

Modul 3 – Android App Entwicklung mit Kotlin

Wiederholung



Gliederung

- Was haben wir zuletzt gelernt?
- Einsatz in der ShoppingApp

```
if (polite) {  
    greetingText.text = "Guten Tag $name $surname! \nUnser Servicepersonal steht Ihnen  
} else {  
    greetingText.text = "Hallo $name $surname! \nGib bescheid wenn du was brauchst  
}
```

Do not concatenate text displayed with setText. Use resource string with placeholders. ⋮

String literal in setText can not be translated. Use Android resources instead.

Suppress: Add @SuppressWarnings("SetText18n") annotation ↗ ↘ More actions... ↗ ↘

```
2022-03-17 16:57:38.023 16953-16953/com.example.databinding D/MainActivity: counter: 3  
2022-03-17 16:57:38.276 16953-16953/com.example.databinding D/MainActivity: counter: 2  
2022-03-17 16:57:38.525 16953-16953/com.example.databinding D/MainActivity: counter: 1  
2022-03-17 16:57:38.757 16953-16953/com.example.databinding D/MainActivity: counter: 0  
2022-03-17 16:57:38.757 16953-16953/com.example.databinding D/AndroidRuntime: Shutting down VM  
2022-03-17 16:57:38.759 16953-16953/? E/AndroidRuntime: FATAL EXCEPTION: main
```

```
<com.google.android.material.card.MaterialCardView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:cardCornerRadius="18dp"  
    app:cardElevation="5dp"  
    app:cardPreventCornerOverlap="false"  
    app:contentPadding="32dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent">
```

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

```
-- <EditText --  
--     android:id="@+id/vorname_input0LD"-->  
--     android:layout_width="220dp"-->  
--     android:layout_height="wrap_content"-->  
--     android:hint="Vorname"-->  
--     android:inputType="textPersonName"/>-->
```

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/vorname_input"  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox
```

Vorname

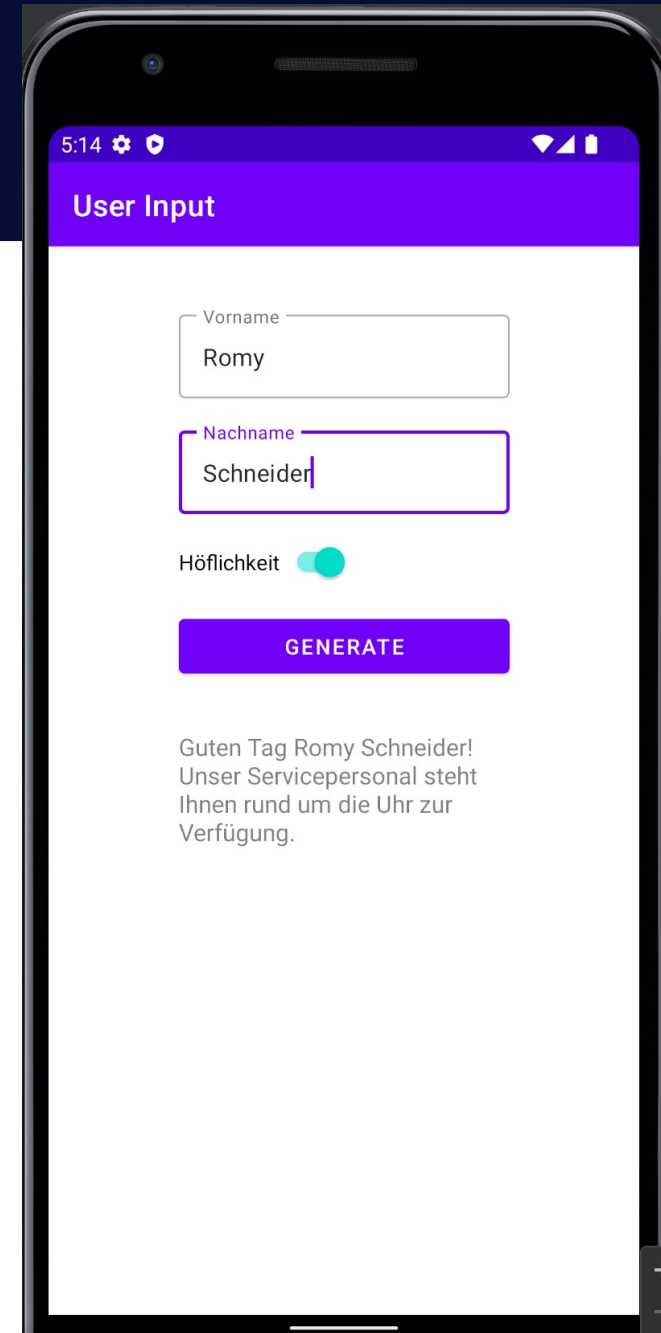
Anna

Nachname

Droid

Wiederholung

- UI Elemente für Benutzereingabe
- Strings immer von Resources abrufen um Übersetzung zu ermöglichen



DataBinding

- mittels DataBinding können sämtliche Elemente das Layouts über eine **binding** Variable angesprochen werden

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    // setContentView(R.layout.activity_main)
    val binding: ActivityMainBinding =
        DataBindingUtil.setContentView<ActivityMainBinding>(activity: this, R.layout.activity_main)

    binding.counting = "Counter"

    var counter = 0

    binding.textView.text = counter.toString()

    binding.button.setOnClickListener { it: View! ->
        if (counter < 10) {
            counter ++
            binding.textView.text = counter.toString()
        } else {
            binding.button.visibility = View.GONE
            binding.counting = "END"
        }
    }
}
```

