

Übung 8: Jetpack Compose

In dieser Übung geht es um einen Einstieg in Jetpack Compose, also um das neue deklarative GUI-Toolkit für Android. Sie werden dazu ein Android Code-Lab absolvieren und dann die dabei entwickelte App weiterentwickeln. Dabei wird es vor allem darum gehen, wie Zustand in Jetpack Compose gehandhabt wird, Stichworte "State Hoisting" und "Source of Truth".

1. Neue Compose-App: FirstCompose

Google bietet mit dem Code-Lab "Jetpack Compose basics" eine gute Einführung in's Thema ein, besuchen Sie also die folgende Seite:

<https://developer.android.com/codelabs/jetpack-compose-basics>

Spielen Sie dieses Code-Lab komplett durch, Sie sollten danach eine App haben, die ca. aussieht wie im Screenshot rechts.

Hinweise:

- Nennen Sie ihr im Rahmen von diesem Code-Lab erstelltes Compose-Projekt "**FirstCompose**" (und **nicht** "Basics Codelab")
- Verwenden Sie als Basis-Package-Name **ch.hslu.mobpro.firstCompose** (und **nicht** **com.codelab.basics**)

Finden sie beim Ausführen von diesem Code-Lab Antworten auf die folgenden Fragen:

- Was ist eine "Composable Function"?
- Ist eine "Composable Function" dasselbe wie eine View im bisherigen Android? Oder worin unterscheiden sich die beiden?
- Erläutern Sie in eigenen Worten, was im Kontext von Jetpack Compose "Source of truth" und "State hoisting" bedeuten.

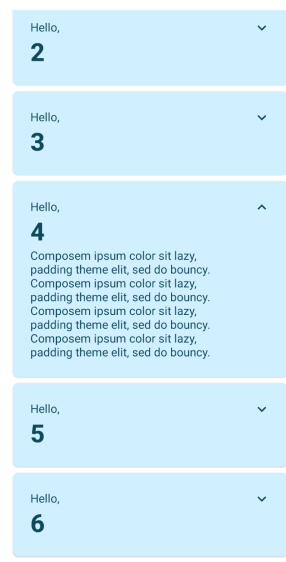
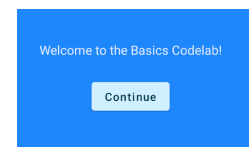
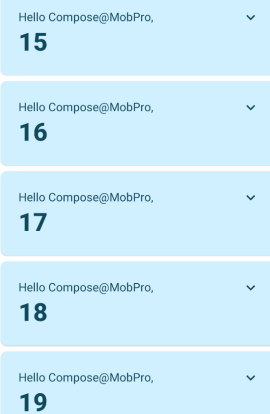
2. Layout anpassen: dynamischer Titel (und Zähler)

Der Hauptbildschirm der App soll neu oben einen Titel haben und darunter eine Reihe mit zwei Knöpfen, erst danach soll die bisherige Liste mit den "Hello..."-Elementen folgen. Ergänzen sie die App entsprechend, indem sie die vorhandene Composable-Funktion Greetings aus dem Code-Lab wie folgt erweitern:

```
@Composable
private fun Greetings(names: List<String> = List(1000) { "$it" }) {
    Column(modifier = Modifier.fillMaxHeight()) {
        MainTitle(text = lastText)
        CounterButtonRow()
        LazyColumn( ...
```

Last clicked #42

I've been clicked 7 times reset counter



Hinweis: LazyColumn soll dabei die aus dem Code-Lab vorhandene Liste sein und nicht verändert werden.

MainTitle solle dabei einen Bildschirmtitel darstellen. Verwenden sie dazu eine Composable-Funktion mit folgender Signatur und formatieren sie diesen Text analog zum Screenshot rechts.

Last clicked #42

```
@Composable
fun MainTitle(text: String) {
    Text( ... )
}
```

Im Titel soll immer die zuletzt angezeigte Zeile der Liste angezeigt werden. Verwenden sie dazu "State hoisting" und rufen sie dazu wie in der Vorlesung gezeigt an geeigneter Stelle die Funktion Modifier.clickable auf. Wo deklarieren sie die dazu notwendige Variable, welchen den Zustand vom Titel hält?

I've been clicked 7 times

reset counter

3. Zähler- und Reset-Knopf

Die App soll weiter zwei Knöpfe haben für eine einfache Zähler-Funktionalität. Mit dem ersten Knopf soll durch Drücken ein Zähler hochgezählt werden können. Dieser Knopf soll mit einem Text "I've been clicked X times" anzeigen, wie oft er gedrückt wurde. Falls der Knopf (seit dem letzten Reset) weniger als 6-mal gedrückt wurde, soll er weiss sein, sonst grün. Hinweis: Sie finden dazu Hinweise auf den Folien. Ergänzen Sie dann die App weiter um einen zweiten Knopf "reset counter", mit welchem der Zähler auf 0 zurück gesetzt werden kann. Die beiden Knöpfe sollen wie im Screenshot rechts oben nebeneinander angeordnet sein. Implementieren Sie dazu eine neue Composable-Funktion CounterButtonRow, die wie folgt aufgebaut ist:

I've been clicked 0 times

```
fun CounterButtonRow() {
    var counterState by rememberSaveable { mutableStateOf(0) }
    Row {
        CounterButton( ... )
        ResetButton( ... )
    }
}
```

Diese Composable Function ist also die "Source of truth" für den Zählerzustand und merkt sich diesen Zustand in counterState. Konsultieren Sie dazu allenfalls die Einführung zu "State and Jetpack Compose", welche unter <https://developer.android.com/jetpack/compose/state> zu finden ist. Auf dieser Seite finden Sie u.a. Informationen zu "rememberSaveable" oder "State hoisting".

Gratulation, damit haben Sie eine erste eigene App mit Jetpack Compose erstellt! 🎉