

LABORATORIJSKA VJEŽBA BR. 5

Uvod u Jetpack Compose

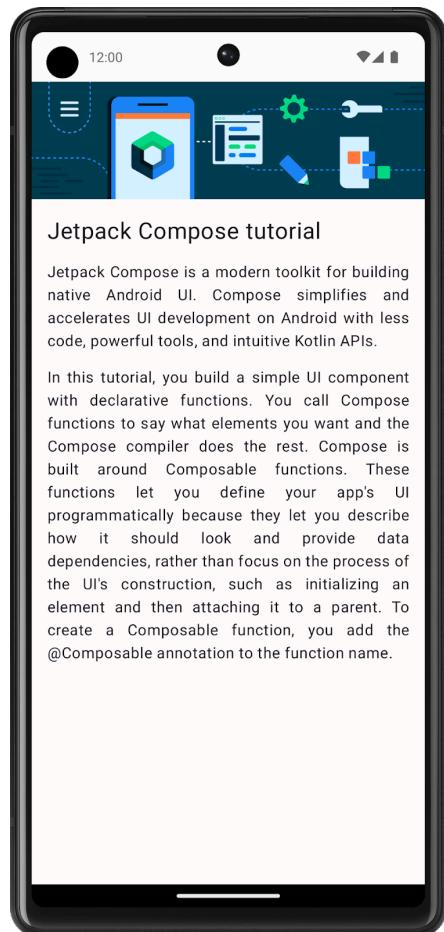
Cilj vježbe

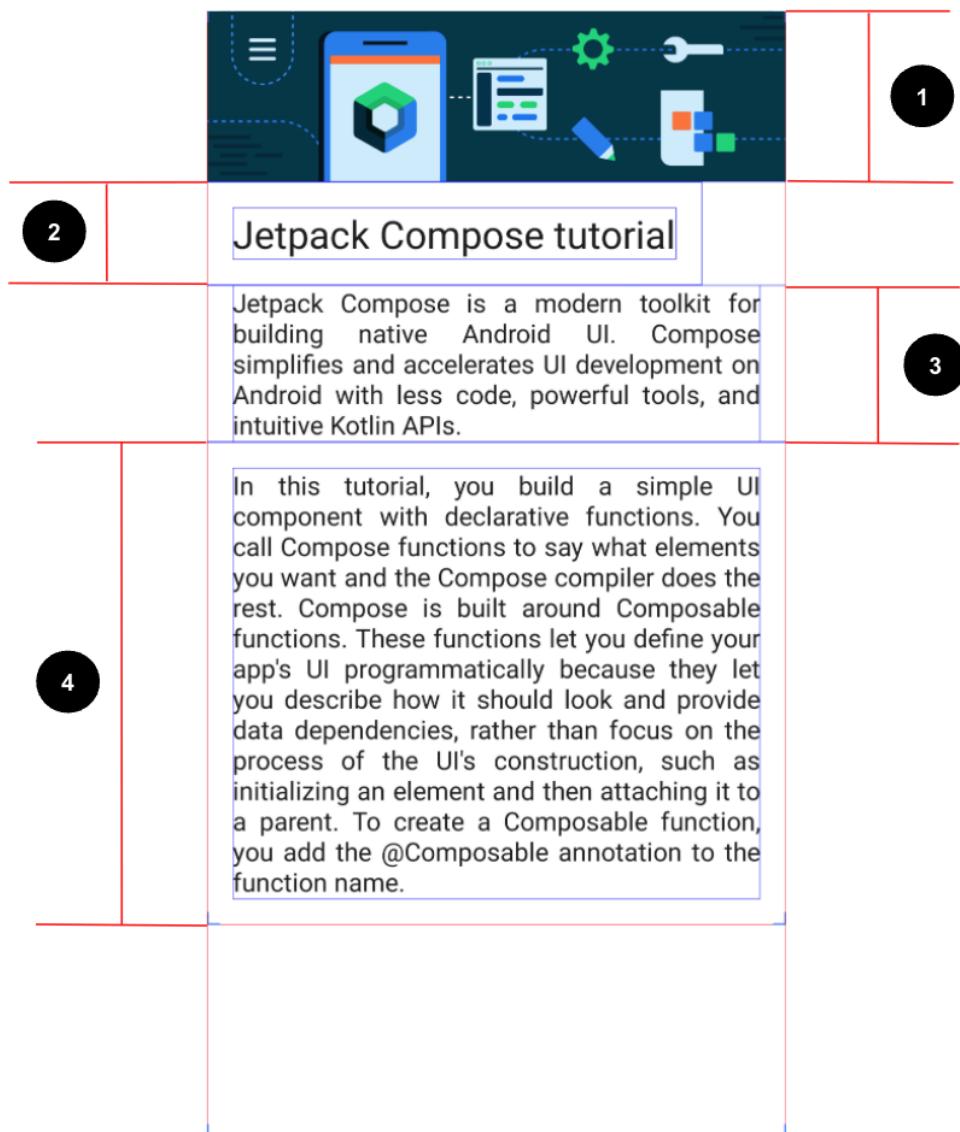
Cilj ove laboratorijske vježbe je upoznati studente sa osnovama Jetpack Compose-a, modernog Android alata za izradu korisničkog interfejsa. Compose pojednostavljuje i ubrzava razvoj Android aplikacija uz manje koda, intuitivne Kotlin API-je i moćne alate.

