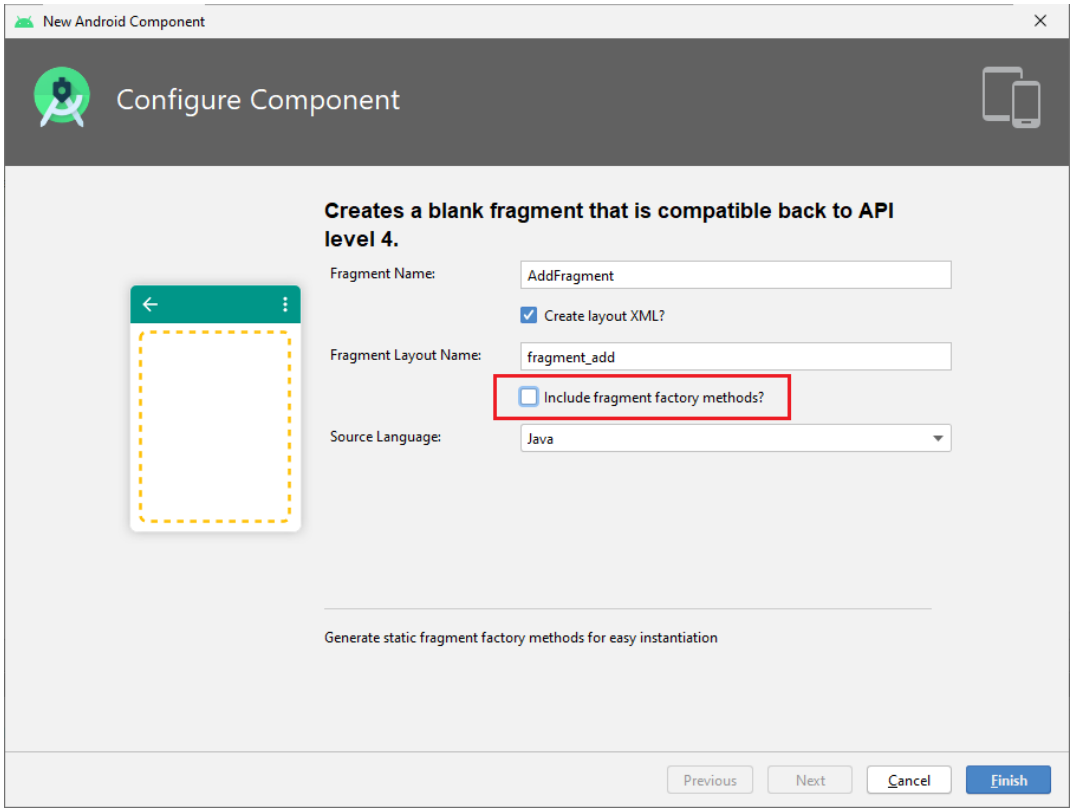


Abbildung 119 Eintragen des Namens des neuen Fragments

Hinweis:

- Falls Sie ein ViewModell benutzen wollen, müssen Sie den Eintrag „Fragment mit ViewModel“ auswählen.

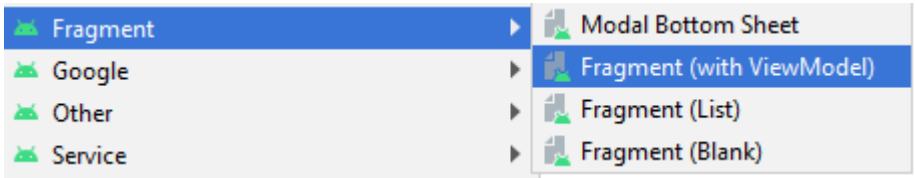


Abbildung 120 ViewModel Auswahl für ein neues Fragment

11.1.5.4 mobile_navigation.xml

In der Datei „mobile_navigation.xml“ wird nun ein neuer Eintrag „add“ eingefügt:

```
<fragment
    android:id="@+id/navigation_add"
    android:name="com.example.mybottomnavigatioactivity.ui.add.AddFragmnent"
    android:label="@string/title_add"
    tools:layout="@layout/fragment_add" />
```

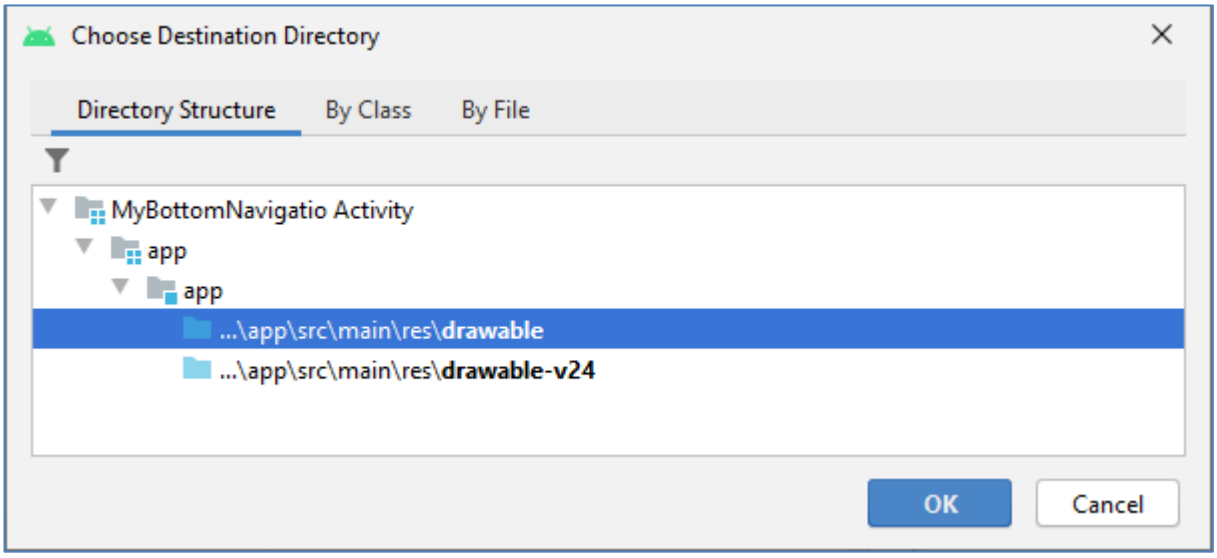
Zum einen die id	@+id/navigation_add
Dann das Package mit der JavaDatei:	mybottomnavigatioactivity.ui.add.AddFragment
Der Titel	title_home

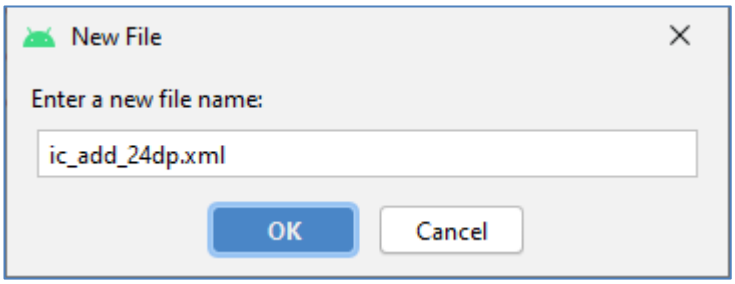
Und die xml-Datei fragment_add

11.1.5.5 drawable/ ic_add_black_24dp.xml

Im Baumeintrag/Ordner drawable wird nun das dazugehörige Symbol eingetragen:

- Anklicken des Eintrags „res/drawable“
- Rechte Maustaste
- Eintrag „New“
- Eintrag „File“
- Auswahl des Ordners „drawable“
- Name ic_add_24dp.xml

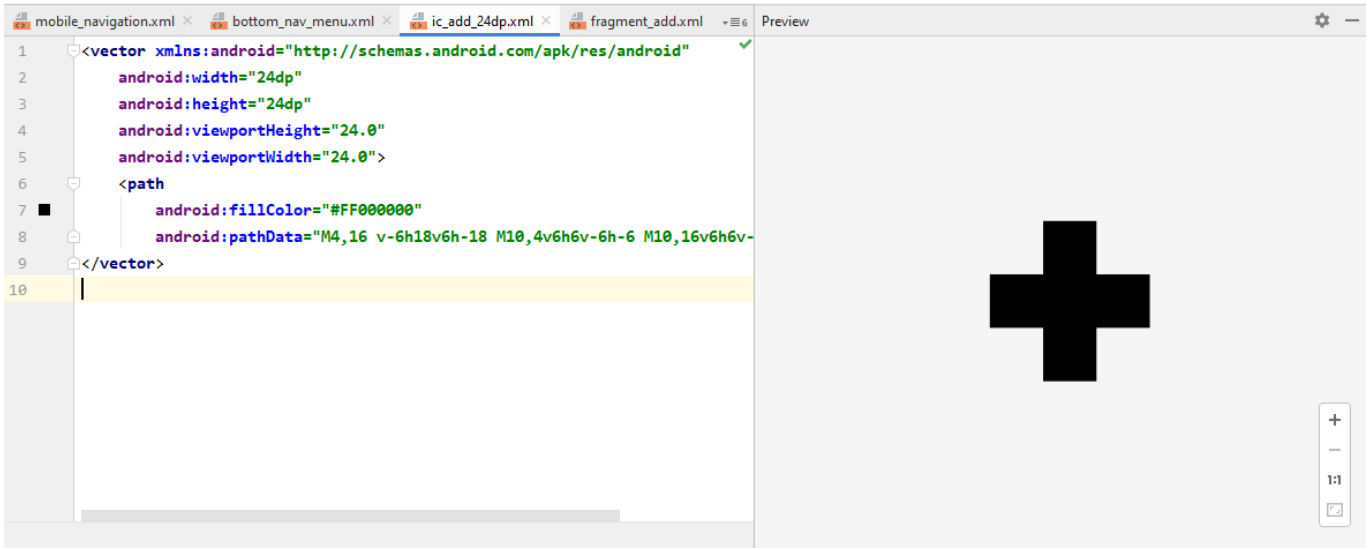




Nun den unteren Inhalte kopieren und einfügen:

ic_add_24dp.xml:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M4,16 v-6h18v6h-18 M10,4v6h6v-6h-6 M10,16v6h6v-6h-6  z"/>
</vector>
```



Alternativ kann man auch ein Symbol aus der vorgefertigten Symbolliste nehmen.

11.1.5.6 bottom_nav_menu.xml

Hier wird der Eintrag in der unteren Navigation-Bar eingefügt

```
<item
    android:id="@+id/add_home"
    android:icon="@drawable/ic_add_24dp"
    android:title="@string/title_add" />
```

Zum einen die id	@+id/navigation_add
Dann das Symbol	ic_add_24dp
Und die Bezeichnung	title_add

11.1.5.7 MainActivity

Nun muss noch das neue Fragment in die Liste der Fragemnte eingetragen werden:

Datei MainActivity.java

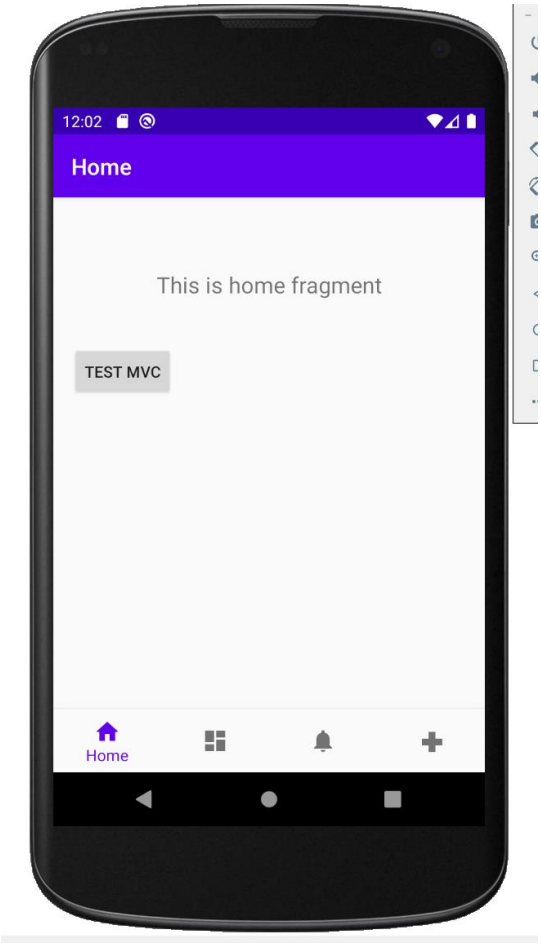
Alte Version:

```
AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
  R.id.navigation_home, R.id.navigation_dashboard,
  R.id.navigation_notifications)
.build();
```

Neue Version:

```
AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
  R.id.navigation_home, R.id.navigation_dashboard,
  R.id.navigation_notifications, R.id.navigation_add)
.build();
```

11.1.6 Testlauf



11.1.7 Add-Fragment als Start-Fragemnt setzen.

11.1.7.1 Setzen des Startfragments

- Öffnen der Datei „mobile_navigation.xml“
- Ändern des Eintrags:

```
app:startDestination="@+id/navigation_home"
```

in

```
app:startDestination="@+id/navigation_add"
```

11.1.7.2 Setzen des Addfragments an den Anfang

- Öffnen der Datei „bottom_nav_menu.xml“.
- Verschieben Sie den item-Block vor den navigation_home-Block

11.1.8 Aktueller Stand

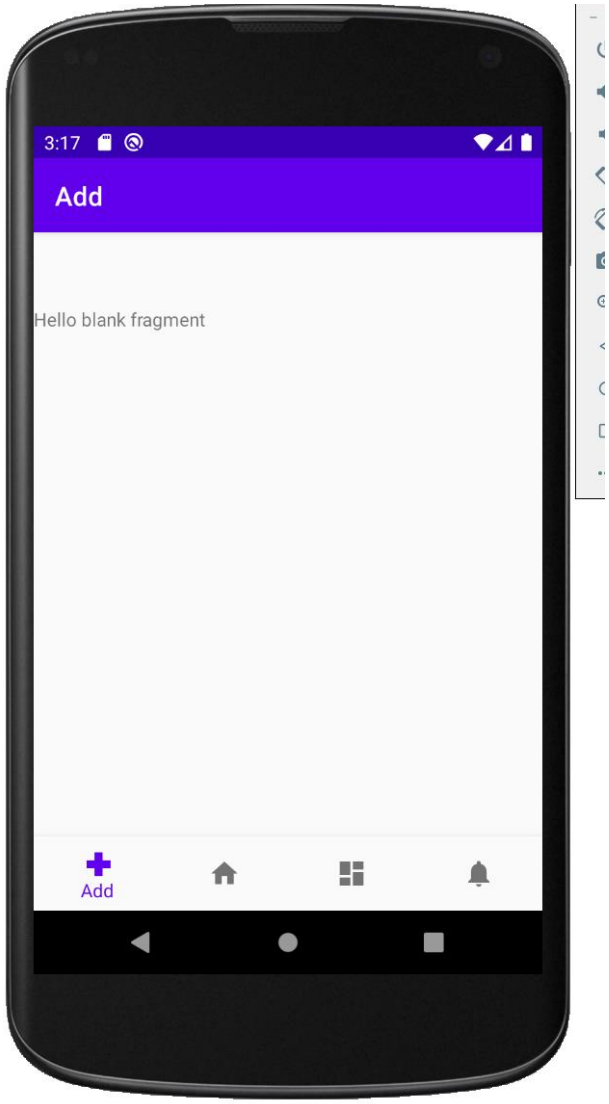


Abbildung 121 Aktueller Stand des Bottom-Navigation-Beispiels (mit Add-Fragment)

11.1.9 Weitere Symbole

Dateiname: ic_add_24dp

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M4,16 v-6h18v6h-18 M10,4v6h6v-6h-6 M10,16v6h6v-6h-6  z"/>
</vector>
```

Dateiname: ic_sub_24dp

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
```

```
        android:fillColor="#FF000000"
        android:pathData="M4,15 v-6h18v6h-18 z" />
</vector>
```

Dateiname: ic_mult_24dp

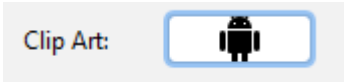
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M19,6.41L17.59,5 12,10.59 6.41,5 5,6.41 10.59,12 5,17.59 6.41,19
12,13.41 17.59,19 19,17.59 13.41,12z"/>
</vector>
```

11.1.10 Icon aus der Androidliste einfügen

- Anklicken des Eintrags „res“
- Anklicken des Eintrags „drawable“
- Menü „File“
- Eintrag „New“

11.1.10.1 Erstellen eines „Plus“-Icons

- Auswahl des Plus-Symbol
- Aktivieren des Eintrags „res/drawable“
    - Anklicken des Eintrags „res“
    - Anklicken des Eintrags „drawable“
    - Menü „File“
        - Eintrag „New“
        - Eintrag „Vector Asset“



        - Anklicken des Clip Art Symbol (Android“
        - Anklicken des „+“-Symbols

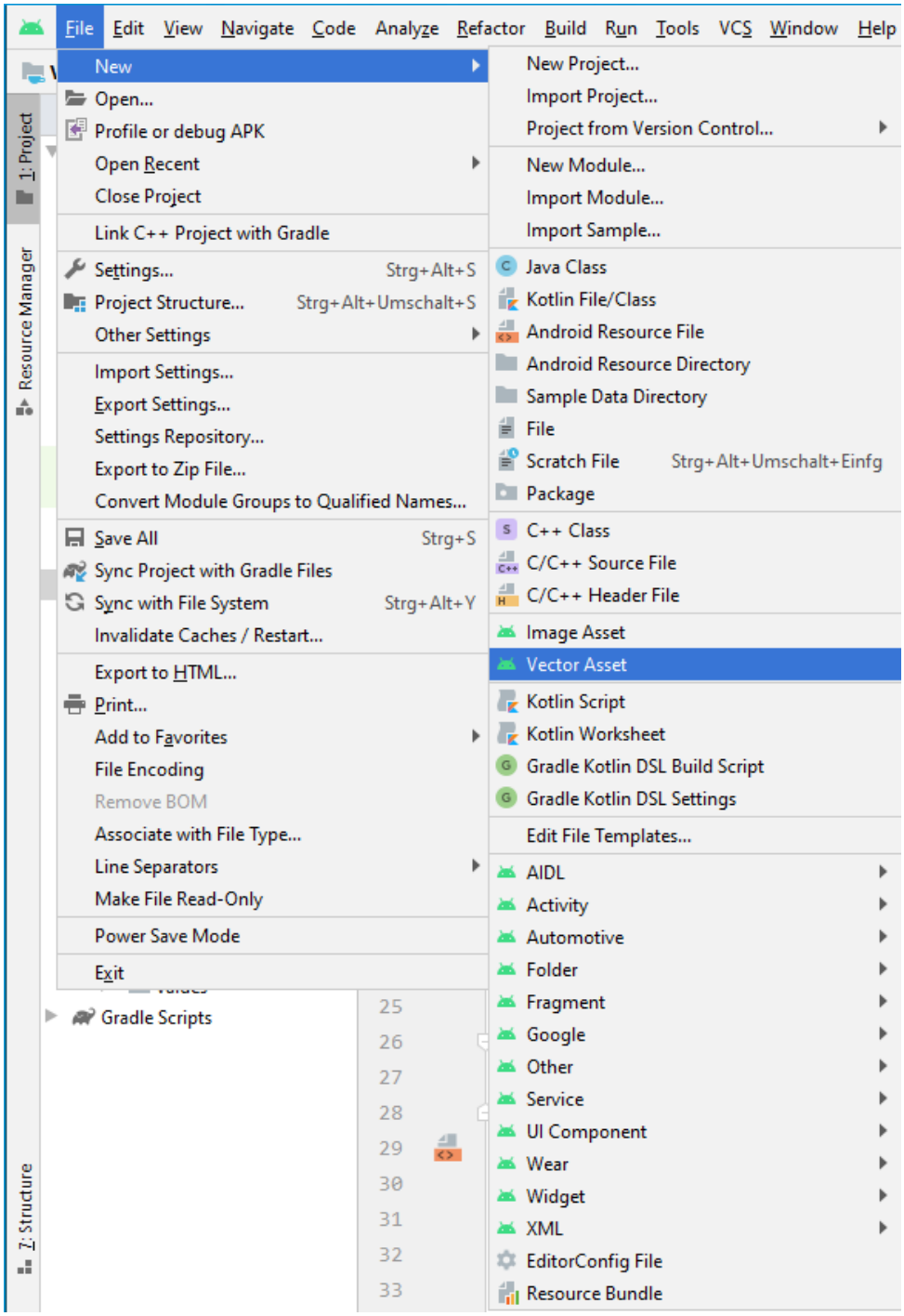


Abbildung 122 Auswahl des Menueintrags „drawable Symbol“

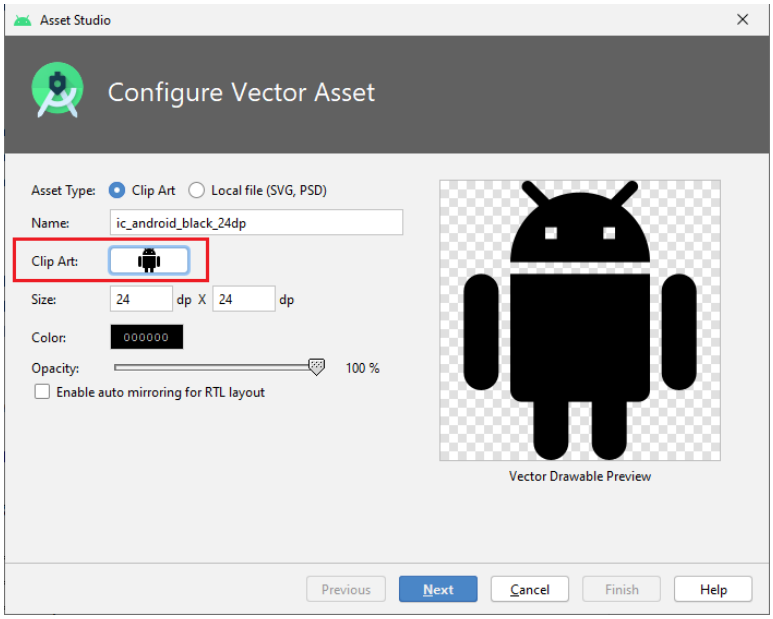


Abbildung 123 Auswahldes neuen Vektore-Symbols

Weiter mit dem Anklicken des kleinen Android-Symbols.

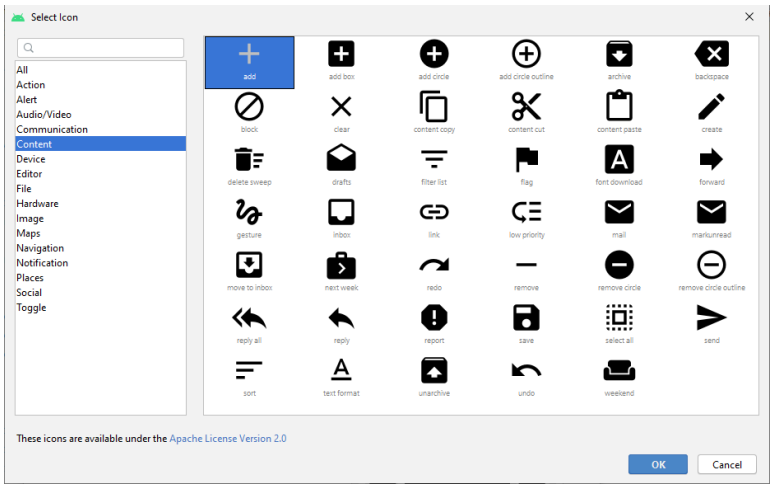


Abbildung 124 Auswahl des Plus-Symbols

Am Schluss zweimal „ok“ drücken.

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="@string/title_home" />

    <item
        android:id="@+id/navigation_dashboard"
        android:icon="@drawable/ic_dashboard_black_24dp"
        android:title="@string/title_dashboard" />

    <item
        android:id="@+id/navigation_notifications"
        android:icon="@drawable/ic_notifications_black_24dp"
```

```
        android:title="@string/title_notifications" />

</menu>
```

Nun fehlen noch die ui-Elemente im Add-Fragment und der Java-Code.

11.1.11 Texte für die Fragmente

```
<string name="title_add">Add</string>
<string name="title_sub">Sub</string>
<string name="title_mult">Mult</string>
<string name="title_div">Div</string>

<string name="captionadd">Addition</string>
<string name="captionsub">Substraktion</string>
<string name="captionmult">Multiplikation</string>
<string name="captiondiv">Division</string>

<string name="number1">1. Zahl</string>
<string name="number2">2. Zahl</string>
<string name="number3">Ergebnis</string>

<string name="bntest">Testen</string>
<string name="bnagain">Noch einmal</string>
<string name="comment">Kommentar</string>
<string name="wrongnumberformat">Bitte eine korrekte Zahl eingeben</string>
<string name="wrongnumber">Falsche Lösung</string>
<string name="correctnumber">Korrekte Lösung</string>
```

11.1.12 fragment_add aufbauen

Aktueller Inhalt:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
    tools:context=".ui.add.AddFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/hello_blank_fragment" />

</FrameLayout>
```

Nun muss man dieses Framelayout in einen Constraint-Layout umwandeln:

Vorgehensweise:

- Umschalten in den kompletten Grafikmodus .

- Anklicken des Eintrags „FrameLayout“
- Der Cursor muss über der Schrift liegen!
- Rechte Maustaste: „Convert FrameLayout to ConstraintLayout“

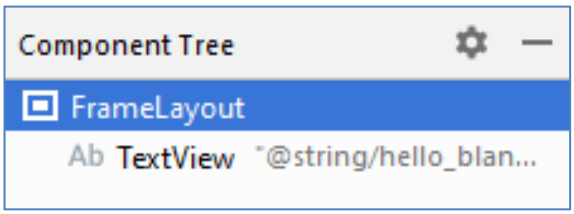


Abbildung 125 Framelayout umwandeln in ein Constraint-Layout

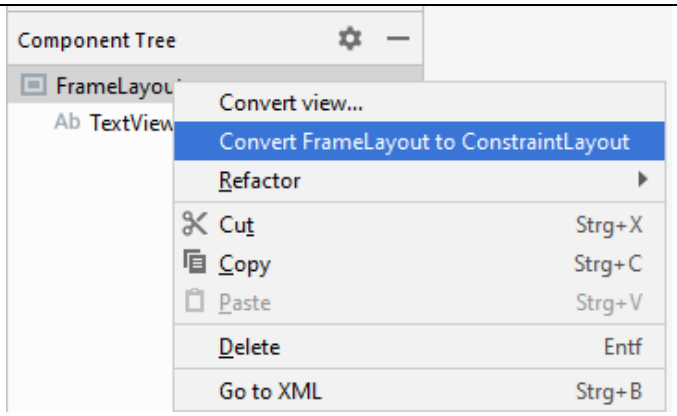


Abbildung 126 Popupmenu zum Umwandeln ins Constraint-Layout

Änderungsmeldung:

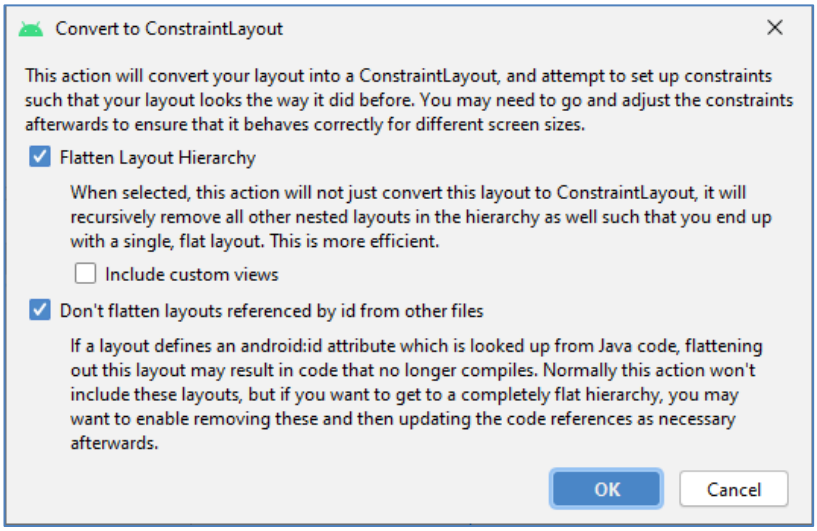


Abbildung 127 Änderungsmeldung bezüglich des Constraint-Layout

OriginalLayout:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
  tools:context=".ui.add.AddFragment">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="@string/hello_blank_fragment" />
```

```
</FrameLayout>
```

Constraint-Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".ui.add.AddFragment">

  <TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:text="@string/hello_blank_fragment"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

11.1.13 Ändern des activity_main-Layout:

In der Datei „activity_main.xml“ muss man nun noch eine Kleinigkeit ändert:

Aktueller Quellcode:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">
```

Das PaddingTop ist überflüssig. Dadurch wird eine zusätzliche Höhe am Anfang des Fragment eingetragen (siehe untere Abbildung, gelebes Rechteck):



Abbildung 128 actionBarSize

Geänderter Quellcode:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
```



Abbildung 129 Bottom-Navigation ohne ActionBar-Padding

11.1.14 Quellcode von fragment_add.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".ui.add.AddFragment">

  <TextView
    android:id="@+id/tvtitleadd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="@string/captionadd"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#1E88E5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvnumber1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="32dp"
    android:text="@string/number1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitleadd" />

  <EditText
    android:id="@+id/tnumber1"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginStart="64dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
    app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

  <TextView
    android:id="@+id/tvoperation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="+"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintEnd_toEndOf="@+id/tnumber1"
    app:layout_constraintStart_toStartOf="@+id/tnumber1"
    app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

  <TextView
    android:id="@+id/tvnumber2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/number2"
    app:layout_constraintStart_toStartOf="@+id/tvnumber1"
    app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

  <EditText
    android:id="@+id/tnumber2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="number"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
    app:layout_constraintEnd_toEndOf="@+id/tnumber1"
    app:layout_constraintStart_toStartOf="@+id/tnumber1" />

  <TextView
    android:id="@+id/tvequation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="="
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintEnd_toEndOf="@+id/tnumber1"
    app:layout_constraintStart_toStartOf="@+id/tnumber1"
    app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

  <TextView
    android:id="@+id/tvnumber3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="@string/number3"
    android:textColor="#D81B60"
    app:layout_constraintStart_toStartOf="@+id/tvnumber2"
    app:layout_constraintTop_toBottomOf="@+id/tvequation" />

  <EditText
    android:id="@+id/tnumber3"
    android:layout_width="0dp"
```

```
    android:layout_height="wrap_content"
    android:inputType="number"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
    app:layout_constraintEnd_toEndOf="@+id/tnumber2"
    app:layout_constraintStart_toStartOf="@+id/tnumber2" />

  <LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tnumber3">

  <Button
    android:id="@+id/bntest"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="5dp"
    android:layout_weight="1"
    android:text="@string/bntest" />

  <Button
    android:id="@+id/bnagain"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:layout_weight="1"
    android:text="@string/bnagain" />
  </LinearLayout>

  <TextView
    android:id="@+id/tvcomment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:text="TextView"
    android:textColor="#24A598"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

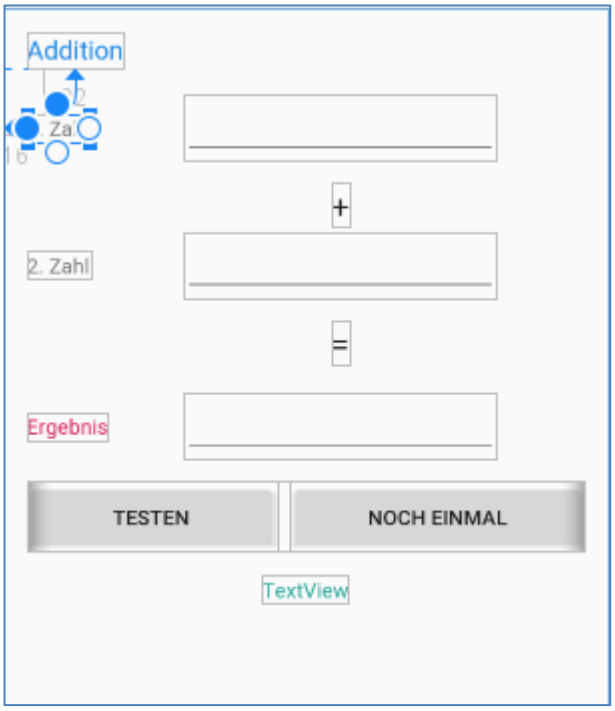


Abbildung 130 Struktur der fragment-add-xml-Datei

Das Programm erzeugt zwei Zufallszahlen, der Anwender muss die Summe im Ergebnis eintragen.

11.1.15 Add-Fragment-Javacode

Bitte beachten Sie, dass man die on-Click-Events der Schalter mit Code-Behind erstellen muss.

```
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.*;
import android.widget.*;

  // Da die Fragmente in einem Extra-Package liegen,
  // muss man die Basis.java und MainActivity.java hier eintragen
import com.example.mybottomnavigatioactivity.Basis;
import com.example.mybottomnavigatioactivity.MainActivity;
import com.example.mybottomnavigatioactivity.R;

import java.util.Random;

import android.view.inputmethod.InputMethodManager;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;


public class AddFragment extends Fragment {

    private static final String TAG = MainActivity.class.getSimpleName();
```

```
    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;

    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public AddFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_add, container,
false);

        tvtitle = view.findViewById(R.id.tvtitleadd);

        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);

        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1+zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
```

```
        tnumber3.setText("?");

        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }


    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tvcomment.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }

    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.1.16 Klasse Basis.java

Bitte den Quellcode im Kapitel 19.8, Seite 368.

11.1.17 Tastatur entfernen

- Wenn mandie Tastatur beim Anklicken des „Hakens“ verschwinden lassen will, muss man mit einem „InputMethodManager“ und der Methode „getSystemService“ arbeiten.
- Dabei ist zu beachten, dass die Fragmente keine Activitäten sind. Das „this“ ist die falsche Klasse.
- Deshalb wird in MainActivity derInputManager statisch deklariert, so dass man in den Fragmenten daruaf zugreifen kann.

11.1.18 MainActivity.java

Eintragen der statischen Variable:

```
public static InputMethodManager in=null;
```

Die Zuweisung eintragen in onCreate:

```
MainActivity.in = (InputMethodManager) this.getSystemService(Context.INPUT_METHOD_SERVICE);
```

11.1.19 AddFragment.java

```
tnumber3.setOnEditorActionListener(new TextView.OnEditorActionListener() {
  @Override
  public boolean onEditorAction(TextView v, int actionId, KeyEvent event){

    if (actionId == EditorInfo.IME_ACTION_GO) {
      performYourAction("go");
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_DONE) {
      performYourAction("done");  // Haken
      // Aufruf der statischen Variable
      MainActivity.in.hideSoftInputFromWindow(
        tnumber3.getApplicationWindowToken(),
        InputMethodManager.HIDE_NOT_ALWAYS);
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_NEXT) {
      performYourAction("next");
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_SEND) {
      performYourAction("send");
      return true;
    }

    return false;
  }
  });
  return view;
}

private void performYourAction(String item) {
  Log.i(TAG,"************  performYourAction: "+item);
  //Basis.showToast(null,"performYourAction: "+item);
}
```

11.1.20 Musterlösung

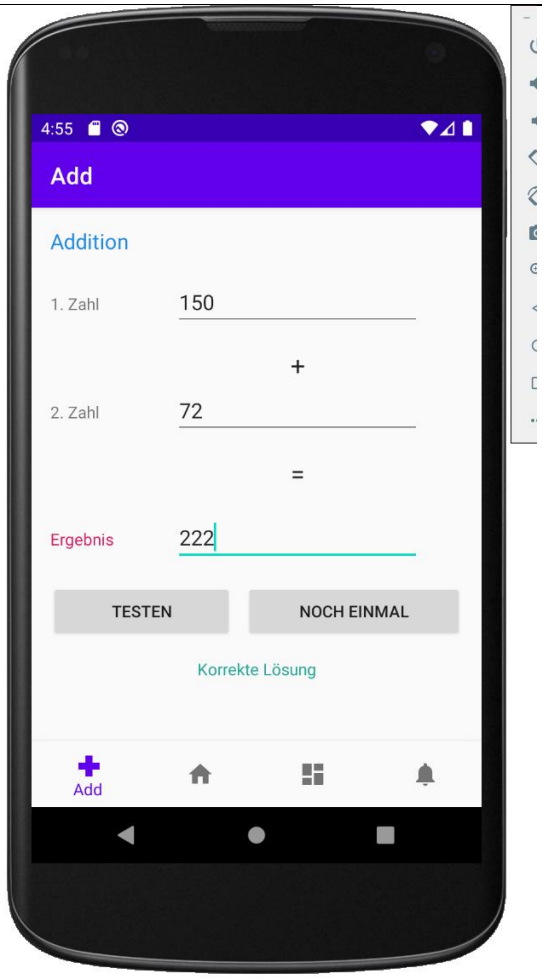


Abbildung 131 Anzeige der Meldung bei korrekter Lösung

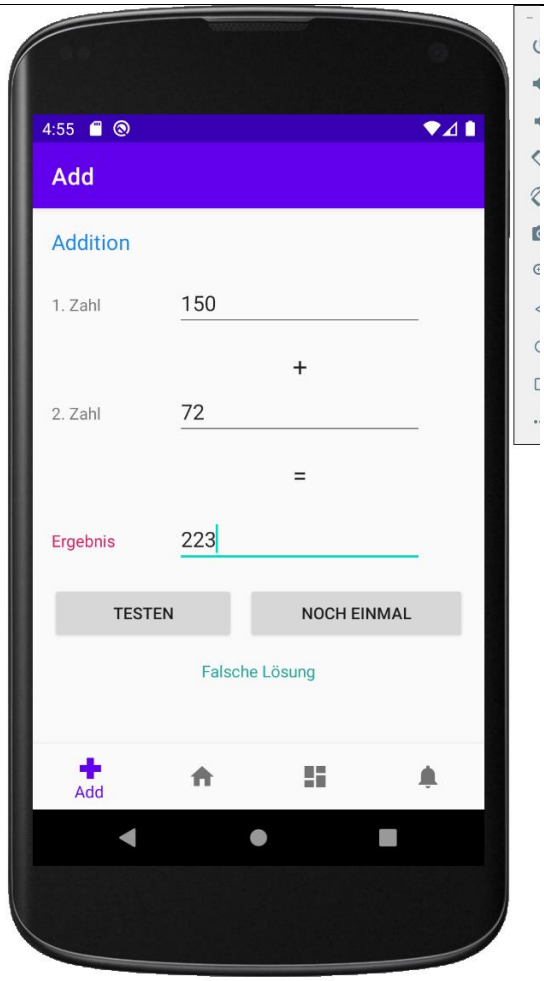


Abbildung 132 Anzeige der Meldung bei falscher Lösung

11.2 Tabbed Activity

Dieses Kapitel zeigt, wie man die App „AddSubMult“ in einem Tabbed Activity einfügen kann.
Bei einer Tabbed Activity werden die Menüs oben angezeigt.



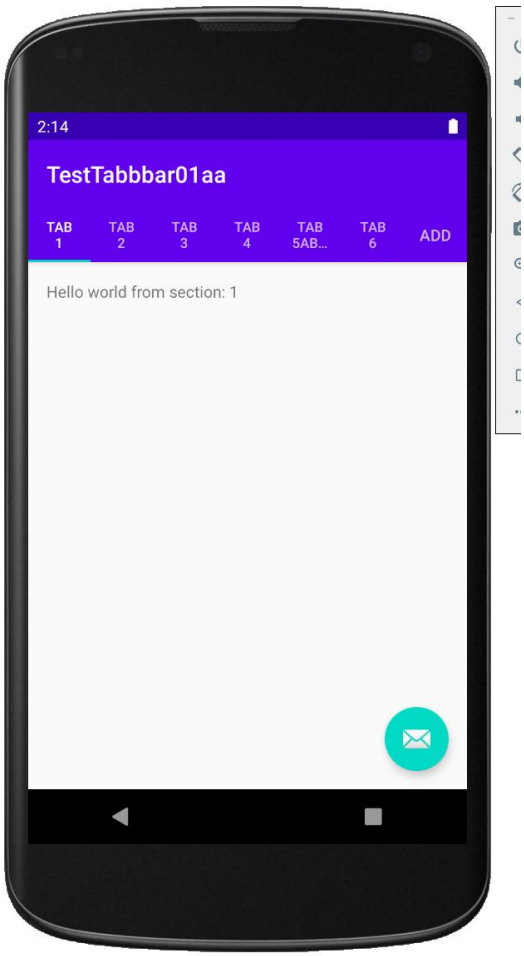


Abbildung 133 Ansicht des Vorlagen-Projektes (leicht verändert)

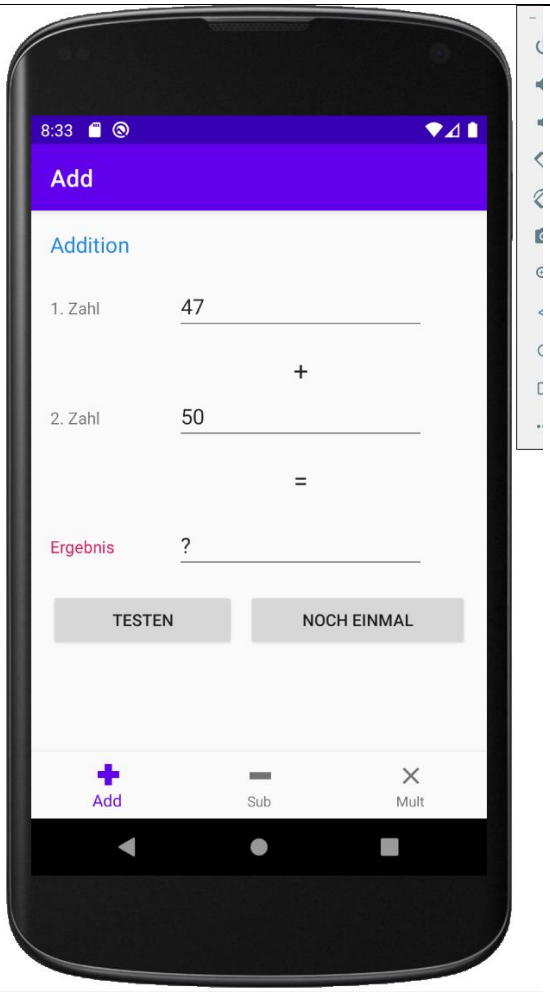


Abbildung 134 Fertiges Projekt

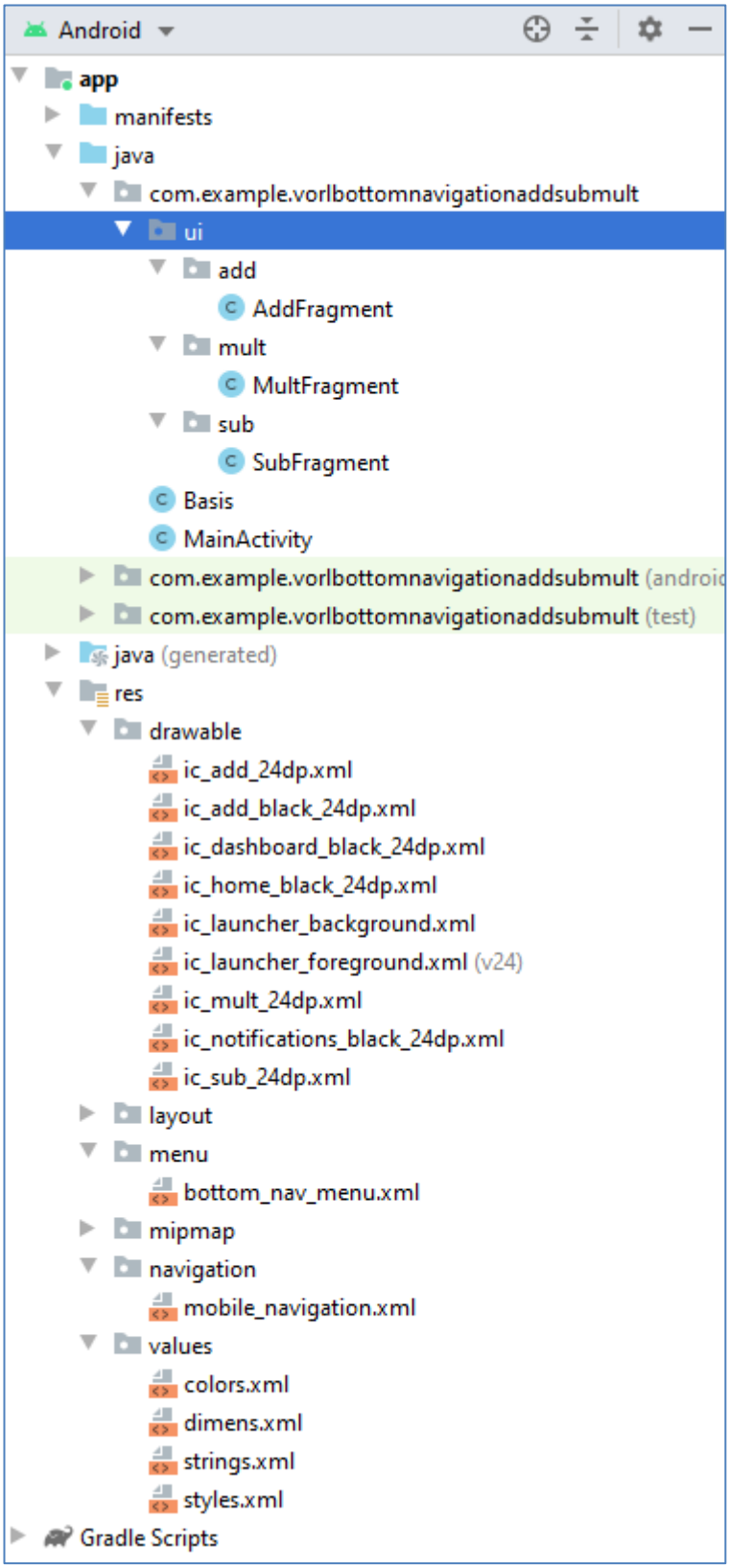


Abbildung 135 Komplette Struktur der Musterlösung

11.2.1 Neues Projekt erstellen

Als Vorlage wird die Tabbed Activity ausgewählt.

