

IN2000 V24 - Obligatorisk oppgave 1

Innleveringsfrist: Tirsdag 30. januar, kl. 23:59

Innledning

I denne obligatoriske oppgaven skal du lage en Android-app. Hensikten med denne oppgaven er å få deg i gang med verktøyene som senere skal benyttes i prosjektarbeidet. Android Studio skal benyttes som IDE, Kotlin som programmeringsspråk og Jetpack Compose til å lage brukergrensesnitt. Det er ingen krav til hvordan appen skal se ut utover det som er spesifisert i oppgaveteksten. Denne obligatoriske innleveringen skal løses individuelt, det vil si at du skal programmere alle deler av løsningen selv. Dette betyr ikke at vi fraråder samarbeid – tvert imot. Vi oppfordrer deg til å utveksle faglige erfaringer med andre.

For å få oppgaven godkjent må alle deloppgaver være gjort og appen skal kunne kjøres i en emulator med API-level 33 **uten å krasje**. Under retting vil en Resizable emulator eller tilsvarende med API level 33 benyttes. Hvis du tester på egen fysisk Android-enhet, vennligst spesifiserer Android-versjonen i kommentarfeltet på [Devilry](#).

Oppgaven

Oppgaven består av følgende tre deler:

1. Appens funksjonalitet
2. Komponenter til appen
3. Brukergrensesnitt

Levering

Trykk på “Build” → “Clean Project”, deretter skal mappen som inneholder Android Studio-prosjektet komprimeres til en zip/tar-fil, og leveres i [Devilry](#) innen fristen.

Nyttige ressurser

- [Jetpack Compose Tutorial](#)
- [Material3 komponenter for Compose](#)
- [Material3 Dropdown-meny](#)
- [Navigasjon i Compose](#)

Komme i gang

Sørg for at du har nyeste versjonen av Android Studio (Hedgehog) før du starter.

1. Opprett et nytt Android Studio-prosjekt, velg “Empty Activity”
2. “Name” settes til [**UiO-bruker**navn]_oblig1
3. “Package name” settes til **no.uio.ifl.in2000.[ditt uio-bruker**navn].oblig1
4. Velg API 24 som minimum SDK
5. Sørg for at det er “Kotlin DSL (build.gradle.kts) [Recommended]” som er valgt som språk for bygg-konfigurasjonen.

Del 1 - Appens funksjonalitet

I denne deloppgaven skal du implementere appens funksjonalitet ved bruk av programmeringsspråket Kotlin. Du skal implementere følgende to funksjoner:

1. `isPalindrome`, som sjekker om en tekststreng er et [palindrom](#) eller ikke.
2. `converter`, som konverterer mellom ulike enheter.

Disse to funksjonene er det laget tester for som spesifiserer forventet oppførsel. Testene finner du på [ressurssiden](#). Legg dem i test-mappen i prosjektet (samme mappe som fra før inneholder *ExampleUnitTest*). Før du begynner å lage funksjonalitetene kan du prøve testene. Alle skal først feile. Du er ferdig med del 1 når alle testene passerer.

Hvordan komme i gang:

- Les README.md i de [tilhørende testene](#).
- Du skal lage en funksjon som heter *isPalindrome*. Den skal ta inn en String *tekst* og returnere en Boolean. Denne skal ligge i kotlin-filen *Palindrome.kt*.

- Du skal også lage en funksjon som heter *converter*. Den skal ta inn en Int verdi og en ConverterUnits *enhet* og returnere en Int. Denne skal ligge i kotlin-filen Converter.kt

Krav til deloppgave:

- Funksjonene skal være implementert så alle de [tilhørende testene](#) passerer.
- Funksjonene skal ligge i hovedmappen i prosjektet (samme som MainActivity.kt) i kotlin-filer med navn Palindrome.kt og Converter.kt.
- Det er **ikke tillatt** å implementere funksjonaliteten med tradisjonelle løkker (f.eks. for/foreach og while). Du kan helt fint benytte deg av innebygde funksjoner i Kotlin.

Del 2 - Komponenter til appen

I denne deloppgaven skal du lage UI-komponentene til appen. Her skal du benytte deg av funksjonene du lagde i del 1. Du skal også lage Preview-funksjoner for å sjekke hvordan komponentene ser ut og fungerer i isolasjon.

Del 2.1 PalindromeChecker

Denne komponenten skal hete *PalindromeChecker*, og skal kort forklart:

1) ta input fra bruker, 2) sjekke om input er et palindrom (ved bruk av funksjonen fra forrige deloppgave) og 3) vise resultatet til brukeren.

Alle komponentene i PalindromeChecker skal være midtstilt og den skal inneholde følgende:

- Et [input-felt](#) der bruker kan taste inn tekst.
- En knapp for å starte palindromsjekken. Når knappen trykkes på eller “enter” i tastaturet, skal skjermtastaturet lukkes. Hint: se på [SoftwareKeyboardController](#).
- Når “enter” trykkes på skal også palindromsjekken gjøres
- En tekst som forteller om input er et palindrom eller ikke etter at knappen er trykket på.

Del 2.2 - Konverterer

I denne deloppgaven skal du lage composable-funksjonen `UnitConverter`. Her skal du benytte deg av funksjonaliteten du implementerte i del 1 for enhetskonvertering og lage en komponent der du kan konvertere fra fluid ounce (*fl oz*), cup (*cp*), gallon (*gal*) og hogshead til liter.

De fire ulike imperiske enhetene skal vises i en [dropdown-meny](#) slik at brukeren enkelt kan velge hvilken enhet det ønskes å konvertere fra.

Og skal ihvertfall inneholde:

- Et input-felt der bruker kan taste inn antall av enheten som skal konverteres.
- En dropdown-meny der du kan velge mellom de 4 enhetene
- Et tekstfelt som gir det konverterte resultatet

Krav til deloppgave:

- Ved inntasting i input-feltet skal brukeren få talltastatur og det skal kun være mulig å taste inn tall.
- Ved trykk på “Enter” i tastaturet skal skjermtastaturet lukke seg.

Del 3 - Skjermer og navigasjon

De to komponentene skal plasseres i hver sine skjermer `PalindromeScreen` og `UnitConverterScreen`. Vi skal kunne navigere mellom de to skjermene. For å navigere mellom skjermene skal du benytte deg av [compose sitt navigasjonsbibliotek](#).

Det skal settes opp 2 navigasjonsruter:

- “palindrome” skal assosieres med `PalindromeScreen()`-funksjonen
- “unitconverter” skal assosieres med `UnitConverterScreen()`-funksjonen

Hver skjerm skal inneholde komponentene midtstilt og i bunn av skjermen ha en knapp som lar deg gå til den andre skjermen. Når appen starter skal Palindrom-skjermen vises.

Krav til deloppgave:

- Komponentene og skjermene skal plasseres i hver sin fil `PalindromeScreen.kt` og `UnitConverterScreen.kt` i hver sin pakke `ui.palindrome` og `ui.unitconverter`.
- Innholdet i hver skjerm skal være sentrert på midten av skjermen.
- Knappen i bunn skal fylle hele bredden av skjermen.