
Velkommen til gruppetime i IN2000!

Gruppe 1 med Ida, Jamila & Jørgen

Viktige beskjeder!

- Husk å sende inn skjema for team! [Lenke til skjemaet](#)
 - if (du ikke har avtalt team med noen) {
 - svar på skjemaet og meld inn deg selv
 - } else if {
 - (du blir meldt inn av andre medstudenter){
 - ikke** svar på skjemaet
 - } else {
 - Frist: **fredag 2. februar, kl. 23:59**
 - }
 - Oblig 1: frist **tirsdag 30. januar kl.23.59**
-

Agenda

- Hva er forskjell på Android Studio, Kotlin og Jetpack Compose?
 - Livekoding med Jørgebassen
 - La oss programmere en handleliste app! :)
 - Tips og triks underveis
 - Etter livekoding har vi lab/oblighjelp
-

Android Studio? Kotlin? Jetpack Compose? What?

KOTLIN

Programmeringsspråket. Hele programmet ditt er skrevet i Kotlin. Måten du skriver koden på. Det er det du bruker for å bestemme hva som skal skje når og hvordan.

<https://kotlinlang.org/>



ANDROID STUDIO

En IDE (integrated development environment). Egentlig bare programmet du laget appen din, hvor alle verktøyene du trenger for å lage appen ligger (kodeeditor, mapper, emulator).

<https://developer.android.com/studio>



JETPACK COMPOSE

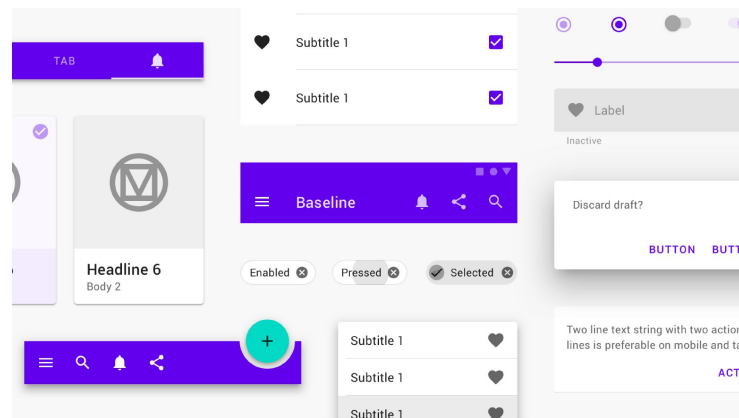
Et Android Studio rammeverk. En måte å lage apper på i Android Studio som gjør det enklere for oss som lager den. Det gjør at du kan lage appen din ved å fortelle den hva den skal gjøre, i stedet for å fortelle den hvordan den skal gjøre det. Basert på bla. et bibliotek med forhåndsdefinerte Composables.

<https://developer.android.com/jetpack/compose>



Komponenter / composables

- Komponenter er interaktive byggesteiner/deler
 - delene gjennomfører konkrete/små funksjoner
 - vi setter sammen løst koblete uavhengige komponenter til en helhet
- Compose tilbyr en rekke av disse
- Fordeler med komponentbasert utvikling:
 - forenkler utviklingen
 - akselererer utviklingen
 - mindre kode fordi gjenbruk av kode
 - skaper kontinuitet



LIVEKODING

Vi lager en handlelisteapp med:

- `NavController + NavHost`
- `remember`
- `ExposedDropDownMenuBox`

Oppsummering - livekoding og docs

- var enVariabel by remember { mutableStateOf(defaultVariabel) }
 - For å "huske" state mellom recomposition (at en composable blir laget på nytt)
 - <https://developer.android.com/jetpack/compose/state>
 - NavController styrer navigasjon mellom skjermer, NavHost renderer en skjerm basert på Route
 - <https://developer.android.com/jetpack/compose/navigation>
 - ExposedDropDownMenuBox - husk både TextField og ExposedDropDownMenu inni
 - [https://developer.android.com/reference/kotlin/androidx/compose/material3/package-summary#ExposedDropDownMenuBox\(kotlin.Boolean,kotlin.Function1,androidx.compose.ui.Modifier,kotlin.Function1\)](https://developer.android.com/reference/kotlin/androidx/compose/material3/package-summary#ExposedDropDownMenuBox(kotlin.Boolean,kotlin.Function1,androidx.compose.ui.Modifier,kotlin.Function1))
 - <https://m3.material.io/components/menus/overview>
-

Tips fra livekoding

- @Composables er bare gjenbrukbare blokker med kode
Bruk dokumentasjon "AppBar jetpack compose" inn i google og let til du finner frem
 - Sjekk imports med dokumentasjonen, og fjern imports som ikke brukes
 - NB! Sjekk at du har med riktige dependencies + husk da å **sync gradle!**
 - @Preview gjør **IKKE** at du kjører appen, men viser en "state" / en slags "skisse" av hvordan koden **ser** ut med de gitte variablene (ikke hvordan det faktisk fungerer)
 - Bruk heller log.cat enn printsetninger for feilsøking (lurt å lære seg og begynne med tidlig)
 - <https://developer.android.com/studio/debug/logcat>
-

Tentativ plan for neste uke

- Oppsummering oblig, oblighjelp hvis noen har utsatt
- Log cat gjennomgåing
- Evt. livekoding av en artig vitseapp :P (kortere denne gangen)

Husk at oblig 2 er stor, og at det er en grunn til at dere får lang tid på den. Begynn tidlig! ;)

Send oss en mail:
idakenene@uio.no
jamilakm@uio.no
jorgenao@uio.no