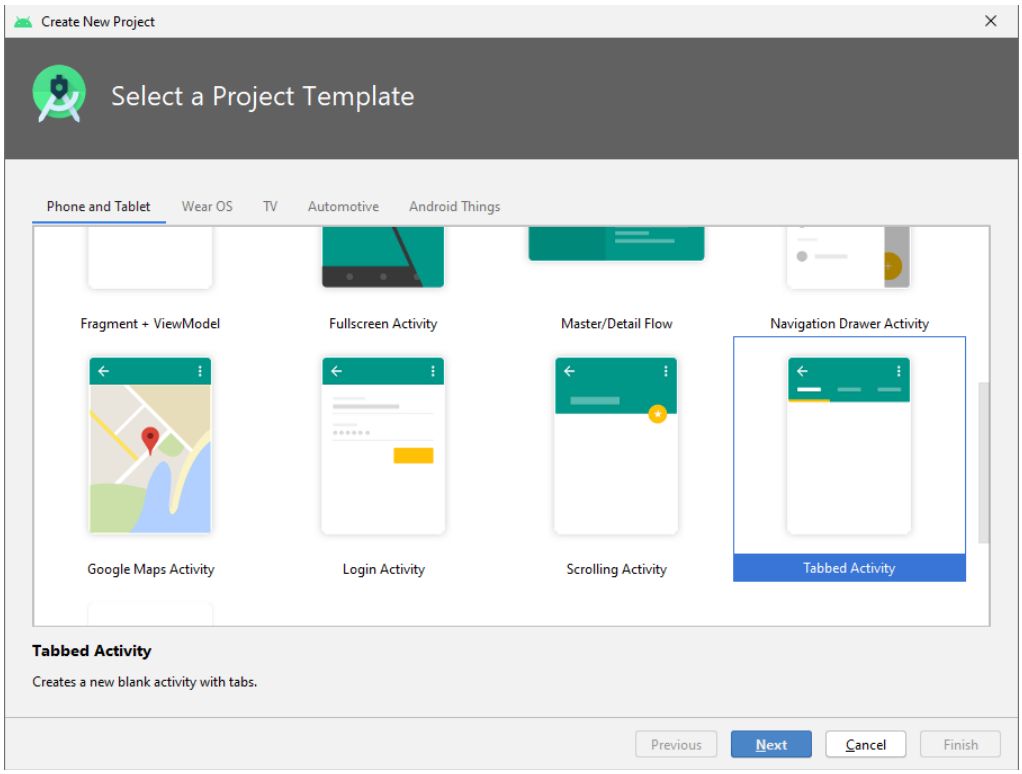
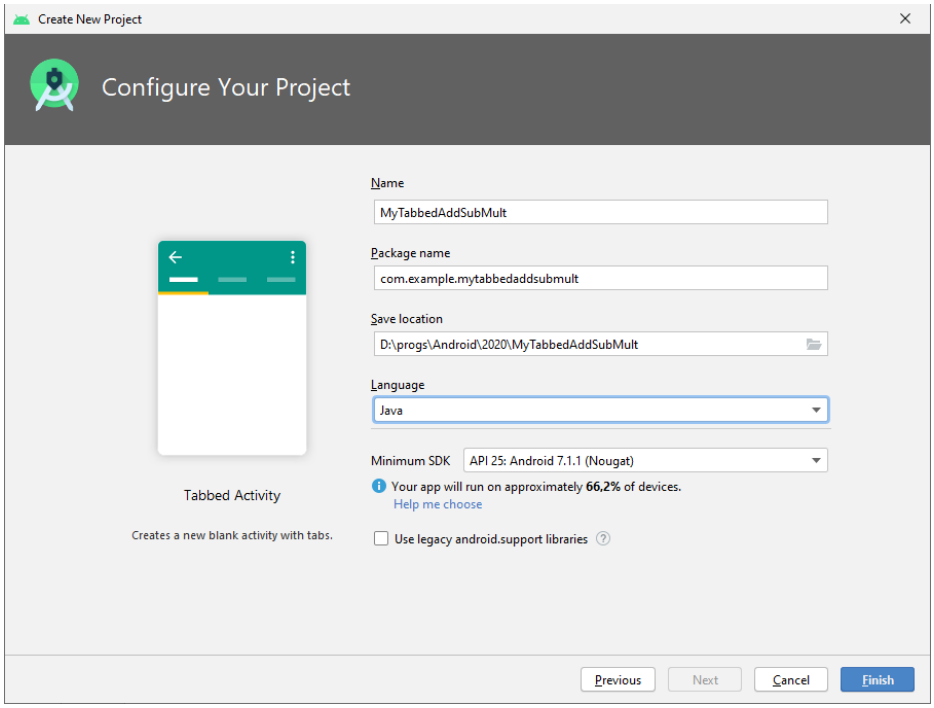


Abbildung 136 Bottom Navigation Activity

Der Projektname lautet: MyTabbedAddSubMult



11.2.2 Struktur des Vorlagen-Projektes

Die Struktur hat sich zum letzten Jahr geändert.

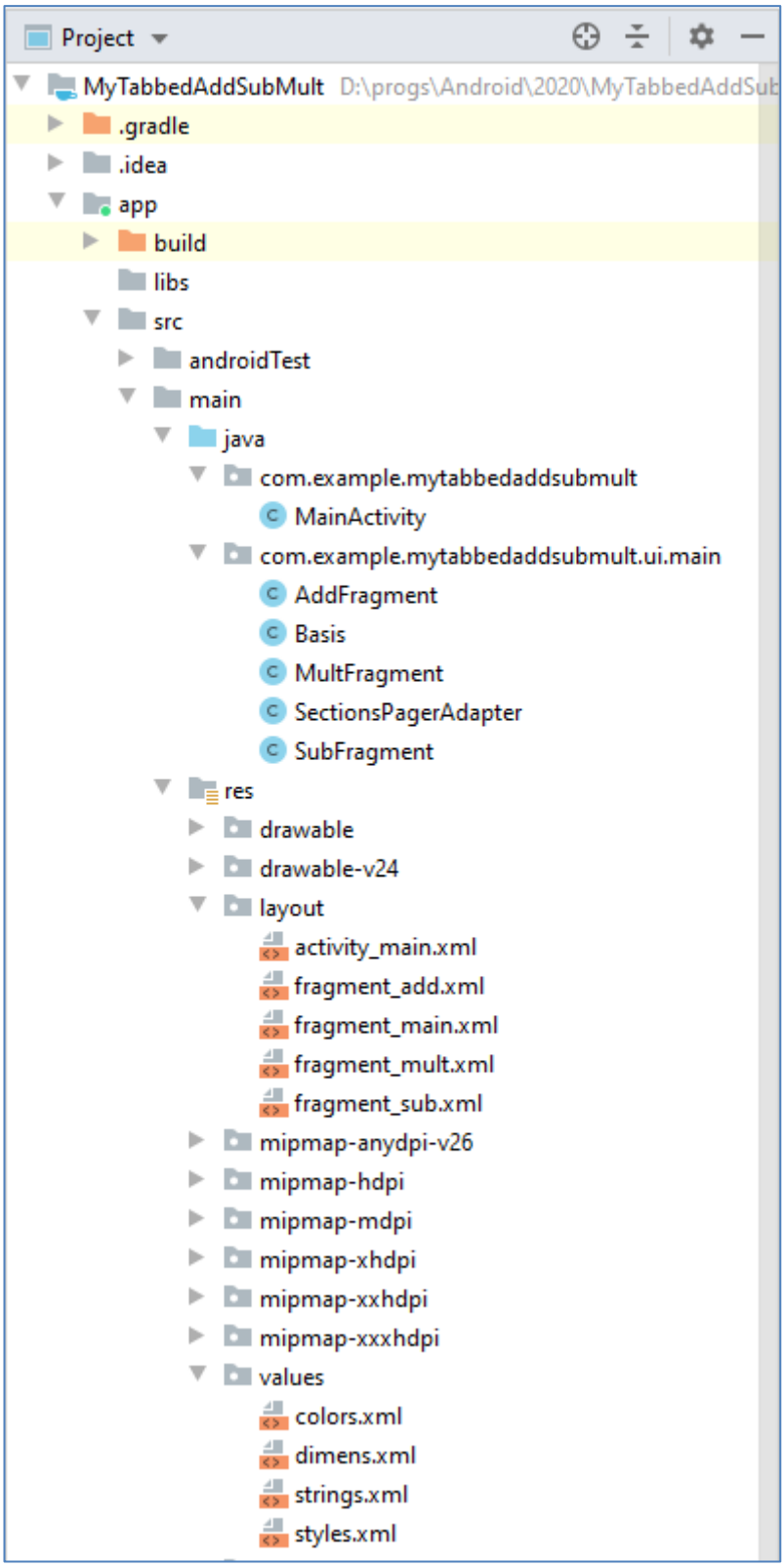


Abbildung 137 Vorlagenstruktur des Tabbed Projektes

- Die Verwaltung der Seiten geschieht hier mit einem „SectionPageAdapter“.
- Die Vorlage ist leider nicht sehr glücklich gewählt. Es gibt nur **EIN** Fragment, bei dem der Text der TextView geändert wird. Die Struktur von mehreren Fragmenten ist nicht sichtbar.
- Jede Variante ist ein Fragment und keine Activity!
- Man muss also pro Variante ein neues Fragment erstellen.
- Jedes Fragment besteht aus einer xml-Datei und einer Java-Datei.

- Die neuen Dateien, inklusive Basis.java, werden alle in das Package „ui.main“ eingetragen.

11.2.3 Plätze der Fragmente

Im Eintrag „layout“ sind wie immer alle Fragment-xml-Dateien eingetragen.

Die Daten der Fragmente sind auf nur vier „Plätze“ verteilt:

- strings
- fragment.java
- fragment.xml
- sectionpageadapter

11.2.4 strings.xml

```
<resources>
    <string name="app_name">MyTabbedAddSubMult</string>
    <string name="tab_text_1">Tab 1</string>
    <string name="tab_text_2">Tab 2</string>
</resources>
```

In der Datei string.xml werden alle Titel eingetragen. Diese Tab-Bezeichnungen werden oben in der App-Bar angezeigt. Sinnvoll sind aber nur drei bis sieben Fragmente.

Neuer Inhalt:

```
<resources>
  <string name="app_name">TabbedBarAddSubMultOptionsmenu</string>

  <string name="tab_text_1">Tab 1</string>
  <string name="tab_text_2">Tab 2</string>
  <string name="tab_add">Add</string>
  <string name="tab_add">Sub</string>
  <string name="tab_add">Mult</string>

  <string name="captionadd">Addition</string>
  <string name="captionsub">Substraktion</string>
  <string name="captionmult">Multiplikation</string>
  <string name="captiondiv">Division</string>
  <string name="number1">1. Zahl</string>
  <string name="number2">2. Zahl</string>
  <string name="number3">Ergebnis</string>
  <string name="bntest">Testen</string>
  <string name="bnagain">Noch einmal</string>
  <string name="comment">Kommentar</string>
<string name="wrongnumberformat">Bitte eine korrekte Zahl eingeben</string>
  <string name="wrongnumber">Falsche Lösung</string>
  <string name="correctnumber">Korrekte Lösung</string>

</resources>
```

11.2.5 Anlegen des neuen AddFragment

Das BottomBar-Activity erfordert ein jeweiliges Package. Bei der Tabbed Bar gibt es Package, ui.main. Deshalb sollte man die neuen Fragmenten in dises Package eintragen.

11.2.6 strings.xml

Der Tab-Eintrag ist schon oben eingetragen:

```
<string name="tab_add">Add</string>
<string name="captionadd">Addition</string>
```

11.2.7 Einfügen des AddFragmentes mit dem Menü.

- Anklicken des Eintrags „SectionsPagerAdapter“ im Eintrag „ui.main“.
- Eintrag „New“
- Eintrag „Fragment“
- Eintrag „Fragment (Blank)“

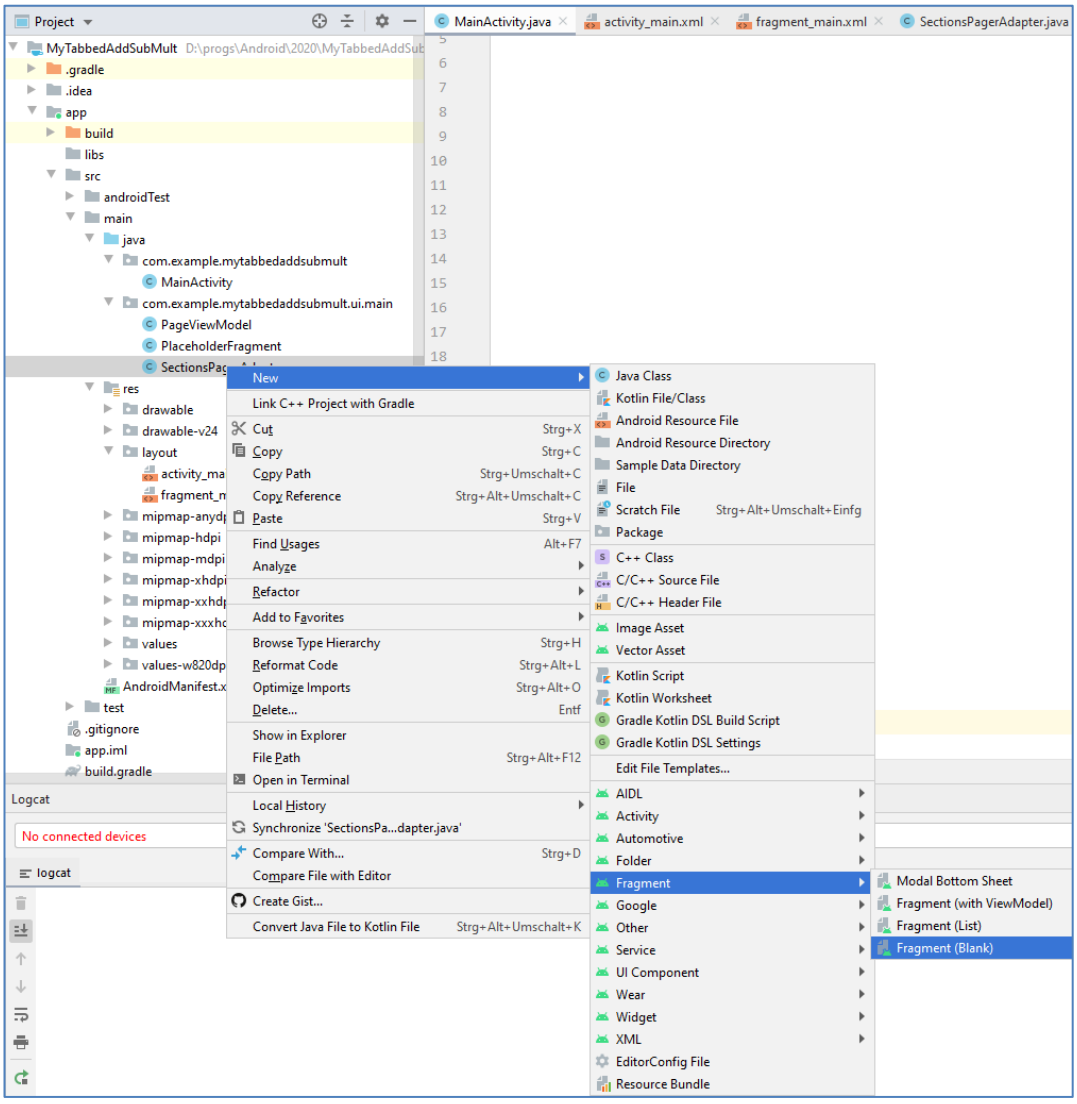


Abbildung 138 Einfügen eines neuen Fragments

Es erschenint folgendes Fenster:

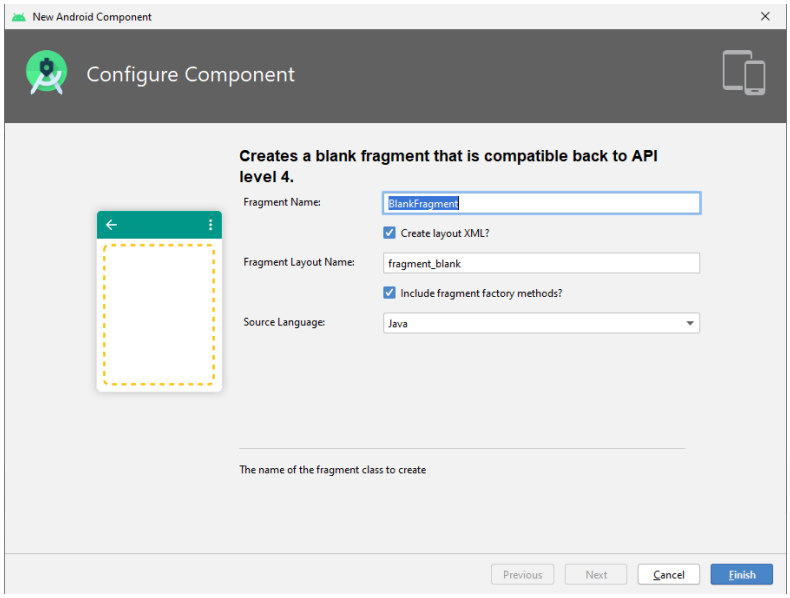


Abbildung 139 Dialog zum Eintragen des Namens des neuen Fragments

- Name: AddFragment
- Ohne „include fragment factory methods“

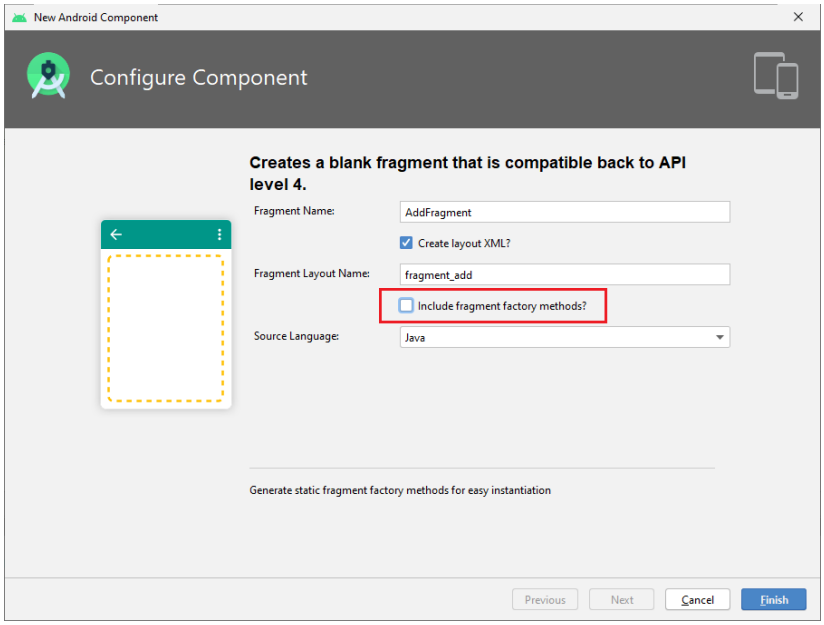


Abbildung 140 Eintragen des Namens des neuen Fragments

Hinweis:

- Falls Sie ein ViewModell benutzen wollen, müssen Sie den Eintrag „Fragment mit ViewModel“ auswählen.



Abbildung 141 ViewModel Auswahl für ein neues Fragment

11.2.8 SectionsPagerAdapter

11.2.8.1 Tab‘s

In Eintrag „SectionsPagerAdapter“ werden die Tab’s verwaltet.

Änderung im Quellcode:

Alte Variante:

```
private static final int[] TAB_TITLES = new int[]{
  R.string.tab_text_1,
  R.string.tab_text_2
};
```

Neue Variante:

```
@StringRes
private static final int[] TAB_TITLES = new int[]{
  R.string.tab_text_1,
  R.string.tab_text_2,
  R.string.tab_add

};
```

Sub und mult kommen später noch dazu.

11.2.8.2 getItem

Alter Quellcode:

```
@Override
public Fragment getItem(int position) {
  return PlaceholderFragment.newInstance(position + 1);
}
```

Der obige Quellcode zeigt nicht unbedingt das Verfahren der TabbedBar.

Neuer Quellcode:

```
@Override
public Fragment getItem(int position) {
  switch (position) {
    case 0:
    case 1:
      return PlaceholderFragment.newInstance(position + 1);

    case 2:
      AddFragment fragment = new AddFragment();
      return fragment;

  }  // switch
  return null;
}
```

11.2.8.3 getCount

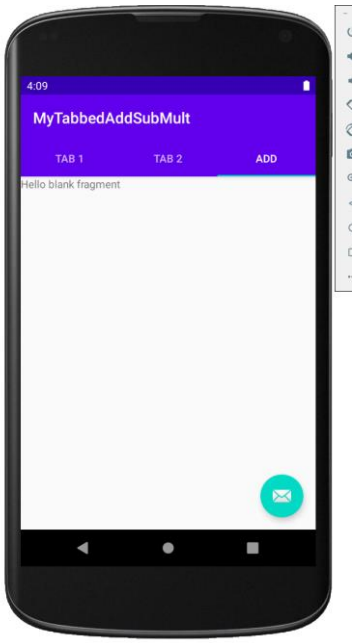
Bitte unbedingt ändern:

```
@Override
public int getCount() {
   return 2;
}
```

Der obige Code ist ein SEHR schlechter Quellcode !!

```
@Override
public int getCount() {
   return TAB_TITLES.length;
}
```

11.2.9 Testlauf



11.2.10 Eintragen der XML-Daten in fragment_add.xml

Hier muss wieder der Context auf das package gesetzt werden:

```
tools:context="ui.main.AddFragment"
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.main.AddFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitleadd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionadd"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/number1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitleadd" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="+"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
```

```
        android:layout_height="wrap_content"

        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

11.2.11 Eintragen des Quellcodes in Addfragment.java

```
package com.example.mytabbedaddsubmult.ui.main;

import com.example.mytabbedaddsubmult.MainActivity;
import com.example.mytabbedaddsubmult.R;


import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import java.util.Random;

import android.view.inputmethod.InputMethodManager;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;

/**
 * A simple {@link Fragment} subclass.
 */
public class AddFragment extends Fragment {

    private static final String TAG = MainActivity.class.getSimpleName();
```

```
    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;

    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public AddFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_add, container,
false);

        tvtitle = view.findViewById(R.id.tvtitleadd);
        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);


        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();


        tnumber3.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
            @Override
            public boolean onEditorAction(TextView v, int actionId,
KeyEvent event) {
                if (actionId == EditorInfo.IME_ACTION_GO) {
```

```
                    performYourAction("go");
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_DONE) {
                    performYourAction("done");  // Haken

MainActivity.in.hideSoftInputFromWindow(tnumber3.getApplicationWindowToken(
),InputMethodManager.HIDE_NOT_ALWAYS);
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_NEXT) {
                    performYourAction("next");
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_SEND) {
                    performYourAction("send");
                    return true;
                }
                return false;
            }
        });
        return view;
    }

    private void performYourAction(String item) {
        Log.i(TAG,"************  performYourAction: "+item);
        //Basis.showToast(null,"performYourAction: "+item);
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1+zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");

        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }



    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tvcomment.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }
```

```
    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.2.12 Klasse Basis.java

Bitte den Quellcode im Kapitel 19.8, Seite 368.

11.2.13 Aktueller Stand

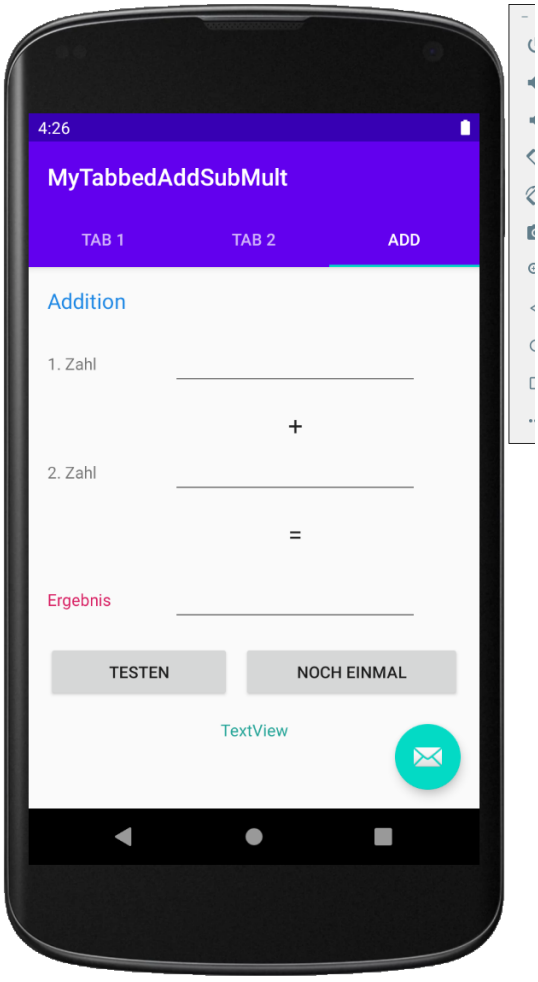
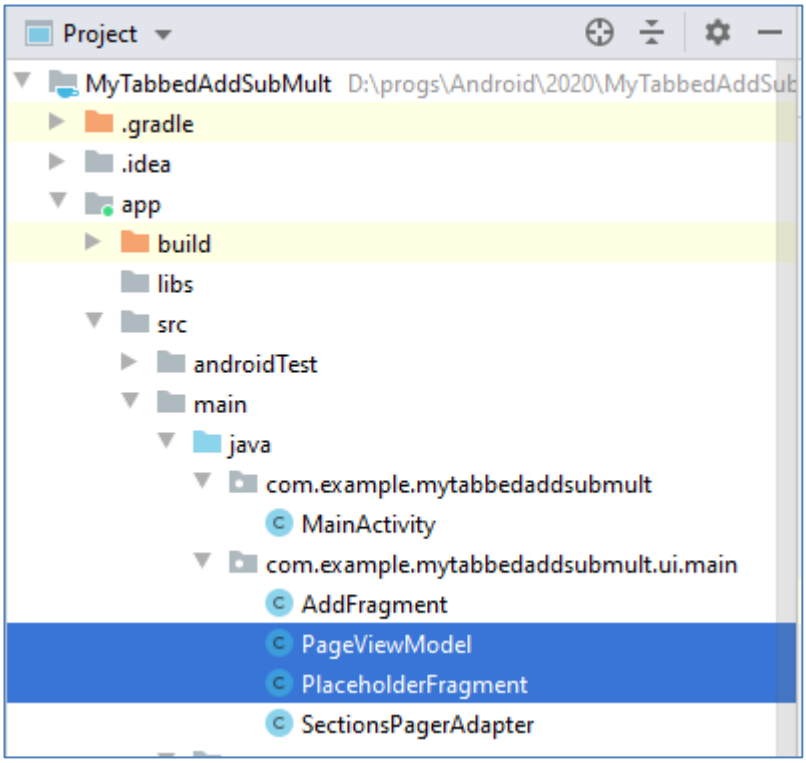


Abbildung 142 Aktueller Stand des TabbedBar-Beispiels (mit Add-Fragment)

11.2.14 Löschen der vorgefertigten Fragmente

Nun müssen die beiden Eintragungen gelöscht werden:



11.2.15 Ändern in SectionsPageAdapter

```
@StringRes
  private static final int[] TAB_TITLES = new int[]{
    R.string.tab_add
  };


@Override
public Fragment getItem(int position) {
  switch (position) {
    case 0:
      AddFragment fragment = new AddFragment();
      return fragment;

    default:
      Log.e("SectionsAdapter, getItem","Fehlerhafte Position"+position);
    }  // switch
    return null;
}
```

Nun die Tab's für Sub und Mult eintragen!

```
@StringRes
private static final int[] TAB_TITLES = new int[]{
  R.string.tab_add,
  R.string.tab_sub,
  R.string.tab_mult
};

@Override
public Fragment getItem(int position) {
```

```
  switch (position) {
    case 0:
      return new AddFragment();

    case 1:
      return new SubFragment();

    case 2:
      return new MultFragment();

    default:
      Log.e("SectionsAdapter, getItem","Fehlerhafte Position"+position);
    }  // switch
    return null;
}
```

11.2.16 Erstellen der restlichen Fragmente

Erstellen des Fragments SubFragment im Ordner ui.main

Erstellen des Fragments MultFragment im Ordner ui.main

11.2.17 fragment_sub.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.main.SubFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitlesub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionsub"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
```

```
        android:text="@string/number1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitlesub" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="-"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_marginTop="32dp"
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

11.2.18 fragment_mult.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ui.main.MultFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitlemult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionmult"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/number1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitlemult" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="*"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />
```

```
    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

11.2.19 Quellcode von SubFragment.java

```
import com.example.mytabbedaddsubmult.R;

import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import java.util.Random;

/**
 * A simple {@link Fragment} subclass.
 */
public class SubFragment extends Fragment {
```

```
    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;

    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public SubFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_sub, container,
false);

        tvtitle = view.findViewById(R.id.tvtitlesub);
           // ohne die 1 gab es einen Fehler

        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);

        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();

        return view;
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
```

```
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1-zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");

        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }



    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tvcomment.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }

    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.2.20 Quellcode von MultFragment.java

```
import com.example.mytabbedaddsubmult.R;

import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import java.util.Random;

public class MultFragment extends Fragment {
```

```
    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;

    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public MultFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_mult, container,
false);

        tvtitle = view.findViewById(R.id.tvtitlemult);
           // ohne die 1 gab es einen Fehler

        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);


        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();

        return view;
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
```

```
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1*zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");

        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }



    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tvcomment.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }

    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

Bitte beachten:

```
import com.example.mytabbedaddsubmult.R;
```

- Wenn mandie Tastatur beim Anklicken des „Hakens“ verschwinden lassen will, muss man mit einem „InputMethodManager“ und der Methode „getSystemService“ arbeiten.
- Dabei ist zu beachten, dass die Fragmente keine Activitäten sind. Das „this“ ist die falsche Klasse.
- Deshalb wird in MainActivity derInputManager statisch deklariert, so dass man in den Fragmenten daruaf zugreifen kann.

11.2.21 MainActivity.java

Eintragen der statischen Variable:

```
public static InputMethodManager in=null;
```

Die Zuweisung eintragen in onCreate:

```
MainActivity.in = (InputMethodManager) this.getSystemService(Context.INPUT_METHOD_SERVICE);
```

11.2.22 AddFragment.java

```
tnumber3.setOnEditorActionListener(new TextView.OnEditorActionListener() {
  @Override
  public boolean onEditorAction(TextView v, int actionId, KeyEvent event){

    if (actionId == EditorInfo.IME_ACTION_GO) {
      performYourAction("go");
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_DONE) {
      performYourAction("done");  // Haken
      // Aufruf der statischen Variable
      MainActivity.in.hideSoftInputFromWindow(
        tnumber3.getApplicationWindowToken(),
        InputMethodManager.HIDE_NOT_ALWAYS);
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_NEXT) {
      performYourAction("next");
      return true;
    }

    if (actionId == EditorInfo.IME_ACTION_SEND) {
      performYourAction("send");
      return true;
    }

    return false;
  }
  });
  return view;
}

private void performYourAction(String item) {
  Log.i(TAG,"************  performYourAction: "+item);
  //Basis.showToast(null,"performYourAction: "+item);
}
```

11.2.23 Musterlösung

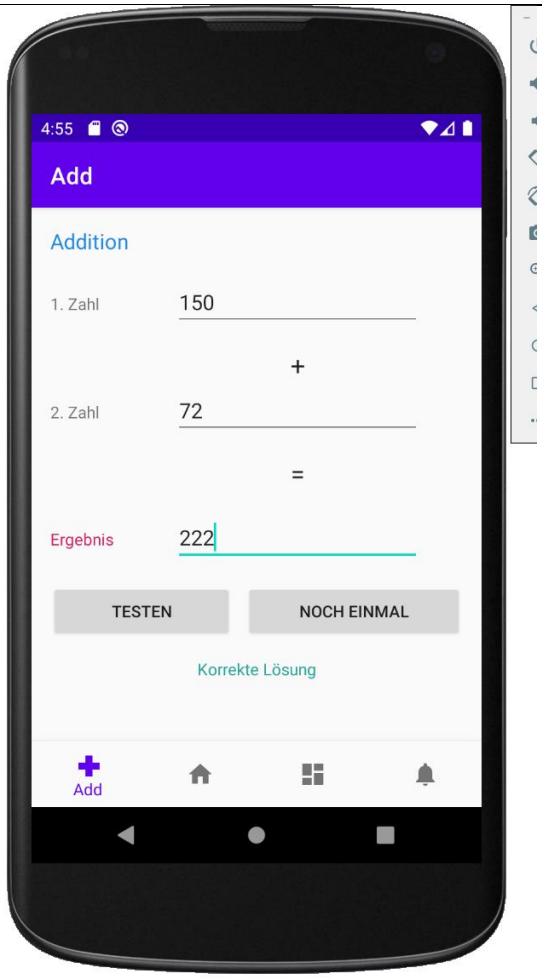


Abbildung 143 Anzeige der Meldung bei korrekter Lösung

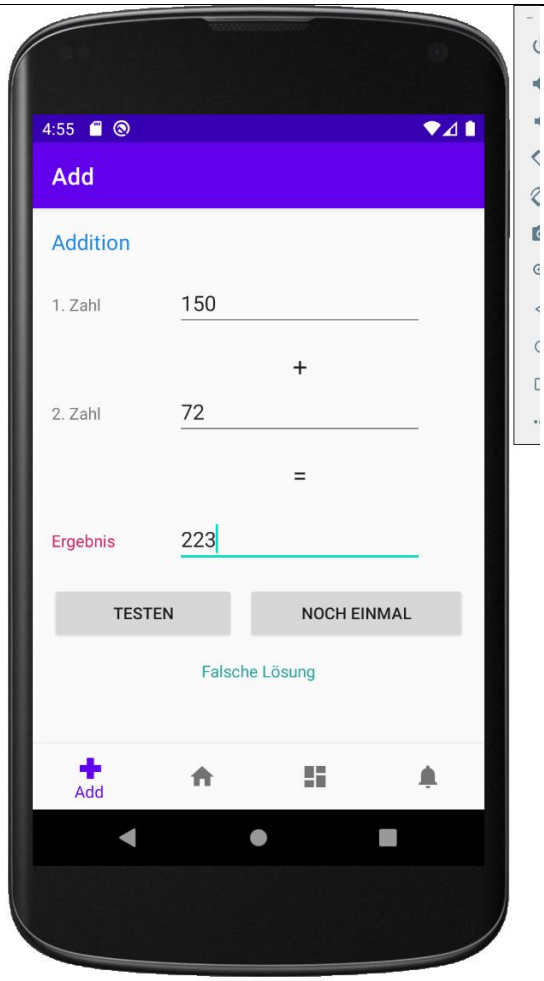


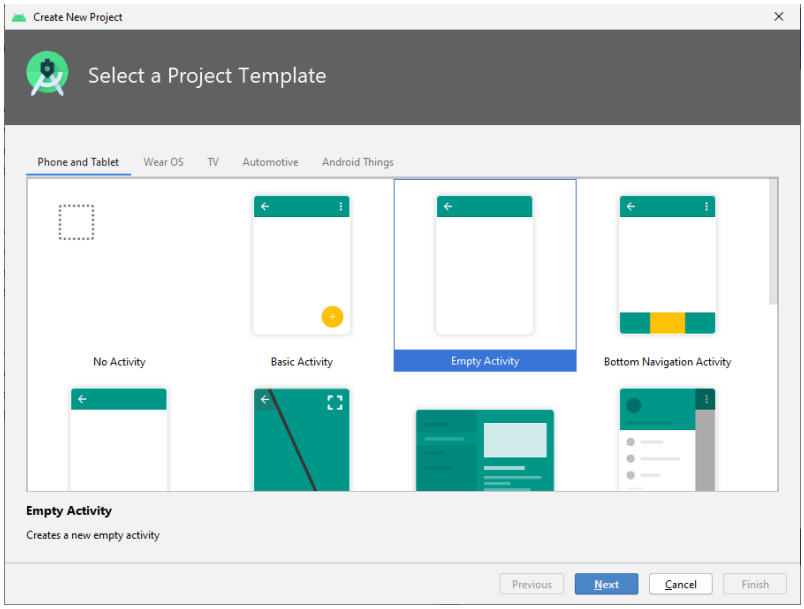
Abbildung 144 Anzeige der Meldung bei falscher Lösung

11.3 Tabbed Activity mit Optionsmenu

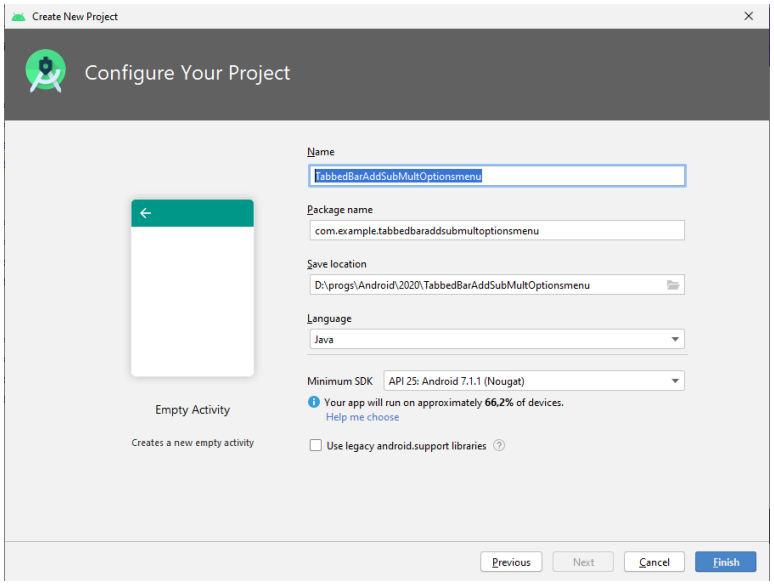
Dieses Kapitel zeigt ein „TabbedBar“, bei der die Funktionen nicht aus einer TabbedBar sondern aus einem Optionsmenü kommen.

11.3.1 Neues Projekt

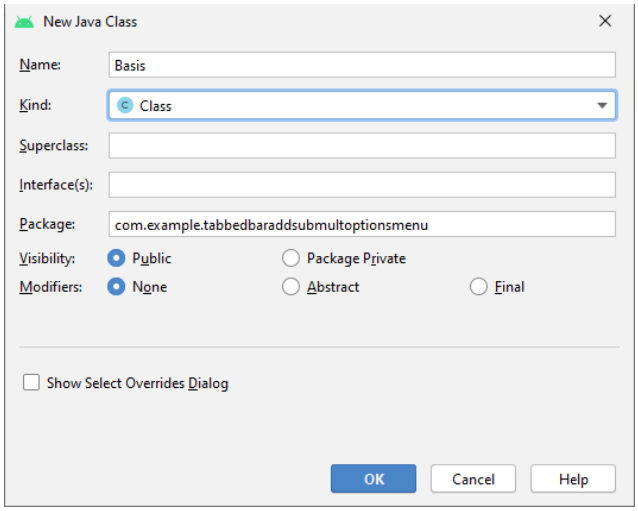
Anlegen eines neuen Projektes mit einer leeren Activity.



