

Završni ispit - praktični dio

Pozivnica za GitHub Classroom: <https://classroom.github.com/a/sP1FsOZA>

Ukupno bodova: 20 (19 bodova će se dodijeliti proporcionalno broju uspješnih testova, pri čemu se početnih 15 testova ne računaju, ali će se oduzimati bodovi ako ti testovi budu padali. Ostalih 1 bod će se dodijeliti ručnim pregledanjem.) ~~Studenti predmeta Razvoj softvera rade i zadatak 2 koji nosi dodatnih 10 bodova (5 na osnovu testova a 5 ručnim pregledanjem).~~ Svi rade sve.

Na repozitoriju se nalazi gotov projekat koji predstavlja tutorijal 9 (RPR), odnosno tutorijal 8 (Razvoj softvera), sa Java FX korisničkim interfejsom. Projekat sadrži i 15 testova koji rade (služe za provjeru regresije), te 19 testova u klasama koje se zovu Ispit* koje su zakomentarisane, a koji se tiču zadataka 1 i 2 navedenih ispod. Kada uradite zadatak trebate odkomentarisati klase kako bi testovi prošli. Na poledini ovog ispita imate detaljniji opis programa koji bi vam mogao pomoći da se snađete u kodu.

Zadatak 1 (9 bodova):

Na formular za unos grada dodati tekstualno polje sa oznakom "Regija" (fx:id **fieldRegija**) koje omogućuje korisniku da unese naziv regije kojoj pripada grad. Kako postojeći testovi ne bi padali, treba biti dozvoljeno da se unese prazan naziv regije. Napravite sve potrebne prepravke u bazi podataka i pripadajućim klasama kako bi ovaj podatak bio trajno zapisan.

Zadatak 2 (10 bodova):

Ispod navedenog tekstualnog polja dodati dugme "Referendum" (fx:id **btnReferendum**). Ovo dugme ima isti efekat kao dugme Ok s tom razlikom što, ako polje regija nije prazno, unesena regija postaje država, a njen glavni grad je trenutni grad, te treba napraviti sve prepravke kako bi se to desilo (kreirati novu državu, pridružiti joj sve gradove koji imaju isto ime regije itd.) a ako regija nije unesena prikazati prozor sa greškom (Alert tipa ERROR).

Pridržavanje pravila dobrog dizajna pri rješavanju zadataka 1 i 2 nosi ukupno 1 bod (za oba zadatka!) što će biti ručno pregledano.

Sarajevo, 9. 9. 2021

Vedran Zuborić

Opis projekta

SQLite baza podataka **baza.db** (nalazi se u korijenskom direktoriju projekta) sadrži dvije tabele:

- Tabela grad sadrži kolone: id (int, primarni ključ), naziv (text), broj_stanovnika (int), drzava (int, strani ključ)
- Tabela drzava sadrži kolone: id (int, primarni ključ), naziv (text), glavni_grad (int, strani ključ)

Dump baze nalazi se u datoteci **baza.db.sql**.

Na osnovu ove dvije tabele formirane su DTO klase **Grad** i **Drzava** koje se pridržavaju JavaBean specifikacije, sadrže sve pobrojane attribute, gettere, settere, konstruktor bez parametara i konstruktor sa svim atributima kao parametrima. Atribut drzava u klasi Grad je tipa Drzava, a atribut glavniGrad u klasi Drzava je tipa Grad (reference na klase).

Klasa **GeografijaDAO** predstavlja tipičnu DAO klasu, ova klasa je singleton što znači da je konstruktor privatan, a metodom getInstance se dobija referenca na ovu klasu. getInstance poziva konstruktor. Konstruktor klase najprije pokušava izvršiti jedan upit da ustanovi da li datoteka baza.db postoji. Ako ne postoji, poziva se metode regenerisiBazu() koja kreira novu bazu iz dump datoteke, izvršavajući upite koji se u njoj nalaze. Zatim se u konstruktoru kreiraju svi pripremljeni upiti koji su potrebni za ostatak klase.

Pored ovih, klasa GeografijaDAO sadrži metode iz tutorijala 9:

- Grad glavniGrad(String nazivDrzave) - vraća null ako država ne postoji
- void obrisiDrzavu(String nazivDrzave) - briše i sve gradove u toj državi
- ArrayList<Grad> gradovi() - vraća spisak gradova sortiranih po broju stanovnika u opadajućem redoslijedu
- void dodajGrad(Grad grad) i void dodajDrzavu(Drzava drzava)
- void izmijeniGrad(Grad grad) - ažurira slog u bazi za dati grad
- Drzava nadjiDrzavu(String nazivDrzave) - vraća null ako država ne postoji

Kao i sljedeće metode kojih nema u tutorijalu 9:

- Grad nadjiGrad(String nazivGrada)
- void obrisiGrad(Grad grad)
- ArrayList<Drzava> drzave()
- void vratiBazuNaDefault() - služi da bi se baza podataka vratila na stanje koje je dato u datoteci baza.db.sql (zbog testova koji mijenjaju sadržaj baze).

U folderu **resources/fxml** nalaze se sljedeći prozori:

- **glavna.fxml** sadrži TableView (tableViewGradovi) sa spiskom gradova, te četiri dugmeta za dodavanje, promjenu, brisanje grada i dodavanje države:
 - GlavnaController je kontroler za ovu formu. Ona popunjava tableViewGradovi, te sadrži action* evente za klik na četiri dugmeta.
 - actionDodajGrad, actionIzmijeniGrad i actionDodajDrzavu otvaraju forme grad.fxml i drzava.fxml. Kod zatvaranja forme izvršava se lambda (setOnHiding) koja poziva metodu getGrad / getDrzava kontrolera te poziva metodu dodajGrad / izmijeniGrad / dodajDrzavu u DAO klasi (ako nije vraćen null).
 - actionObrisiGrad prikazuje Alert tipa Confirmation, te ako je korisnik kliknuo na OK poziva se metoda obrisiGrad iz DAO klase.

- **grad.fxml** i **drzava.fxml** su forme za unos novog grada i države / promjenu postojećih, sadrže polja koja odgovaraju atributima klasa Grad i Drzava, te dva dugmeta Ok i Cancel
 - GradController i DrzavaController imaju konstruktor sa dva parametra, prvi je Grad ili Drzava a drugi je GeografijaDAO. Ako je prvi parametar null, onda je u pitanju dodavanje novog, a ako nije onda je izmjena. DAO služi isključivo kako bi se padajuća lista choiceGrad/choiceDrzava popunila svim gradovima/državama iz baze, što se obavlja u metodi initialize(). Ova metoda također popunjava formu u slučaju izmjene.
 - Metoda clickCancel zatvara prozor i postavlja atribut grad na null.
 - clickOk vrši validaciju forme, dodaje CSS klasu poljeIspravno ili poljeNeispravno, te ako su sva polja ispravna zatvara formu. Ova metoda *ne dodaje* ništa u bazu. Umjesto toga, ona ažurira privatni atribut grad / drzava. GlavnaController će pozvati metodu getGrad / getDrzava kako bi dobili vrijednost tog privatnog atributa. U slučaju cancel privatni atribut će biti postavljen na null.