Tokom vježbe studenti analiziraju jednostavan primjer aplikacije koja prikazuje naslov, sliku i tekst, te kroz njega razumiju osnovne koncepte deklarativnog programiranja u Jetpack Compose-u.

Zadatak 1.

U ovoj laboratorijskoj vježbi studenti su dužni kreirati izgleda UI-a, kao što je prikazan na slici korištenjem osnovnih Composable funkcija Text i Image. Neophodno je koristiti jedan od layout-a (Row, Column ili Box) and uz pomoć modifikatora (Modifier) postaviti tražene zahtjeve za poravnanje, padding, veličinu teksta i dr.





Upustva

1. Postavi sliku tako da zauzima cijelu širinu ekrana.
2. Postavi prvi Text element da ima veličinu fonta 24sp i unutrašnje margine (padding) od 16dp sa svih strana (lijevo, desno, gore i dole).
3. Postavi drugi Text element sa default veličinom fonta, padding-om od 16dp (lijevo i desno) i poravnanjem teksta s obje strane (Justify).
4. Postavi treći Text element sa default veličinom fonta, padding-om od 16dp (lijevo, desno, gore i dole) i poravnanjem teksta s obje strane (Justify).

Ovo su vrijednosti stringova koje je potrebno staviti u strings.xml -> u folderu app/src/main/res/values..

Jetpack Compose tutorial

Jetpack Compose is a modern toolkit for building native Android UI. Compose simplifies and accelerates UI development on Android with less code, powerful tools, and intuitive Kotlin APIs.

In this tutorial, you build a simple UI component with declarative functions. You call Compose functions to say what elements you want and the Compose compiler does the rest. Compose is built around Composable functions. These functions let you define your app's UI programmatically because they let you describe how it should look and provide data dependencies, rather than focus on the process of the UI's construction, such as initializing an element and then attaching it to a parent. To create a Composable function, you add the @Composable annotation to the function name.

Pokušajte zadatak riješiti sami - rješenje možete naći na [GitHub repozitoriju](#) u folderu Labs/L5/P5ex1solution.

Slika se nalazi u prilogu.

Zadatak 2.

Napravite vašu vizit karticu kao na slici. Slobodno ubacite vašu sliku i podatke.

Napomene o različitim Composable elementima koje možete koristiti

- Row ili Column composables

Isprobajte različite vrijednosti parametara horizontalArrangement i verticalAlignment u Row i Column elementima kako biste postigli raspored koji odgovara željenom dizajnu.

- Image composable

Ne zaboravite popuniti parametar contentDescription. Ako se Image koristi samo u dekorativne svrhe, ili ako već postoji Text element koji opisuje sliku, možete postaviti contentDescription = null. Također možete prilagoditi veličinu slike pomoću modifikatora height i width u okviru parametra modifier.

- Icon composable

Možete koristiti Icon composable za dodavanje ikona iz [Material Design biblioteke](#). Parametrom [Tint](#) možete promijeniti boju ikone kako bi odgovarala stilu vaše aplikacije ili vizitke. Slično kao i kod Image elementa, ne zaboravite popuniti contentDescription parametar.

- Text composable

Eksperimentišite s različitim vrijednostima parametara fontSize, textAlign, color i fontWeight kako biste postigli željeni izgled i stil teksta.

- Razmak i poravnanje

Pomoću Modifier argumenata, kao što su padding i weight, možete kontrolisati raspored elemenata u interfejsu. Također možete koristiti Spacer composable kako biste eksplicitno dodali razmake između elemenata.



- Prilagođavanje boja

Možete koristiti prilagođene boje uz pomoć klase Color i heksadecimalnog koda (RGB zapis boje u obliku hexa). Na primjer, zelena boja Androida ima heksadecimalni kod #3DDC84. Svoj tekst možete obojiti tom nijansom pomoću sljedećeg koda:

```
color = Color(0xFF3DDC84)
```