Projektname: TabbedBarAddSubMultOptionsmenu



11.3.2 Anlegen und Kopieren der Klasse Basis.java



11.3.3 strings.xml

```
<resources>
    <string name="app_name">VorlBottomNavigationAddSubMult</string>

    <string name="title_add">Add</string>
    <string name="title_sub">Sub</string>
    <string name="title_mult">Mult</string>
    <string name="title_div">Div</string>

    <string name="captionadd">Addition</string>
    <string name="captionsub">Substraktion</string>
    <string name="captionmult">Multiplikation</string>
    <string name="captiondiv">Division</string>

    <string name="number1">1. Zahl</string>
    <string name="number2">2. Zahl</string>
    <string name="number3">Ergebnis</string>

    <string name="bntest">Testen</string>
    <string name="bnagain">Noch einmal</string>
    <string name="comment">Kommentar</string>
    <string name="wrongnumberformat">Bitte eine korrekte Zahl
eingeben</string>
    <string name="wrongnumber">Falsche Lösung</string>
    <string name="correctnumber">Korrekte Lösung</string>

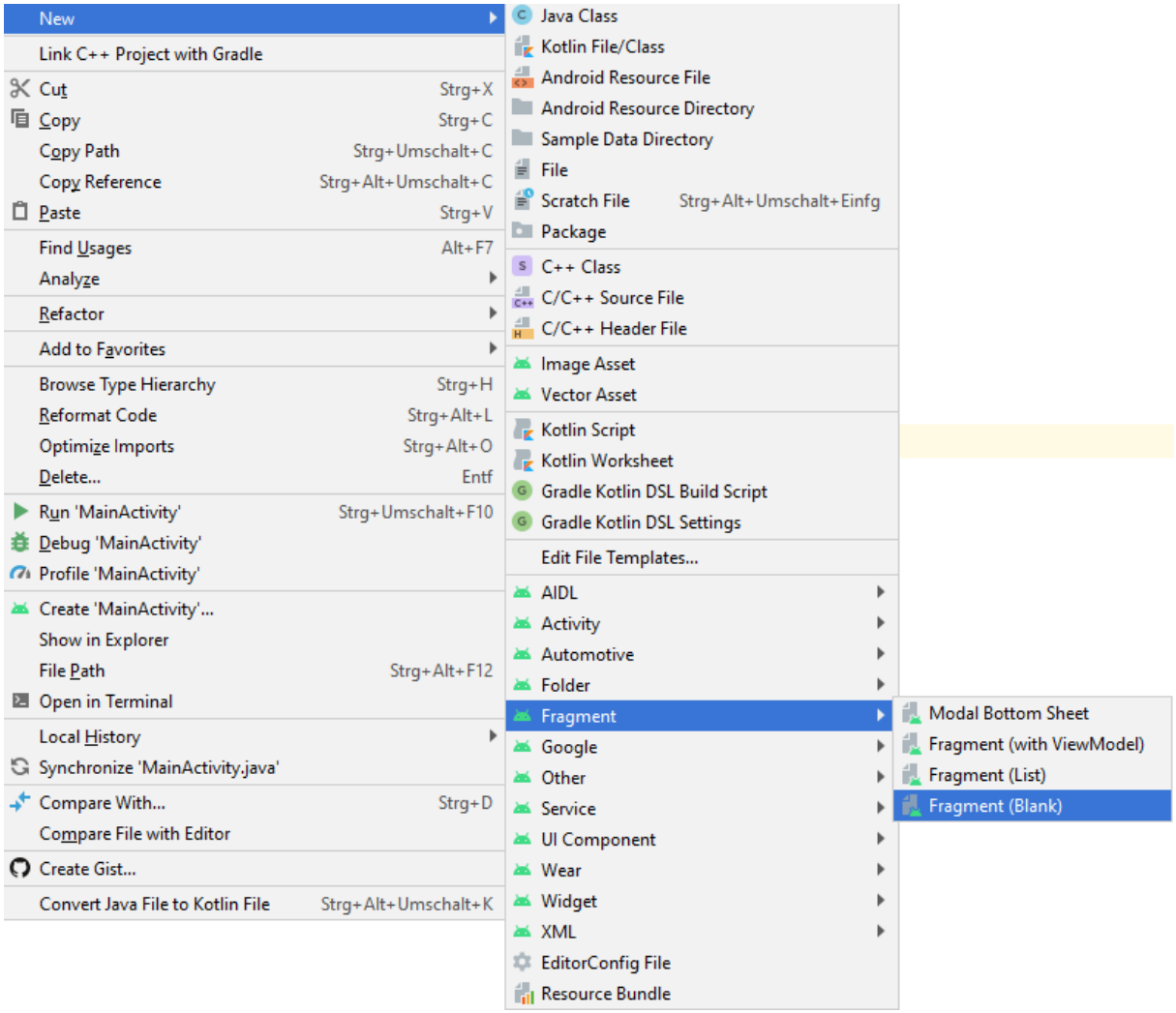
</resources>
```

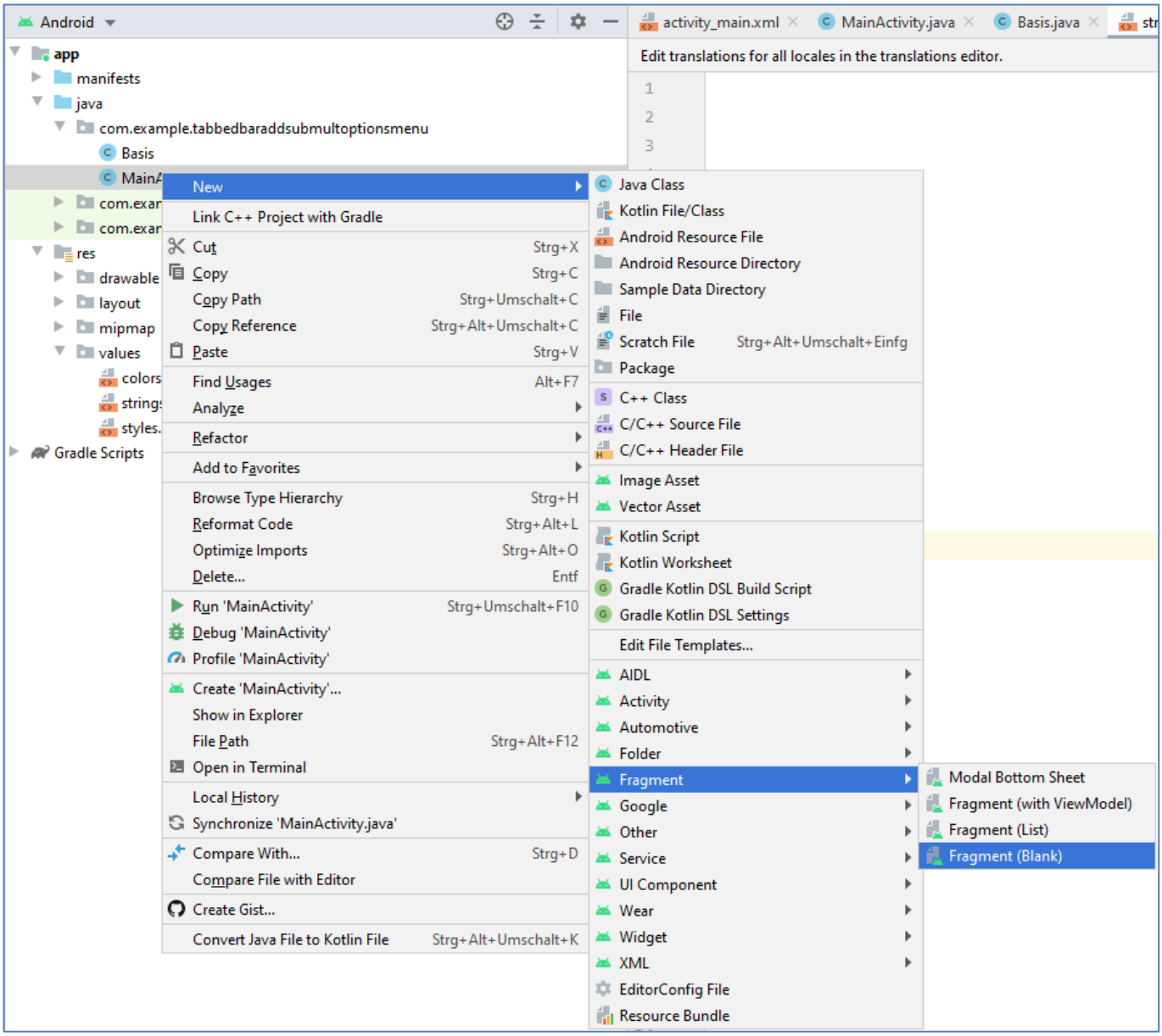
11.3.4 Anlegen der drei Fragmente Add, Sub, Mult

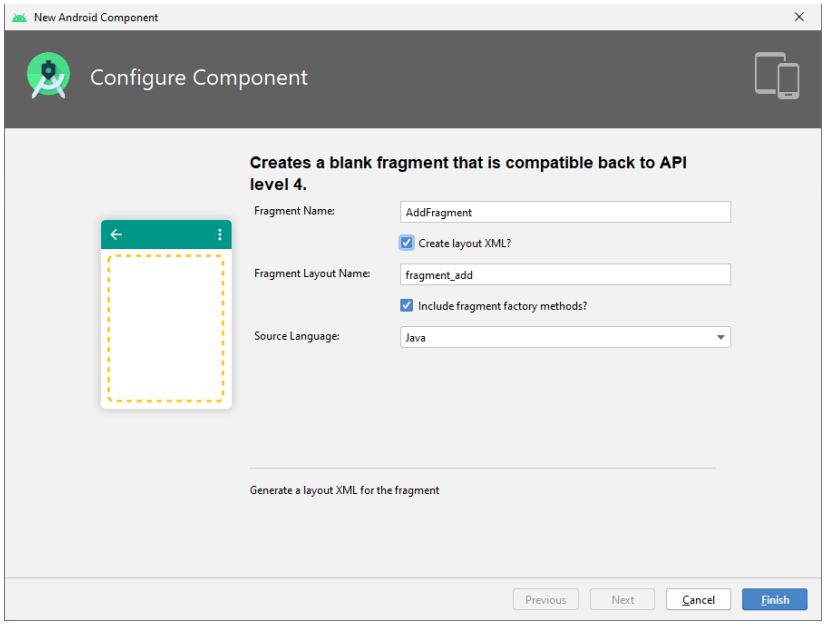
Hier werden jetzt keine Pakacge erzeugt. Die Java-Fragmente werden im gleichen Ordner wie mainAcivity angelegt.

Anklicken von „MainActivity“

Aufruf eines neuen Fragments:

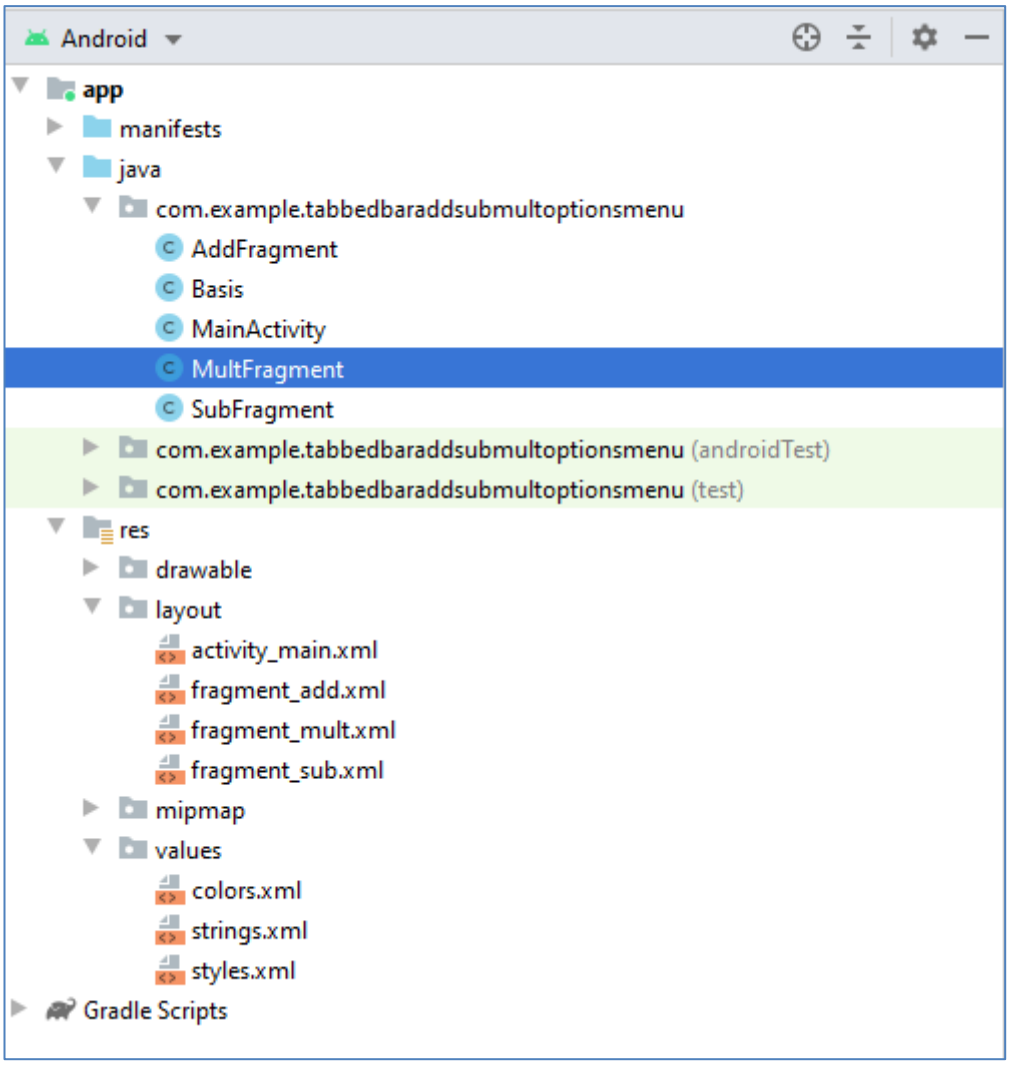






Name: AddFragment

Aktueller Stand:



11.3.5 Ändern der Texte in den Fragmenten:

```
<TextView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:text="@string/captionadd" />
```

```
<TextView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:text="@string/captionsub" />
```

```
<TextView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:text="@string/captionmult" />
```

11.3.6 activity_main.xml

Eintragen des Fragments-Containers.

```
<fragment
   android:id="@+id/nav_host_fragment"
   android:name="androidx.navigation.fragment.NavHostFragment"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   app:defaultNavHost="true"
   app:layout_constraintLeft_toLeftOf="parent"
   app:layout_constraintRight_toRightOf="parent"
   app:layout_constraintTop_toTopOf="parent"
   app:navGraph="@navigation/mobile_navigation"
/>
```

11.3.7 Navigation

Erstellen eines Ordners „res/navigation“

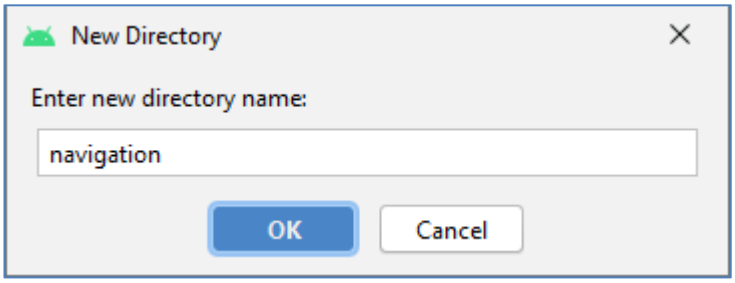
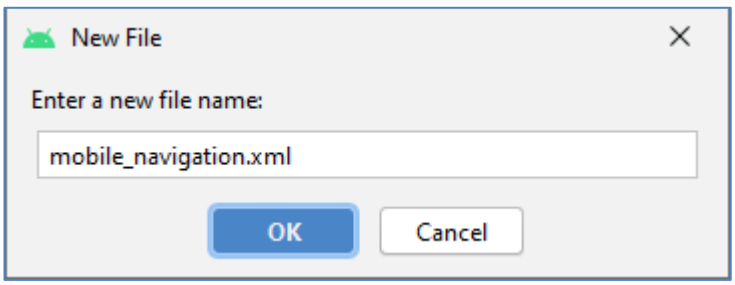
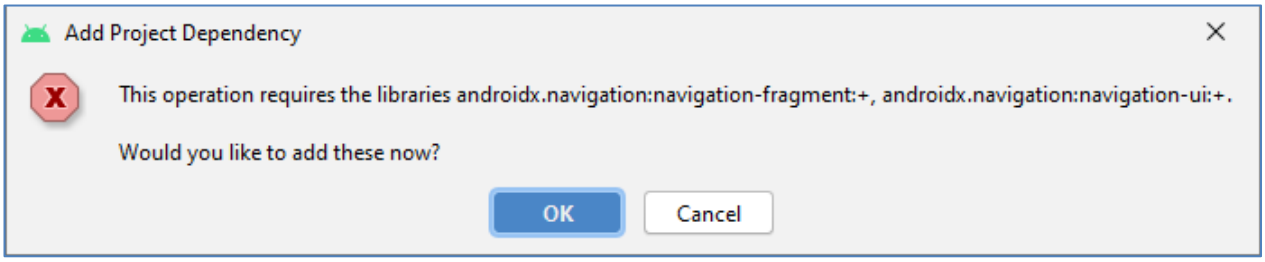


Abbildung 145 Anlegen des Ordners navigation im Eintrag "res"

Einfügen einer neuen Datei im Ordner „navigation“

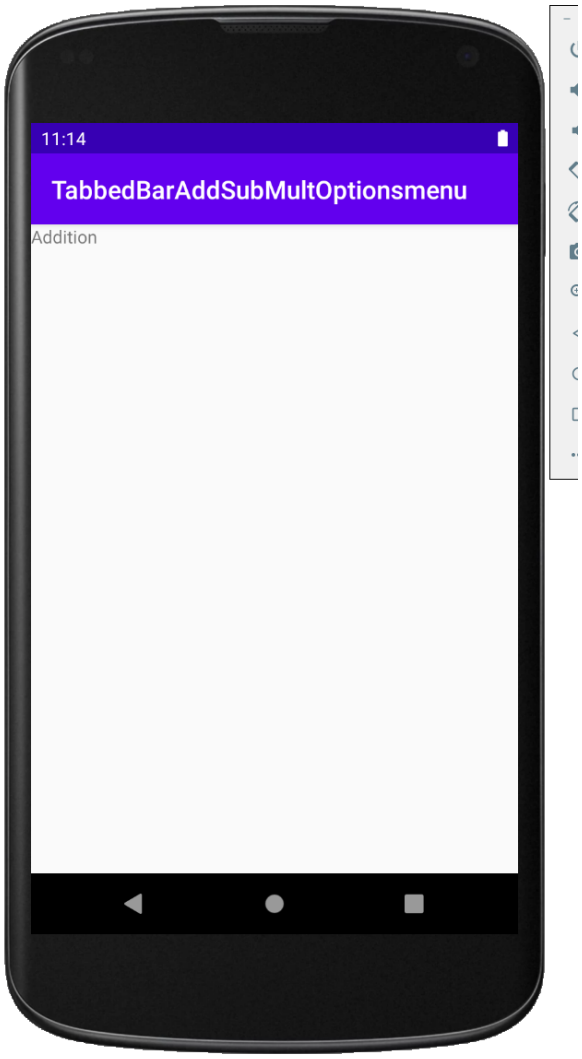
Name: mobile_navigation.xml





Die Abfrage mit „Ok“ beantworten.

11.3.8 Testlauf



11.3.9 Optionsmenu erstellen

11.3.9.1 Menü erstellen

- Anlegen eines Ordners „menu“ im Eintrag „res“

- Menü File
- Eintrag „New“
- Eintrag „Directory“

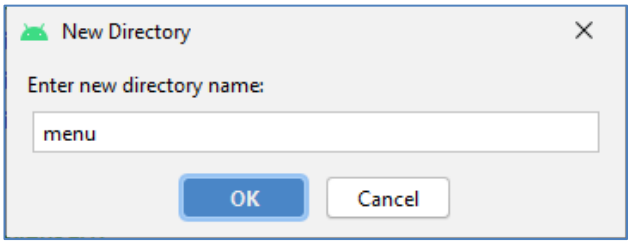


Abbildung 146 Anlegen eines Ordners "menu"

11.3.9.2 Datei menu.xml anlegen

- Anklicken des Ordners „menu“
- Menü File
- Eintrag „New“
- Eintrag „File“
- Eintragen „menu.xml“

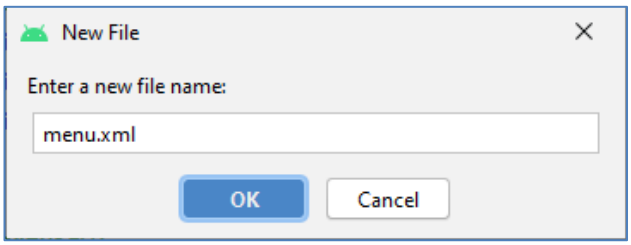


Abbildung 147 Erstellen der xml-Datei "menu.xml"

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

11.3.9.3 Menüeinträge erstellen

In diesem Menü werden dann die einzelnen Punkte eingetragen.
Jeder Eintrag enthält:

- id menu_menu1_item
- Symbol (optional) ic_menu1_24dp.xml
- Beschriftung menu1_item

Die untere Lösung zeigt auch die Symbole an.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:title="Main" app:showAsAction="always"
        android:icon="@android:drawable/ic_menu_preferences"
        >

        <menu>
            <item android:id="@+id/menu_add_item"
                android:icon="@drawable/ic_add_24dp"
                android:title="@string/title_add"
                app:showAsAction="always"
                />

            <item android:id="@+id/menu_sub_item"
                android:icon="@drawable/ic_sub_24dp"
                android:title="@string/title_sub"
                app:showAsAction="always"
                />

            <item android:id="@+id/menu_mult_item"
                android:icon="@drawable/ic_mult_24dp"
                android:title="@string/title_mult"
                app:showAsAction="always"
                />

        </menu>
    </item>

</menu>
```

Bitte beachten:

- Es gibt zwei „Orte“ aus denen man Symbole holen kann.
  - @android:drawable/ic_menu_add
  - @drawable/ic_add_24dp
- Eigene Symbole werden immer mit der zweiten Varainte adressiert.

11.3.10 Optionsmenu einfügen in die mainActivity

In mainActivity wird die Methode onStart überschrieben:

// import android.app.ActionBar;

```
@Override
protected void onStart() {
  super.onStart();
  ActionBar ab = getActionBar();
  if (ab != null) {
    ab.setDisplayHomeAsUpEnabled(true);
  }
}
```

11.3.11 Optionsmenu-Start-Event abfangen

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.menu, menu);

  return super.onCreateOptionsMenu(menu);

}
```

11.3.12 Optionsmenu-Klick-Event abfangen

```
@Override
public boolean onOptionsItemSelected(MenuItem mnItem) {
  switch (mnItem.getItemId()) {
    case R.id.menu_menu1_item:
      Basis.showToast(this, "Menü new angeklickt");
      // Action
      break;

  } // switch
  return true;
}
```

11.3.13 Navigationscontroller

Um die Navigation in Eigenregie zu realisaieren, benötigt man wenig Code:

Globale Variablen:

```
private NavController navController = null;
private HashMap<Integer,Integer> hashIds = new HashMap<>();
```

Die HashMap dient dem schnellen Finden der Navagations-Id. Der Schlüssel ist die Menü-Id.

Neuer Code in onCreate:

```
navController = Navigation.findNavController(this, R.id.nav_host_fragment);

hashIds.put(R.id.menu_add_item,R.id.navigation_add);
hashIds.put(R.id.menu_sub_item,R.id.navigation_sub);
hashIds.put(R.id.menu_mult_item,R.id.navigation_mult) ;
```

Die erste Anweisungholt sich den NavigationsController und speichert ihn global ab.

Methode onOptionsItemSelected:

Übergeben wird eine MenüId und diese wird umgewandelt in eine Fragment-Id.
Diese wird dann benutzt, um das neue Fragment anzuzeigen.

```
@Override
  public boolean onOptionsItemSelected(MenuItem mnItem) {

     Integer idFragment = hashIds.get(mnItem.getItemId());
     if (idFragment!=null) {
       navController.navigate(idFragment.intValue());
     }
     return true;
}
```

11.3.14 Fragment Add

Im Prinzip kann man die Lösung aus den vorletzten Kapitel nehmen. Nur muss man folgende Code ersetzen:

- tools:context=".ui.add.AddFragment"

durch

- tools:context="AddFragment"

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="AddFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitleadd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionadd"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/number1"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitleadd" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="+"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
```

```
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

11.3.15 AddFragment.java

```
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import com.example.vorlbottomnavigationaddsubmult.Basis;
import com.example.vorlbottomnavigationaddsubmult.MainActivity;
import com.example.vorlbottomnavigationaddsubmult.R;

import java.util.Random;

import android.view.inputmethod.InputMethodManager;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;

public class AddFragment extends Fragment {

    private static final String TAG = MainActivity.class.getSimpleName();

    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;
    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public AddFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_add, container,
false);

        tvtitle = view.findViewById(R.id.tvtitleadd);

        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
```

```
        tvcomment = view.findViewById(R.id.tvcomment);

        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                    @Override
                                    public void onClick(View v) {
                                          bntest_Click();
                                    }
                                }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                    @Override
                                    public void onClick(View v) {
                                        bnagain_Click();
                                    }
                                }
        );

        setRamdomNumbers();


        tnumber3.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
            @Override
            public boolean onEditorAction(TextView v, int actionId,
KeyEvent event) {
                if (actionId == EditorInfo.IME_ACTION_GO) {
                    performYourAction("go");
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_DONE) {
                    performYourAction("done");  // Haken

MainActivity.in.hideSoftInputFromWindow(tnumber3.getApplicationWindowToken(
),InputMethodManager.HIDE_NOT_ALWAYS);
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_NEXT) {
                    performYourAction("next");
                    return true;
                }
                if (actionId == EditorInfo.IME_ACTION_SEND) {
                    performYourAction("send");
                    return true;
                }
                return false;
            }
        });
        return view;
    }

    private void performYourAction(String item) {
        Log.i(TAG,"************  performYourAction: "+item);
        //Basis.showToast(null,"performYourAction: "+item);
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
```

```
        this.zahl1 = rand.nextInt(200)+1;
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1+zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");
        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }



    private void bntest_Click() {
       Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
       if (Zahl3==null) {
           tvcomment.setText(getText(R.string.wrongnumberformat));
           return;
       }
       int eingabeErgebnis = Zahl3.intValue();
       if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
       }
       else {
           tvcomment.setText(getString(R.string.wrongnumber));
       }

    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.3.16 Fragment Sub

Im Prinzip kann man die Lösung aus den vorletzten Kapitel nehmen. Nur muss man folgende Code ersetzen:

- tools:context=".ui.add.AddFragment"

durch

- tools:context="AddFragment"

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.sub.SubFragment">
```

```
    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitlesub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionsub"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/number1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitlesub" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="-"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

        android:inputType="number"
```

```
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
```

```
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />



</androidx.constraintlayout.widget.ConstraintLayout>
```

11.3.17 SubFragment.java

```
import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import com.example.vorlbottomnavigationaddsubmult.Basis;
import com.example.vorlbottomnavigationaddsubmult.R;

import java.util.Random;


public class SubFragment extends Fragment {

    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;
    private TextView tverror = null;
    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;
```

```
    public SubFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_sub, container,
false);

        tvtitle = view.findViewById(R.id.tvtitlesub);

        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);
        tverror = view.findViewById(R.id.tverror);

        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();

        return view;
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1-zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");
        tverror.setText("");
        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }
```

```
    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tverror.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }

    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.3.18 Fragment Mult

Im Prinzip kann man die Lösung aus den vorletzten Kapitel nehmen. Nur muss man folgende Code ersetzen:

- tools:context=".ui.add.AddFragment"

durch

- tools:context="AddFragment"

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.mult.MultFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/tvtitlemult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/captionmult"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#1E88E5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
    <TextView
        android:id="@+id/tvnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="@string/number1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvtitlemult" />

    <EditText
        android:id="@+id/tnumber1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber1"
        app:layout_constraintStart_toEndOf="@+id/tvnumber1" />

    <TextView
        android:id="@+id/tvoperation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="*"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber1" />

    <TextView
        android:id="@+id/tvnumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/number2"
        app:layout_constraintStart_toStartOf="@+id/tvnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tvoperation" />

    <EditText
        android:id="@+id/tnumber2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber2"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
        app:layout_constraintStart_toStartOf="@+id/tnumber1" />


    <TextView
        android:id="@+id/tvequation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="="
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintEnd_toEndOf="@+id/tnumber1"
```

```
        app:layout_constraintStart_toStartOf="@+id/tnumber1"
        app:layout_constraintTop_toBottomOf="@+id/tnumber2" />

    <TextView
        android:id="@+id/tvnumber3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/number3"
        android:textColor="#D81B60"
        app:layout_constraintStart_toStartOf="@+id/tvnumber2"
        app:layout_constraintTop_toBottomOf="@+id/tvequation" />

    <EditText
        android:id="@+id/tnumber3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        app:layout_constraintBaseline_toBaselineOf="@+id/tvnumber3"
        app:layout_constraintEnd_toEndOf="@+id/tnumber2"
        app:layout_constraintStart_toStartOf="@+id/tnumber2" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tnumber3">

        <Button
            android:id="@+id/bntest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:text="@string/bntest" />

        <Button
            android:id="@+id/bnagain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/bnagain" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvcomment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:text="TextView"
        android:textColor="#24A598"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />



</androidx.constraintlayout.widget.ConstraintLayout>
```

11.3.19 MultFragment.java

```
import android.content.Intent;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;

import com.example.vorlbottomnavigationaddsubmult.Basis;
import com.example.vorlbottomnavigationaddsubmult.R;

import java.util.Random;

public class MultFragment extends Fragment {

    private EditText tnumber1 = null;
    private EditText tnumber2 = null;
    private EditText tnumber3 = null;
    private TextView tvcomment = null;

    private Button bntest = null;
    private Button bnagain = null;
    private int zahl1=0;
    private int zahl2=0;
    private int ergebnis=0;


    public MultFragment() {
        // Required empty public constructor
    }

    private TextView tvtitle = null;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_mult, container,
false);

        tvtitle = view.findViewById(R.id.tvtitlemult);
```

```
        tnumber1 = view.findViewById(R.id.tnumber1);
        tnumber2 = view.findViewById(R.id.tnumber2);
        tnumber3 = view.findViewById(R.id.tnumber3);
        tvcomment = view.findViewById(R.id.tvcomment);


        bntest = (Button) view.findViewById(R.id.bntest);
        bnagain = (Button) view.findViewById(R.id.bnagain);

        bntest.setOnClickListener(new View.OnClickListener() {
                                      @Override
                                      public void onClick(View v) {
                                          bntest_Click();
                                      }
                                  }
        );
        bnagain.setOnClickListener(new View.OnClickListener() {
                                       @Override
                                       public void onClick(View v) {
                                           bnagain_Click();
                                       }
                                   }
        );

        setRamdomNumbers();

        return view;
    }

    private void setRamdomNumbers() {
        Random rand = new Random();
        this.zahl1 = rand.nextInt(200)+1;
        this.zahl2 = rand.nextInt(200)+1;
        this.ergebnis = zahl1*zahl2;
        tnumber1.setText(Integer.toString(zahl1));
        tnumber2.setText(Integer.toString(zahl2));
        tnumber3.setText("?");

        tvcomment.setText("");
        Log.e("------------------ Zahl1",Integer.toString(zahl1));
        Log.e("------------------ Zahl2",Integer.toString(zahl2));
    }



    private void bntest_Click() {
        Integer Zahl3 =
Basis.getIntegerNumber(tnumber3.getText().toString());
        if (Zahl3==null) {
            tvcomment.setText(getText(R.string.wrongnumberformat));
            return;
        }
        int eingabeErgebnis = Zahl3.intValue();
        if (eingabeErgebnis==this.ergebnis) {
            tvcomment.setText(getString(R.string.correctnumber));
        }
        else {
            tvcomment.setText(getString(R.string.wrongnumber));
        }
```

```
    }

    private void bnagain_Click() {
        setRamdomNumbers();
    }

}
```

11.3.20 Abbildung der Musterlösung

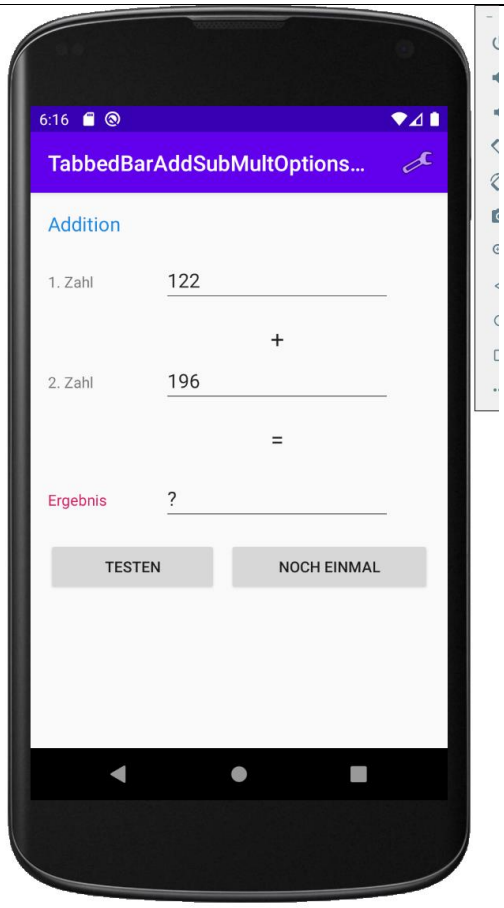


Abbildung 148 AddSubMult in Optionsmenu

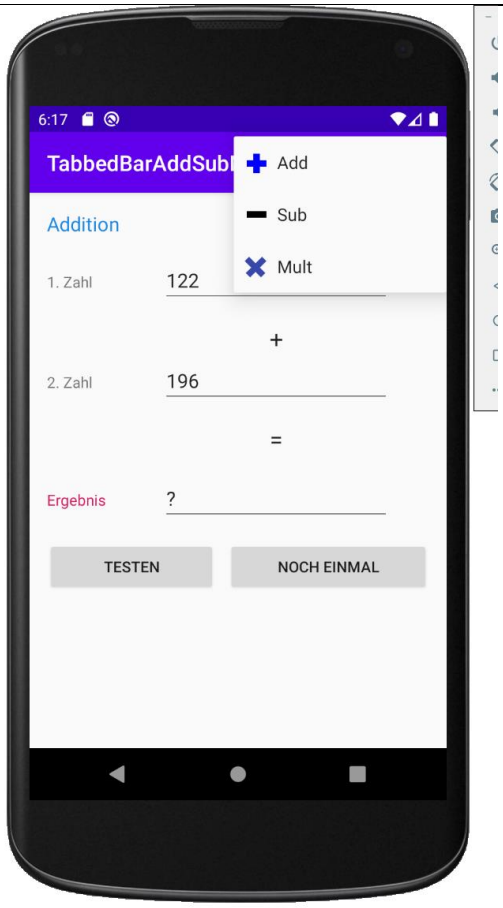


Abbildung 149 AddSubMult in Optionsmenu

11.4 Swipe-Activity

Dieses Kapitel zeigt, wie man eine Tabbed-Activity als Swipe-Activity benutzen kann. Zwischen den Seiten wird nach links und rechts gewüscht.

11.4.1 Anlegen des Projektes

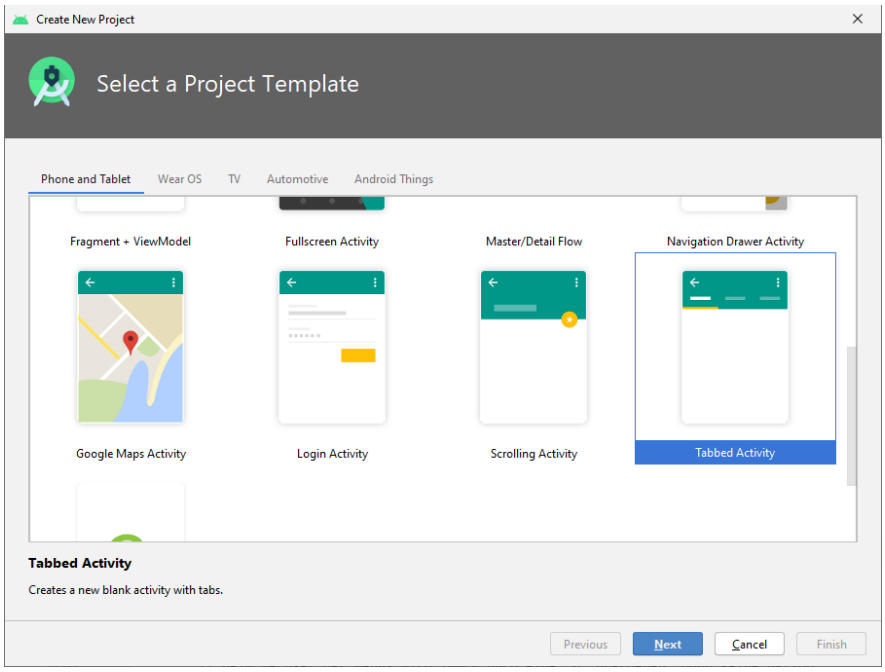


Abbildung 150 Auswahl einer Tabbed Activity

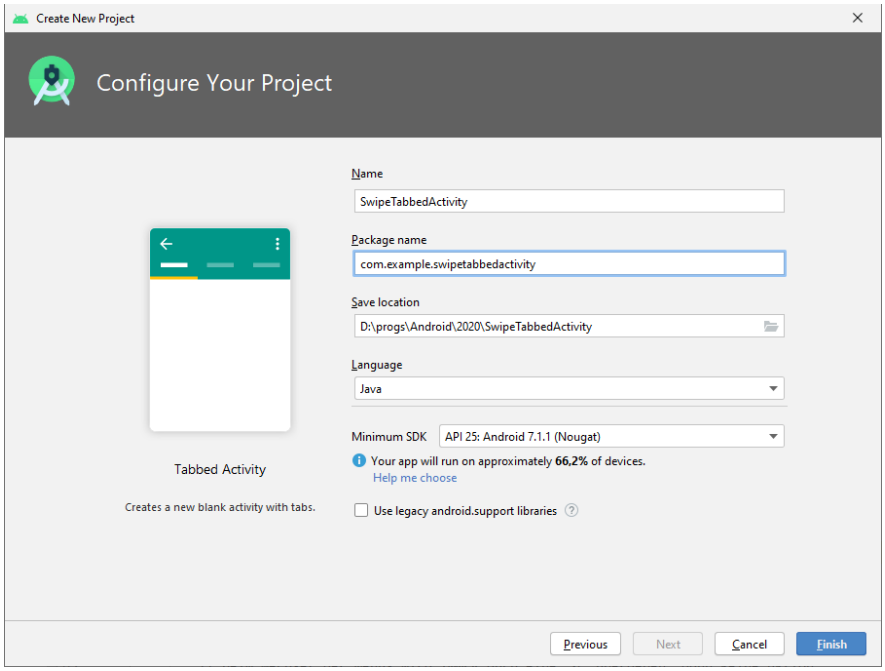
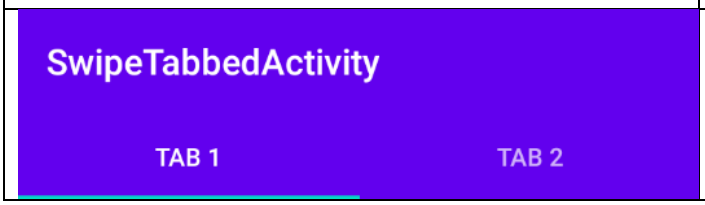
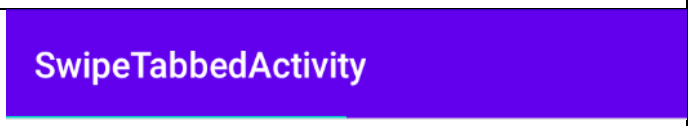


Abbildung 151 Name der Tabbed Activity: SwipeTabbedActivity

Name: SwipeTabbedActivity

11.4.2 activity_main.xml

Höhe der ActionBar wrap_content	Höhe der ActionBar „1dp“
<com.google.android.material.tabs.TabLayout android:id="@+id/tabs" android:layout_width="match_parent" android:layout_height="wrap_content" android:background="?attr/colorPrimary" />	<com.google.android.material.tabs.TabLayout android:id="@+id/tabs" android:layout_width="match_parent" **android:layout_height="1dp"** android:background="?attr/colorPrimary" />
	

Wie man sieht, verschwinden die Beschriftungen, so dass man zum Einen mehr Platz hat und zum Andern bei großer Anzahl der „Tab’s“ diese sowieso nicht sieht. Dieses Projekt ist also nur dann sinnvoll, wenn man mehr als 10 Tab’s hat.

11.4.3 Kopieren der Bilder

Nun kopiert man die 305 Beispielbilder in das Verzeichnis „res/drawable“.

11.4.4 Anlegen des Fragments

- Anklicken des Eintrags „java/…/ui.main“
- Rechte Maustaste
- Eintrag „New“
- Eintrag „Fragment“
- Eintrag „Fragment (Blank)“
- Name: PictureFragment
- **Mit Factory**

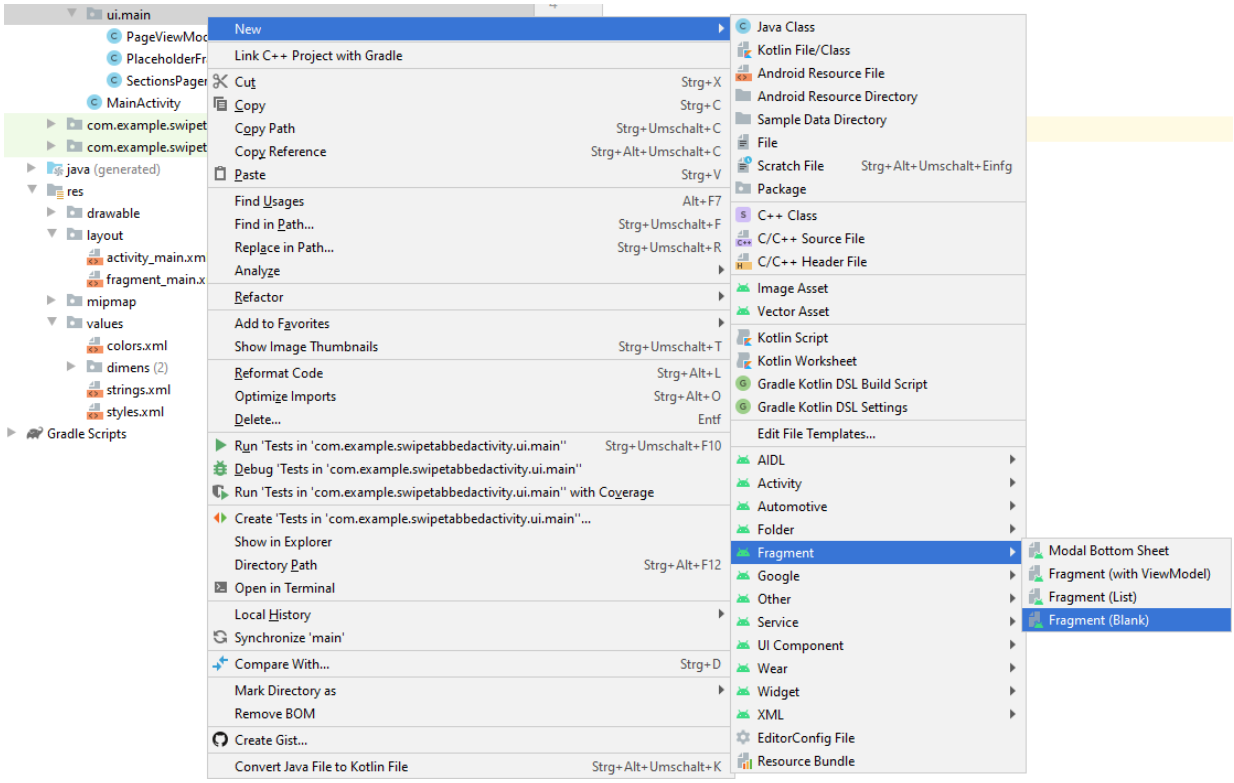


Abbildung 152 Anlegen eines Pictures-Fragment

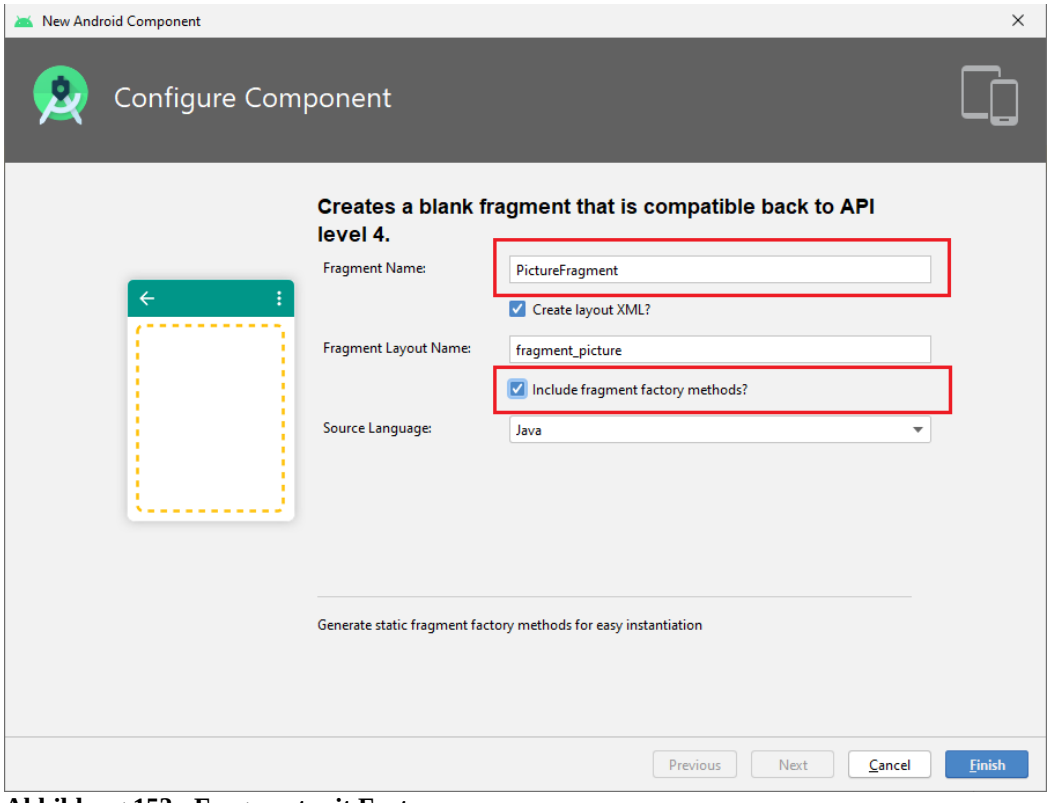


Abbildung 153 Fragment mit Factory

11.4.5 Ändern der Factory

Die Factory dient dazu, immer das gleiche Fragment zu Verfügung zu stellen. Der Java-Code des neuen Fragment braucht aber Parameter, die anzeigen, welche Daten resp. Bilder angezeigt werden sollen. Standardmäßig werden zwei String übergeben. Für dieses Beispiel benötigen wir aber die Picturenummer.

