

# Tutorijal 10

## Priprema za tutorijal

Potrebno je da završite sve zadatke koji se tiču tutorijala 9 na repozitoriju koji ste dobili za tutorijal 9. Rok za dovršetak zadataka je 5 dana od dana kada ste imali tutorijal (do kraja navedenog dana u ponoć), npr. grupe koje su imale tutorijal u ponedjeljak imaju rok do subote u 23:59 da dovrše tutorijal.

Šestog dana će biti kreirana kopija ovog repozitorija pod nazivom:

<https://github.com/RPR-2019/tutorijal-10-vašekorističkoime>

Klonirajte ovaj repozitorij na vaš računar i uradite zadatak opisan ispod. Napominjem da ako ste dovršili tutorijal 9 ranije, možete u istom projektu nastaviti raditi pripremu za tutorijal 10, a kada bude kreiran repozitorij kopirajte to što ste uradili u novi projekat (nemojte ipak mijenjati remote jer će na novom repozitoriju biti dodati i novi testovi za tutorijal 10).

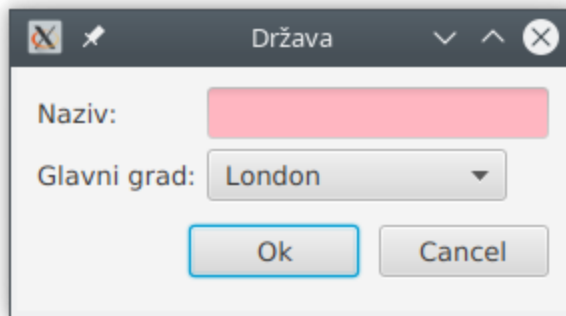
### Gradovi svijeta

ID	Naziv	Stanovnika	Država
2	London	8825000	Velika Britanija
1	Pariz	2206488	Francuska
3	Beč	1899055	Austrija
4	Manchester	545500	Velika Britanija
5	Graz	280200	Austrija

Kreirajte grafički korisnički interfejs sa sljedećom specifikacijom:

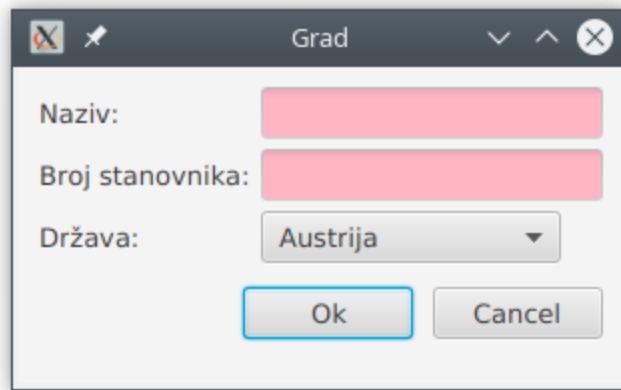
- Forma **glavna.fxml**. Ova forma treba biti otvorena prilikom pokretanja programa, što znači da postojeću Main klasu trebate izbaciti. Forma sadrži:
  - TableView (fx:id "tableViewGradovi") sa kolonama: ID (colGradId), Naziv (colGradNaziv), Stanovnika (colGradStanovnika), Država (colGradDrzava).

- U desnom dijelu prozora treba se nalaziti vertikalni red dugmadi: "Dodaj grad" (btnDodajGrad), "Dodaj državu" (btnDodajDrzavu), "Izmijeni grad" (btnIzmijeniGrad), "Obriši grad" (btnObrisiGrad)
- Prilikom promjene veličine prozora, treba se mijenjati veličina TableView dok kolona sa dugmadima treba biti široka fiksno 120 piksela. Slika na prethodnoj stranici
- Forma **drzava.fxml** treba se otvoriti klikom na dugme "Dodaj državu". Ovo treba biti standardni formular za unos podataka o državi sa sljedećim poljima:
  - TextField "Naziv" (fx:id fieldNaziv)
  - ChoiceBox "Glavni grad" (fx:id choiceGrad)
  - ButtonBar u kojem se nalaze dva dugmeta "Ok" i "Cancel" (btnOk i btnCancel). btnCancel samo zatvara prozor bez ikakvih izmjena, a btnOk treba da validira polje fieldNaziv. Jedini uslov validnosti je da polje ne smije biti prazno. Ako je polje prazno, treba postaviti pozadinu boje "lightpink", a ako nije boje "greenyellow" koristeći CSS kako smo radili na predavanjima. Slika:



- Forma **grad.fxml** treba da služi za dodavanje novog grada i izmjenu postojećeg, dakle treba da se otvori klikom na dugme "Dodaj grad" i "Izmijeni grad" (ali za sada neka forma bude prazna u oba slučaja). Također standardni formular za unos podataka sa poljima:
  - TextField "Naziv" (fx:id fieldNaziv)
  - TextField "Broj stanovnika" (fx:id fieldBrojStanovnika)
  - ChoiceBox "Država" (fx:id choiceDrzava)
  - ButtonBar u kojem se nalaze dva dugmeta "Ok" i "Cancel" (btnOk i btnCancel). Sve je kao i kod države, s tim što btnOk pored fieldNaziv treba da validira i fieldBrojStanovnika. Uslov za validnost ovog polja je da je unesen pozitivan cijeli broj (integer). Slika na sljedećoj stranici.

Naglašavam još jednom da, pored validacija i otvaranja novih formi klikom na dugme, grafički korisnički interfejs za pripremu ne treba da radi ništa! Potrebno je kreirati klase GlavnaController, DrzavaController i GradController. Obavezno koristiti propisnu organizaciju koda za JavaFX aplikacije opisanu na predavanjima (koristeći resources folder). U skladu sa *dependency injection* principom, pobrišite vezu između fxml datoteke i kontrolerske klase kada završite pripremu.



Radi izvršavanja testova, u klasu GlavnaController trebate dodati sljedeću metodu koju smo spominjali na predavanjima:

```
// Metoda za potrebe testova, vraća bazu u polazno stanje
public void resetujBazu() {
    GeografijaDAO.removeInstance();
    File dbfile = new File("baza.db");
    dbfile.delete();
    dao = GeografijaDAO.getInstance();
}
```

## Zadatak 1

Postaviti sadržaj kontrole tableViewGradovi na spisak gradova koji se nalazi u bazi (koji vraća metoda gradovi() klase GeografijaDAO).

## Zadatak 2

Dovršiti formu **grad.fxml** - dodavanje novog grada. choiceDrzava treba biti popunjen svim državama koje su trenutno registrovane u bazi podataka. Kada korisnik klikne na btnOk, ako je grad validan, treba biti dodat u bazu podataka.

Radi *separation of concerns*, poželjno je da se GradController ne bavi bazom tj. da ne pristupa instanci DAO klase.

*Savjet:* konstruktor GradController može primati spisak država. Možete dodati metodu getGrad koja vraća podatke o gradu ako je korisnik kliknuo btnOk, a null ako je kliknuo btnCancel. Zatim pozovite ovu metodu iz GlavnaController koristeći metodu Stage.setOnHiding koja vam omogućuje da zadate lambda funkciju koja se izvršava u trenutku kada se određena forma zatvori.

### Zadatak 3

Dovršiti izmjenu postojećeg grada. Kada korisnik klikne na dugme "Izmijeni grad", treba se ne samo otvoriti grad.fxml nego i sva polja trebaju biti popunjena podacima za trenutno izabrani grad. Pazite da btnOk sada neće izazvati dodavanje novog grada u bazu nego izmjenu postojećeg.

*Savjet:* Još jedan parametar konstruktoru klase GradController - prima grad koji se edituje ili null ako je u pitanju dodavanje novog grada.

### Zadatak 4

Dovršite dodavanje nove države klikom na dugme "Dodaj državu". choiceGrad treba biti popunjen svim gradovima koji su trenutno registrovane u bazi podataka. Kada korisnik klikne na btnOk, ako je država validna, treba je dodati u bazu podataka. Sve napomene važe kao za zadatak 2. Za rješenje ovog zadatka biće potrebno da u DAO klasu dodamo metodu **dodajDrzavu** koju ranije nismo imali.

### Zadatak 5

Dovršite brisanje grada klikom na dugme "Obriši grad". Najprije treba prikazati Alert tipa CONFIRMATION koji pita korisnika da li je siguran. Ako korisnik potvrdi, treba obrisati grad iz baze. Za ove potrebe, u DAO klasu dodajte metodu **obrisiGrad**.

*Izmjene:*

-