Debugging

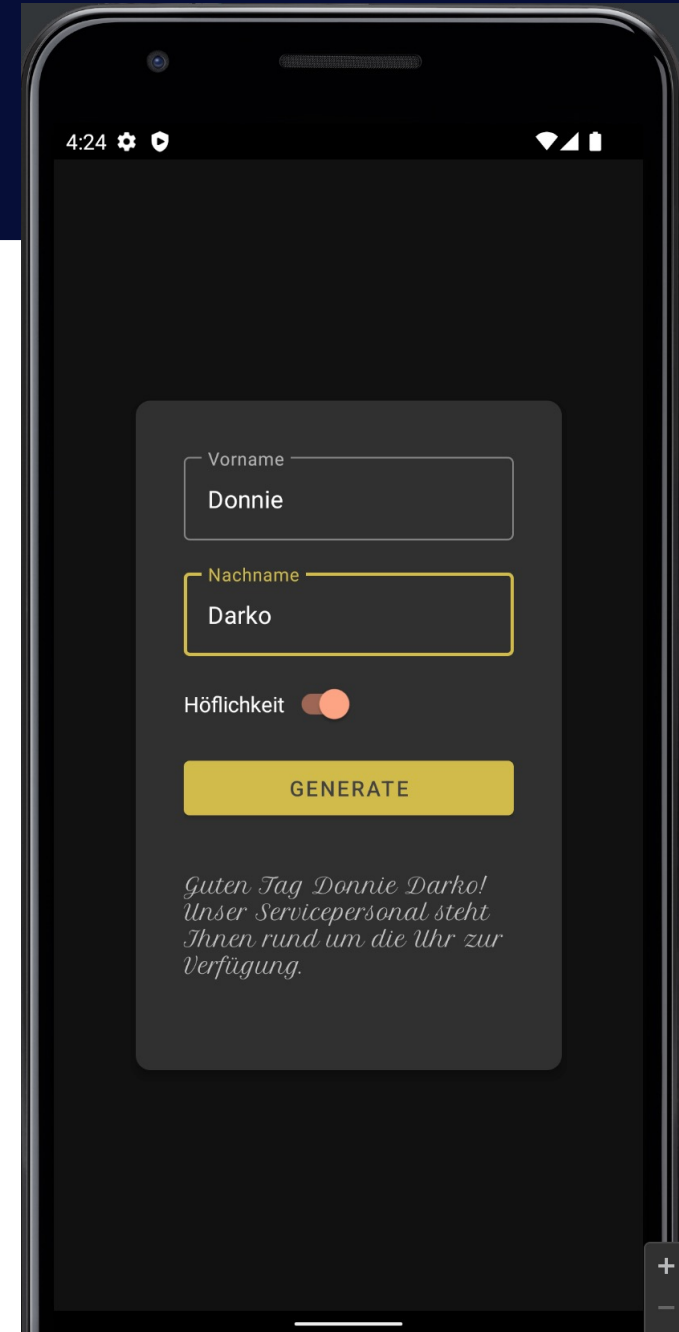
- Lognachrichten in der Konsole ausgeben
- Fehler mittels Exception Handling abfangen

```
try{...} catch{...}
```

```
2022-03-17 16:57:38.023 16953-16953/com.example.databinding D/MainActivity: counter: 3
2022-03-17 16:57:38.276 16953-16953/com.example.databinding D/MainActivity: counter: 2
2022-03-17 16:57:38.525 16953-16953/com.example.databinding D/MainActivity: counter: 1
2022-03-17 16:57:38.757 16953-16953/com.example.databinding D/MainActivity: counter: 0
2022-03-17 16:57:38.757 16953-16953/com.example.databinding D/AndroidRuntime: Shutting down VM
2022-03-17 16:57:38.759 16953-16953/? E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.example.databinding, PID: 16953
    java.lang.ArithmeticException: divide by zero
        at com.example.databinding.MainActivity.onCreate$lambda-0(MainActivity.kt:41)
        at com.example.databinding.MainActivity.$r8$lambda$IwsRH-8SG95i6TXVXHDGHmEiAFo(Unknown Source:0)
        at com.example.databinding.MainActivity$$ExternalSyntheticLambda0.onClick(Unknown Source:4)
        at android.view.View.performClick(View.java:7455)
        at com.google.android.material.button.MaterialButton.performClick(MaterialButton.java:1131)
        at android.view.View.performClickInternal(View.java:7432)
```

Wiederholung

- **Themes** bestimmen das grundlegende Erscheinungsbild der App (Farben, Schriften, usw)
- Farben sind in `res/values/colors.xml` definiert
- Ein dunkles Theme wird in einem separaten `values-night` Ordner erstellt



Wiederholung

- Ein **App-Icon** kann im Image Asset Studio erstellt werden
- Ein dynamisches Icon besteht aus Vordergrund und Hintergrund
- **Material Design Components** bietet anpassbare, fertige Bausteine für die UI Gestaltung

A UI mockup of a form on a light yellow background. The form is a light orange rounded rectangle. It contains a text input field labeled 'Vorname', a text input field labeled 'Nachname' with a yellow border, a toggle switch labeled 'Höflichkeit' which is currently turned on (red), and a yellow button labeled 'GENERATE'.



Viel Spaß!

Quelle: <https://www.chatelaine.com/food/grocery-costs-canada/>