Damit sieht der Quellcode folgendermaßen aus:

```
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.example.swipetabbedactivity.R;


public class PictureFragment extends Fragment {

    private static final String ARG_PICT_NUMBER = "number";


    private int number;

    public PictureFragment() {
        // Required empty public constructor
    }


    public static PictureFragment newInstance(int number) {
        PictureFragment fragment = new PictureFragment();
        Bundle args = new Bundle();
        args.putInt(ARG_PICT_NUMBER, number);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            number = getArguments().getInt(ARG_PICT_NUMBER);
        }
        else {
            number=1;
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_picture, container,
false);
    }
}
```

11.4.6 SectionsPageAdapter

Als erstes werden die beiden Java-Dateien gelöscht:

- PageViewModel.java
- PlaceholderFragment .java

Danach wird das Fragment gelöscht:

- fragment_main.xml

Nun wird die Datei SectionsPageAdapter.java umgebaut:

- Die Anzahl ist bei 310 Bilder
- Die Factory muss eingerichtet werden.

```
import android.content.Context;

import androidx.annotation.Nullable;
import androidx.annotation.StringRes;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;

import com.example.swipetabbedactivity.R;

public class SectionsPagerAdapter extends FragmentPagerAdapter {

  private final int MAXPICTURES = 310;

  private final Context mContext;

  public SectionsPagerAdapter(Context context, FragmentManager fm) {
    super(fm);
    mContext = context;
  }

  @Override
  public Fragment getItem(int position) {
    return PictureFragment.newInstance(position + 1);
  }

  @Nullable
  @Override
  public CharSequence getPageTitle(int position) {
    // return mContext.getResources().getString(TAB_TITLES[position]);
    return "Picture no: "+position;
  }

  @Override
  public int getCount() {
    return MAXPICTURES;
  }
}
```

Änderungen:

- Löschen der TAB_TITLES
- Einfügen einer Konstanten „MAXPICXUTRES“

- Ändern von getCount
- Der Titel „getPageTitle“ wird auch durch die Nummer bestimmt.
- Der Factory wird nun die „position+1“ übergeben.

11.4.7 fragment_picture.xml

Hier wird ein Title und ein ImageView eingetragen:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="PictureFragment">


  <TextView
    android:id="@+id/tvtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="-"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#1E88E5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <ImageView
    android:id="@+id/ivpicture"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:src="@drawable/picture_001"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitle" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

11.4.8 PictureFragment.java

Änderungen:

- Der Parameter der Factory, bildnummer, muss an das neue Fragment per Bundle übergeben werden.
- In onCreate wird die Nummer aus dem Bundle abgefragt.

- In onCreateView holt man die Referenzen der beiden UI-Elemente und setzt den Title und das Bild.
- Hier muss nun die Bildnummer im Title eingetragen werden und das Bild sollte sichtbar sein.

Parameterübergabe:

```
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
  private static final String ARG_PICT_NUMBER = "number";

  private int number;  / speichert die Bildnummer

  private TextView tvtitle = null;
  private ImageView ivpicture = null;

  public PictureFragment() {
    // Required empty public constructor
  }


    // TODO: Rename and change types and number of parameters
    // FACTORY !
  public static PictureFragment newInstance(int number) {
    PictureFragment fragment = new PictureFragment();
    Bundle args = new Bundle();
    args.putInt(ARG_PICT_NUMBER, number);
    fragment.setArguments(args);
    return fragment;
  }

  @Override
  public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
    if (getArguments() != null) {
      number = getArguments().getInt(ARG_PICT_NUMBER);
    }
    else {
      number=1;
    }
  }
```

Aufbau der View:

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                  Bundle savedInstanceState) {
  View view = inflater.inflate(R.layout.fragment_picture,container,false);

  tvtitle = view.findViewById(R.id.tvtitle);
  ivpicture = view.findViewById(R.id.ivpicture);

  tvtitle.setText( Integer.toString(number)+". Bild");

  switch (number) {
    case 1:
      ivpicture.setImageResource(R.drawable.picture_001);
      break;
    case 2:
      ivpicture.setImageResource(R.drawable.picture_002);
```

```
        break;
    }


    return view;
}
```

Für das Setzen des Bildes gibt es nun mehrere Möglichkeiten:

- Switch / case mit 310 Eintragungen (optiomal)
- Man holt den Pfad der Bilder und setzt das Bild mittels
    - setImageDrawable
    - Drawable.createFromPath(filename);
- Man holt sich die Resourcen-ID und nimmt die bekannte Methode.

setImageDrawable:

- String filename = String.format("@drawable/picture_%1$03d",number);
- Log.i("----------Picture",filename+" "+Drawable.createFromPath(filename));
- ivpicture.setImageDrawable(Drawable.createFromPath(filename));

Die erste Zeile formatiert nur den Dateiname:

- picture_001 bis picture_310

Der Aufruf mit createPath funktionierte leider nicht bei mir. ES kam immer null zurück.

Resourcen-ID:

- **int imageResource = getResources().getIdentifier(uri, null, MainActivity.packageName);**
- Log.i("----------uri",uri+" resid: "+imageResource);
- **Drawable res = getResources().getDrawable(imageResource);**
- **ivpicture.setImageDrawable(res);**

Mit der Methode „getResources().getIdentifier“ kann man eine Resourcen-Id bestimmen. Für den letzten Parameter muss man den Packagenamen angeben. Leider funktioniert das nicht in einem Fragment. Deshalb gibt es in MainActivity eine statische Variable, die man hier benutzen kann (siehe unteren Quellcode. Aus der Id „imageResource“ bestimtm die dritte Zeile die Resource, mit der dann in der letzten Zeile das Bild gesetzt wird.

```
public class MainActivity extends AppCompatActivity {

    public static String packageName="";

  @Override
  protected void onCreate(Bundle savedInstanceState) {
     ...
     //context.getPackageName()
     MainActivity.packageName = this.getPackageName();

  }
}
```

Der obere Quellcode (Ausschnitt) speiochert den Packagenamen für die Zuweisung der Bilder im Fragment.

Kompletter Quellcode des Fragments:

```
import android.graphics.drawable.Drawable;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.example.swipetabbedactivity.MainActivity;
import com.example.swipetabbedactivity.R;


public class PictureFragment extends Fragment {
  // TODO: Rename parameter arguments, choose names that match
  private static final String ARG_PICT_NUMBER = "number";

  private int number;
  private TextView tvtitle = null;
  private ImageView ivpicture = null;

  public PictureFragment() {
    // Required empty public constructor
  }


  public static PictureFragment newInstance(int number) {
    PictureFragment fragment = new PictureFragment();
    Bundle args = new Bundle();
    args.putInt(ARG_PICT_NUMBER, number);
    fragment.setArguments(args);
    return fragment;
  }

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
      number = getArguments().getInt(ARG_PICT_NUMBER);
    }
    else {
      number=1;
    }
  }

  @Override
  public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
  View view = inflater.inflate(R.layout.fragment_picture, container,false);

    tvtitle = view.findViewById(R.id.tvtitle);
    vpicture = view.findViewById(R.id.ivpicture);
```

```
        tvtitle.setText( Integer.toString(number)+". Bild");
        String uri = String.format("@drawable/picture_%1$03d",number);

        int imageResource = getResources().getIdentifier(uri, null,
                                                MainActivity.packageName);
        Log.i("------------------uri",uri+"        resid: "+imageResource);
        Drawable res = getResources().getDrawable(imageResource);
        ivpicture.setImageDrawable(res);

        return view;
    }
}
```

Musterlösung:

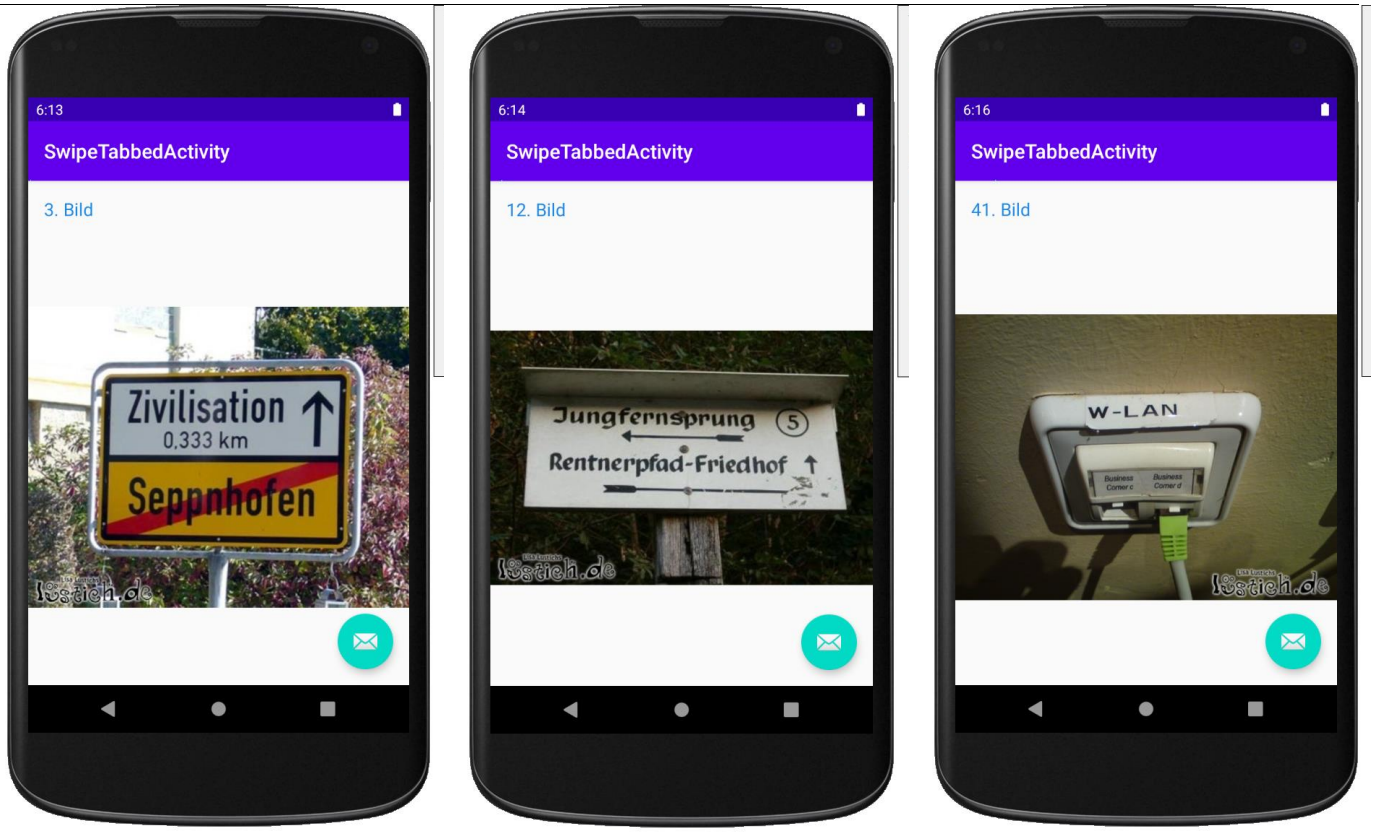
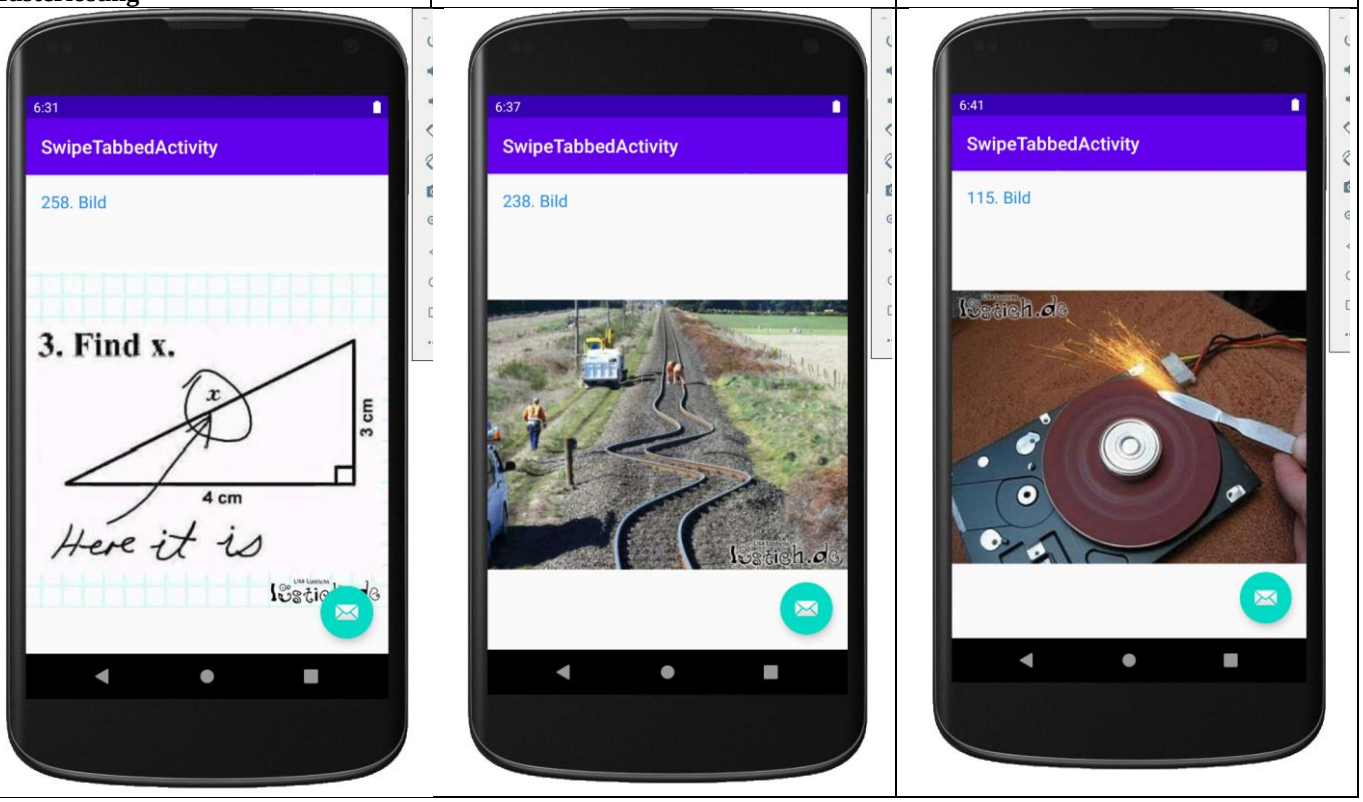


Abbildung 154 Anzeige der Swipe-Musterlösung



11.5 Master Detail Activity

11.6 Eigene Navigation

Dieses Kapitel zeigt, wie man ohne mit eigenen Mitteln eine Navigation erreicht. Dazu gibt es auf der Startseite drei Schalter, mit denen man weitere Navigations-Seiten aufrufen kann:

- Bootom-Navigation
- TabbedBar Navigation
- ListView

11.6.1 Musterlösung



Abbildung 155 Startbildschirm

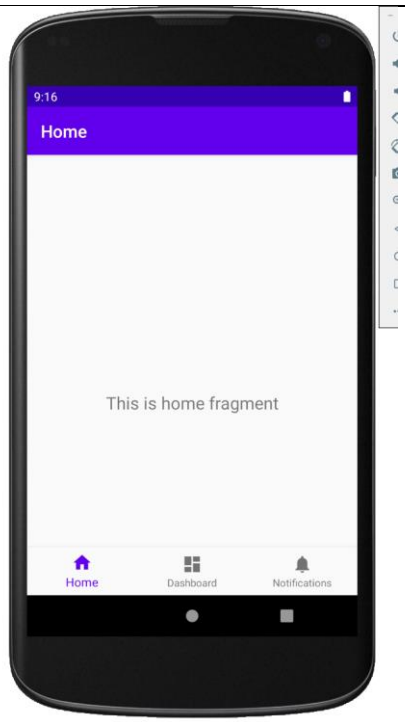


Abbildung 156 Vorspeisen

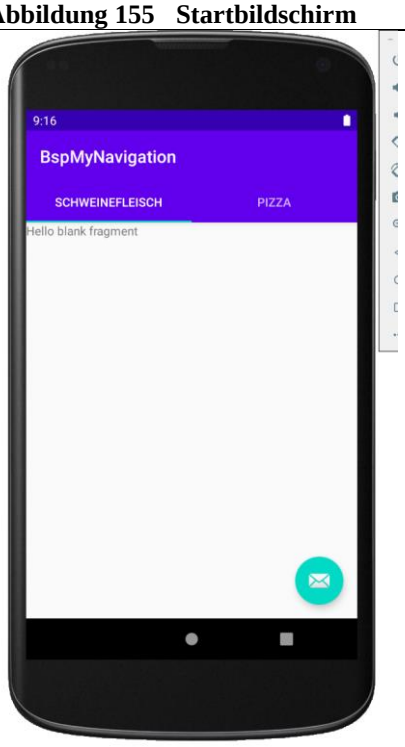


Abbildung 157 Hauptspeise

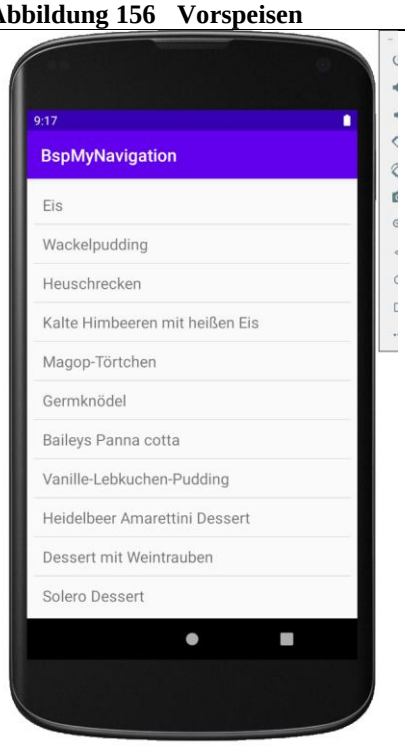


Abbildung 158 Nachspeise

11.6.2 Projekt anlegen

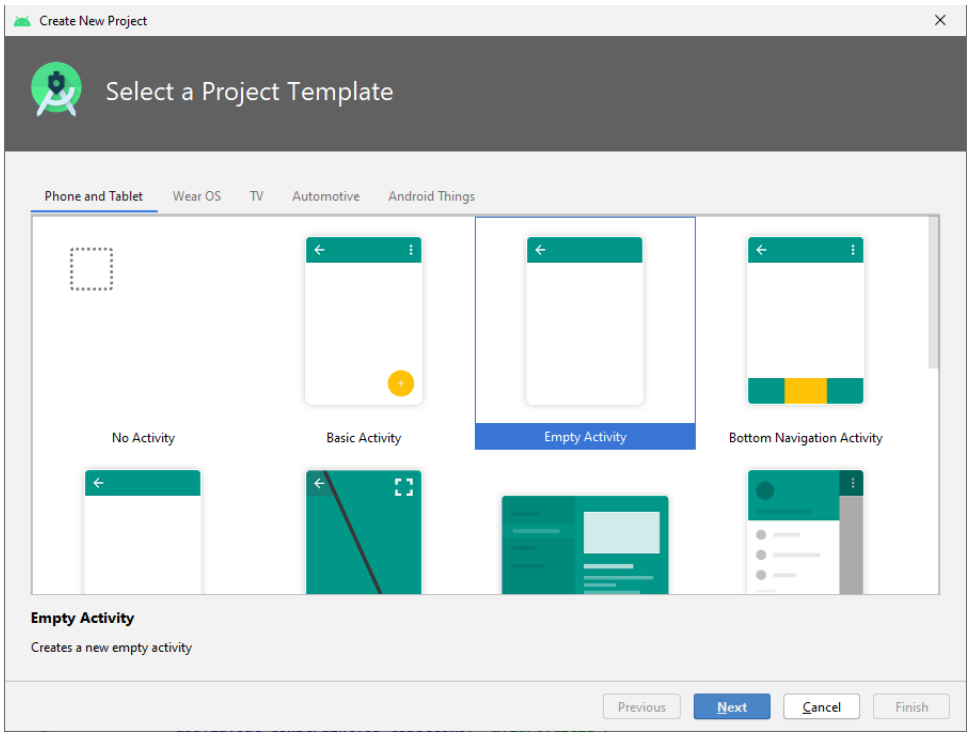


Abbildung 159 Projekttyp "Empty Activity"

Projektname: BspMyNavigation

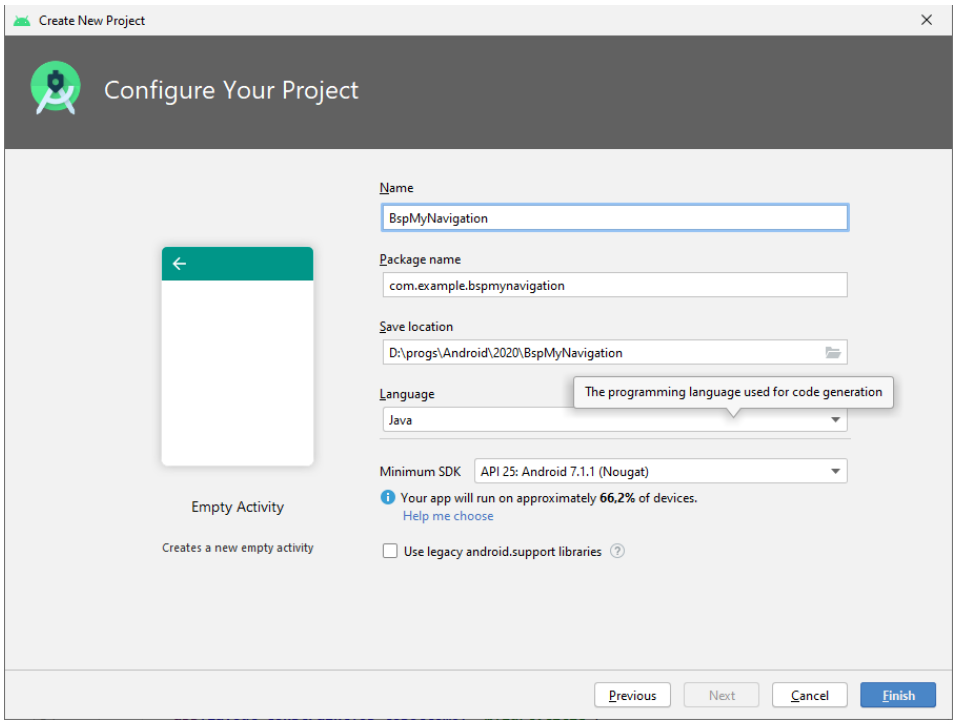


Abbildung 160 Projektname BspMyNavigation

11.6.3 Vorarbeiten

Anlegen einer Datei im Eintrag „drawable“
Name: rounded_button_settings.xml

Inhalt:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval" >
    <solid android:color="#DEFF59"></solid>
</shape>
```

Eintragen einer Farbe:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#6200EE</color>
  <color name="colorPrimaryDark">#3700B3</color>
  <color name="colorAccent">#03DAC5</color>
  <color name="colorButton">#536DFE</color>
</resources>
```

strings.xml:

```
<resources>
  <string name="app_name">BspMyNavigation</string>
  <string name="tvtilte">Eigene Navigation</string>

  <string name="bnvorspeisen">Vorspeisen</string>
  <string name="bnhauptspeisen">Hauptspeisen</string>
  <string name="bnnachspeisen">Nachspeisen</string>
  <string name="title_activity_vorspeisen">VorspeisenActivity</string>
  <string name="title_activity_hauptspeisen">HauptspeisenActivity</string>

  <string name="tab_zucchini">Zucchini</string>
  <string name="tab_pizza">Pizza</string>


</resources>
```

11.6.4 Startbildschirm

Das Startview besteht aus drei Schaltern, die aber durch eine Style verändert wurden.

Activity_main.xml

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

```
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <TextView
    android:id="@+id/tvtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:text="@string/tvtilte"
    android:textColor="#FF5252"
    android:textSize="36dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="32dp"
    android:layout_marginTop="64dp"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="24dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitle">

    <Button
      android:id="@+id/bnvorspeisen"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginLeft="40dp"
      android:layout_marginTop="20dp"
      android:layout_marginRight="40dp"
      android:layout_marginBottom="20dp"
      android:layout_weight="1"
      android:background="@drawable/rounded_button_settings"
      android:onClick="bnvorspeisen_Click"
      android:text="@string/bnvorspeisen"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:textColor="@color/colorButton" />

    <Button
      android:id="@+id/bnhauptspeisen"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginLeft="40dp"
      android:layout_marginTop="20dp"
      android:layout_marginRight="40dp"
      android:layout_marginBottom="20dp"
      android:layout_weight="1"
      android:background="@drawable/rounded_button_settings"
```

```
      android:onClick="bnhauptspeisen_Click"
      android:text="@string/bnhauptspeisen"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:textColor="@color/colorButton" />

    <Button
      android:id="@+id/bnnachspeisen"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginLeft="40dp"
      android:layout_marginTop="20dp"
      android:layout_marginRight="40dp"
      android:layout_marginBottom="20dp"
      android:layout_weight="1"
      android:background="@drawable/rounded_button_settings"
      android:onClick="bnnachspeisen_Click"
      android:text="@string/bnnachspeisen"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:textColor="@color/colorButton" />
  </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

//import com.example.bspmynavigation.vorspeisen.VorspeisenActivity;
//import com.example.bspmynavigation.hauptspeise.HauptspeisenActivity;
//import com.example.bspmynavigation.nachspeise.NachspeisenActivity;

public class MainActivity extends AppCompatActivity {

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


  public void bnvorspeisen_Click(View view) {
    // Intent intent = new Intent(this,  VorspeisenActivity.class);
    // startActivity(intent);
    }


  public void bnhauptspeisen_Click(View view) {
    // Intent intent = new Intent(this,  HauptspeisenActivity.class);
    // startActivity(intent);
  }


  public void bnnachspeisen_Click(View view) {
```

```
        // Intent intent = new Intent(this,  NachspeisenActivity.class);
        // startActivity(intent);
    }
}
```

Einige Zeilen sind auskommentiert, da diese Activites noch nicht eingebaut sind.

Die drei neuen Activities saollten / müssen in jeweils ein neues Package eingefügt werden.

11.6.5 Neue Packages

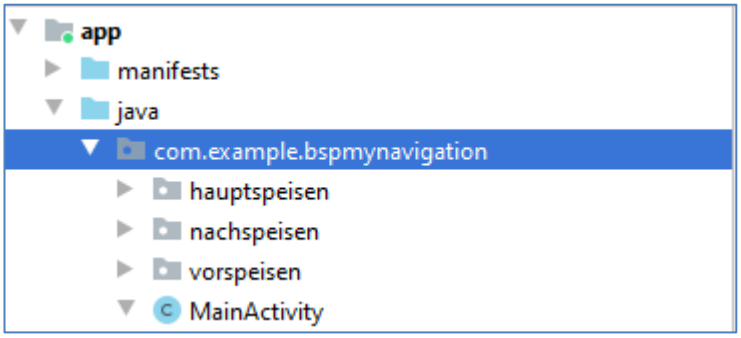


Abbildung 161 Anzeige der drei neuen Packages

Vorgehensweise:

- Anklicken des Eintrags „com.example.bspmynavigation
- Rechte Maustaste
- Eintrag „New“
- Package
- Eingabe „vorspeisen“

Vorgehensweise:

- Anklicken des Eintrags „com.example.bspmynavigation
- Rechte Maustaste
- Eintrag „New“
- Package
- Eingabe „hauptspeisen“

Vorgehensweise:

- Anklicken des Eintrags „com.example.bspmynavigation
- Rechte Maustaste
- Eintrag „New“
- Package
- Eingabe „nachspeisen“

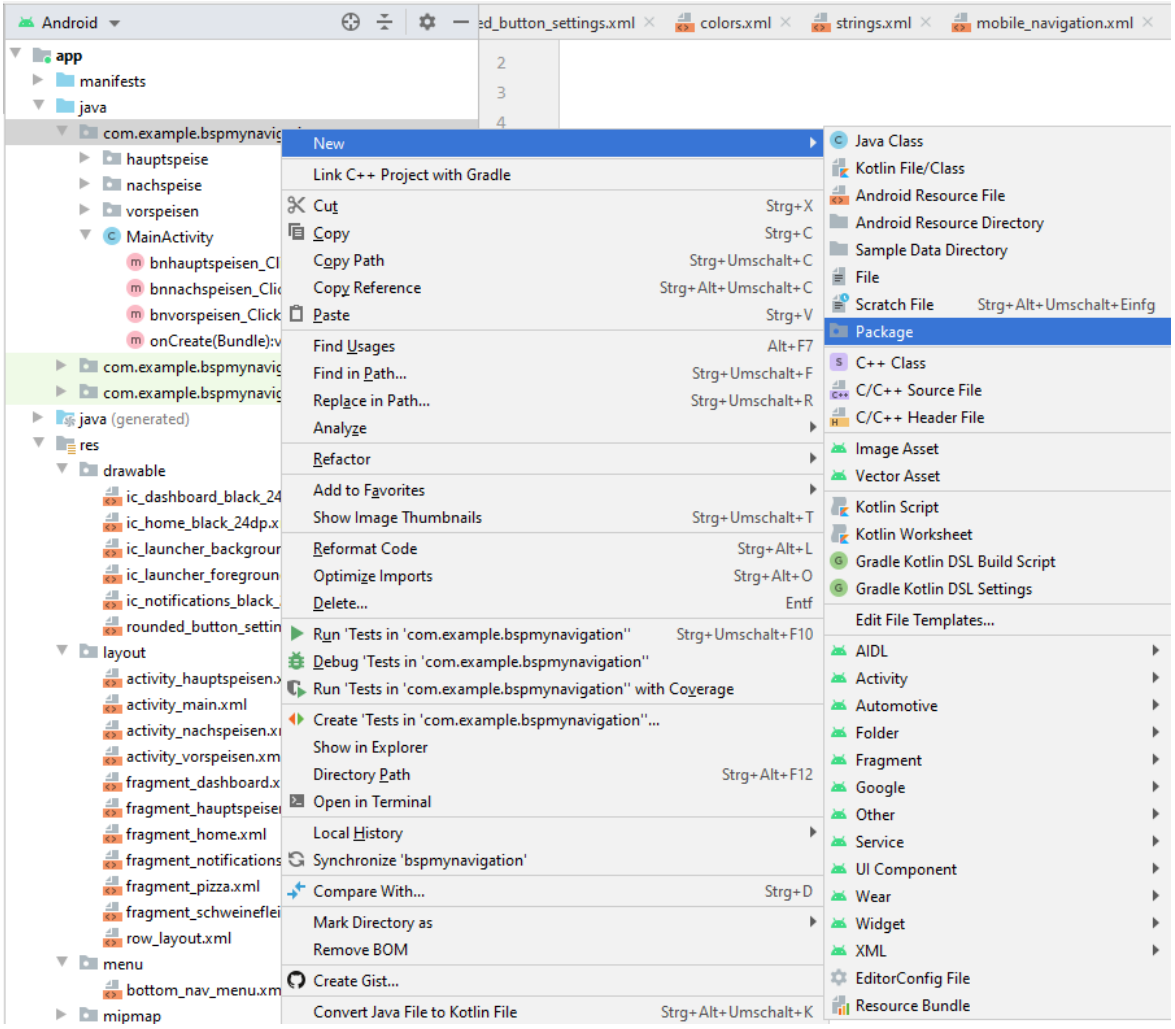


Abbildung 162 Neues Package

Es erscheint folgendes Dialogfenster:

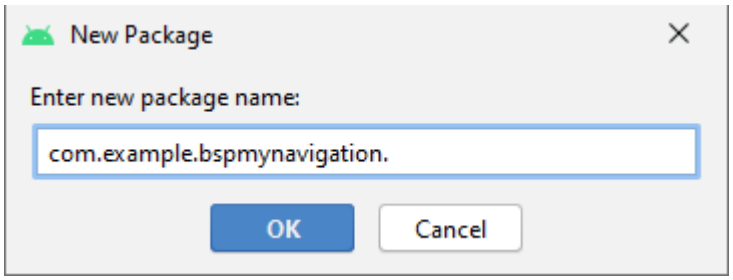


Abbildung 163 Eingabe des Namens des neuen Package

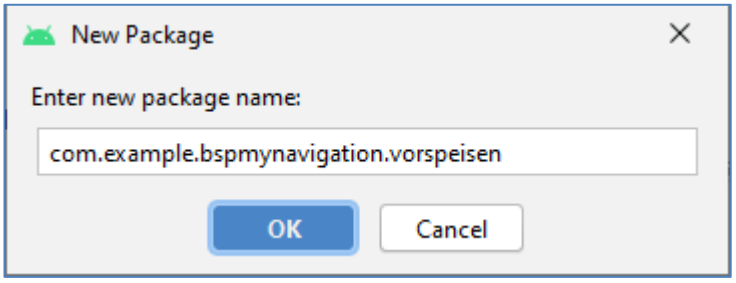


Abbildung 164 Eingabe des Namens "vorspeisen"

Nun kann man die drei neuen Activities einfügen.

11.6.6 Bottom Navigation Activity

Eintragen der Bottom-Navigation:

- Anklicken des Package „vorspeisen.
- Rechte Maustaste
- Eintrag „New“
- Eintrag Activity
- Eintrag „Bottom Navigation Activity“

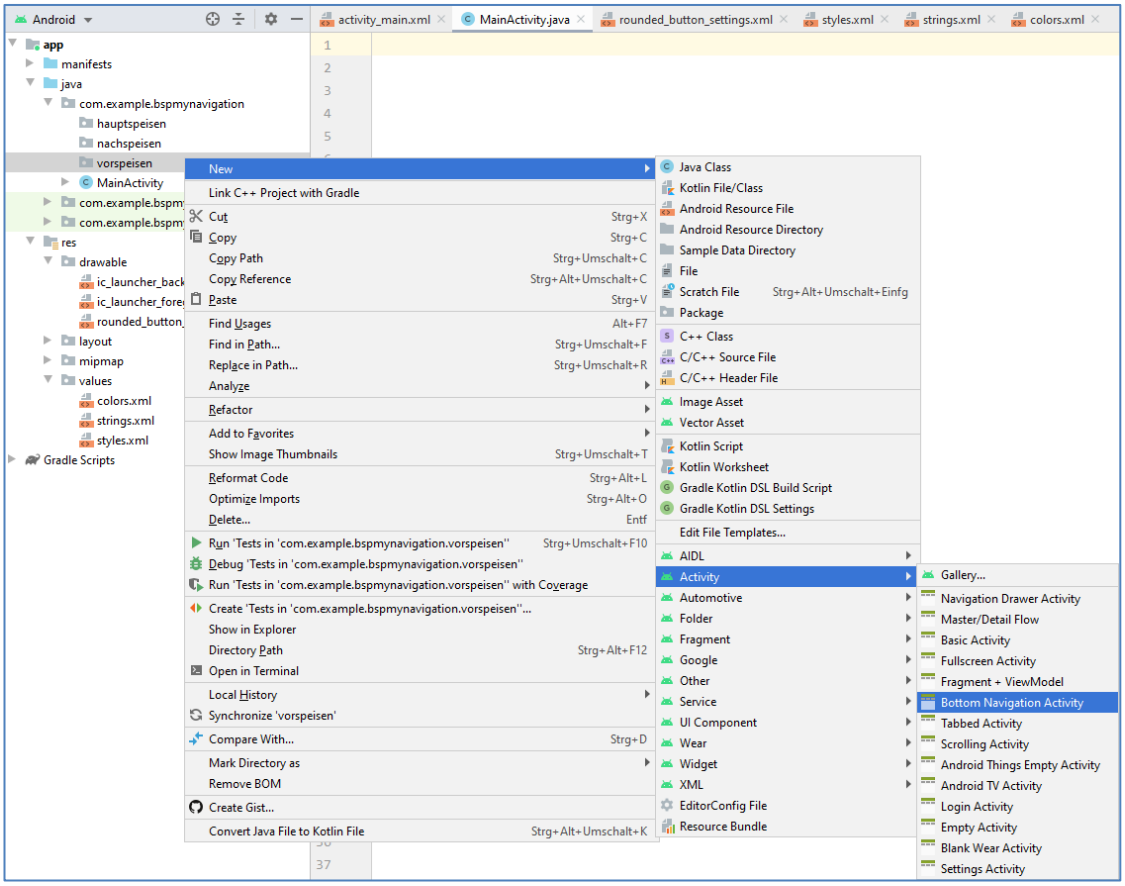


Abbildung 165 Eintragen der Bottom-Navigation

Name der neuen Activity: VorspeisenActivity

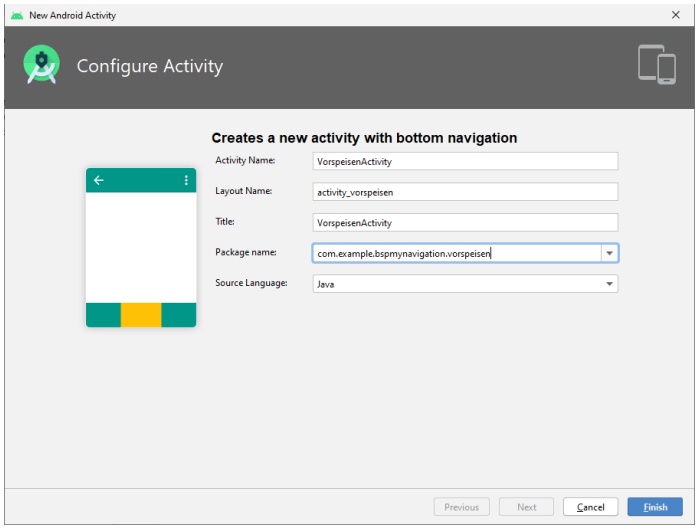


Abbildung 166 Einfügen der Activity VorspeisenActivity

Durch die neue Activity sind die bekannten Elemente hinzugefügt:

- Datei „strings.xml“
  - <string name="title_home">Home</string>
  - <string name="title_dashboard">Dashboard</string>
  - <string name="title_notifications">Notifications</string>
- Eintrag „res/menu“
  - Datei „bottom_nav_menu.xml“
- Eintrag „res/layout“
  - Datei „activity_vorspeisen.xml“
  - Datei „fragment_dashboard.xml“
  - Datei „fragment_home.xml“
  - Datei „fragment_notifications.xml“
- Eintrag „res/drawable“
  - Datei „ic_dashboard_black_24dp.xml“
  - Datei „ic_home_black_24dp.xml“
  - Datei „ic_notifications_black_24dp.xml“
- Eintrag „java/vorspeisen“
  - Datei „VorspeisenActivity.java“
  - Ein neues Packages „ui“
  - Drei neue Packages im Package „ui“
    - dashboard
    - home
    - notification

In den drei Java-Dateien „**DashboardFragment.java**“, „**HomeFragment .java**“ und „**NotificationsFragment .java**“ muss die Verknüpfung zum Resourcen-Ordner **eingetragen** werden.

```
import com.example.bspmynavigation.R;
```

Gelöscht werden muss folgende Zeile:

```
import com.example.bspmynavigation.vorspeisen.R;
```

Nun kann man in MainActivity den Quellcode der Vorspeisen aktivieren.

