

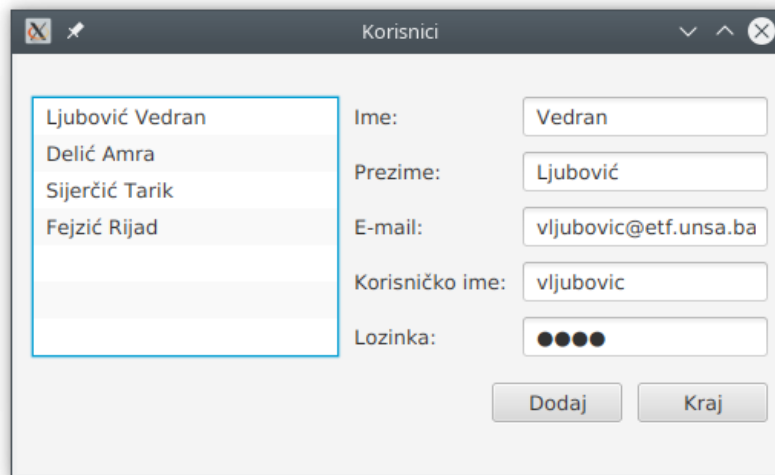
Tutorijal 7

Priprema za tutorijal

Prihvatite sljedeću pozivnicu koja sadrži prazan repozitorij:

<https://classroom.github.com/a/DxoA9cGX>

Na ovaj repozitorij trebate postaviti novi JavaFX projekat koji ćete kreirati. Projekat treba da sadrži formu za unos i izmjenu korisnika kao na slici:



U lijevom dijelu forme nalazi se ListView sa spiskom korisnika, desno se nalaze polja formulara: ime, prezime, email, korisničko ime i lozinka. Klikom na korisnika u listi, polja forme se popunjavaju podacima tog korisnika. Dok mijenjate podatke u poljima forme, oni se automatski odražavaju na listu. U dnu se nalazi ButtonBar sa dva dugmeta: dugme **Dodaj** kreira novog korisnika sa svim poljima praznim i otvara ga tako da možete popuniti ime, prezime itd. Dugme **Kraj** završava program. Implementirajte funkcionalnost za dugme Kraj (Dodaj će biti urađeno kasnije).

Zadatak 1

Kreirajte klasu Korisnik koja predstavlja JavaFX Bean i sadrži sva polja potrebna da se opiše jedan korisnik za aplikaciju iznad. (Podsjetimo se, to znači da treba koristiti polja tipa Property npr. SimpleStringProperty umjesto klase String). Kreirajte i gettere/settere.

Zadatak 2

Kreirajte klasu KorisniciModel koja predstavlja kompletan model za aplikaciju iznad:

- Klasa treba sadržavati ObservableArrayList korisnika, te atribut za trenutno izabranog korisnika koji treba biti tipa SimpleObjectProperty.
- Za oba ova atributa treba kreirati gettere/settere.
- Dodajte i metodu **napuni** koja će popuniti listu navedenim podacima. Radi lakšeg rada, poželjno je da u klasu Korisnik dodate konstruktor koji prima String-ove umjesto SimpleStringProperty-a.

Zadatak 3

U klasi Controller dodajte metodu:

```
@FXML
public void initialize() {
}
```

te u ovoj metodi izvršite dvosmjerno povezivanje svakog od polja sa odgovarajućim property-em trenutnog korisnika, npr:

```
fldIme.textProperty().bindBidirectional(model.getTrenutniKorisnik().imeProperty());
```

Pored toga izvršite povezivanje ListView kontrole sa spiskom korisnika, npr:

```
listKorisnici.setItems(model.getKorisnici());
```

Primijetićete da kada pokrenete program, u listi imate neke čudne stringove. To možete popraviti tako što u klasi Korisnik dodate toString metodu.

```
@Override
public String toString() {
    ...
}
```

Konačno, klikom na neku opciju u listi treba da se promijeni trenutni korisnik. Potrebno je da dodamo listener koji mijenja izabranog korisnika (kada počnete kucati ponudiće vam se ispravne opcije, pa samo potvrdite izbor):

```
listKorisnici.getSelectionModel().selectedItemProperty().addListener((obs,
oldKorisnik, newKorisnik) -> {
    model.setTrenutniKorisnik(newKorisnik);
    listKorisnici.refresh();
});
```

Napravite i akciju za dugme Dodaj. Dugme Dodaj treba pozivom metode klase KorisniciModel da doda novog korisnika u model čiji su svi podaci prazni te da postavi tog korisnika za trenutnog korisnika, što će izvesti sve ostale potrebno promjene.

Zadatak 4

U vaše rješenje Tutorijala 7 dodajte JUnit 5 testove koji testiraju model klase. Ovako modificiran projekat pošaljite nazad na GitHub u istom repozitoriju.

Dodajte i TestFX testove koji testiraju korisnički interfejs. I ove promjene postavite u isti repozitorij na GitHubu.