```
import com.example.bspmynavigation.vorspeisen.VorspeisenActivity;

  public void bnvorspeisen_Click(View view) {
    Intent intent = new Intent(this,  VorspeisenActivity.class);
    startActivity(intent);
  }
```

11.6.7 Tabbed Bar Activity

Eintragen der Tabbed Activity:

- Anklicken des Package „hauptspeisen.
- Rechte Maustaste
- Eintrag „New“
- Eintrag Activity
- Eintrag „Tabbed Activity“

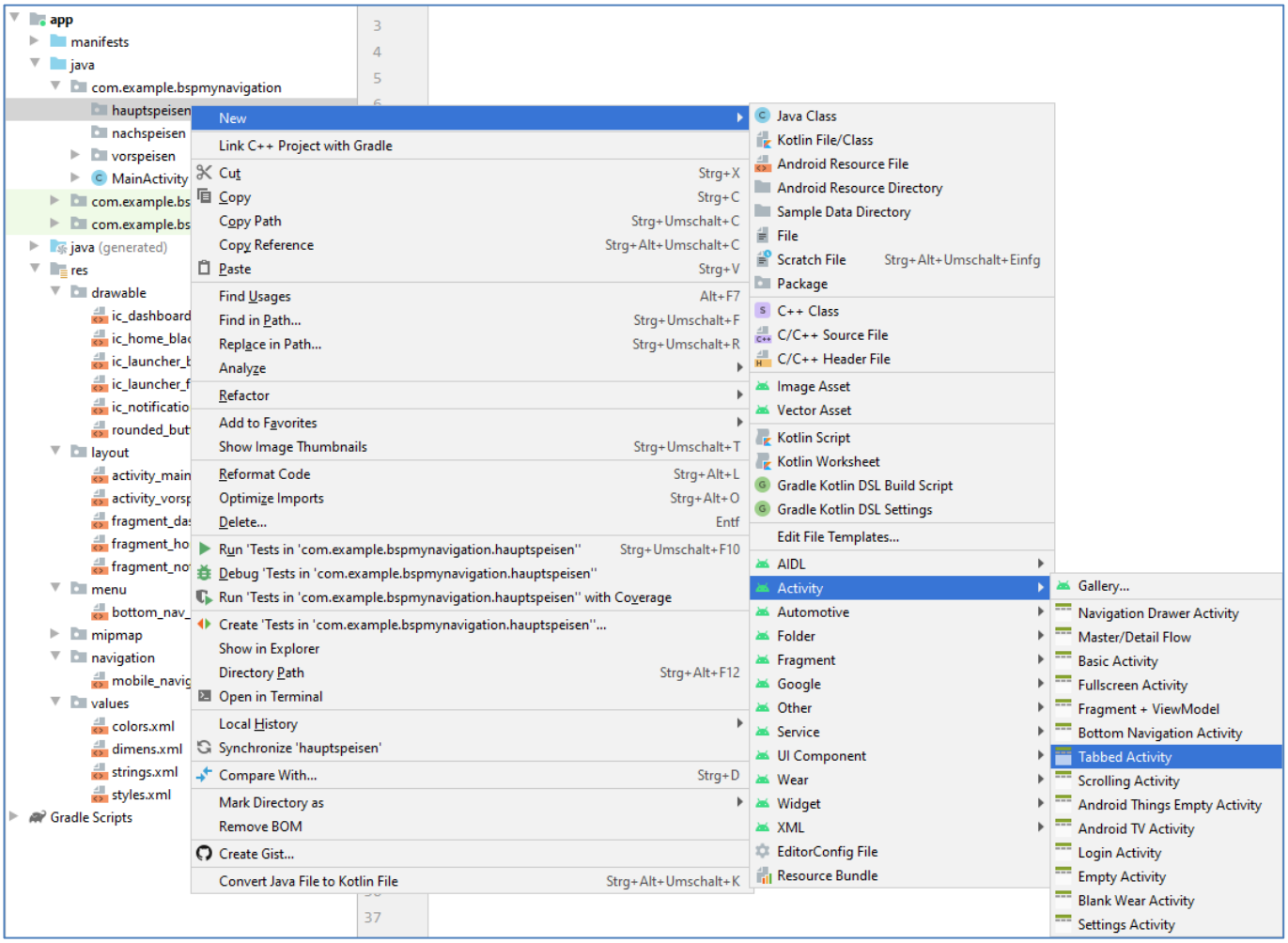


Abbildung 167 Eintragen der Tabbed Activity

Name der neuen Activity: HauptspeisenActivity

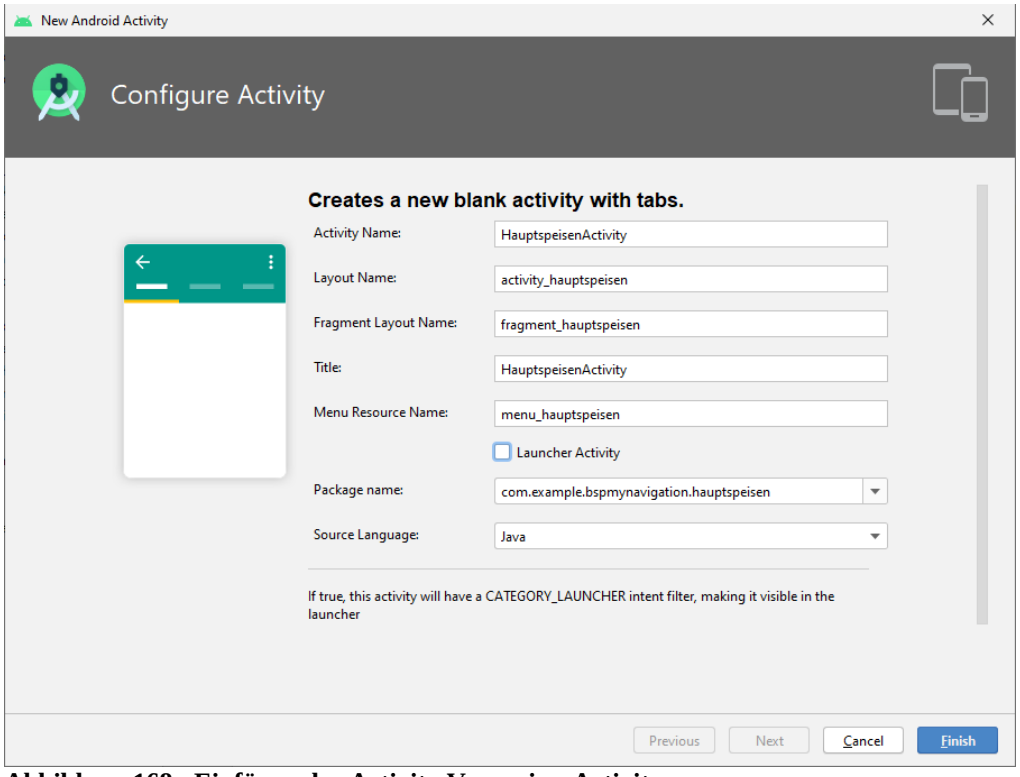


Abbildung 168 Einfügen der Activity VorspeisenActivity

Durch die neue Activity sind die bekannten Elemente hinzugefügt:

- Datei „strings.xml“
  - <string name="tab_text_1">Tab 1</string>
  - <string name="tab_text_2">Tab 2</string>
- Eintrag „res/layout“
  - Datei „activity_hauptspeisen.xml“
  - Datei „fragment_notifications.xml“
- Eintrag „java/hauptspeisen“
  - Datei „HauptspeisenActivity.java“
  - Ein neues Packages „ui.main“
  - Drei Dateien
    - PageViewModel.java
    - PlaceholderFragment.java
    - SectionsPageAdapter.java

Am einfachsten ist es, nun zwei neue Fragmente zu erstellen und die PlaceHolder-Variante zu löschen.

- Anklicken „hauptspeisen/ui.main“
- Anlegen eines neuen Fragmentes
- Name: PizzaFragment

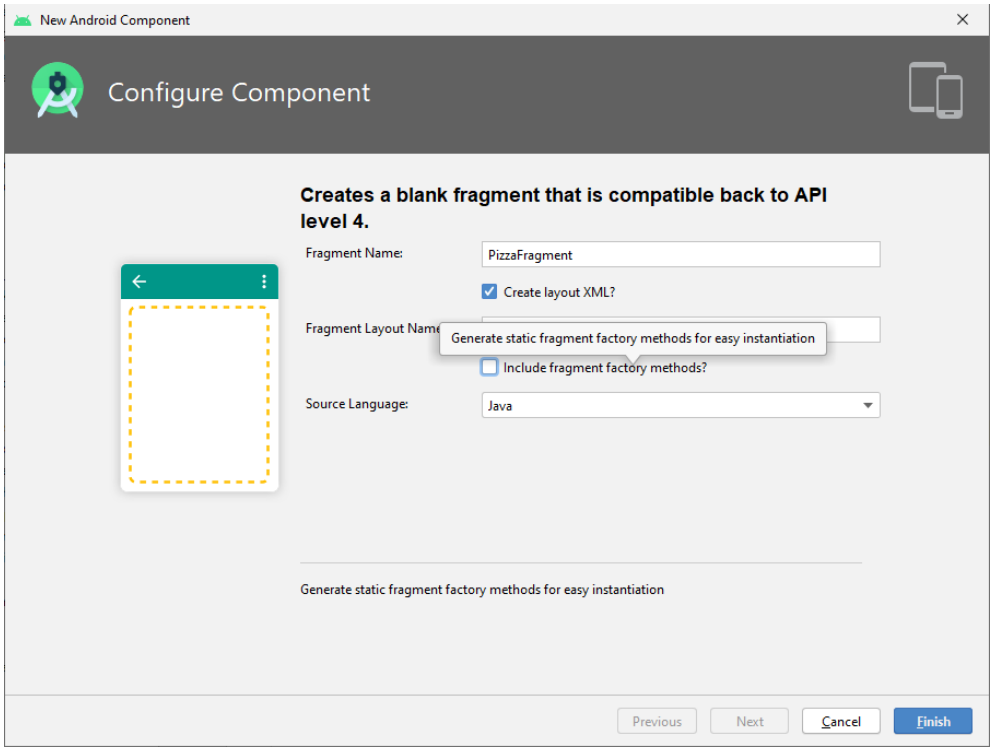


Abbildung 169 PizzaFragment erstellen

- Anklicken „hauptspeisen/ui.main“
- Anlegen eines neuen Fragmentes
- Name: ZucchiniFragment

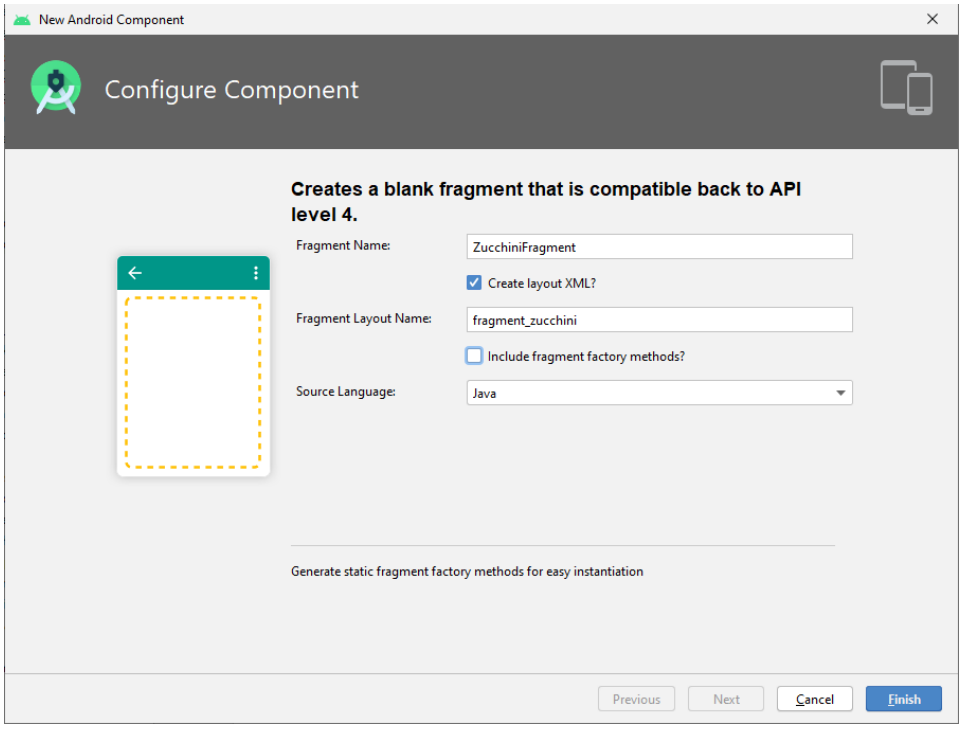


Abbildung 170 ZucchiniFragment erstellen

Nun löschen der beiden Placeholder-Dateien

- Placeholder.java
- PageViewModel.java

In den Java-Dateien „**PizzaFragment.java**“, „**ZucchiniFragment.java**“, „**SectionsPagerAdapter.java**“ und „**HauptspeisenActivity.java**“ muss die Verknüpfung zum Resourcen-Ordner **eingetragen** werden.

```
import com.example.bspmynavigation.R;
```

Gelöscht werden muss folgende Zeile:

```
import com.example.bspmynavigation.vorspeisen.R;
```

Ändern der Fragmente in SectionsPagerAdapter.java (**Ausschnitte**):

```
import com.example.bspmynavigation.R;


    @StringRes
    private static final int[] TAB_TITLES = new int[]{
            R.string.tab_pizza,
            R.string.tab_zucchini
    };

    @Override
    public Fragment getItem(int position) {
        // getItem is called to instantiate the fragment for the given
page.
        // Return a PlaceholderFragment (defined as a static inner class
below).
        switch (position) {
            case 0:
                return new ZucchiniFragment();
```

```
            case 1:
                return new PizzaFragment();
            default:
                return null;
        }
    }

    @Override
    public int getCount() {
        // Show 2 total pages.
        return TAB_TITLES.length;
    }
}
```

Nun kann man in MainActivity den Quellcode der Hauptpeisen aktivieren.

```
import com.example.bspmynavigation.hauptspeisen.HauptspeisenActivity;

  public void bnvorspeisen_Click(View view) {
    Intent intent = new Intent(this,  HauptspeisenActivity.class);
    startActivity(intent);
  }
```

11.6.8 ListView

Erstellen der row_layout.xml Datei

In Ornder „layout“ die Datei „row_layout.xml“ erstellen.

Eintragen des Inhalts:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rowTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:textSize="18sp"
    >
</TextView>
```

Eintragen der Empty Activity und Einfügen einer ListView:

- Anklicken des Package „nachspeisen.
- Rechte Maustaste
- Eintrag „New“
- Eintrag Activity
- Eintrag „Empty Activity“

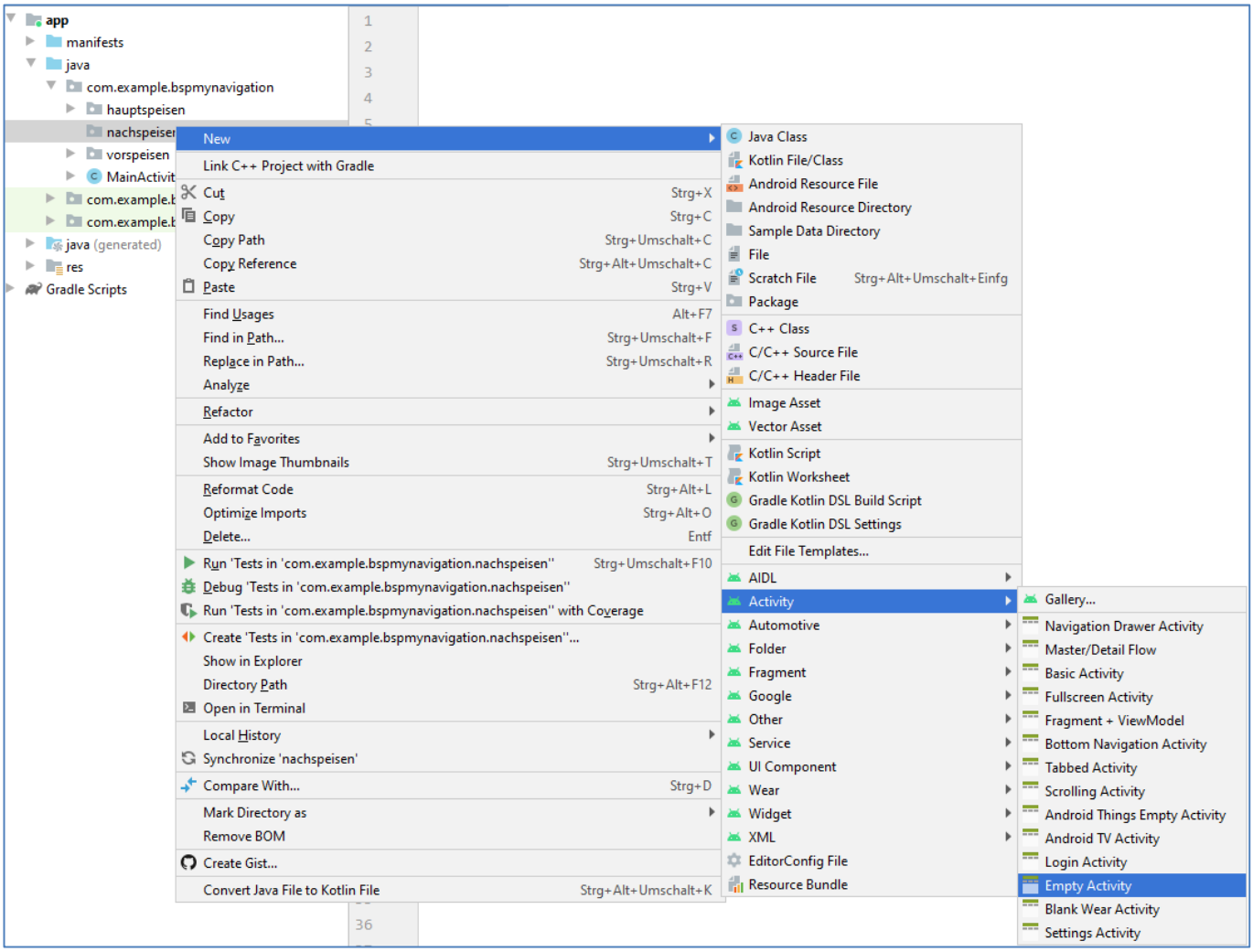


Abbildung 171 Eintragen der Empty Activity

Name der neuen Activity: NachspeisenActivity

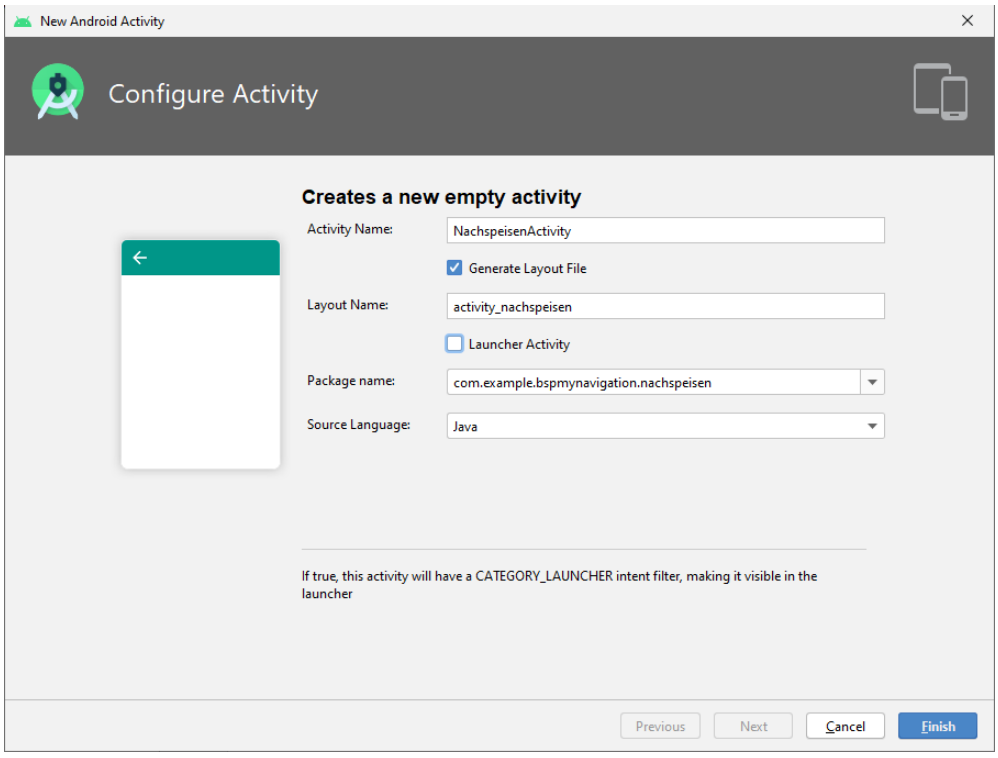


Abbildung 172 Einfügen der Activity NachspeisenActivity

Durch die neue Activity sind die bekannten Elemente hinzugefügt:

- Datei „strings.xml“
  - <string name="tab_text_1">Tab 1</string>
  - <string name="tab_text_2">Tab 2</string>
- Eintrag „res/layout“
  - Datei „activity_hauptspeisen.xml“
  - Datei „fragment_notifications.xml“
- Eintrag „java/hauptspeisen“
  - Datei „HauptspeisenActivity.java“
  - Ein neues Packages „ui.main“
  - Drei Dateien
    - PageViewModel.java
    - PlaceholderFragment.java
    - SectionsPageAdapter.java

Am einfachsten ist es, nun zwei neue Fragmente zu erstellen und die PlaceHolder-Variante zu löschen.

- Anklicken „hauptspeisen/ui.main“
- Anlegen eines neuen Fragmentes
- Name: PizzaFragment

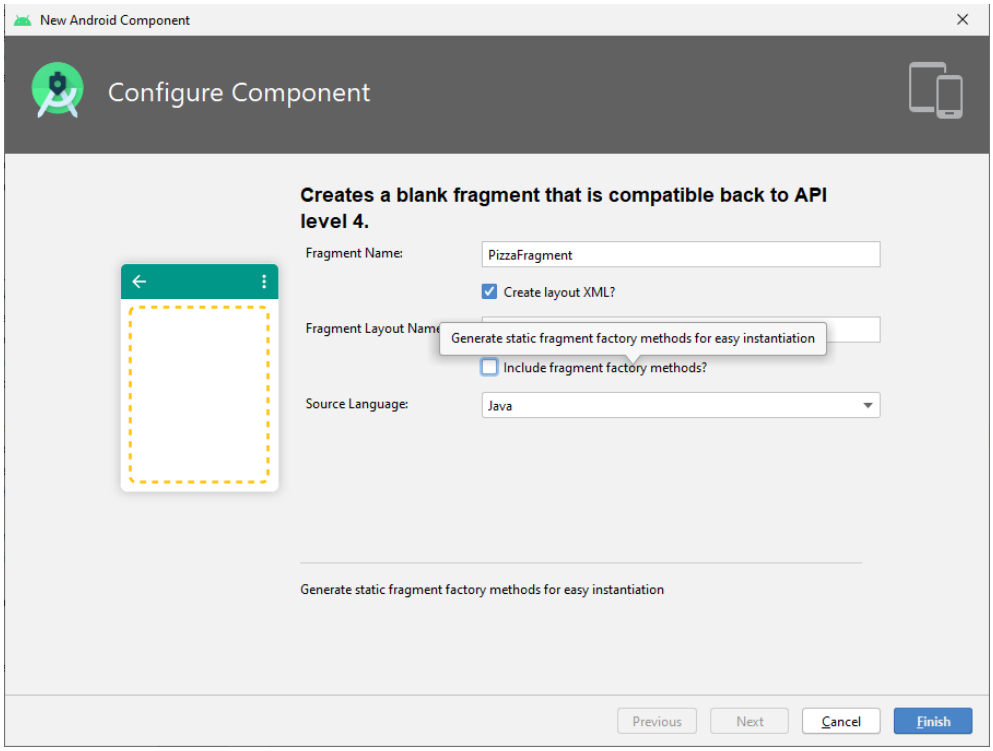


Abbildung 173 PizzaFragment erstellen

In der Java-Datei „**NachspeisenActivity.java**“ muss die Verknüpfung zum Resourcen-Ordner **eingetragen** werden.

```
import com.example.bspmynavigation.R;
```

Im Ordner „res/layout“ gibt es nun die Datei „activity_nachspeisen.xml“.

Nun kann man in MainActivity den Quellcode der Hauptpeisen aktivieren.

```
import com.example.bspmynavigation.nachspeisen.NachspeisenActivity;

  public void bnvorspeisen_Click(View view) {
    Intent intent = new Intent(this,  NachspeisenActivity.class);
    startActivity(intent);
  }
```

Beim Testen der Nacxhspeise erscheint nun eine leere Seite.

activity_nachspeisen.xml:

Hier wird erstmal eine ListView eingetragen.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

In der Java Datei wird nun eine einfache Liste zur Anzeige gebracht.

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.*;

import com.example.bspmynavigation.R;

public class NachspeisenActivity extends AppCompatActivity {
```

```
  private String[] favoriteLanguages = new String[] {
    "Eis", "Wackelpudding", "Heuschrecken",
    "Kalte Himbeeren mit heißen Eis", "Magop-Törtchen", "Germknödel",
    "Baileys Panna cotta", "Vanille-Lebkuchen-Pudding",
    "Heidelbeer Amarettini Dessert",  "Dessert mit Weintrauben",
    "Solero Dessert", "Erdbeer-Kokos-Dessert",  "Giotto-Creme-Dessert",
    "Bratapfeleis",  "Crema Catalana", "Vanille-Lebkuchen-Pudding"
  };


  private ListView listview = null;
  private ArrayAdapter<String> listAdaper;


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

    listview = findViewById(R.id.listview);
    listAdaper = new ArrayAdapter<String>(this,
          R.layout.row_layout, this.favoriteLanguages);
    listview.setAdapter(listAdaper);

   listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick(AdapterView<?> parent,
                              View view, int position, long id) {
        Log.i("********************* clicked","position: " + position);
        String rezept = (String) listAdaper.getItem(position);
        showRezept(rezept);
      }

    });  // setOnItemClickListener

  }  // onCreate

  private void showRezept(String rezept) {
      showToast(this,"Rezept: "+rezept);
  }

    // besser wäre die Klasse Basis.java
  public void showToast(Context context, String text) {
    Toast.makeText(context, text, Toast.LENGTH_LONG).show();
  }
}
```

12 Activity aufrufen

Vorgehensweise:

12.1 Neue Activity erstellen, mittels Menü)

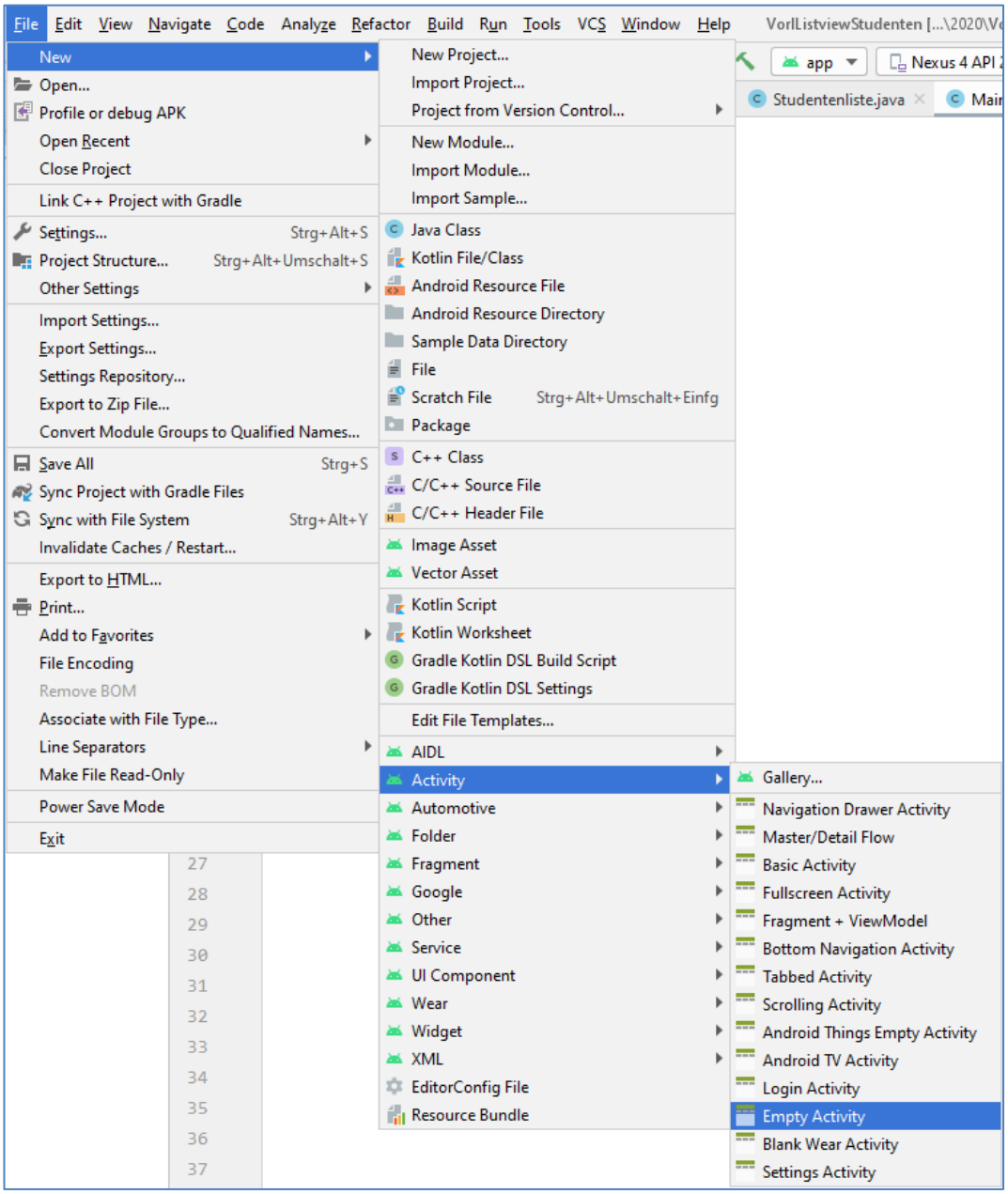


Abbildung 174 Erstellen einer neuen Activity

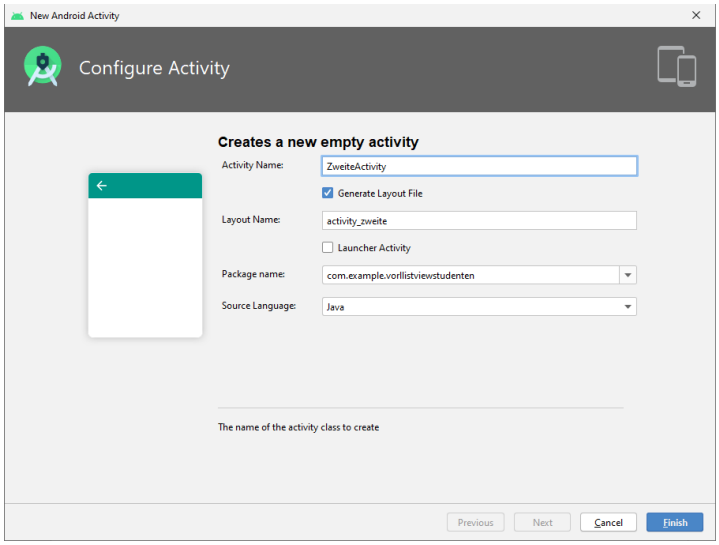


Abbildung 175 Eingabe des Namens für die neue Activity

Nun hat man eine Layoutdatei mit dem Namen „activity_zweite.xml“ und eine Javadatei mit dem Namen „ZweiteActivity.java“.

In der Haupt-Activity muss man nun ein Event auslösen können:

- Schalter-Click
- Long-Press
- Short-Press
- Optionsmenu (drei Punkte)
- Gesten

In der „Click-Methode“ muss man nun folgenden Code eintragen:

Die Parameter sind optional:

- action
  - eine Nummer, um der zweiten Activity den Actiontyp anzuzeigen
  - Change, Delete, New
  - Drucken, Senden
- MyData data
  - Eine Instanz einer Klasse, die Daten für die zweite Activity bereitstellt.
  - Die Klasse „ZweiteActivity „ hat eine Konstante für den Parameter
    - `public static final String DATA = "MYDATA";`
  - Die Klasse sollte die Schnittstelle Serializable haben
    - `implements Serializable`

```
private void showZweiteActity(int action, MyData data  ) {
    // Erzeugen der zweiten View, Parameter erstellen  und starten
  Intent intent = new Intent(this, ZweiteActivity.class);
  Log.i("********** in showZweiteActity ","action: "+action);

    // füllen der Parameter, es gibt leider keinen Konstruktor
  intent.putExtra("ZweiteActivity.ACTION, action);
  intent.putExtras(MyData.DATA, data);
  startActivity(intent);
}
```

Falls man eine Rücksprung abfangen will (callback-Methode), so nimmt man diese Variante.

```
private void showZweiteActity(int action, MyData data  ) {
    // Erzeugen der zweiten View, Parameter erstellen  und starten
  Intent intent = new Intent(this, ZweiteActivity.class);
  Log.i("********** in showZweiteActity ","action: "+action);

    // füllen der Parameter, es gibt leider keinen Konstruktor
  intent.putExtra("ZweiteActivity.ACTION, action);
  intent.putExtras(MyData.DATA, data);
  int sendernr = 1;
  startActivityForResult(intent, sendernr);
}

@Override
protected void onActivityResult(
         int requestCode, int resultCode, @Nullable Intent data) {

  int sendernr = 1;
  super.onActivityResult(requestCode, resultCode, data);
  if (requestCode == sendernr) {  // korrekte Sender-Adresse?
    switch (resultCode) {
      case Activity.RESULT_OK:
        // Action, irgendwas neuladen
        break;

      case Activity.RESULT_CANCELED:
        Basis.showToast(this,"Schade, dass Sie Cancel gedrückt haben");
        break;
      }
    }  // switch
}
```

13 Optionsmenü

Dieses Kapitel zeigt die Verwendung eines Optionsmenü, also Anzeige eines Menüs nach dem Anklicken der drei Punkte.

13.1 Menü erstellen

- Anlegen eines Ordners „menu“ im Eintrag „res“
- Menü File
- Eintrag „New“
- Eintrag „Directory“

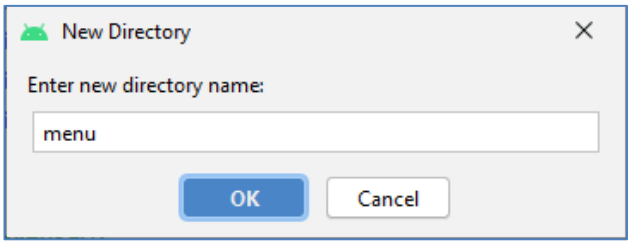


Abbildung 176 Anlegen eines Ordners "menu"

13.2 Datei menu.xml anlegen

- Anklicken des Ordners „menu“
- Menü File
- Eintrag „New“
- Eintrag „File“
- Eintragen „menu.xml“

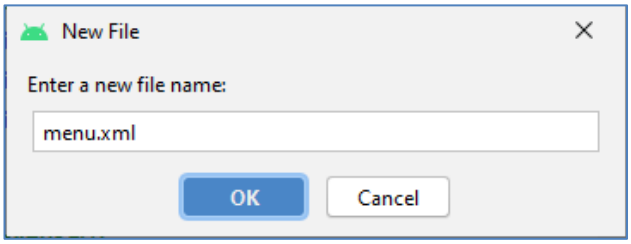


Abbildung 177 Erstellen der xml-Datei "menu.xml"

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

13.3 Menüeinträge erstellen

In diesem Menü werden dann die einzelnen Punkte eingetragen.
Jeder Eintrag enthält:

- id menu_menu1_item
- Symbol (optional) ic_menu1_24dp.xml
- Beschriftung menu1_item

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:id="@+id/menu_new_item"
    android:icon="@android:drawable/ic_menu_add"
    android:title="@string/new_item"
    app:showAsAction="always"

/>
```

Hinweise:

- Bitte immer nur „ic_menu“ Symbole auswahlen. Sonst kommt es zu unterschiedlichen Größen.

Optionale Attribute:
app:showAsAction="always"

Auswahl der Werte für die Einträge der ActionBar:

- Always
- collapseActionView
- ifRoom Anzeige links, wenn Platz ist
- never
- withText

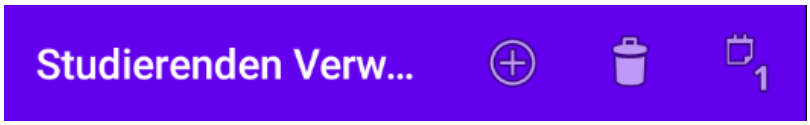


Abbildung 178 Beispiel für Symbole des Optionsmenus

Mit der obigen Lösung werden die Symbole nur aufgeklappt angezeigt. Bei mehreren Einträgen funktioniert das nicht.

Dann muss man die Variante nehmen, in der die Symbole bei Anklicken sichtbar sind:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:title="Main" app:showAsAction="always"
    android:icon="@android:drawable/ic_menu_preferences"
  >

    <menu>
      <item android:id="@+id/menu_new_item"
```

```
        android:icon="@android:drawable/ic_menu_add"
        android:title="@string/new_item"
        app:showAsAction="always"
      />

      <item android:id="@+id/menu_delete_item"
        android:icon="@android:drawable/ic_menu_delete"
        android:title="@string/delete_item"
        app:showAsAction="always"
      />

    </menu>
  </item>

</menu>
```



Abbildung 179 Neues Symbol im Optionsmenu

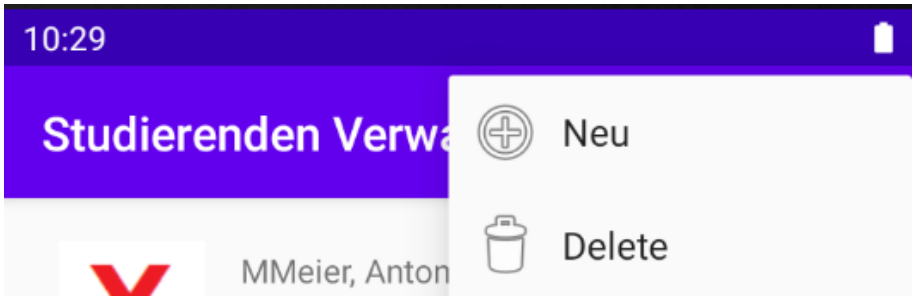


Abbildung 180 Anzeige der Einträge im Optionsmenu (mit Symbolen)

Bitte beachten:

- Es gibt zwei „Orte“ aus denen man Symbole holen kann.
  - @android:drawable/ic_menu_add
  - @drawable/ic_add_24dp
- Eigene Symbole werden immer mit der zweiten Varainte adressiert.

13.4 Optionsmenu einfügen in die mainActivity

In mainActivity wird die Methode onStart überschrieben:

// import android.app.ActionBar;

```
@Override
protected void onStart() {
  super.onStart();
  ActionBar ab = getActionBar();
  if (ab != null) {
    ab.setDisplayHomeAsUpEnabled(true);
  }
```

```
}
```

13.5 Optionsmenu-Start-Event abfangen

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.menu, menu);

  return super.onCreateOptionsMenu(menu);

}
```

13.6 Optionsmenu-Klick-Event abfangen

```
@Override
public boolean onOptionsItemSelected(MenuItem mnItem) {
  switch (mnItem.getItemId()) {
    case R.id.menu_menu1_item:
      Basis.showToast(this, "Menü new angeklickt");
      // Action
      break;

  } // switch
  return true;
}
```

14 Speichern mittels DataOutputStream und DataInputStream

Die Speicherung von Daten kann mittels Datenbank oder Binärdateien implementiert werden. Dieses Kapitel zeigt die Verwendung der Klassen DataOutputStream und DatainputStream. Diese Benutzung ist einfacher strukturiert, als die Verwendung einer Datenbank. Man hat hier die volle Kontrolle.

14.1 DataOutputStream

Mit Hilfe der Klasse „DataOutputStream“ kann man Daten binär speichern. Dies kann ein Datentyp oder ein Array einer Klasse sein.

14.1.1 Pures Java

```
FileOutputStream fout = null;
DataOutputStream dout = null;
try {
  fout = new  FileOutputStream(filename);
  dout = new DataOutputStream(fout);
  dout.writeInt(123);
  dout.writeDouble(123.34);
}
catch (IOException e) {
  System.error.println("saveData, Error in saveInput",e);
}
finally {
  try {
    if (fout!=null) fout.close();
    if (dout!=null) dout.close();
  }
  catch (IOException e) {
    System.error.println("Error in finally-block",e);
  }
}
```

Bemerkungen:

- fout und dout müssen „global“ deklariert sein, da diese im finally-Block benutzt werden.
- Der finally-Block wird immer aufgerufen! Deshalb muss man if Abfragen auf null einbauen.
- Da man die fout und dout schließt, braucht man einen zusätzlichen try/catch-Block.

14.1.2 DataOutputStream in Android

In Android ist das Speichern mittels DataOutputStream fast identisch mit der obigen Version. Der Unterschied liegt im filename. In Java kann man einen beliebigen Pfad angeben. In Android muss man den lokalen Pfad der Applikation benutzen.

```
FileOutputStream fout = null;
DataOutputStream dout = null;
try {
  FileOutputStream fout = context.openFileOutput(filename, MODE_PRIVATE);
  dout = new DataOutputStream(fout);
  dout.writeInt(123);
```

```
  dout.writeDouble(123.34);
}
catch (IOException e) {
  System.error.println("saveData, Error in saveInput",e);
}
finally {
  try {
    if (fout!=null) fout.close();
    if (dout!=null) dout.close();
  }
  catch (IOException e) {
    System.error.println("Error in finally-block",e);
  }
}
```

Bemerkungen:

- Der filename darf keinen Pfad haben, sondern nur den Dateiname:
  - test.bin
  - Optional könnte man eine lokale Ordnerstruktur anlegen.
- fout und dout müssen „global“ deklariert sein, da diese im finally-Block benutzt werden.
- Der finally-Block wird immer aufgerufen! Deshalb muss man if Abfragen auf null einbauen.
- Da man die fout und dout schließt, braucht man einen zusätzlichen try/catch-Block.
- Neuer Import:
  - import static android.content.Context.MODE_PRIVATE;

14.1.3 Methoden der Klasse DataOutputStream

- writeBoolean
- writeByte (Schreiben einer 8-Bit Vorzeichenzahl)
- writeChar (Schreiben einer 16-Bit vorzeichenlosen Zahl)
- writeDouble (Schreiben einer Double-Zahl)
- writeFloat (Schreiben einer Single-Zahl)
- writeInt (Schreiben einer 32-Bit Vorzeichenzahl)
- writeLong (Schreiben einer 64-Bit Vorzeichenzahl)
- writeShort (Schreiben einer 16-Bit Vorzeichenzahl)
- writeUTF (Schreiben von Strings) // Unicode Transformation Format
- writeChars

14.2 DataInputStream

Mit Hilfe der Klasse „DataInputStream“ kann man Daten binär lesen. Dies kann ein Datentyp oder ein Array einer Klasse sein.

14.2.1 Pures Java

```
FileInputStream fin = null;
DataInputStream din = null;
try {
  fin = new  FileInputStream(filename);
  din = new DataInputStream(fin);
  int i = din.readInt();
  syso(i);
  double d = din.readDouble();
  syso(d)
}
catch (IOException e) {
  System.error.println("loadData, Error in loadInput",e);
}
finally {
  try {
    if (fin!=null) fin.close();
    if (din!=null) din.close();
  }
  catch (IOException e) {
    System.error.println("Error in finally-block",e);
  }
}
```

Bemerkungen:

- fout und dout müssen „global“ deklariert sein, da diese im finally-Block benutzt werden.
- Der finally-Block wird immer aufgerufen! Deshalb muss man if Abfragen auf null einbauen.
- Durch den finally-Block wird sichergestellt, dass die Datei immer geschlossen wird.
- Da man die fout und dout schließt, braucht man einen zusätzlichen try/catch-Block.

14.2.2 DataInputStream in Android

In Android ist das Speichern mittels DataInputStream fast identisch mit der obigen Version. Der Unterschied liegt im filename. In Java kann man einen beliebigen Pfad angeben. In Android muss man den lokalen Pfad der Applikation benutzen.

```
FileInputStream fin = null;
DataInputStream din = null;
try {
  fin = context.openFileInput(filename);
  din = new DataInputStream(fin);
  int i = din.readInt();
  syso(i);
  double d = din.readDouble();
  syso(d)
}
catch (IOException e) {
  System.error.println("loadData, Error in loadInput",e);
}
finally {
  try {
    if (fin!=null) fin.close();
    if (din!=null) din.close();
  }
  catch (IOException e) {
```

```
    System.error.println("Error in finally-block",e);
  }
}
```

Bemerkungen:

- Der „filename“ darf keinen Pfad haben, sondern nur den Dateiname:
  - test.bin
  - Optional könnte man eine lokale Ordnerstruktur anlegen.
- fin und din müssen „global“ deklariert sein, da diese im finally-Block benutzt werden.
- Der finally-Block wird immer aufgerufen! Deshalb muss man if Abfragen auf null einbauen.
- Durch den finally-Block wird sichergestellt, dass die Datei immer geschlossen wird.
- Da man die fin und din schließt, braucht man einen zusätzlichen try/catch-Block.
- Neuer Import:
  - import static android.content.Context.MODE_PRIVATE;

14.2.3 Methoden der Klasse DataInputStream

- readBoolean
- readByte () Lesen einer 8-Bit Vorzeichenzahl
- readChar () Lesen einer 16-Bit vorzeichenlosen Zahl
- readDouble () Lesen einer Double-Zahl
- readFloat () Lesen einer Single-Zahl
- readInt () Lesen einer 32-Bit Vorzeichenzahl
- readLong () Lesen einer 64-Bit Vorzeichenzahl
- readShort ( ) Lesen einer 16-Bit Vorzeichenzahl
- readUTF() Lesen eines Strings // Unicode Transformation Format

14.3 Beispiele

14.3.1 Speichern und laden einer Instanz einer Klasse

Beim Speichern und laden einer Klasse muss man nur zwei zusätzliche Methoden einbauen:

- loadFromFile
- save2File

Beispiel:

```
// Neuer Import:
import static android.content.Context.MODE_PRIVATE;
import java.io.*;
import android.content.Context;
import android.util.Log;


public class Student {
  private static final String TAG = MainActivity.class.getSimpleName();

  public String name="";
  public int matrnr=1;
```

```
public double note=1.0;

public Student() {
}

public Student(String name, int matrnr, double note) {
  this.name=name;
  this.matrnr =matrnr;
  this.note = note;
}

public boolean loadFromFile(Context context, String filename) {
  FileInputStream fin = null;
  DataInputStream din = null;
  boolean ok=false;
  try {
    fin = context.openFileInput(filename);
    din = new DataInputStream(fin);
    name = din.readUTF();
    matrnr = din.readInt();
    note = din.readDouble();
    ok = true;
  }
  catch (IOException e) {
    Log.e(TAG, "loadFromFile, Error in loadfromFile student");
  }
  finally {
    try {
      if (fin!=null) fin.close();
      if (din!=null) din.close();
    }
    catch (IOException e) {
      System.error.println("Error in finally-block",e);
    }
  }  // finally

  return ok;
}  // loadFromFile


public boolean save2File(Context context, String filename) {
  FileOutputStream fout = null;
  DataOutputStream dout = null;
  boolean ok=false;
    try {
      fout = context.openFileOutput(filename, Context.MODE_PRIVATE);
      dout = new DataOutputStream(fout)
      dout.writeUTF(name);
      dout.writeInt(matrnr);
      dout.writeDouble(note);
      ok = true;
    }
    catch (IOException e) {
      Log.e(TAG, "save2File, Error in save2File student");
    }
  finally {
    try {
      if (fout!=null) fout.close();
      if (dout!=null) dout.close();
    }
```

```
        catch (IOException e) {
          System.error.println("Error in finally-block",e);
        }
      }  // finally

      return ok;
}
```

14.3.2 Speichern und laden von mehreren Instanzen einer Klasse

Bei einer ListView, einer grafischen Liste, hat man mehrere Studenten, Rezepte oder ähnliches. Dann kann man die obige Struktur als Vorlage verwenden, muss aber einige Änderungen vornehmen:

- Das Öffnen und Schließen darf nicht in den Methoden „save“ oder „load“ sein.
- „Save“ und „load“ bekommen einen DataOutputStream bzw. eine DataInputStream übergeben.
- Es müssen zwei neue Methoden programmiert werden, die den Dateinamen übergeben bekommen und das Array zum Speichern benutzen.

Lösung:

- Man kann nun in der Klasse „Student“ mehrere statische Methoden und ein statische Array implementieren.
- Man entwickelt eine separate Klasse „StudentenListe“, in der die Methoden gespeichert sind.

14.3.2.1 Geänderte Klasse Student

```
// Neuer Import:
import static android.content.Context.MODE_PRIVATE;
import java.io.*;
import android.util.Log;

public class Student {
  private static final String TAG = MainActivity.class.getSimpleName();

  public String name="";
  public int matrnr=1;
  public double note=1.0;

  public Student() {
  }

  public Student(String name, int matrnr, double note) {
    this.name=name;
    this.matrnr =matrnr;
    this.note = note;
  }

    // quasi ein Kopy-Konstruktor
  public void init(Student std) {
    this.nachname=std.nachname;
    this.matrnr=std.matrnr;
    this.semester = std.semester;
  }
```

```
  public boolean loadFromFile(DataOutputStream dout) {
    boolean ok=false;
    try {
      name = din.readUTF();
      matrnr = din.readInt();
      note = din.readDouble();
      ok = true;
    }
    catch (IOException e) {
      Log.e(TAG, "loadFromFile, Error in loadfromFile student");
    }

    return ok;
  }  // loadFromFile


  public boolean save2File(DataInputStream din) {
    boolean ok=false;
      try {
        dout.writeUTF(name);
        dout.writeInt(matrnr);
        dout.writeDouble(note);
        ok = true;
      }
      catch (IOException e) {
        Log.e(TAG, "save2File, Error in save2File student");
      }
    }  // finally

    return ok;
}
```

Änderungen:

- Die Methode „loadFromFile“ hat nun den Parameter vom Typ „DataInputStream“.
- Die Methode „loadFromFile“ öffnet nicht die Datei.
- Die Methode „loadFromFile“ schließt nicht die Datei.
- Die Methode „save2File“ hat nun den Parameter vom Typ „DataOutStream“.
- Die Methode „save2File“ öffnet nicht die Datei.
- Die Methode „save2File“ schließt nicht die Datei.

14.3.2.2 Studentenverwaltung des Speichern und Ladens

Bemerkungen:

- Man braucht einen festen Datennamen.
- Es gibt ein internes Array.
- Es gibt die Methode „save“.
- Es gibt die Methode „load“.
- Man kann einen neuen Studenten einfügen.
- Man kann einen Studenten löschen.
- Man kann einen Studenten ändern.
- Die Klasse ist als **Singleton** implementiert.

Quellcode:

```
// package

import android.content.Context;

import java.io.*;
import java.util.ArrayList;

public class Studentenliste {

  private final String FILENAME = "studenten.bin";

    // instance  ist DER BigMÄÄÄÄg
  private static Studentenliste instance = new Studentenliste();

  public static Studentenliste getInstance() {
    return instance;
  }
      // interne Array der Studenten
  public ArrayList<Student> list = new ArrayList<>();

  // Konstruktor, erzeugt eine Structure
  private Studentenliste() {
    insertStd2Liste();  // fügt Default-Werte zum Testen ein
  }

    // simuliert die Datenbankabfrage
  private void insertStd2Liste() {
    list.add(new Student("Meier",1,12365));
    list.add(new Student("Schmidt",3,12745));
  }

  public void save(Context context) {
    FileOutputStream fout = null;
    DataOutputStream dout = null;
    try {
      fout = context.openFileOutput(FILENAME, Context.MODE_PRIVATE);
      dout = new DataOutputStream(fout);
      dout.writeInt(list.size()); // Anzahl der Studneten!!!
      for (Student std : list) {
        std.write2File(dout);
      }
    }
    catch (IOException e) {
      android.widget.Toast.makeText(context,"Fehler beim Schreiben",
        android.widget.Toast.LENGTH_LONG).show();
    }
    finally {
      try {
        if (fout!=null) fout.close();
        if (dout!=null) dout.close();
      }
      catch (IOException e) {
      android.widget.Toast.makeText(context,"Fehler im finally-Block",
        android.widget.Toast.LENGTH_LONG).show();
      }
    }  // finally

} // save

  public void load(Context context) {
```

```
    FileInputStream fin = null;
    DataInputStream din = null;
    try {
      fin = context.openFileInput(FILENAME);
      din = new DataInputStream(fin);
      list.clear(); // eigentlich überflüssig
      int n = din.readInt();  // Anzahl der Studenten
      for (int i=0; i<n; i++) {
        Student std = new Student(); // erstz anlegen
        std.readFromFile(din); // dann lesen
        list.add(std);   // dann einfügen
      } // for
    }  // try
    catch (IOException e) {
      android.widget.Toast.makeText(context,"Fehler beim Laden",
        android.widget.Toast.LENGTH_LONG).show();
    }
    finally {
      try {
        if (fin!=null) fin.close();
        if (din!=null) din.close();
      }
      catch (IOException e) {
      android.widget.Toast.makeText(context,"Fehler im finally-Block",
        android.widget.Toast.LENGTH_LONG).show();
      }
    }  // finally

}  // load

  public void newStudent(Student std) {
    list.add(std);
  }

  public void deleteStudent(Context context,Student std) {
    list.remove(std);
  }

  public void changeStudent(Student parameterStd,Student stdChange) {
    for (Student std : list) {
      if (std==parameterStd) {
        std.init(stdChange);  // neu
        break;
      } // if
    }
}  // changeStudent

}
```

Aufruf:

- Der Aufruf geschieht in einer Klasse Adapter, die zu einer ListView gehört. Nähere Details im Kapitel Listview, Seite 120.

15 Die Datenbank SQLite

15.1 SQLite

15.1.1 Eigenschaften

- Einfache SQL-Datenbank
  - Eine Datei
  - 100 kB Größe
- Eigenschaften
  - Vollständig in Android integriert.
  - Einfach zu implementieren.
  - Select-Anweisung „anders“
  - Wrapper-Klasse: SQLiteOpenHelper
  - Cursor-Adapter für ListViews
  - Ausgewählte Datentypen
  - Integrierte RowID 64 Bit, ulong

15.1.2 Datentypen

SQLIte hat nur wenige Datentypen. Sie können aber alle „bekannten“ Datentypen benutzen. Diese Bezeichner werden dann in die internen Bezeichner (Datentypen) umgewandelt.

- INTEGER 1,2,4,6,8 Bytes
- FLOAT Double 8 Bytes
- NUMERIC
- TEXT UTF-8, UTF-16BE or UTF-16LE
- BLOB

15.1.3 Gemappte Datentypen

- BOOLEAN INTEGER 0=false, 1=true
- DATE/TIMESTAMP Real, Integer, Text
- VARCHAR Text
- CBLOB Blob

DATE / TIMESTAMP

- TEXT as ISO8601 strings
  - ("YYYY-MM-DD HH:MM:SS.SSS").
- REAL
  - as Julian day numbers, the number of days since noon in Greenwich on November 24, 4714 B.C. according to the proleptic Gregorian calendar.
- INTEGER as Unix Time,
  - the number of seconds since 1970-01-01 00:00:00 UTC.

Integer:

- INT
- INTEGER
- TINYINT
- SMALLINT
- MEDIUMINT
- BIGINT
- UNSIGNED BIG INT
- INT2
- INT8

Text:

- CHARACTER(20)
- VARCHAR(255)
- VARYING CHARACTER(255)
- NCHAR(55)
- NATIVE CHARACTER(70)
- NVARCHAR(100)
- TEXT
- CLOB

Real:

- REAL
- DOUBLE
- DOUBLE PRECISION
- FLOAT

Numeric:

- NUMERIC
- DECIMAL(10,5)

15.1.4 Aufbau in Android

In Android ist das Datenbanksystem SQLite fest integriert. Man benötigt nur zwei Klassen und wenig Code um auf eine Datenbank zugreifen zu können. Die Syntax ist aber **gewöhnungsbedürftig**! SQLite verzichtet auf einen Parser der SQL-Befehle und benutzt eigene Methoden inklusive einer Hashtable für die Parameter.

Um in Adroid eine Datenbank anzusprechen benötigt man nur eine Klasse, die von der Klasse „SQLiteOpenHelper“ abgeleitet ist. Beim ersten Ansprechen wird die Datenbank **automatisch** angelegt! Die SQL-Befehle sollte man aber in einer separaten Klasse auslagern!

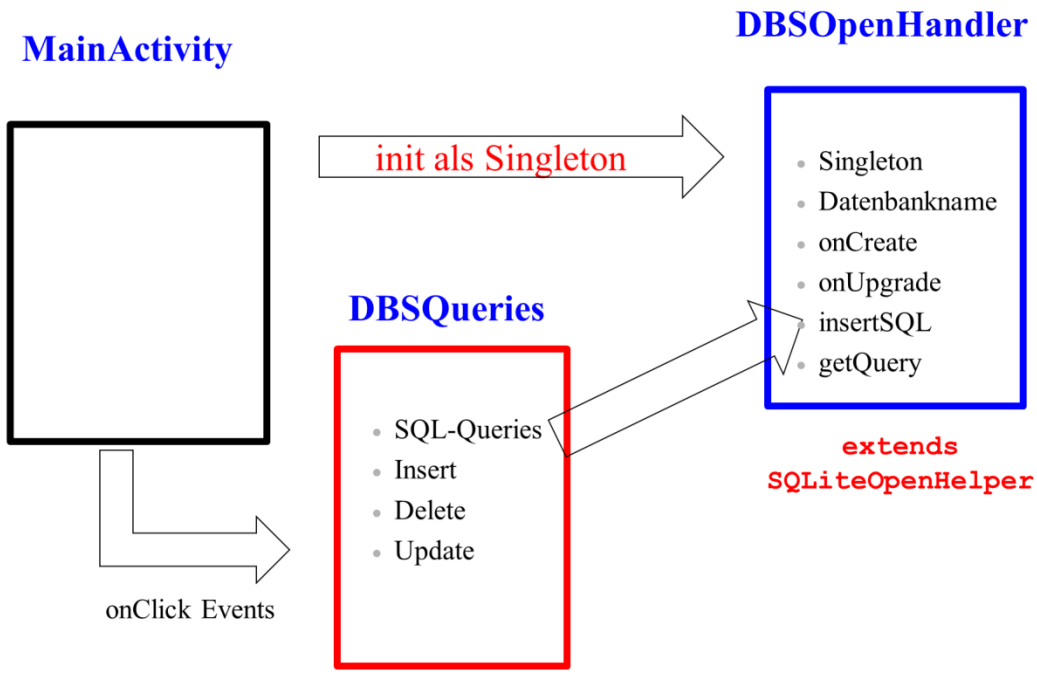


Abbildung 181 Übersicht über SQLIte in Android

Die Klasse DBSOpenHandler sollte als Singleton definiert oder mit statischen Methoden definiert werden. Man muss nur folgende Methoden überschreiben:

- public void onCreate(SQLiteDatabase db)
- public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

onCreate-Methode:
Die onCreate-Methode wird automatisch beim ersten Insert- oder Query-Befehl aufgerufen. Hier holt man sich die „Create"-Befehle und fügt diese mit „db.execSQL" aus.

Die Update-Methode ist nicht so wichtig. **Bitte beachten, bei einem Update werden automatisch alle Daten gelöscht!** Man sollte also eine etwas andere Updatetechnik verwenden!

Insert:
Die Insert-Methode ist ungewöhnlich. Hier erhält man eine Liste von Hashwerten und den Namen der Tabelle. So spart man sich den Parser.
Beispiel:

- values.put(NAME, std.name);
- values.put(MATRNR, std.matrnr);

getQuery:
Die getQuery-Anweisung ist genauso gewöhnungsbedürftig. Es gibt mehrere Methodenaufrufe!

15.1.5 Datenbank erzeugen

Das Datenbanksystem ruft automatisch onCreate beim ersten „Kontakt" auf.
In der Klasse DBSQueries wird eine Methode aufgerufen, die alle Tabelle erzeugt.

Beispiel:

```
public static String getCreateTables() {
  String s = "CREATE TABLE TEST ("+
    "_id  INTEGER PRIMARY KEY AUTOINCREMENT, "+
    "  NAME+" VARCHAR(30) DEFAULT '', "+
    "  PERSONALNR + "  INTEGER DEFAULT 0 "+
  " );";
  return s;
} // getCreateTables"
```

Bemerkungen:

- Sinnvollerweise sollte man Konstanten für die Attribute verwenden!
- Verwenden Sie **immer** für die normalen Attriburte Default-Werte!

Aufruf aus onCreate:

```
@Override
public void onCreate(SQLiteDatabase db) {
  String sql= DBSQueries.getCreateTables();
  Log.i(TAG, "********** String  getCreateTables");
  try {
    db.execSQL(sql);
    Log.i(TAG, "********** database are created (1)");
  }
  catch (SQLiteException e) {
    Log.i(TAG, "**********           no database are created", e);
  } finally {
    Log.i(TAG, "********** in finally onCreate");
  }
}
```

15.1.6 Insert

Wie oben erwaähnt, erhält die Insert-Methode eine Liste mit Hashwerten (Spaltenname/Inhalt) und den Tabellennamen. Diese Liste muss natürlich in DQBQueries aufgebaut werden:

```
public static long insertTest(String name, int personalnummer) {
    // holen der Datenbankverbindung
  DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();

  // Füllen der Liste mit den Hashwerten
  ContentValues values = new ContentValues(); // à la Hashmap
  values.put("NAME", name);
  values.put("PERSONALNR", personalnummer);

  // Aufrufen der insert-Methode
  return dbshandler.insertSQL(values, "TEST");
}
```

```
public long insertSQL(ContentValues values, String dbtable) {
  long rowId = -1;
  try {
    // open database
    SQLiteDatabase db = getWritableDatabase();
```

```
    Log.i(TAG, "Path: " + db.getPath());
    rowId = db.insert(dbtable, null, values);
  }
  catch (SQLiteException e) {
    Log.e(TAG, "insert()", e);
  } finally {
    Log.d(TAG, "insert(): rowId=" + rowId);  // wenn -1, dann Fehler!
  }
  return rowId;
}
```

15.1.7 getQuery

Die Query-Anweisung ist sehr ungewöhnlich!
SQLIte hat eine Methode, bei der alle Elemente einer normalen SQL-Anweisung als Parameter übergeben werden.

Parameter der getQuery-Methode:

- String table — Datenbanktabelle
- String[] columns — Spalten oder null
- String selection (where) — "name=? AND email=?"
- String[] selectionArgs — Parameter für die Where-Bedingungen
- String groupBy — Spalte(n)
- String having — Group by –Where Bedingung
- String orderBy — Sortierreihenfolge

Man muss nicht alle Parameter benutzen. Dann übergibt man den Wert „null“.

Beispiele:

Normale Syntax:

SELECT firstname, lastname FROM customer;
SELECT * FROM customer;

1. query("customer", null, null, null, null, null, null);
   - SELECT * FROM customer;

Der obige Befehl übergibt nur den Tabellennamen. SQLite nimmt dann alle Spalten und Zeilen!

2. query("customer", new String[] {lname, fname},null, null, null, null, null, null);
   - SELECT lname, fname FROM customer;

Der obige Befehl übergibt neben dem Tabellennamen auch ein **Array** mit den Spalten. SQLite nimmt dann die beiden Spalten, aber alle Zeilen!

3. query("customer", null, "pid>100", null, null, null, null);
   - SELECT * FROM customer WHERE pid>100;

Der obige Befehl übergibt den Tabellennamen und eine Where-Bedingung als String. SQLite nimmt dann alle Spalten. Für die Zeilen gilt die Where-Bedingung!

4. query("customer", null, "pid>?", new String[] {"100"},null, null, null);
    - SELECT * FROM customer WHERE pid>100;

Der obige Befehl übergibt den Tabellennamen und eine Where-Bedingung als String, die aber parametrisiert ist. Das Fragezeichen in der Where-Bedingung wird durch den String „100“ ersetzt. SQLite nimmt dann alle Spalten. Für die Zeilen gilt die Where-Bedingung!

5. query("customer", new String[] {lastname, fistname}, "salary>2000", null, "department", null, " department, lastname ASC");
    - SELECT lastname, firstname FROM employee WHERE salary>2000 GROUP BY department ORDER BY department , lastname ASC ;

Der obige Befehl übergibt den Tabellennamen und eine Where-Bedingung als String, ein Group By-Anweisung nach den Abteilungen. Sortiert wird nach den Abteilungen und den Nachnamen.

15.1.8 Löschen der Datenbank

Mit folgenden Befehlen können Sie die Datenbank resp. Den Simulator löschen:

- Menü Tools
- Eintrag AVD-Manager
- Auswählen des Simulators
- Den Pfeil-nach-unten mit der linken Maus anklicken
- Auswahl des Eintrags „Wipe Data“

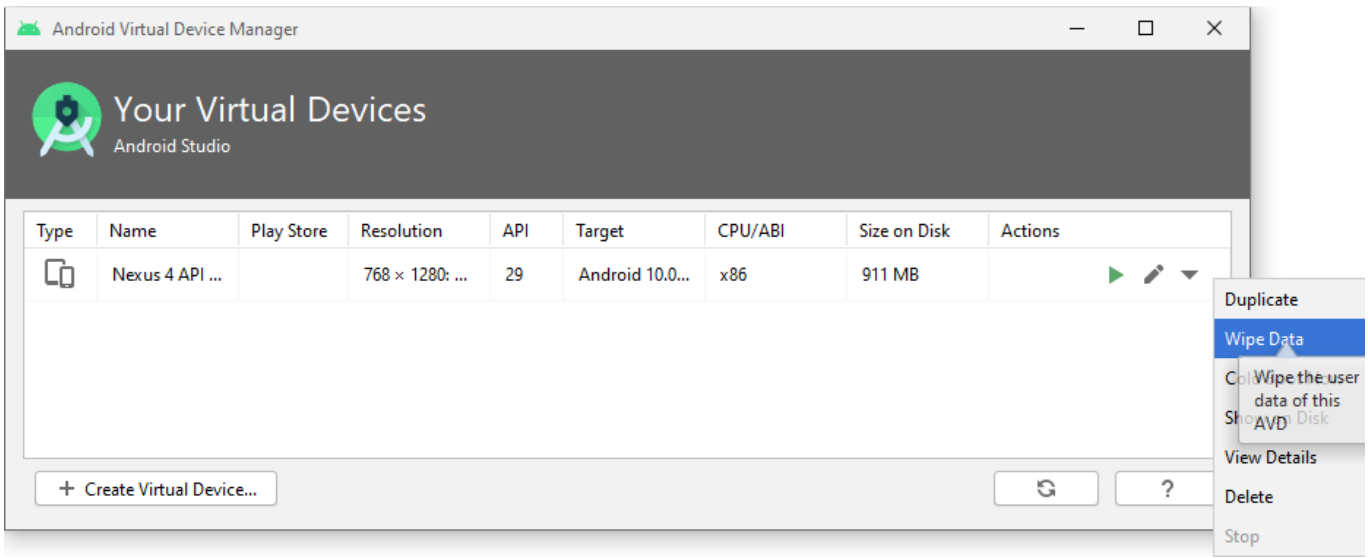


Abbildung 182 Löschen der Daten eines Simulators

15.2 Beispiel einer einfachen SQLite-App

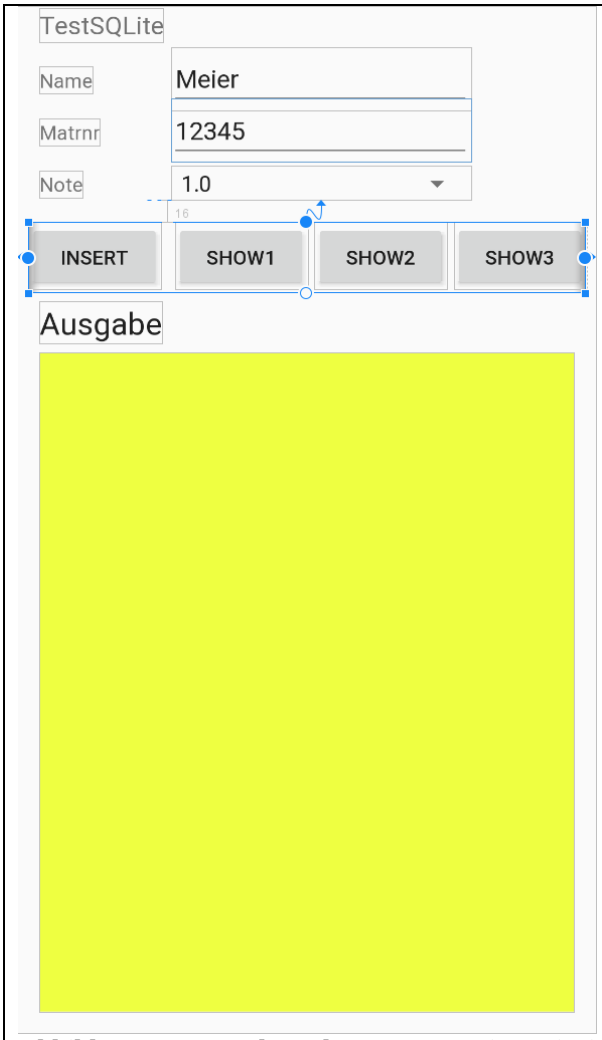


Abbildung 183 Struktur des ersten SQLite-Beispiels

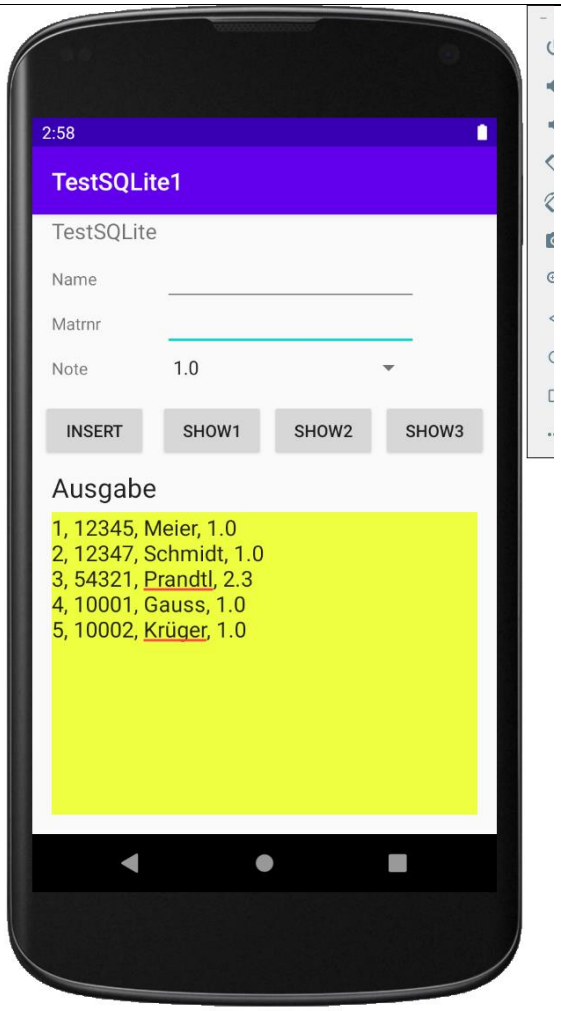


Abbildung 184 Musterlösung des ersten SQLIte-Beispiels

15.2.1 Anlegen eines Projektes

Anlegen eines neuen Projektes:
Name: TestSQLite1

Auswahl einer EmptyActivity

15.2.2 colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <color name="colorPrimary">#6200EE</color>
   <color name="colorPrimaryDark">#3700B3</color>
   <color name="colorAccent">#03DAC5</color>
   <color name="colorbackground">#EEFF41</color>
   <color name="colorbackgroundlistview">#E6EABD</color>
</resources>
```

15.2.3 strings.xml

```
<resources>
  <string name="app_name">TestSQLite1</string>
  <string name="tvtitle">TestSQLite</string>
  <string name="tvshowstdtitle">Liste der Studierenden</string>
  <string name="bninsert">Insert</string>
  <string name="bnshowstd1">Show1</string>
  <string name="bnshowstd2">Show2</string>
  <string name="bnshowstd3">Show3</string>

  <string name="tvstudent">Name</string>
  <string name="tvmatrnr">Matrnr</string>
  <string name="tvnote">Note</string>
  <string name="tveditor">Ausgabe</string>


  <string-array name="noten_arrays">
    <item>1.0</item>
    <item>1.3</item>
    <item>1.7</item>
    <item>2.0</item>
    <item>2.3</item>
    <item>2.7</item>
    <item>3.0</item>
    <item>3.3</item>
    <item>3.7</item>
    item>4.0</item>
    <item>4.0</item>
    <item>5.0</item>
  </string-array>
</resources>
```

15.2.4 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <TextView
    android:id="@+id/tvtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="1dp"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/tvtitle"
    app:layout_constraintStart_toStartOf="parent"
     app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvname"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="18dp"
    android:text="@string/tvstudent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitle" />

  <EditText
    android:id="@+id/tname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="55dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Meier"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvname"
    app:layout_constraintStart_toEndOf="@+id/tvname" />

  <TextView
    android:id="@+id/tvmatrnr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:text="@string/tvmatrnr"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvname" />

  <EditText
    android:id="@+id/tmatrnr"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="12345"
    android:inputType="number"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvmatrnr"
    app:layout_constraintEnd_toEndOf="@+id/tname"
    app:layout_constraintStart_toStartOf="@+id/tname" />


  <TextView
    android:id="@+id/tvnote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:text="@string/tvnote"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvmatrnr" />

  <Spinner
    android:id="@+id/spnote"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:entries="@array/noten_arrays"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvnote"
    app:layout_constraintEnd_toEndOf="@+id/tname"
    app:layout_constraintStart_toStartOf="@+id/tname" />

  <LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="50dp"
```

```
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spnote">

      <Button
        android:id="@+id/bninsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp"
        android:layout_weight="1"
        android:onClick="bninsert_Click"
        android:text="@string/bninsert" />

      <Button
        android:id="@+id/bnshowstd1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:onClick="bnshowStd1_Click"
        android:text="@string/bnshowstd1" />

      <Button
        android:id="@+id/bnshowstd2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:onClick="bnshowStd2_Click"
        android:text="@string/bnshowstd2" />

      <Button
        android:id="@+id/bnshowstd3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:onClick="bnshowStd3_Click"
        android:text="@string/bnshowstd3" />

    </LinearLayout>

    <TextView
      android:id="@+id/tveditor"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginStart="16dp"
      android:layout_marginTop="6dp"
      android:text="@string/tveditor"
      android:textAppearance="?android:attr/textAppearanceLarge"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/linearLayout1" />

  <EditText
    android:id="@+id/teditor"
    android:layout_width="match_parent"
```

```
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="6dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:background="@color/colorbackground"
    android:gravity="start|top"
    android:inputType="textMultiLine"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/tveditor"
    app:layout_constraintTop_toBottomOf="@+id/tveditor" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

15.2.5 Student.java

Anlegen der Klasse „Student“

Die Klasse selber ist übersichtlich.

```
public class Student  {

  public int pindex = -1;  // _id
  public String name = "";
  public int matrnr = -1;
  public double note = 0.0;

  public Student(String name, int matrnr,  double note) {
    this.name = name;
    this.matrnr = matrnr;
    this.note = note;
  }

  public Student() {
  }


    @Override
    public String toString() {
        return pindex+", "+matrnr+", "+name+", "+note;
    }


}
```

15.2.6 DBSOpenHandler.java

Die Klasse DBSOpenHandler ist immer gleich:

Man könnte den Datenbanknamen auch aus DBSQueries entnehmen:

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.*;
import android.util.Log;


public class DBSOpenHandler extends SQLiteOpenHelper {
  private static final String TAG = DBSOpenHandler.class.getSimpleName();

  private static DBSOpenHandler instance = null;


    // Singleton
  public static void createInstance(Context context,
                    String databasename, int databaseVersion) {
    if (instance==null) {
      instance = new DBSOpenHandler(context,
                          databasename, databaseVersion);
    }
  }


  public static DBSOpenHandler getInstance() {
    return instance;
  }


    // Konstruktor
  private DBSOpenHandler (Context context,
    String databasename, int databaseVersion) {
    super(context, databasename, null, databaseVersion);
  }


  @Override
  public void onCreate(SQLiteDatabase db) {
    String sql= DBSQueries.getCreateTables();
    Log.i(TAG, "********** String  getCreateTables");
    try {
      db.execSQL(sql);
      Log.i(TAG, "********** database are created (1)");
    }
    catch (SQLiteException e) {
      Log.i(TAG, "**********          no database are created", e);
    }
    finally {
        Log.i(TAG, "********** in finally onCreate");
    }
  }

    @Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
  Log.i(TAG, "Upgrade of the database from version "
        + oldVersion + " to "
        + newVersion + "; all datas are deleted");
  String sql= DBSQueries.getCreateTables();
  db.execSQL(sql);
  onCreate(db);
}
```

```
public long insertSQL(ContentValues values, String dbtable) {
  long rowId = -1;
  try {
    // open database
    SQLiteDatabase db = getWritableDatabase();
    Log.i(TAG, "Path: " + db.getPath());
    rowId = db.insert(dbtable, null, values);
  }
  catch (SQLiteException e) {
    Log.e(TAG, "insert()", e);
  }
  finally {
    Log.d(TAG, "insert(): rowId=" + rowId);  // kann auch -1 sein
  }
  return rowId;
}

    // select * from  =>  tableColumns=null
    // es sind nur die wichtigsten parameter vorhanden!
  public Cursor getQuery(String table, String[] tableColumns,
            String whereClause, String orderBy)  {
    try{
      SQLiteDatabase db = getWritableDatabase();
      return db.query(table, tableColumns, whereClause,
                null, null, null, orderBy);
    }
    catch(SQLiteException e) {
      Log.e("Database", "getQueryAutomarke", e);
      return null;
    }
  }  // getQuery


}  // DBSOpenHandler
```

15.2.7 DBSQueries

Die Klasse DBSQueries verwaltet alle Strings, inklusive der Konstanten für die Tabellen und Attributnamen.

```
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

import java.util.ArrayList;
import java.util.List;


public class DBSQueries {

  private static final String TAG = DBSQueries.class.getSimpleName();

  private static final String TABLE1 = "STUDENT";
```

```
    // Attribute für die Tabelle Student
  private static final String ID = "_id";
  private static final String NAME = "NAME";
  private static final String MATRNR = "MATRNR";
  private static final String NOTE = "NOTE";

  private static final String DATABASE_NAME = "database1.db";
  private static final int DATABASE_VERSION = 1;

  // verwendet in MainActivity
  public static String getDatabasename() {
    return DATABASE_NAME;
  }

  public static int getDatabaseVersion() {
    return DATABASE_VERSION;
  }


//  für FOREIGN KEY (table2_id) REFERENCES table2(id),
// http://www.sqlite.org/lang_createtable.html

  public static String getCreateTables() {
    String s = "CREATE TABLE "+TABLE1+" ("+
       ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "+
       NAME+" VARCHAR(30) DEFAULT '', "+
       MATRNR + "  INTEGER DEFAULT 0, "+
       NOTE + " VALUE1 DOUBLE DEFAULT 0.0"+
      " );";
    Log.i(TAG, "**********\n"+s);
    return s;
  } // getCreateTables"

    // Parameter ein Student
    // Erstellen der HashKay-Liste: ContentValues
    // dann einfügen und Aufrufen von insertSQL
  public static long insertStudent(Student std) {
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    ContentValues values = new ContentValues(); // à la Hashmap
    values.put(NAME, std.name);
    values.put(MATRNR, std.matrnr);
    values.put(NOTE, std.note);
    return dbshandler.insertSQL(values, TABLE1);
  }

  // hier gibt es den Adapter CURSOR !
public static ArrayList<Student> getStudentItems()    {
  String[] tableColumns = {ID,NAME, MATRNR, NOTE};
  DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
  ArrayList<Student> liste = new ArrayList<Student>();
  Cursor cursor = dbshandler.getQuery(TABLE1, tableColumns, null, null);
  if (cursor==null) {
    return null;
  }
  else {
    cursor.moveToFirst();
    while (!cursor.isAfterLast()) {
      Student std = new Student();
      std.pindex = cursor.getInt(0);
```

```
      std.name = cursor.getString(1);
      std.matrnr = cursor.getInt(2);
      std.note = cursor.getDouble(3);
      liste.add(std);
      cursor.moveToNext();
    }
    cursor.close();
  }
  return liste;
}

  // Rückgabe des Cursors für die ListViews
public static Cursor  getStudentCursor()    {
  String[] tableColumns = {ID,NAME, MATRNR, NOTE};
  DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
  Cursor cursor = dbshandler.getQuery(TABLE1, tableColumns, null, null);
  if (cursor==null) {
    Log.e(TAG,"Der cursor (getStudentCursor) ist null");
    return null;
  }
  else {
    int anz = cursor.getCount();
    Log.e(TAG,"(getStudentCursor)  Anzahl der Zeilen: "+anz);
    return cursor;
  }
}


  private static String quoted(String s) {
    return "'"+s+"'";
  }

  public static void updateSQL(Student std) {
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    SQLiteDatabase db = dbshandler.getWritableDatabase();
    ContentValues cVals = new ContentValues();
    cVals.put(NAME, std.name);
    cVals.put(MATRNR, std.matrnr);
    cVals.put(NOTE, std.note);
    int count = db.update(TABLE1, cVals, PINDEX+" = ?",new
String[]{String.valueOf(std.pindex)});
  }

  public static void deleteSQL(int pindex) {
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    SQLiteDatabase db = dbshandler.getWritableDatabase();
    db.delete(TABLE1, ID+" = ?",new String[]{String.valueOf(pindex)});
  }

}  // DBSQueries
```

Normale Fassung des Updates-Befehls:

```
" SET "+
   NAME+"="+quoted(std.name) + ", "+
   MATRNR + "="+std.matrnr + ", " +
   NOTE + "="+std.note+" " +
  " WHERE "+ PINDEX +"="+std.pindex+";";
```

Am Schluss folgt die Datei MainActivity

15.2.8 Basis.java

Anlegen der Klasse Basis.java und kopieren des Inhalts aus dem Anhang.

15.2.9 MainActivity

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

  private EditText tname = null;
  private EditText tmatrnr = null;
  private Spinner spnote = null;

  private EditText teditor = null;

  private List<Double> notenliste = null;

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

    tname = findViewById(R.id.tname);
    tmatrnr =  findViewById(R.id.tmatrnr);
    spnote =    findViewById(R.id.spnote);
    teditor = findViewById(R.id.teditor);

    notenliste = new ArrayList<Double>();
    notenliste.add(1.0);
    notenliste.add(1.3);
    notenliste.add(1.7);
    notenliste.add(2.0);
    notenliste.add(2.3);
    notenliste.add(2.7);
    notenliste.add(3.0);
    notenliste.add(3.3);
    notenliste.add(3.7);
    notenliste.add(4.0);
    notenliste.add(5.0);

    DBSOpenHandler.createInstance(this,
        DBSQueries.getDatabasename(),
```

```
        DBSQueries.getDatabaseVersion());
  }

    // aus ListViewStudenten
  private Student checkInputs(){
    String dummy;
    int matrnr=0;
    double note=1.0;
    String name = tname.getText().toString();
    if (name.length()==0) {
      Basis.showToast(this,"Der Namen muss mindestens einen Buchstaben");
      return null;
    }
    Integer Int = Basis.getIntegerNumber(tmatrnr.getText().toString());
    if (Int == null) {
      Basis.showToast(this,
          "Die Matrikelnummer muss numerisch sein (ganzzahlig, >0)");
      return null;
    }
    matrnr = Int.intValue();
    if (matrnr<1) {
      Basis.showToast(this,"Die Matrikelnummer muss größer 0 sein!");
      return null;
    }

    int indexNote =  spnote.getSelectedItemPosition();
    note = notenliste.get(indexNote);
    return new Student(name, matrnr, note );
  }  // checkInputs

    // hier wird's spannend
  public void bninsert_Click(View view)   {
    Student std = checkInputs();
    if (std == null) {
      return;
    }
    long rowId = DBSQueries.insertStudent(std);
    Basis.showToast(this,"rowId: "+rowId);
  }

    // Anzeigen der aktuellen Einträge in der Tabelle Student
  public void bnshowStd1_Click(View view) {
    ArrayList<Student> liste = DBSQueries.getStudentItems();
    StringBuilder sb = new StringBuilder();
    for (Student std : liste) {
    sb.append(std);
    sb.append("\n");
  }
  teditor.setText(sb.toString());
}

   // hier wird eine neue Seite aufgerufen
  public void bnshowStd2_Click(View view) {
    Intent intent = new Intent(this,  ShowStudentsActivity1.class);
    startActivity(intent);
  }

   // hier wird eine neue Seite aufgerufen
  public void bnshowStd3_Click(View view) {
    Intent intent = new Intent(this,  ShowStudentsActivity2.class);
```

```
    startActivity(intent);
  }


}
```

Die nächsten beiden Kapitel zeigen jeweils einen View, in dem eine ListView einen DatenbankCursor entgegennimmt und damit die Studierenden darstellt.

15.2.10 ShowStudentsActivity1

Anlegen einer neuen leeren Activity
Name: ShowStudentsActivity1

15.2.10.1.1 activity_show_students1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">

  <TextView
    android:id="@+id/tvshowstdtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="1dp"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/tvshowstdtitle"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <EditText
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="6dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"

    android:background="@color/colorbackgroundlistview"
    android:gravity="start|top"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/tvshowstdtitle"
    app:layout_constraintTop_toBottomOf="@+id/tvshowstdtitle" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

15.2.10.2 row_layout.xml

Anlegen der Datei „row_layout.xml“, die den Inhalt einer Zeile anzeigt.



Abbildung 185 row_layout für die Datenbank-Anzeige

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="?android:attr/listPreferredItemHeight"
    android:padding="1dp"
    >

  <TextView
    android:id="@+id/tvpindex"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="4dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:text="0"
    android:textAppearance="?android:attr/textAppearanceMedium"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="5dp"
    android:text="tvname"
    android:textAppearance="?android:attr/textAppearanceMedium"
    app:layout_constraintStart_toEndOf="@+id/tvpindex"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvmatrnr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:text="@string/tvmatrnr"
    android:textAppearance="?android:attr/textAppearanceSmall"
    app:layout_constraintStart_toStartOf="@+id/tvname"
```

```
    app:layout_constraintTop_toBottomOf="@+id/tvname" />

  <TextView
    android:id="@+id/tvnote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="5dp"
    android:text="@string/tvnote"
    android:textAppearance="?android:attr/textAppearanceSmall"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

15.2.10.2.1 ShowStudentsActivity1.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.*;
import java.util.ArrayList;

public class ShowStudentsActivity1 extends AppCompatActivity {

  private ListView listview = null;
  private SimpleCursorAdapter dataAdapter;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_show_students1);
    listview = findViewById(R.id.listview);
    insertAdapter();
  }

  private void insertAdapter() {
    // The desired columns to be bound
    String[] columns = new String[] {
      DBSQueries.ID,
      DBSQueries.NAME,
      DBSQueries.MATRNR,
      DBSQueries.NOTE
    };

      // the XML defined views which the data will be bound to
    int[] to = new int[] {
      R.id.tvpindex,
      R.id.tvname,
      R.id.tvmatrnr,
      R.id.tvnote,
    };
```

```
    Cursor cursor = DBSQueries.getStudentCursor();
      // create the adapter using the cursor pointing to the desired data
      //as well as the layout information
    dataAdapter = new SimpleCursorAdapter(
      this, R.layout.row_layout,
      cursor,
      columns,
      to,
      0);

    listview.setAdapter(dataAdapter);

//   SimpleCursorAdapter(Context context, int layout, Cursor c,
//       String[] from, int[] to, int flags)

// flags   int:  CursorAdapter.FLAG_REGISTER_CONTENT_OBSERVER
//   FLAG_AUTO_REQUER   FLAG_REGISTER_CONTENT_OBSERVER;

  }  // insertAdapter

}
```

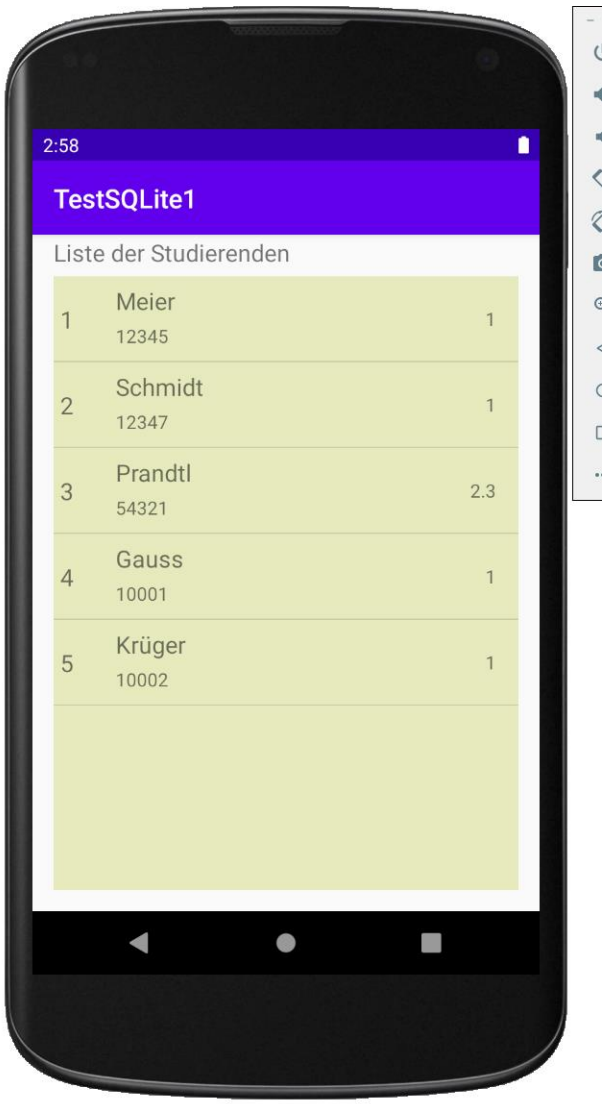


Abbildung 186 LiustView mit einem SimpleCursorAdapter

15.2.11 ShowStudentsActivity2

Dieses Kapitel legt einen separaten Cursoradapter an.

Anlegen einer neuen leeren Activity
Name: ShowStudentsActivity2

15.2.11.1.1 activity_show_students2.xml

Anlegen der Datei „row_layout.xml“, die den Inhalt einer Zeile anzeigt.



Abbildung 187 row_layout für die Datenbank-Anzeige

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="?android:attr/listPreferredItemHeight"
    android:padding="1dp"
    >

  <TextView
    android:id="@+id/tvpindex"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="4dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:text="0"
    android:textAppearance="?android:attr/textAppearanceMedium"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/tvname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="5dp"
    android:text="tvname"
    android:textAppearance="?android:attr/textAppearanceMedium"
    app:layout_constraintStart_toEndOf="@+id/tvpindex"
    app:layout_constraintTop_toTopOf="parent" />
```

```
  <TextView
    android:id="@+id/tvmatrnr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:text="@string/tvmatrnr"
    android:textAppearance="?android:attr/textAppearanceSmall"
    app:layout_constraintStart_toStartOf="@+id/tvname"
    app:layout_constraintTop_toBottomOf="@+id/tvname" />

  <TextView
    android:id="@+id/tvnote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="5dp"
    android:text="@string/tvnote"
    android:textAppearance="?android:attr/textAppearanceSmall"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

15.2.11.2 MyCursorAdapter

Diese Klasse ist ähnlich der BaseAdapter der ListView. Es gibt aber keine interne Holder-Klasse und die Methode „getView“ wird ersetzt durch die Methoden „newView“ und „bindView“.

```
package com.example.testsqlite1;

import android.content.Context;
import android.database.Cursor;
import android.view.*;
import android.widget.*;

public class MyCursorAdapter extends CursorAdapter {

  public MyCursorAdapter(Context context, Cursor cursor) {
    super(context, cursor, 0);
  }

    // newView baut die Zeile aus dem layout row_layout
    // es werden keine Daten hinzugefügt
  @Override
  public View newView(Context context, Cursor cursor, ViewGroup parent) {
    return LayoutInflater.from(context).inflate(
                            R.layout.row_layout, parent, false);
  }

    // Prinzip à la ListViewStudenten, ohne Holder
    // such as setting the text on a TextView.
  @Override
  public void bindView(View view, Context context, Cursor cursor) {
```

```
    // Find fields to populate in inflated template
    TextView tvpindex = (TextView) view.findViewById(R.id.tvpindex);
    TextView tvname = (TextView) view.findViewById(R.id.tvname);
    TextView tvmatrnr = (TextView) view.findViewById(R.id.tvmatrnr);
    TextView tvnote = (TextView) view.findViewById(R.id.tvnote);

      // Holen der Daten aus den Cursor
    int id = cursor.getInt(cursor.getColumnIndexOrThrow(DBSQueries.ID));
    String name= cursor.getString(cursor.getColumnIndexOrThrow(
                                                    DBSQueries.NAME));
    int matrnr = cursor.getInt(cursor.getColumnIndexOrThrow(
                                                    DBSQueries.MATRNR));
    double note= cursor.getDouble(cursor.getColumnIndexOrThrow(
                                                    DBSQueries.NOTE));
    // setzen der UI-Elemente aus derm Datensatz
    tvpindex.setText(Integer.toString(id));
    tvname.setText(name);
    tvmatrnr.setText(Integer.toString(matrnr));
    tvnote.setText(Double.toString(note));
  }
}
```

15.2.11.2.1 ShowStudentsActivity2.java

```
package com.example.testsqlite1;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.*;

import java.util.ArrayList;

public class ShowStudentsActivity2 extends AppCompatActivity {

    private ListView listview = null;
    private MyCursorAdapter dataAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_show_students1);

        listview = findViewById(R.id.listview);
        insertAdapter();
    }


    private void insertAdapter() {
        Cursor cursor = DBSQueries.getStudentCursor();
        dataAdapter = new MyCursorAdapter(this, cursor);
        listview.setAdapter(dataAdapter);

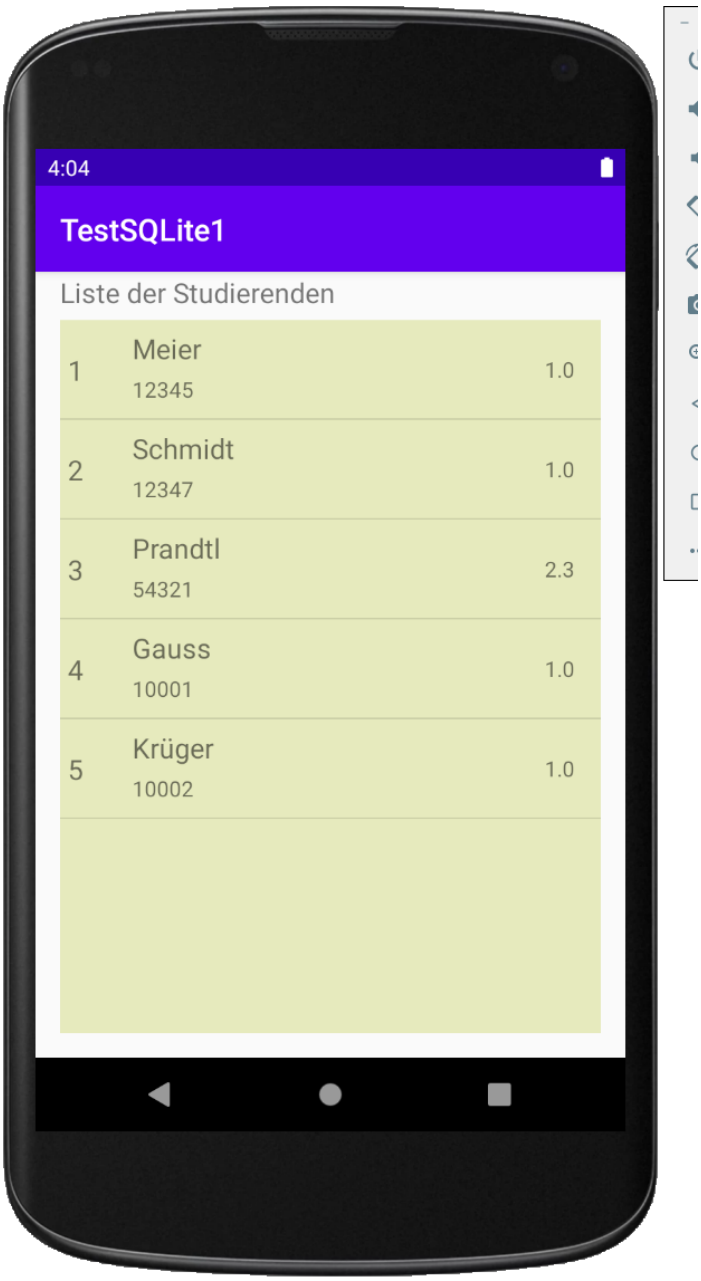
    }  // insertAdapter

}
```

Beide Lösungen sind weitgehend identisch. Die zweite Lösung mit der externen Adapter-Klasse zeigt in der Klasse „ShowStudentsActivity2“ einen übersichtlichen Code.

Wenn man Löschen und Einfügen will, muss man die zweite Variante nehmen.

Lösung:


4:04
TestSQLite1
Liste der Studierenden
1
Meier
12345
1.0
2
Schmidt
12347
1.0
3
Prandtl
54321
2.3
4
Gauss
10001
1.0
5
Krüger
10002
1.0

16 ListView und die Studentendatenbank

Dieses Kapitel zeigt die Benutzung der SQLite-Datenbank mit einer ListrView. Es basiert auf den Quellcode der ListView mit DataInput und DataOutputStream auf.

16.1 Einfügen der Klasse DBSOpenHandler

Diese Klasse ist komplett unabhängig von den verwendeten Tabellen.
Ist als Singleton implementiert.

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.*;
import android.util.Log;


public class DBSOpenHandler extends SQLiteOpenHelper {
  private static final String TAG = DBSOpenHandler.class.getSimpleName();

  private static DBSOpenHandler instance = null;

  // diese Methode muss in MainActivity aufgerufen werden
  public static void createInstance(Context context, String databasename,
    int databaseVersion) {
    if (instance==null) {
      instance = new DBSOpenHandler(context, databasename,
                      databaseVersion);
    }
  }

  public static DBSOpenHandler getInstance() {
    return instance;
  }

  private DBSOpenHandler (Context context, String databasename,
          int databaseVersion) {
    super(context, databasename, null, databaseVersion);
  }


  @Override
  public void onCreate(SQLiteDatabase db) {
    String sql= DBSQueries.getCreateTables();
    Log.i(TAG, "********** String  getCreateTables");
    try {
      db.execSQL(sql);
      Log.i(TAG, "********** database are created (1)");
    } catch (SQLiteException e) {
      Log.i(TAG, "**********     no database are created", e);
    } finally {
      Log.i(TAG, "********** in finally onCreate");
    }
  }
```

```
  @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
    Log.i(TAG, "Upgrade of the database from version "
               + oldVersion + " to "
               + newVersion + "; all datas are deleted");
    String sql= DBSQueries.getCreateTables();
    db.execSQL(sql);
    onCreate(db);
  }

  // Bitte beachten, die rowId wird zurückgegeben
  public long insertSQL(ContentValues values, String dbtable) {
    long rowId = -1;
    try {
      SQLiteDatabase db = getWritableDatabase();
      Log.i(TAG, "Path: " + db.getPath());
      rowId = db.insert(dbtable, null, values);
    } catch (SQLiteException e) {
      Log.e(TAG, "insert()", e);
    } finally {
      Log.d(TAG, "insert(): rowId=" + rowId);
    }
    return rowId;
}

  public Cursor getQuery(String table, String[] tableColumns,
        String whereClause, String orderBy)  {
    try{
      SQLiteDatabase db = getWritableDatabase();
      return db.query(table, tableColumns, whereClause, null, null, null,
             orderBy);
    }
    catch(SQLiteException e)    {
      Log.e("Database", "getQueryAutomarke", e);
      return null;
    }
  }


}  // DBSOpenHandler
```

16.2 Klasse Student

Änderungen:

- Hier wird das neue Attribut „pindex“ eingetragen.
- Für den Adapter benötigt man eine Konvertierung in eine Instanz „Student“.

Attribute:

```
public static final String  STD = "STUDENT";  // serialize

public int  pindex=-1;
public String name="-";
public int matrnr=1;
public int semester=1;
```

```
public double note = 1.0;
```

Methoden:

```
public Student(Cursor cursor) {
  pindex = cursor.getInt(cursor.getColumnIndexOrThrow((DBSQueries.ID)));
  semester = cursor.getInt(cursor.getColumnIndexOrThrow(
        DBSQueries.SEMESTER));
  name  = cursor.getString(cursor.getColumnIndexOrThrow(DBSQueries.NAME));
   matrnr = cursor.getInt(cursor.getColumnIndexOrThrow(
        DBSQueries.MATRNR));
  note = cursor.getDouble(cursor.getColumnIndexOrThrow(DBSQueries.NOTE));
}
```

16.3 DBSQueries

In dieser Datei stehen alle SQL-Befehle und es stehen Methoden zu Verfügung, die die UI-Elemente mit der Datenbank koppelt.

Quellcode:

```
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

import java.util.ArrayList;
import java.util.List;


public class DBSQueries {

  private static final String TAG = DBSQueries.class.getSimpleName();

  private static final String TABLE1 = "STUDENT";

    // Namen der Spalten der tabelle Student
  public static final String ID = "_id";
  public static final String SEMESTER = "SEMESTER";
  public static final String NAME = "NAME";
  public static final String MATRNR = "MATRNR";
  public static final String NOTE = "NOTE";

  private static final String DATABASE_NAME = "database1.db";
  private static final int DATABASE_VERSION = 1;


    Wird von MainActivity aufgerufen
  public static String getDatabasename() {
    return DATABASE_NAME;
  }

    Wird von MainActivity aufgerufen
  public static int getDatabaseVersion() {
    return DATABASE_VERSION;
  }
```

```
    // Struktur der Datenbank, hier nur eine Tabelle
  public static String getCreateTables() {
    String s = "CREATE TABLE  if not exists "+TABLE1+" ("+
        ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "+
        NAME+" VARCHAR(30) DEFAULT '', "+
        MATRNR + "  INTEGER DEFAULT 0, "+
        SEMESTER + "  INTEGER DEFAULT 0, "+
        NOTE + " DOUBLE DEFAULT 0.0"+
    " );";
    Log.i(TAG, "**********\n"+s);
    return s;
} // getCreateTables"



  public static long insertStudent(Student std) {
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    ContentValues values = new ContentValues(); // à la Hashmap
    values.put(NAME, std.name);
    values.put(MATRNR, std.matrnr);
    values.put(SEMESTER, std.semester);
    values.put(NOTE, std.note);
    return dbshandler.insertSQL(values, TABLE1);
  }


    // die se Methode holt ALLE Studenten, aber keine ArrayList<Student>
  public static Cursor  getStudentCursor()    {
    String[] tableColumns = {ID,NAME, MATRNR, SEMESTER, NOTE};
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    Cursor cursor = dbshandler.getQuery(TABLE1, tableColumns, null, null);
    if (cursor==null) {
      Log.e(TAG,"Der cursor (getStudentCursor) ist null");
      return null;
    }
    else {
      int anz = cursor.getCount();
      Log.e(TAG,"(getStudentCursor)  Anzahl der Zeilen: "+anz);
      return cursor;
    }
  }


  public static int updateStudent(Student std) {
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    SQLiteDatabase db = dbshandler.getWritableDatabase();
    ContentValues cVals = new ContentValues();
    cVals.put(NAME, std.name);
    cVals.put(MATRNR, std.matrnr);
    cVals.put(SEMESTER, std.semester);
    cVals.put(NOTE, std.note);
    int count = db.update(TABLE1, cVals, ID+" = ?",new
                          String[]{String.valueOf(std.pindex)});
    Log.i(TAG,"Update: Anzahl der betroffenen Datensätze: "+count+
        "     pindex: "+std.pindex);
    return count;
  }


  public static int deleteStudent(int pindex) {
```

```
    DBSOpenHandler dbshandler = DBSOpenHandler.getInstance();
    SQLiteDatabase db = dbshandler.getWritableDatabase();
    return db.delete(TABLE1, ID+" = ?",
                     new String[]{String.valueOf(pindex)});
  }



    // DummyDaten, werden durch einen OptionsMenu-Eintrag eingefügt
  public static ArrayList<Student>  getInitListe() {
    ArrayList<Student> liste = new ArrayList<Student>();
    Log.i(TAG, "#####################################  in initListe ");
    liste.clear();
    liste.add( new Student(1,"Meier, Antonia", 12345, 8, 1.0)  );
    liste.add( new Student(3,"Schulze, Susanne", 12365, 6, 2.3)  );
    liste.add( new Student(4,"Brandt, Hugo", 22345, 6, 3.3)  );
    liste.add( new Student(6,"Binder, Gabi", 22365, 6, 4.0)  );
    liste.add( new Student(8,"Märtens, Friedrich", 22845, 5, 1.7)  );
    liste.add( new Student(9,"Müller, Andreas", 22349, 5, 1.3)  );
    liste.add( new Student(10,",Baumann, Herbert", 20445, 7, 2.7)  );
    liste.add( new Student(11,"Künneth, Thomas", 324455, 4, 3.3)  );
    liste.add( new Student(13,"Unbereit, Peter", 324355, 7, 2.7)  );
    liste.add( new Student(15,"Meyer, Hubert ", 324355, 3, 2.0)  );
    liste.add( new Student(18,"Altmeier, Jutta", 324355, 3, 3.3)  );
    liste.add( new Student(19,"Gisin, Nicoletta", 33445, 2, 2.7)  );
    liste.add( new Student(22,"Telemann, Dieter", 34645, 2, 5.0)  );
    liste.add( new Student(23,"Wagner, Hans", 38345, 2, 4.0)  );
    liste.add( new Student(24,"Wolff, Bernd", 39345, 1, 3.7)  );
    liste.add( new Student(25,"Prandtl, Daria", 40345, 1, 1.3)  );
    liste.add( new Student(26,"Burton, Michael", 41445, 1, 3.0)  );
    liste.add( new Student(27,"Fehr, Hauke", 42445, 1, 2.0)  );
    liste.add( new Student(33,"Klenke, Mario", 44445, 1, 3.7)  );
    Log.i(TAG,"* Modul Studenteliste,anzahl Studierende: "+liste.size());
    return liste;
  }

}  // DBSQueries
```

16.4 MyCursorAdapter

Der CursorAdapter ersetzt den StudentenAdapter: Bitte löschen
Die Studenliste wird nicht mehr benötigt: Bitte löschen

In der Klasse werden die Daten aus der Datenbank geholt übder dem Konstruktor gesetzt. Es gibt die Methode, mit der das row_layout zusammengesetzt wird und eine Methode, die die aktuelle „Zeile“ erstellt. Hier wird auch das „Tag“ gesetzt, damit die onClick-Event die Daten erhalten.

```
import android.content.Context;
import android.database.Cursor;
import android.view.*;
import android.widget.*;

public class MyCursorAdapter extends CursorAdapter {
```

```
  public MyCursorAdapter(Context context, Cursor cursor) {
    super(context, cursor, 0);
  }


    // Aufbau der Struktur der Zeile
  @Override
  public View newView(Context context, Cursor cursor, ViewGroup parent) {
    return LayoutInflater.from(context).inflate(R.layout.row_layout,
             parent, false);
  }

    // Erstellen der Zeile mit Inhalt
  @Override
  public void bindView(View view, Context context, Cursor cursor) {
    UI-Element-Referenzen holen
    ImageView ivsemester = view.findViewById(R.id.ivsemester);
    TextView tvname = view.findViewById(R.id.tvname);
    TextView tvmatrnr = view.findViewById(R.id.tvmatrnr);
    TextView tvnote = view.findViewById(R.id.tvnote);

    // Extract properties from cursor
    int id = cursor.getInt(cursor.getColumnIndexOrThrow(DBSQueries.ID));
    int semester = cursor.getInt(cursor.getColumnIndexOrThrow(
                                              DBSQueries.SEMESTER));
    String name= cursor.getString(cursor.getColumnIndexOrThrow(
                                              DBSQueries.NAME));
    int matrnr = cursor.getInt(cursor.getColumnIndexOrThrow(
                                              DBSQueries.MATRNR));
    double note= cursor.getDouble(cursor.getColumnIndexOrThrow(
                                              DBSQueries.NOTE));

      // beim Anklicken der Liste wird dieser cursor, besser std der
      // Activity EditNewDelete übergeben !!!
    Student std = new Student(cursor);
    view.setTag(std);  // WICHTIG !!!!!!!!!!

    // Populate fields with extracted properties
    ivsemester.setImageResource(std.getIdForDrawable());
    tvname.setText(name);
    tvmatrnr.setText(Integer.toString(matrnr));
    tvnote.setText(Double.toString(note));
  }
}
```

16.5 MainActivity

In der MainActivity müssen folgende Änderungen durchgeführt werden:

```
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
```

```
import android.util.Log;
import android.view.*;
import android.widget.*;

import android.app.ActionBar;

import java.util.ArrayList;


public class MainActivity extends AppCompatActivity {

  private static final String TAG = MainActivity.class.getSimpleName();

  private MyCursorAdapter adapter = null;
  private ListView listview = null;

  private MenuItem menu_insertdummydatas = null;
  private Cursor anfangscursor=null;

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

    // Erzeugen des Singleton-Objektes der Datenbank
    DBSOpenHandler.createInstance(this,DBSQueries.getDatabasename(),
              DBSQueries.getDatabaseVersion());

    // Liste der "Zeilen"
    this.anfangscursor = DBSQueries.getStudentCursor();

    listview = findViewById(R.id.listview);
    adapter = new MyCursorAdapter(this, anfangscursor);
    listview.setAdapter(adapter);

    // long und short clicks NIEMALS per XML-Code
    // als erstes den long-click, dann  short-click

    listview.setLongClickable(true);
    listview.setOnItemLongClickListener(new
                        AdapterView.OnItemLongClickListener() {
      @Override
      public boolean onItemLongClick(AdapterView<?> parent, View view,
                                     int position, long id) {
        Student std = (Student)view.getTag();
        showEditDeleteNew(EditDeleteNewActivity.ACTION_DELETE, std);
        return true;
      }
    }
  );


    listview.setOnItemClickListener(new AdapterView.OnItemClickListener(){
      @Override
      public void onItemClick(AdapterView<?> parent, View view,
                              int position, long id) {
        Student std = (Student)view.getTag();
        Log.i("******************* clicked","position: "+position);
        showEditDeleteNew(EditDeleteNewActivity.ACTION_EDIT, std);
```

```
      }
    });

  }


    // import android.app.ActionBar
  @Override
  protected void onStart() {
    super.onStart();
    ActionBar ab = getActionBar();
    if (ab!=null) {
      ab.setDisplayHomeAsUpEnabled(true);
    }
  }

    // menu_insertdummydatas ist global deklariert
    // wird ausgeschaltet, wenn es daten schon gibt!
    // diese Methode wird nach dem Anklicken der drei Punkte aktiviert
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu,menu);
    menu_insertdummydatas = menu.findItem(R.id.menu_adddatas);
    menu_insertdummydatas.setEnabled(anfangscursor.getCount()<=0);
    return super.onCreateOptionsMenu(menu);
  }


  @Override
  public boolean onOptionsItemSelected(@NonNull MenuItem mnItem){
    switch (mnItem.getItemId()) {
      case R.id.menu_new_item:
        Student stdNew =  new Student();
        showEditDeleteNew(EditDeleteNewActivity.ACTION_NEW, stdNew);
        break;
      case R.id.menu_adddatas:
        if (anfangscursor.getCount()==0) {
          ArrayList<Student> liste = DBSQueries.getInitListe();
          for (Student std : liste) {
            DBSQueries.insertStudent(std);
          }
          anfangscursor = DBSQueries.getStudentCursor();
          adapter = new MyCursorAdapter(this, anfangscursor);
          listview.setAdapter(adapter);
          menu_insertdummydatas.setEnabled(anfangscursor.getCount()<=0);
        }
        break;
    }
    return true;
  }

  private void showEditDeleteNew(int action, Student std) {
    // Erzeugen der zweiten View, Parameter erstellen  und starten
    Intent intent = new Intent(this, EditDeleteNewActivity.class);
    intent.putExtra(EditDeleteNewActivity.ACTION, action);
    intent.putExtra(Student.STD, std);  // Serialize!!

    startActivityForResult(intent,1);
  }
```

```
  @Override
  protected void onActivityResult(int requestCode, int resultCode,
                                  @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1) {  // ist alles mit der Übertragung in Ordnung?
      switch (resultCode) {
        case Activity.RESULT_OK:
          // die ListView muss neu geladen werden
          Cursor cursor = DBSQueries.getStudentCursor();
          adapter.changeCursor(cursor);
        break;
        case Activity.RESULT_CANCELED:
          Basis.showToast(this,"cancel gedrückt");
          break;
        }
    }  // if
    else {
      Basis.showToast(this,"falscher resultCode:" + resultCode);
    }
  }
}
```

16.6 EditDeleteNewActivity

Hier muss nur das Speichern der daten geändert werden:

- update
- insert
- delete

```
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;


import android.view.View;
import android.widget.*;
import java.text.DecimalFormat;
import java.util.*;

public class EditDeleteNewActivity extends AppCompatActivity {

  private static final String TAG = MainActivity.class.getSimpleName();
  public static final String ACTION = "ACTION";
  public static  final  int ACTION_EDIT = 1;
  public static  final  int ACTION_DELETE = 2;
  public static  final  int ACTION_NEW = 3;
  private int action = ACTION_EDIT;

  private TextView tvtitle=null;
  private EditText tname = null;
```

```
  private EditText tmatrnr = null;
  private Spinner spsemester = null;
  private Spinner spnote = null;
  private Button bnok = null;
  private Button bnesc = null;


  private int pindex=-1;
  private DecimalFormat df = new DecimalFormat("0.0");
  private List<Double> notenliste = null;


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


    Intent intent = super.getIntent();
    if (intent==null) {
      tvtitle.setText("intent ist null");
    }
    else{
      action = intent.getIntExtra(EditDeleteNewActivity.ACTION, -1);
      if (action==-1) {
        Basis.showToast(this,"Fehlerhafte Action");
        return;
      }

      tvtitle = findViewById(R.id.tvtitle);
      tname = findViewById(R.id.tname);
      tmatrnr =  findViewById(R.id.tmatrnr);
      spsemester = findViewById(R.id.spsemester);
      spnote = findViewById(R.id.spnote);
      bnok =  findViewById(R.id.bnok);

      switch (action) {
        case EditDeleteNewActivity.ACTION_EDIT:
          tvtitle.setText(getString(R.string.actiontitle_edit));
          bnok.setText(getString(R.string.bnsavechanges));
          break;
        case EditDeleteNewActivity.ACTION_DELETE:
          tvtitle.setText(getString(R.string.actiontitle_delete));
          bnok.setText(getString(R.string.bndelete));
          break;
        case EditDeleteNewActivity.ACTION_NEW:
          tvtitle.setText(getString(R.string.actiontitle_new));
          bnok.setText(getString(R.string.bnnew));
          break;
        default:
          Basis.showToast(this,"Fehlerhafte Action: "+action);
          return;
      }




      notenliste = new ArrayList<Double>();
      notenliste.add(1.0);
      notenliste.add(1.3);
```

```
      notenliste.add(1.7);
      notenliste.add(2.0);
      notenliste.add(2.3);
      notenliste.add(2.7);
      notenliste.add(3.0);
      notenliste.add(3.3);
      notenliste.add(3.7);
      notenliste.add(4.0);
      notenliste.add(5.0);


      Student std = (Student) intent.getSerializableExtra(Student.STD);
      this.pindex = std.pindex;  // für delete und update


      tname.setText( std.name );
      tmatrnr.setText( Integer.toString(std.matrnr) );
      spsemester.setSelection(std.semester-1);


      ArrayAdapter<Double> notenAdapter = new ArrayAdapter<Double>(this,
                  android.R.layout.simple_spinner_item, notenliste);
      spnote.setAdapter(notenAdapter);

      for (int i=0; i<notenliste.size(); i++) {
        if (std.note==notenliste.get(i)) {  // eventuell eps. Bereich
          spnote.setSelection(i);
          break;
        }
      }

    }

  }  // onCreate


  private Student checkInputs(){
    String dummy;
    int matrnr=0;
    int semester=0;
    double note=1.0;
    String name = tname.getText().toString();
    if (name.length()==0) {
      Basis.showToast(this,"Der Namen muss mindestens einen Buchstaben");
      return null;
    }
    Integer Int = Basis.getIntegerNumber(tmatrnr.getText().toString());
    if (Int == null) {
      Basis.showToast(this,
        "Die Matrikelnummer muss numerisch sein (ganzzahlig, >0)");
      return null;
    }
    matrnr = Int.intValue();
    if (matrnr<1) {
      Basis.showToast(this,"Die Matrikelnummer muss größer 0 sein!");
      return null;
    }

    semester =  spsemester.getSelectedItemPosition()+1;
    int indexNote =  spnote.getSelectedItemPosition();
```

```
    note = notenliste.get(indexNote);
    return new Student(0,name, matrnr, semester, note );
  }  // checkInputs


  public void bnok_Click(View view) {
    switch (action) {
      case EditDeleteNewActivity.ACTION_EDIT: {
        Student std = checkInputs();
        if (std == null) {
          return;
        }
        std.pindex = this.pindex;  // aus CheckInputs ist pindex=0 !!!
        if (DBSQueries.updateStudent(std)==0) {
          Basis.showToast(this,
              "Der Datensatz konnte nicht geändert werden");
        }
        else {
          Basis.showToast(this,"Der Datensatz wurde geändert.");
        }

      }
      break;

      case EditDeleteNewActivity.ACTION_DELETE:
        if (DBSQueries.deleteStudent(this.pindex)==0) {
          Basis.showToast(this,
              "Der Datensatz konnte nicht gelöscht werden");
        }
        else {
          Basis.showToast(this,"Der Datensatz wurde gelöscht.");
        }

        break;

      case EditDeleteNewActivity.ACTION_NEW: {
        Student std = checkInputs();
        if (std == null) {
          return;
        }
        if (DBSQueries.insertStudent(std)==0) {
          Basis.showToast(this,
                    "Der Datensatz konnte nicht eingefügt werden.");
        }
        else {
          Basis.showToast(this,"Der Datensatz wurde eingefügt.");
        }
      }
      break;

      default:
        Basis.showToast(this,"Fehlerhafte Action: "+action);
        return;
    }
    Intent returnIntent = new Intent();
    setResult(Activity.RESULT_OK, returnIntent);
    finish();
  }
```

```
  public void bnesc_Click(View view) {
    Intent returnIntent = new Intent();
    setResult(Activity.RESULT_CANCELED, returnIntent);
    finish();
  }

}
```

17 Sensoren

Auch mit Android Studio kann man auf die internen Sensoren zugreifen.
Benötigt werden:

- Ein Sensormanager
- Ein Sensor mit einem bestimmten Typ
- Ein EventListener
- Eine Callback-Methode

Übergeben wird ein float-Array von einem bis 4 Elementen.

17.1 Sensor-Typen

Folgende Typen gibt es. Es sind aber nicht alle eingebaut.

- Beschleunigung (Accelerator)
- Temperatur
- Gravity (Schwerkraft)
- Gyroskope (Lade, zum Beispiel für Spiele)
- Herzschlag (Heart Beat)
- Herzschlagrate (Heart Beat Rate)
- Licht (Light)
- Linear Beschleunigung (Linear Accelerator)
- Magnetfeld (für Kompass)
- Luftdruck (Pressure)
- Schrittzähler (Step Counter, Step Detector)
- Bewegung (Motion)
- Luftfeuchtigkeit (Humidity)

17.2 Allgemeiner Aufbau

```
SensorManager sensMan = getSystemService(SensorManager.class);

SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_???) {
    }
  }

  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};

Sensor s = sensorManager.getDefaultSensor(Sensor.TYPE_????);
sensMan.registerListener(sensListener,s,SensorManager.SENSOR_DELAY_NORMAL);
```

Als erstes holt man eine Referenz des Sensormanager (sensMan). Nun erstellt man entweder eine Listener, à la obigen QAuellcode oder man implementiert den EventListener in der Klasse. Der vorletzte Schritt erstellt die Sensoren und registriert diese dann im Listener.

17.3 Beispiele der Sensoren

17.3.1 Accelerator (Beschleunigung)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor accel;
private float[] lastaccelField = new float[3];
private float lastaccel = 0;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
      lastaccelField = event.values.clone();
      float x=event.values[0];
      float y=event.values[1];
      float z=event.values[2];
      lastaccel = Math.sqrt(x*x + y*y +z*z);
      sb.append("Accel\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append(Float.toString(lastaccel));
      sb.append("\n");
      sb.append("\n");
      teditor.setText(   sb.toString());
    } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
if(accel == null) {
  teditor.setText(  "accel sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, accel,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.2 Temperatur

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor temp;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor. TYPE_AMBIENT_TEMPERATURE) {
      sb.append("TYPE_AMBIENT_TEMPERATURE\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor.setText(   sb.toString());   } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


temp = sensorManager.getDefaultSensor(Sensor.TYPE_AMBIENT_TEMPERATURE);
if(temp == null) {
  teditor.setText("temp sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, temp,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.3 Gravity (Schwerkraft)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor gravy;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
```

```
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor. TYPE_GRAVITY) {
      sb.append("Gravity\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor.setText(   sb.toString());
    } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


gravy = sensorManager.getDefaultSensor(Sensor. TYPE_GRAVITY);
if(gravy == null) {
  teditor.setText("gravy sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, gravy,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.4 Gyroskope (Lade, zum Beispiel für Spiele)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor gyro;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_GYROSCOPE) {
      sb.append("Gyroskop\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append(Float.toString(lastaccel));
      sb.append("\n");
      sb.append("\n");
      teditor.setText(   sb.toString());
    } // if
  }  // onSensorChanged
```

```
  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


gyro = sensorManager.getDefaultSensor(Sensor. TYPE_GYROSCOPE);
if(gyro == null) {
  teditor.setText(" gyro sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, gyro,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.5 Magnetfeld (für Kompass)

Beim Magnetfeld muss man zwei Sensoren betrachten:

- Beschleunigung
- Magnetfeld

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor accel;
private Sensor magneto;
private float[] lastaccelField = new float[3];
private float lastaccel = 0;
private float[]lastmagnetoField = new float[3];
private float lastmagneto = 0;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
      lastaccelField = event.values.clone();
      float x=event.values[0];
      float y=event.values[1];
      float z=event.values[2];
      lastaccel = Math.sqrt(x*x + y*y +z*z);
      sb.append("Accel\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append(Float.toString(lastaccel));
      sb.append("\n");
      sb.append("\n");
```

```
      teditor.setText(   sb.toString());
    } // if accel

    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
      lastmagnetoField = event.values.clone();
      float[] mR = new float[9];
      float[] mOrientation = new float[3];
      SensorManager.getRotationMatrix(mR, null, lastaccelField,
          lastmagnetoField);
      SensorManager.getOrientation(mR, mOrientation);
      float azimuthRadians = mOrientation[0];  // Bogemmaß
      float azimuthGrad = Math.toDegrees(azimuthInRadians);
      // kann negativ sein
      azimuthGrad = (azimuthGrad+360)%360;

      sb.append("Magnet\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\nWinkel: "+Float.toString(azimuthInDegress));
      sb.append("\n");
      teditor.setText(   sb.toString());
    }

  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


magneto = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
if(magneto == null) {
  teditor.setText(  "magneto sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, magneto,
      SensorManager.SENSOR_DELAY_NORMAL);
}

accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
if(accel == null) {
  teditor.setText("accel sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, accel,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.6 Luftdruck (Pressure)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor pressure;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor. TYPE_PRESSURE) {
      sb.append("Pressure\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor.setText(   sb.toString());
    } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


pressure = sensorManager.getDefaultSensor(Sensor. TYPE_PRESSURE);
if(pressure == null) {
  teditor.setText(" pressure sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, pressure,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.7 Luftfeuchtigkeit (Humidity)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor humidity;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor. TYPE_RELATIVE_HUMIDITY) {
      sb.append("Huminity\n");
```

```
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor.setText(   sb.toString());
    } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};


humidity = sensorManager.getDefaultSensor(Sensor. TYPE_RELATIVE_HUMIDITY);
if(humidity == null) {
  teditor.setText(" humidity sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, humidity,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.3.8 Licht (Light)

Globale Variablen:

```
private SensorManager sensorManager;
private Sensor light;
```

onCreate:

```
sensorManager = getSystemService(SensorManager.class);

// Create listener
SensorEventListener sensorListener = new SensorEventListener() {
  @Override
  public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_LIGHT) {
      sb.append("Light\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor.setText(sb.toString());
    } // if
  }  // onSensorChanged


  @Override
  public void onAccuracyChanged(Sensor sensor, int i) {
  }
};
```

```
light = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
if(light == null) {
  teditor.setText("light sensor not available.");
}
else {
  sensorManager.registerListener(sensorListener, light,
    SensorManager.SENSOR_DELAY_NORMAL);
}
```

17.4 Alle Sensoren

Dieses Kapitel beschreibt kurz ein Projekt, mit dem alle verfügbaren Sensoren getestet werden können.

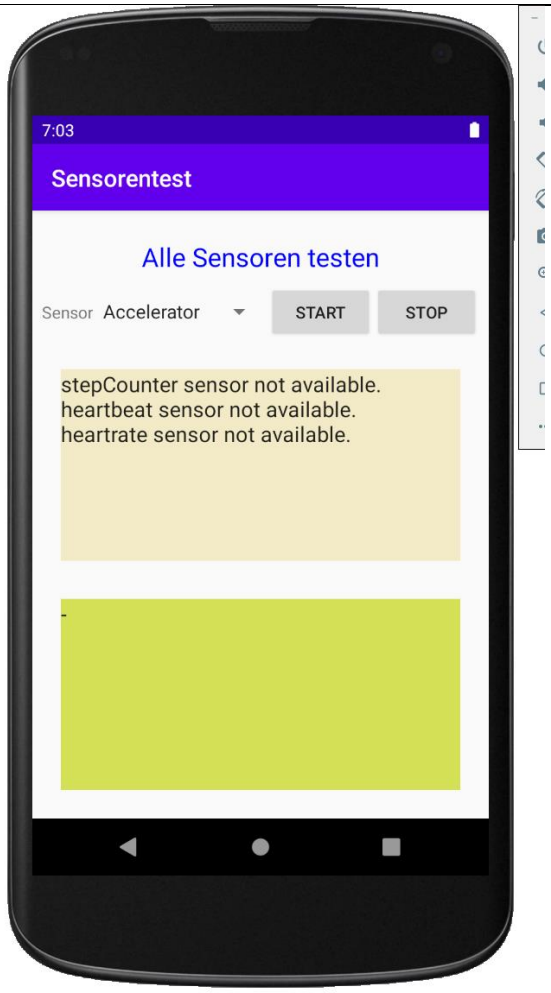


Abbildung 188 Start des Projektes "Alle Sensoren"

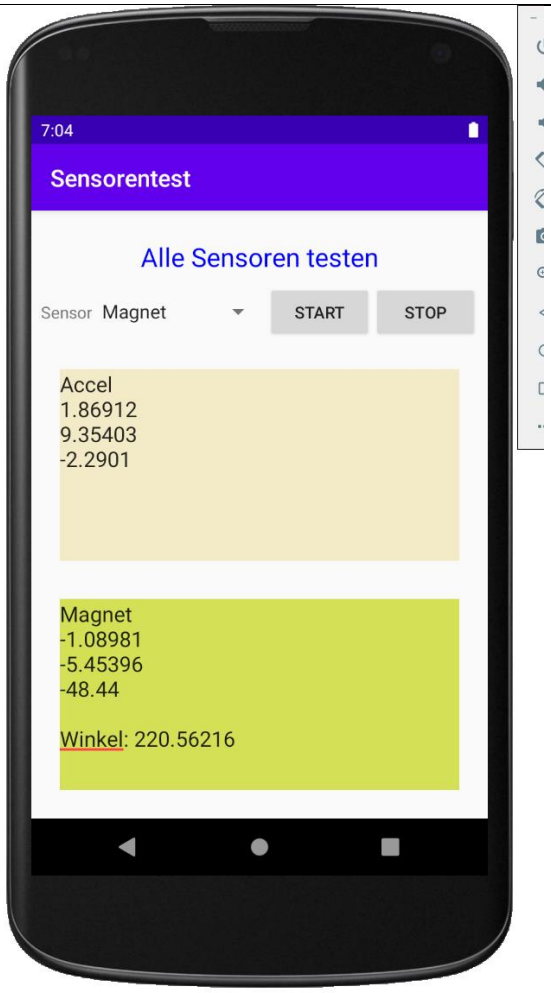


Abbildung 189 Sonsor Magnetfeld + Accelerator

17.4.1 String.xml

```
<resources>
  <string name="app_name">Sensorentest</string>
  <string name="tvtitle">Alle Sensoren testen</string>
  <string name="tvsensor">Sensor</string>
  <string name="bnstart">Start</string>
  <string name="bnstop">Stop</string>
  <string name="tveditor">editor</string>
  <string name="no_sensors">Kein Sensor gefunden</string>
  <string name="wait">Warte auf Bewegung</string>
  <string name="proceed">Weiter</string>
</resources>
```

17.4.2 Color.xml

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

```
<resources>
  <color name="colorPrimary">#6200EE</color>
  <color name="colorPrimaryDark">#3700B3</color>
  <color name="colorAccent">#03DAC5</color>
  <color name="colortitle">#0000FF</color>
  <color name="colorbackground1">#F3EBC7</color>
  <color name="colorbackground2">#D4E157</color>
</resources>
```

17.4.3 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/tvtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="16dp"
    android:text="@string/tvtitle"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="@color/colortitle"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />


  <LinearLayout
    android:id="@+id/linearlayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvtitle">

    <TextView
      android:id="@+id/tvsensor"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:text="@string/tvsensor" />

    <Spinner
      android:id="@+id/spsensor"
```

```
      android:spinnerMode="dropdown"
      app:layout_constraintBaseline_toBaselineOf="@+id/tvsensor"

      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1" />

    <Button
      android:id="@+id/bnstart"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:onClick="bnstart_Click"
      android:text="@string/bnstart" />

    <Button
      android:id="@+id/bnstop"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:onClick="bnstop_Click"
      android:text="@string/bnstop" />
  </LinearLayout>

  <LinearLayout
    android:id="@+id/linearlayout2"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearlayout1">

    <EditText
      android:id="@+id/teditor1"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginStart="16dp"
      android:layout_marginTop="16dp"
      android:layout_marginEnd="16dp"
      android:layout_marginBottom="16dp"
      android:layout_weight="1"

      android:background="@color/colorbackground1"

      android:gravity="start|top"
      android:inputType="textMultiLine"
      android:text="abc"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="@+id/linearlayout2"
      app:layout_constraintTop_toBottomOf="@+id/linearlayout2" />

    <EditText
      android:id="@+id/teditor2"
```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"

        android:layout_weight="1"
        android:background="@color/colorbackground2"

        android:gravity="start|top"
        android:inputType="textMultiLine"
        android:text="-"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/linearlayout2"
        app:layout_constraintTop_toBottomOf="@+id/linearlayout2" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

17.4.4 Klasse SensorItem.java

```
import android.hardware.Sensor;

public class SensorItem {

  public String bez="";
  public int type=0;
  public Sensor sensor;

  public SensorItem(String bez, int type, Sensor sensor) {
    this.bez = bez;
    this.type = type;
    this.sensor = sensor;
  }

  public String toString() {
    return bez;
  }

}
```

17.4.5 MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.util.Log;
```

```
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

  private final String  TAG = "SENSOR";

  private EditText teditor1 = null;
  private EditText teditor2 = null;
  private Spinner spsensor = null;



  private SensorManager sensorManager;
  private Sensor gyrometer;
  private Sensor pressure;
  private Sensor light;
  private Sensor temp;
  private Sensor humity;
  private Sensor magneto;
  private Sensor accel;
  private Sensor gravy;

  private Sensor heartbeat;
  private Sensor heartrate;
  private Sensor stepcounter;


  private Sensor activesensor1=null;
  private Sensor activesensor2=null;

  private float xaccel;
  private float yaccel;
  private float zaccel;
  private double mAccelLast = 0;


  private float[] lastAccelerometer = new float[3];
  private float[] lastMagnetometer = new float[3];

  SensorEventListener sensorListener=null;



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

    teditor1 = findViewById(R.id.teditor1);
    teditor2 = findViewById(R.id.teditor2);
    spsensor = findViewById(R.id.spsensor);

    sensorManager = getSystemService(SensorManager.class);
    //s = sensorManager.getDefaultSensor(Sensor.TYPE_SIGNIFICANT_MOTION);
```

```
    // Create listener
sensorListener = new SensorEventListener() {

  @Override
  public void onSensorChanged(SensorEvent event) {
    //Log.e(TAG, "onSensorChanged. Type: " + event.sensor.getType());
    StringBuilder sb = new StringBuilder();

    if (event.sensor.getType() == Sensor.TYPE_GYROSCOPE) {
      sb.append("TYPE_GYROSCOPE\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor1.setText(   sb.toString());
    }

    if (event.sensor.getType() == Sensor.TYPE_GRAVITY) {
      sb.append("TYPE_GRAVITY\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\n");
      teditor1.setText(   sb.toString());
    }

    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
      lastMagnetometer = event.values.clone();

      float[] mR = new float[9];
      float[] mOrientation = new float[3];
      SensorManager.getRotationMatrix(mR, null, lastAccelerometer,
                                      lastMagnetometer);
      SensorManager.getOrientation(mR, mOrientation);
      float rad = mOrientation[0];
      float grad = (float)(Math.toDegrees(rad)+360)%360;

      sb.append("Magnet\n");
      for (float item : event.values) {
        sb.append(Float.toString(item));
        sb.append("\n");
      }
      sb.append("\nWinkel: "+Float.toString(grad));
      sb.append("\n");
      if (activesensor2!=null) {
        teditor2.setText(   sb.toString());
      }
      else {
        teditor1.setText(   sb.toString());
      }

    }
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
      lastAccelerometer = event.values.clone();
      xaccel=event.values[0];
      yaccel=event.values[1];
      zaccel=event.values[2];
```

```
          mAccelLast = Math.sqrt(xaccel*xaccel + yaccel*yaccel
+zaccel*zaccel);
          sb.append("Accel\n");
          for (float item : event.values) {
            sb.append(Float.toString(item));
            sb.append("\n");
          }
          sb.append("\n");
          teditor1.setText(   sb.toString());

        }

        if (event.sensor.getType() == Sensor.TYPE_PRESSURE) {
          sb.append("TYPE_PRESSURE\n");
          for (float item : event.values) {
            sb.append(Float.toString(item));
            sb.append("\n");
          }
          sb.append("\n");
          teditor1.setText(   sb.toString());
        }

        if (event.sensor.getType() == Sensor.TYPE_LIGHT) {
          sb.append("TYPE_LIGHT\n");
          for (float item : event.values) {
            sb.append(Float.toString(item));
            sb.append("\n");
          }
          sb.append("\n");
          teditor1.setText(   sb.toString());
        }


        if (event.sensor.getType() == Sensor.TYPE_AMBIENT_TEMPERATURE) {
          sb.append("TYPE_AMBIENT_TEMPERATURE\n");
          for (float item : event.values) {
            sb.append(Float.toString(item));
            sb.append("\n");
          }
          sb.append("\n");
          teditor1.setText(   sb.toString());
        }

        if (event.sensor.getType() == Sensor.TYPE_RELATIVE_HUMIDITY) {
          sb.append("TYPE_RELATIVE_HUMIDITY, Feuchtigkeit\n");
          for (float item : event.values) {
            sb.append(Float.toString(item));
            sb.append("\n");
          }
          sb.append("\n");
          teditor1.setText(   sb.toString());
        }

      }


      @Override
      public void onAccuracyChanged(Sensor sensor, int i) {
        String s = "onAccuracy: " + i;
        //teditor.setText(s + teditor.getText().toString());
```

```
      }
    };

    teditor1.setText("");
    StringBuilder sb = new StringBuilder();

    ArrayList<SensorItem>  sensorliste = new ArrayList<SensorItem>();



    accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    if(accel == null) {
      sb.append(  "accel sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Accelerator",Sensor.TYPE_ACCELEROMETER,accel));
    }

    gravy = sensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
    if(gravy == null) {
      sb.append("gravy sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Gravity",Sensor.TYPE_GRAVITY,gravy));
    }



    gyrometer = sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
    if(gyrometer == null) {
      sb.append("mgyro sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Gyroscpe",Sensor.TYPE_GYROSCOPE,gyrometer));
    }

    magneto = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    if(magneto == null) {
      sb.append(  "magneto sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Magnet",Sensor.TYPE_MAGNETIC_FIELD,magneto));
    }

    humity= sensorManager.getDefaultSensor(Sensor.TYPE_RELATIVE_HUMIDITY);
    if(humity == null) {
      sb.append("humity sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Humidity",Sensor.TYPE_RELATIVE_HUMIDITY,humity));
    }


    light= sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
```

```
    if(light == null) {
      sb.append("light sensor not available.\n");
    }
    else {
      sensorliste.add(new SensorItem("Licht",Sensor.TYPE_LIGHT,light));
    }


    pressure= sensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE);
    if(pressure == null) {
      sb.append("pressure sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Pressure",Sensor.TYPE_PRESSURE,pressure));
    }

    temp= sensorManager.getDefaultSensor(Sensor.TYPE_AMBIENT_TEMPERATURE);
    if(temp == null) {
      sb.append("temp sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("Gravity",Sensor.TYPE_AMBIENT_TEMPERATURE, temp));
    }

    stepcounter= sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);
    if(stepcounter == null) {
      sb.append("stepCounter sensor not available.\n");
    }
    else {
      sensorliste.add(new
SensorItem("StepCounter",Sensor.TYPE_STEP_COUNTER,stepcounter));
    }

    heartbeat= sensorManager.getDefaultSensor(Sensor.TYPE_HEART_BEAT);
    if(heartbeat == null) {
      sb.append("heartbeat sensor not available.\n");
    }
    else {
      sensorliste.add(new SensorItem("Heart
Beat",Sensor.TYPE_HEART_BEAT,heartbeat));
    }

    heartrate= sensorManager.getDefaultSensor(Sensor.TYPE_HEART_RATE);
    if(heartrate == null) {
      sb.append("heartrate sensor not available.\n");
    }
    else {
      sensorliste.add(new SensorItem("Heart
Rate",Sensor.TYPE_HEART_RATE,heartrate));
    }


    teditor1.setText(sb.toString());

    ArrayAdapter<SensorItem> sensorAdapter = new
ArrayAdapter<SensorItem>(this,android.R.layout.simple_spinner_item,
sensorliste);
    spsensor.setAdapter(sensorAdapter);
```

```
  }  // onCreate


  public void bnstart_Click(View view) {
    if (activesensor1!=null) {
      sensorManager.unregisterListener(sensorListener,activesensor1);
      activesensor1=null;
    }
    if (activesensor2!=null) {
        // nur im Fall Magneto !
      sensorManager.unregisterListener(sensorListener,activesensor2);
      activesensor2=null;
    }
    SensorItem sensoritem = (SensorItem) spsensor.getSelectedItem();
    if (sensoritem.type==Sensor.TYPE_MAGNETIC_FIELD) {
      sensorManager.registerListener(sensorListener, accel,
SensorManager.SENSOR_DELAY_NORMAL);
      activesensor2 = accel;
    }
    sensorManager.registerListener(sensorListener, sensoritem.sensor,
SensorManager.SENSOR_DELAY_NORMAL);
    activesensor1 = sensoritem.sensor;
  }

  public void bnstop_Click(View view) {
    if (activesensor1!=null) {
      sensorManager.unregisterListener(sensorListener,activesensor1);
      activesensor1=null;
    }
    if (activesensor2!=null) {
      sensorManager.unregisterListener(sensorListener,activesensor2);
      activesensor2=null;
    }
  }


}
```

18 Chronometer

Dieses Kapitel zeigt den Einbau einer einfachen Stoppuhr und eines Countdowns.

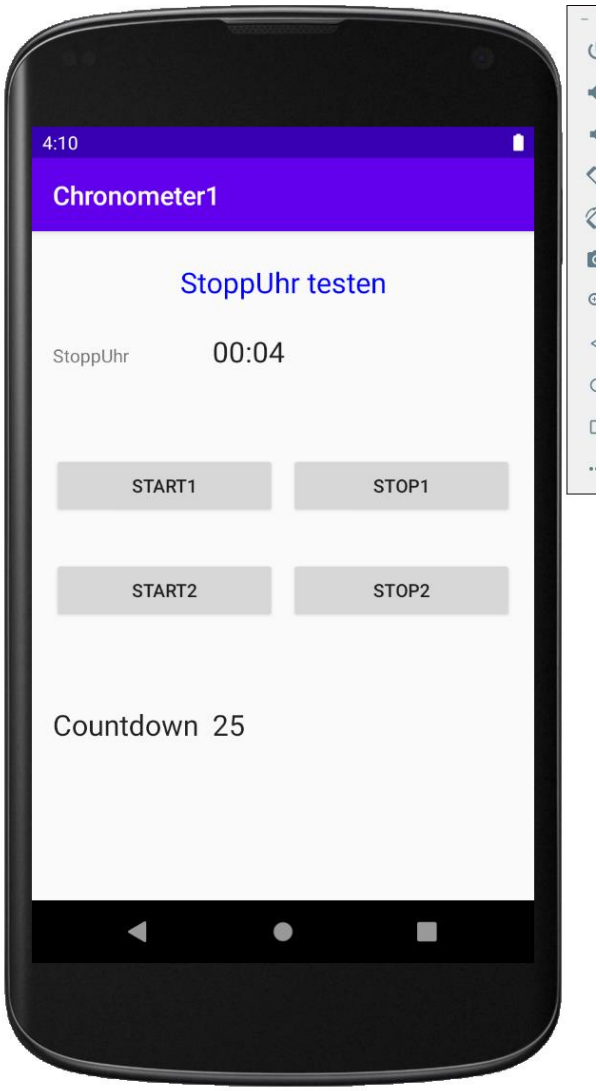


Abbildung 190 Musterlösung einer Stoppuhr und eines Counddown-Zählers

18.1 Projekt-Erstellung

Anlegen eines neuen Projektes.
Name: Chronometer1
Activity: leere Activity

18.2 colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#6200EE</color>
  <color name="colorPrimaryDark">#3700B3</color>
  <color name="colorAccent">#03DAC5</color>
  <color name="colortitle">#0000FF</color>
</resources>
```

18.3 strings.xml

```
<resources>
  <string name="app_name">Chronometer1</string>

  <string name="tvtitle">StoppUhr testen</string>
  <string name="tvstoppuhr">StoppUhr</string>

  <string name="bnstart1">Start1</string>
  <string name="bnstop1">Stop1</string>

  <string name="bnstart2">Start2</string>
  <string name="bnstop2">Stop2</string>

  <string name="tvcountdown">Countdown</string>

  <string name="tvcountdowntime">0</string>
</resources>
```

18.4 Layout:



Abbildung 191 Layout des Chronometers

Quellcode des Layouts:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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="match_parent"
    tools:context=".MainActivity">
```

```
<TextView
  android:id="@+id/tvtitle"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginStart="16dp"
  android:layout_marginTop="24dp"
  android:layout_marginEnd="16dp"
  android:text="@string/tvtitle"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:textColor="@color/colortitle"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent" />

<TextView
  android:id="@+id/tvstoppuhr"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginStart="16dp"
  android:layout_marginTop="32dp"
  android:text="@string/tvstoppuhr"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toBottomOf="@+id/tvtitle" />

<Chronometer
  android:id="@+id/chronometer"
  android:layout_width="200dp"
  android:layout_height="wrap_content"
  android:layout_marginStart="64dp"
  android:text="00:00"
  android:textAppearance="?android:attr/textAppearanceLarge"
  app:layout_constraintBaseline_toBaselineOf="@+id/tvstoppuhr"
  app:layout_constraintStart_toEndOf="@+id/tvstoppuhr" />

<LinearLayout
  android:id="@+id/linearLayout1"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_marginStart="16dp"
  android:layout_marginTop="64dp"
  android:layout_marginEnd="16dp"
  android:orientation="horizontal"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toBottomOf="@+id/chronometer">

    <Button
      android:id="@+id/bnstart1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginRight="5dp"
      android:layout_weight="1"
      android:onClick="bnstart1_Click"
      android:text="@string/bnstart1" />

    <Button android:id="@+id/bnstop1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="5dp"
```

```
        android:layout_weight="1"
        android:onClick="bnstop1_Click"
        android:text="@string/bnstop1" />
  </LinearLayout>

  <LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="16dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout1">

    <Button
      android:id="@+id/bnstart2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginRight="5dp"
      android:layout_weight="1"
      android:onClick="bnstart2_Click"
      android:text="@string/bnstart2" />

    <Button
      android:id="@+id/bnstop2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="5dp"
      android:layout_weight="1"
      android:onClick="bnstop2_Click"
      android:text="@string/bnstop2" />
  </LinearLayout>

  <TextView
    android:id="@+id/tvcountdown"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="64dp"
    android:text="@string/tvcountdown"
    android:textAppearance="?android:attr/textAppearanceLarge"
    app:layout_constraintStart_toStartOf="@+id/tvstoppuhr"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout2" />

  <TextView
    android:id="@+id/tvcountdowntime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tvcountdowntime"
    android:textAppearance="?android:attr/textAppearanceLarge"
    app:layout_constraintBaseline_toBaselineOf="@+id/tvcountdown"
    app:layout_constraintStart_toStartOf="@+id/chronometer" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

18.5 MainActivity.java

```
package com.example.chronometer1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.os.CountDownTimer;
import android.os.SystemClock;
import android.view.View;
import android.widget.*;


public class MainActivity extends AppCompatActivity {


  private Chronometer chronometer=null;
  private TextView tvcountdowntime=null;
  private CountDownTimer countDownTimer = null;

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

    chronometer =  findViewById(R.id.chronometer);
    tvcountdowntime = findViewById(R.id.tvcountdowntime);
  }

  public void bnstart1_Click(View view) {
    chronometer.setBase(SystemClock.elapsedRealtime());
      // chronometer.setFormat("Formated Time - %s");
    chronometer.start();
  }

  public void bnstop1_Click(View view) {
    chronometer.stop();
  }


  public void bnstart2_Click(View view) {
    countDownTimer = new CountDownTimer(30000, 1000) {
      public void onTick(long millisUntilFinished) {
        tvcountdowntime.setText(
                    Long.toString(millisUntilFinished / 1000) );
      }

      public void onFinish() {
        tvcountdowntime.setText("done!");
      }
    };
    countDownTimer.start();

  }

  public void bnstop2_Click(View view) {
    countDownTimer.cancel();
  }
}
```

Der Quellcode ist eigentlich selbsterklärend.

19 Anhang

19.1 toast

a toast show a message only a fewe seconds.

Version a)

```
import android.widget.Toast;
Toast.makeText(MainActivity.this, "My Message", Toast.LENGTH_LONG).show();
```

Version b)

```
android.widget.Toast.makeText(MainActivity.this, "My Message",
android.widget.Toast.LENGTH_SHORT).show();
android.widget.Toast.makeText(MainActivity.this, "My Message",
android.widget.Toast.LENGTH_LONG).show();
```

19.2 decimalFormat

```
private DecimalFormat df = new DecimalFormat("0.0");
double x=123.4567;
tOutput.setText( df.format(x) );
```

19.3 Umwandlung eines Textes in eine Integer-Zahl

```
private Object getIntNumber(String str_number) {
  try {
    int zahl = Integer.parseInt(str_number);
    return new Integer(zahl);
  } catch (NumberFormatException e) {
    return null;
  }
}
```

Die Abfrage erfolgt mit folgendem Code:

```
String dummy = tzahl1.getText().toString();  // nur ascii-text keine Zahl!!
Integer zahl1 = Basis.getIntNumber(dummy);
if (zahl1==null) {
  tverror.setText("Fehlerhafte 1. Zahl");
  tergebnis.setText("-");
  return;
}
```

19.4 Umwandlung eines Textes in eine Float-Zahl

```
try {
      z1 = Double.parseDouble(tZahl.getText().toString());
  }
  catch (NumberFormatException e) {
     tOutput.setText("Fehlerhafte Zahl");
     return;
  }



    private Double getNumber(EditText edText) {
        try {
            String s= edText.getText().toString();
            double zahl = Double.parseDouble(s);
            return new Double(zahl);
        } catch (NumberFormatException e) {
            return Double.NaN;
        }
    }
```

Die Abfrage erfolgt mit folgendem Code:

```
String dummy = tzahl1.getText().toString();  // nur ascii-text keine Zahl!!
double zahl1 = Basis.getDoubleNumber(dummy);
if (Double.isNaN(zahl1)) {
  tverror.setText("Fehlerhafte 1. Zahl");
  tergebnis.setText("-");
  return;
}
```

Neben „isNaN“ gibt es auch noch „isInfinite “, unendlich. Wenn man beide Abfragen einbauen wollte könnte man schreiben:

```
if (Double.isNaN(zahl1)|| Double. isInfinite (zahl1)) {
}
```

Beziehungsweise gibt es noch die Methode „isFinite(…). Diese Methode zeigt an, dass die Zahl gültig ist. Deshalb eine Negation:

```
if (!Double.isFinite(zahl1)) {
  tverror.setText("Fehlerhafte 1. Zahl");
  tergebnis.setText("-");
  return;
}
```

19.5 Random-Zahl erzeugen

```
Random rand = new Random();
int zahl1 = rand.nextInt(50) + 1;
double zahl2 = rand.nextDouble(50) + 1;
```

Constraint-Layout mit Schalter

```
<LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/spnote">

        <Button
            android:id="@+id/bnesc"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:onClick="bninsert_Click"
            android:text="@string/bninsert" />

        <Button
            android:id="@+id/bnok1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:onClick="bnok1_Click"
            android:text="@string/bnok1" />
        <Button
            android:id="@+id/bnok2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:onClick="bnok2_Click"
            android:text="@string/bnok2" />
    </LinearLayout>
```

19.6 ListView, Update

Man muss nur den Adapter neu setzen:

```
listView.setAdapter(adapter);
```

19.7 Constraint-Layout, zwei Editoren untereinander

```
<LinearLayout
  android:id="@+id/linearlayout2"
  android:layout_width="match_parent"
  android:layout_height="0dp"
  android:layout_marginStart="8dp"
```

```
  android:layout_marginTop="8dp"
  android:layout_marginEnd="8dp"
  android:layout_marginBottom="8dp"
  android:orientation="vertical"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toBottomOf="@+id/linearlayout1">

  <EditText
    android:id="@+id/teditor1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:layout_weight="1"

    android:background="@color/colorbackground1"

    android:gravity="start|top"
    android:inputType="textMultiLine"
    android:text="abc"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/linearlayout2"
    app:layout_constraintTop_toBottomOf="@+id/linearlayout2" />

  <EditText
    android:id="@+id/teditor2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"

    android:layout_weight="1"
    android:background="@color/colorbackground2"

    android:gravity="start|top"
    android:inputType="textMultiLine"
    android:text="abc"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/linearlayout2"
    app:layout_constraintTop_toBottomOf="@+id/linearlayout2" />
</LinearLayout>
```

19.8 Klasse Basis.java

```
import android.content.Context;
import android.widget.EditText;
import android.widget.Toast;
```

```
import java.io.*;


public class Basis {

  public static  boolean FileExists(Context context, String filename) {
    try {
      FileInputStream fin = context.openFileInput(filename);
      fin.close();
      return true;
    }
    catch (IOException e) {
      return false;
    }
  }


  public static void showToast(Context context, int stringid) {
    Toast.makeText(context, context.getString(stringid),
                                        Toast.LENGTH_SHORT).show();
  }

  public static void showToast(Context context, String text) {
    Toast.makeText(context, text, Toast.LENGTH_LONG).show();
  }


    // wandelt einen String in eine double zahl um
    // bei Fehler wird double.NaN zurÃ¼ckgegeben
  public static double getDoubleNumber(String input) {
    try {
      return Double.parseDouble(input);
    }
    catch (NumberFormatException e) {
      return Double.NaN;
    }
  }
/*    Abfrage:
    if (Double.isNaN(tzahl.getText().toString())) {
      // hier ist ergebnis NaN !!
      tergebnis.setText("-");
    }

    String dummy = tzahl1.getText().toString();  // nur ascii-text keine
Zahl!!
    double zahl1 = Basis.getDoubleNumber(dummy);
    if (!Double.isFinite(zahl1)) {
      tverror.setText("Fehlerhafte 1. Zahl");
      tergebnis.setText("-");
      return;
    }
*/


 public static Double getDoubleNumber(EditText tInput, String errortext) {
    try {
      String s= tInput.getText().toString();
      double zahl = Double.parseDouble(s);
      return new Double(zahl);
    }
```

```
    catch (NumberFormatException e) {
      tInput.setError(errortext);
      return Double.NaN;
    }
  }

  public static Integer getIntegerNumber(String input) {
    try {
      int zahl = Integer.parseInt(input);
      return new Integer(zahl);
    }
    catch (NumberFormatException e) {
      return null;
    }
  }


 public static Integer getIntegerNumber(EditText tInput, String errortext){
    try {
      String s= tInput.getText().toString();
      int zahl = Integer.parseInt(s);
      return new Integer(zahl);
    }
    catch (NumberFormatException e) {
      tInput.setError(errortext);
      return null;
    }
  }

}  // class Basis.java
```

19.9 onClick, Code Behind

```
bntest.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    bntest_Click();
  }
});

private void bntest_Click() {
  // Aktion
}
```

19.10 Erstellen eines „Plus“-Icons

- Auswahl des Plus-Symbol
- Aktivieren des Eintrags „res/drawable“
  - Anklicken des Eintrags „res“
  - Anklicken des Eintrags „drawable“
  - Menü „File“
    - Eintrag „New“

- Eintrag „Vector Asset“



- Anklicken des Clip Art Symbol (Android“
- Anklicken des „+“-Symbols

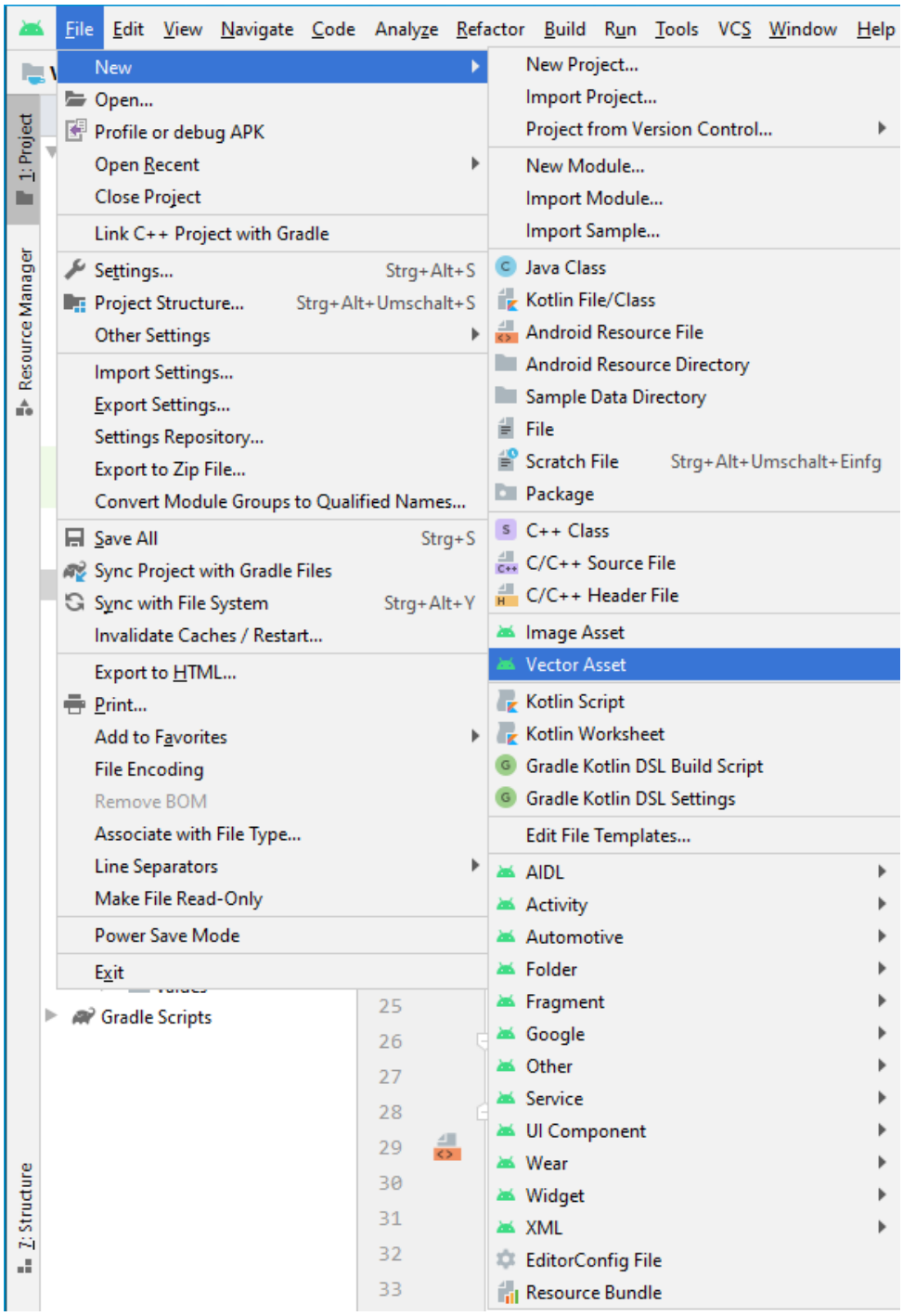


Abbildung 192 Auswahl des Menueintrags „drawable Symbol“

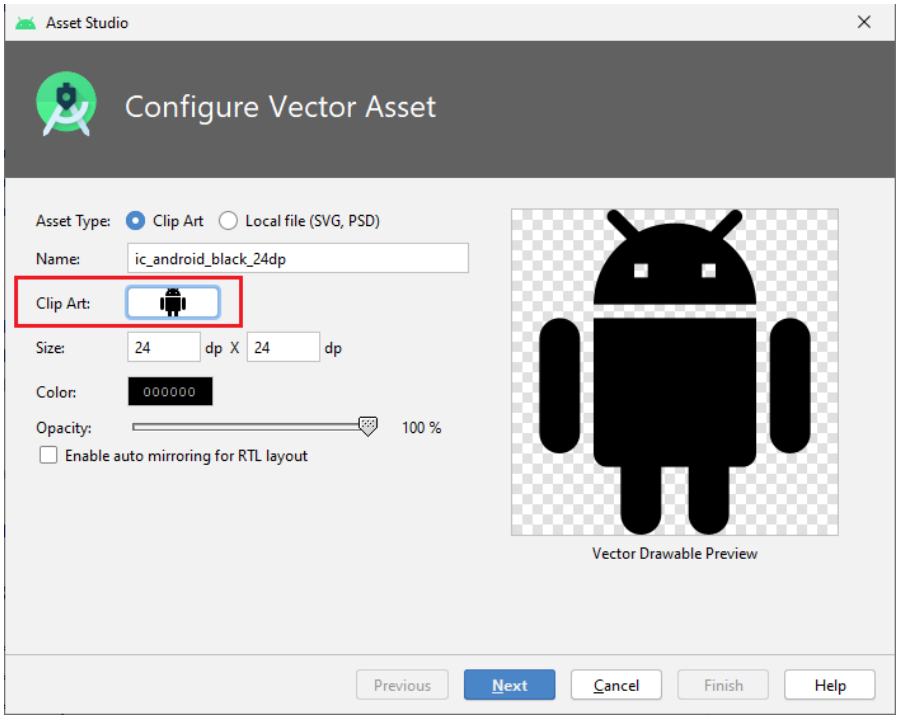


Abbildung 193 Auswahldes neuen Vektore-Symbols

Weiter mit dem Anklicken des kleinen Android-Symbols.

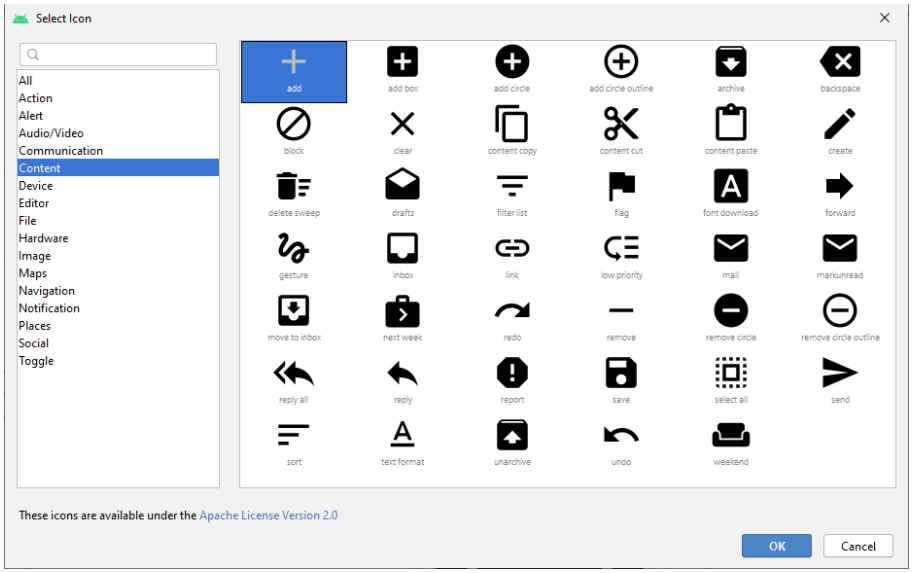


Abbildung 194 Auswahl des Plus-Symbols

Am Schluss zweimal „ok“ drücken.

Inhalt der Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="@string/title_home" />

    <item
        android:id="@+id/navigation_dashboard"
```

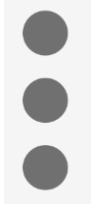
```
        android:icon="@drawable/ic_dashboard_black_24dp"
        android:title="@string/title_dashboard" />

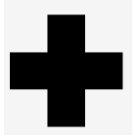
    <item
        android:id="@+id/navigation_notifications"
        android:icon="@drawable/ic_notifications_black_24dp"
        android:title="@string/title_notifications" />

</menu>
```

19.11 Beispiel Symbole

Bild	xml-Texte

	`<?xml version="1.0" encoding="utf-8"?>`
	`<vector xmlns:android="http://schemas.android.com/apk/res/android"`
	`android:width="24dp"`
	`android:height="24dp"`
	`android:viewportWidth="24.0"`
	`android:viewportHeight="24.0"`
	`android:tint="?attr/colorControlNormal">`
	`<path`
	`android:pathData="M12,8c1.1,0 2,-0.9 2,-2s-0.9,-2 -2,-2c-1.1,0 -2,0.9 -2,2S10.9,8 12,8zM12,10c-1.1,0 -2,0.9 -2,2s0.9,2 2,2c1.1,0 2,-0.9 2,-2S13.1,10 12,10zM12,16c-1.1,0 -2,0.9 -2,2s0.9,2 2,2c1.1,0 2,-0.9 2,-2S13.1,16 12,16z"`
	`android:fillColor="@android:color/white"/>`
	`</vector>`

	`<vector xmlns:android="http://schemas.android.com/apk/res/android"`
	`android:width="24dp"`
	`android:height="24dp"`
	`android:viewportHeight="24.0"`
	`android:viewportWidth="24.0">`
	`<path`
	`android:fillColor="#FF000000"`
	`android:pathData="M4,16 v-6h18v6h-18 M10,4v6h6v-6h-6 M10,16v6h6v-6h-6 z"/>`
	`</vector>`

	`<vector xmlns:android="http://schemas.android.com/apk/res/android"`
	`android:width="24dp"`
	`android:height="24dp"`
	`android:viewportHeight="24.0"`
	`android:viewportWidth="24.0">`
	`<path`
	`android:fillColor="#FF000000"`
	`android:pathData="M4,15 v-6h18v6h-18 z" />`
	`</vector>`

	`<?xml version="1.0" encoding="utf-8"?>`
	
	`<vector xmlns:android="http://schemas.android.com/apk/res/android"`
	`android:width="24dp"`
	`android:height="24dp"`
	`android:viewportWidth="24.0"`
	`android:viewportHeight="24.0">`
	
	`<path`
	`android:fillColor="#3949AB"`

```
        android:pathData="M2,7 L6,3 L12,9 L18,3 L22,7 L16,13 L21,18 L17,22 L12,17
L7,22 L3,18 L8,13 z "/>

</vector>
```

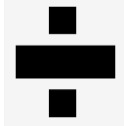
```
Punkte:  2/7     6/3     12/9     18/3     22/7     16/13    21/18
         17/22   12/17   7/22     3/18     8/13
```



```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M2,7 L6,3 L21,18 L17,22 z M3,18 L7,22 L22,7 L18,3 z "/>
</vector>
```

Zwei Rechtecke !

```
R1:      Punkte:          2/7     6/3     21/18    17/22
R2:      Punkte:          3/18    7/22    22/7     18/3
```



```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M4,15 v-6h18v6h-18 M10,2v5h5v-5h-5 M10,17v5h5v-5h-5  z"
/>
</vector>
```



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

<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportHeight="24"
        android:viewportWidth="24">

    <path
        android:fillColor="@android:color/black"
        android:pathData="M12,7c2.76,0 5,2.24 5,5 0,0.65 -0.13,1.26 -
0.36,1.83l2.92,2.92c1.51,-1.26 2.7,-2.89 3.43,-4.75 -1.73,-4.39 -6,-7.5 -11,-7.5
-1.4,0 -2.74,0.25 -3.98,0.7l2.16,2.16C10.74,7.13 11.35,7 12,7zM2,4.27l2.28,2.28
0.46,0.46C3.08,8.3 1.78,10.02 1,12c1.73,4.39 6,7.5 11,7.5 1.55,0 3.03,-0.3 4.38,-
0.84l0.42,0.42L19.73,22 21,20.73 3.27,3 2,4.27zM7.53,9.8l1.55,1.55c-0.05,0.21 -
0.08,0.43 -0.08,0.65 0,1.66 1.34,3 3,3 0.22,0 0.44,-0.03 0.65,-0.08l1.55,1.55c-
0.67,0.33 -1.41,0.53 -2.2,0.53 -2.76,0 -5,-2.24 -5,-5 0,-0.79 0.2,-1.53 0.53,-
2.2zM11.84,9.02l3.15,3.15 0.02,-0.16c0,-1.66 -1.34,-3 -3,-3l-0.17,0.01z"/>

</vector>
```



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

<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0"
```

```
        android:tint="?attr/colorControlNormal">
    <path

android:pathData="M19,6.41L17.59,5,12,10.59,6.41,5,5,6.41,10.59,12,5,17.59,6.41,1
9,12,13.41,17.59,19,19,17.59,13.41,12z"
            android:fillColor="@android:color/white"/>
</vector>
```



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


<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:height="24dp"
        android:viewportHeight="24.0"
        android:viewportWidth="24.0"
        android:width="24dp"
        android:tint="?attr/colorControlNormal"
        android:autoMirrored="true">

    <group
            android:name="arrow"
            android:rotation="90.0"
            android:pivotX="12.0"
            android:pivotY="12.0">
        <path android:fillColor="@android:color/black" android:pathData="M7,14
L12,9 L17,14 L7,14 Z" />
        <path android:pathData="M0,0 L24,0 L24,24 L0,24 L0,0 Z" />
    </group>
</vector>
```

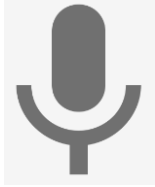


```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M17,3L5,3c-1.11,0 -2,0.9 -2,2v14c0,1.1 0.89,2
2,2h14c1.1,0 2,-0.9 2,-2L21,7l-4,-4zM12,19c-1.66,0 -3,-1.34 -3,-3s1.34,-3 3,-3
3,1.34 3,3 -1.34,3 -3,3zM15,9L5,9L5,5h10v4z"/>
</vector>
```

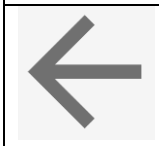


```
<?xml version="1.0" encoding="utf-8"?>

<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0"
        android:tint="?attr/colorControlNormal">
    <path
        android:pathData="M15.5,14l-0.8,0l-0.3,-0.3c1,-1.1 1.6,-2.6 1.6,-
4.2C16,5.9 13.1,3 9.5,3C5.9,3 3,5.9 3,9.5S5.9,16 9.5,16c1.6,0 3.1,-0.6 4.2,-
1.6l0.3,0.3l0,0.8l5,5l1.5,-1.5L15.5,14zM9.5,14C7,14 5,12 5,9.5S7,5 9.5,5C12,5
14,7 14,9.5S12,14 9.5,14z"
        android:fillColor="@android:color/white"/>
</vector>
```



```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0"
        android:tint="?attr/colorControlNormal">
    <path
        android:pathData="M12,14c1.7,0 3,-1.3 3,-3l0,-6c0,-1.7 -1.3,-3 -3,-3c-
1.7,0 -3,1.3 -3,3l0,6C9,12.7 10.3,14 12,14zM17.299999,11c0,3 -2.5,5.1 -5.3,5.1c-
2.8,0 -5.3,-2.1 -5.3,-5.1L5,11c0,3.4 2.7,6.2 6,6.7L11,21l2,0l0,-3.3c3.3,-0.5 6,-
3.3 6,-6.7L17.299999,11.000001z"
        android:fillColor="@android:color/white"/>
</vector>
```



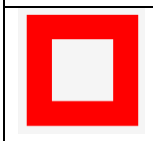
```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0"
        android:autoMirrored="true"
        android:tint="?attr/colorControlNormal">
    <path
            android:pathData="M20,11L7.8,11l5.6,-5.6L12,4l-8,8l8,8l1.4,-
1.4L7.8,13L20,13L20,11z"
            android:fillColor="@android:color/white"/>
</vector>
```



```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0"
        android:tint="?attr/colorControlNormal">
    <path
        android:pathData="M10,6l-1.4,1.4 4.599999,4.6 -4.599999,4.6 1.4,1.4 6,-
6z"
        android:fillColor="@android:color/white"/>
</vector>
```



```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M12,22c1.1,0 2,-0.9 2,-2h-4c0,1.1 0.89,2 2,2zM18,16v-
5c0,-3.07 -1.64,-5.64 -4.5,-6.32L13.5,4c0,-0.83 -0.67,-1.5 -1.5,-1.5s-1.5,0.67 -
1.5,1.5v0.68C7.63,5.36 6,7.92 6,11v5l-2,2v1h16v-1l-2,-2z" />
</vector>
```



```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="#FFFF0000"
        android:pathData="M4,4 v18h18 v-18h-18 M18,8 v10h-10 v-10h10 z"
```

	/>`</vector>` Hier werden zwei Vierecke übereinander gezeichnet. Die Schnittfläche ist dann weiß. Siehe das obere Multiplkations-Symbol.

20 Stichwortverzeichnis

0

0dp 102

A

Activity
  aufrufen 285
Activity aufrufen 285
Adapter
  ArrayAdapter 120
  BaseAdapter 120
  Cursor 314
  CursorAdapter 120
  CustomArrayAdapter 120
  CustomSimpleAdapter 120
  SimpleAdapter 120
Anhang 365
  decimalFormat 365
  Random Zahl 366
  Text vs. Float 366
  Text vs. Integer 365
  toast 365
AVD Manager 27

B

BaseAdpater 134
BaseLine
  Spinner 101
Basis.java 67, 138, 199, 216, 368
Beispiele
  Spritverbrauch 69
Bottom Navigation Activity 173
Button 81
  Code Behind 370

C

callback-Methode 286
CheckBox 84
  ListView 165
Chronometer 359
Constraint
  Button 367
  Schalter 367
ConstraintLayout
  Barriers 99
  Group 99
  Leeres Row-Layout 122
  Prozent 99
  Ratio 98
Constraint-Layout 96
Constraint-Layout
  baseline 100
Countdown 359
Cursor 314

D

DataInputStream 292, 293
DataOutputStream 292
Datenbank
  EditDeleteNewActivity 335
  ListView 327
    DBSOpenHandler 327
    DBSQueries 329
    Student 328
  MainActivity 332
  MyCursorAdapter 331
Datenbanken 301
Datenbase
  ListView
    DBSQueries 329
    Student 328
  MyCursorAdapter 331
decimalFormat 365
drawable 373
Drawable Symbol erstellen 188, 370

E

EditText 78
Eigene Navigation 268

F

Factory 258
Float 366

G

getDoubleNumber 67
getIntegerNumber 67
Gravity-Konstanten 108
GridLayout 116

I

IME_ACTION_DONE 200, 228
IME_ACTION_GO 200, 228
IME_ACTION_NEXT 200, 228
IME_ACTION_SEND 200, 228
Installation 13
Integer 365
Intent 285
isFinite 366
isInfinite 366
isNaN 366

K

Kamera 340

L

Layout 96
  Baseline 100
  Constraint-Layout 96
  Gravity-Konstanten 108
  GridLayout 116

LinearLayout 105
numberDecimal 103
RelativeLayout 108
TableLayout 112
Leeres Row-Layout
ConstraintLayout 122
LinearLayout 105
ListSelector 126
ListView 120
CheckBox 165
Refresh 367
StudentenAdapter 125
Update 367

M

Master Detail 267
Menü 93
Mikrofon 340
MultiPages
Bottom Navigation Activity 173
MultiPages 173
MultiPages
Tabbed Activity 202
MultiPages
Swipe-Activity 256
MultiPages
Master Detail 267
MultiPages
Eigene Navigationl 268

N

numberDecimal 103
numberSigned 103

O

onActivityResult 164
onClick 370
Optionsmenu
Eigene Symbole 238

P

package 178
Parameter
serialize 146
Parameterübergabe
Serialize 125

R

RadioButton 85
RadioGroup 86
orientation 86
Random Zahl 366
RelativeLayout 108
resultCode 164

S

SDK 23, 26
SectionsPageAdapter 260

Sensoren ... 340
Accelerator ... 341
Alle Sensoren ... 349
Allgemeiner Aufbau ... 340
Beschleunigung ... 341
Gravity ... 342
Gyroscope ... 343
Humidity ... 346
Licht ... 347
Light ... 347
Luftdruck ... 345
Luftfeuchtigkeit ... 346
Magnetfield ... 344
Pressure ... 345
Schwerkraft ... 342
Sensor Typen ... 340
Temperatur ... 342
serialize ... 146
Serialize ... 125, 147
setOnEditorActionListener ... 200, 228
showtoast ... 67
Singleton ... 133
Spinner ... 90
BaseLine ... 101
SQLite ... 301
Aufbau in Android ... 302
Datenbank erzeugen ... 303
Datentypen ... 301
Eigenschaften ... 301
Gemappte Datentypen ... 301
getQuery ... 305
Insert ... 304
Löschen der Datenbank ... 306
Stoppuhr ... 359
Swipe-Activity ... 256
Switch ... 88
Symbol erstellen ... 188, 370

T

Tabbed Activity ... 202
TableLayout ... 112
Text vs. Float ... 366
Text vs. Integer ... 365
TextView ... 77
toast ... 365

U

UI-Elemente ... 77
Button ... 81
CheckBox ... 84
EditText ... 78
RadioButton ... 85
Spinner ... 90
Switch ... 88
TextView ... 77

V

ViewModel ... 178, 183, 208
Virtual Device ... 27

W

width
0dp 102