

Popravni parcijalni ispit

Pozivnica za GitHub Classroom: https://classroom.github.com/a/TfW_vyoR

Link za Google Meet sobu:

- <https://meet.google.com/eqb-mwzu-gfn> - studenti čije prezime počinje slovom A-K
- <https://meet.google.com/vjp-dswb-sug> - studenti čije prezime počinje slovom L-Ž

Ukupno bodova: 30 (Bodovi 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.)

Na repozitoriju se nalazi gotov projekat koji predstavlja tutorijal 9 sa Java FX korisničkim interfejsom. Projekat sadrži i 15 testova koji rade (služe za provjeru regresije), te 25 testova u klasama koje se zovu Ispit* koje su zakomentarisane, a koji se tiču zadatka navedenog 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:

Potrebno je na glavnu formu dodati dugme “Izmijeni državu” (**btnIzmijeniDrzavu**) koje omogućuje da se izmijene podaci o državi u kojoj se nalazi trenutno selektovani grad. Da bi to bilo moguće, dodajte u DAO klasu metodu **izmijeniDrzavu** koja radi slično kao postojeća metoda izmijeniGrad.

Zadatak 2:

Na formu za izmjenu države dodajte padajuću listu “Uređenje” (**choiceUredjenje**) sa opcijama: “Republika”, “Kraljevina”, “Federacija” i “Diktatura”. Dodajte i odgovarajuće polje u klasu Drzava (obavezno **enum UREDJENJE**) te izmijenite sve potrebne klase kako bi podešavanje uređenja radilo i bilo sačuvano u bazi podataka. Pored konstruktora klase Drzava koji prima tri parametra, treba dodati i konstruktor koji ima i četvrti parametar (uređenje), a podrazumijevana vrijednost za uređenje je Republika.

Zadatak 3:

Na glavnu formu dodajte dugmad “Zapiši JSON” (**btnZapisiJSON**) i “Učitaj JSON” (**btnUcitajJSON**). Ova dugmad trebaju otvoriti standardne open/save dijaloge za odabir datoteke, a zatim zapisati odnosno učitati JSON datoteku u formatu koji je opisan na sljedećoj stranici. U slučaju neispravno formatirane datoteke, dugme “Učitaj JSON” treba prikazati Alert tipa ERROR.

Za potrebe realizacije ovog zadatka kreirajte klasu **JSONFormat** koja ima: privatne attribute **ArrayList<Drzava> drzave** i **ArrayList<Grad> gradovi** sa setterima i getterima, metodu **ucitaj** koja prima **File** i iz datoteke puni podatke u privatne attribute, te metodu **zapisi** koja prima **File** i u datoteku upisuje podatke iz privatnih atributa. Metoda ucitaj treba baciti izuzetak za sve vrste neispravnih JSON datoteka, uključujući situacije kada neko od polja označenih kao **obavezno** ne postoji u datoteci.

Zadatak 3 nosi 10 bodova, od čega će 5 biti dodijeljeno pomoću testova u klasi IspitJSONTest, a 5 ručnim pregledanjem glavnog prozora, tražene dugmadi i pratećih akcija.

Sarajevo, 10. 7. 2020

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 privatn, 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 nadjDrzavu(String nazivDrzave) - vraća null ako država ne postoji
- void removeInstance() - služi da bi se prekinula konekcija na bazu, kako bi se fajl baza.db mogao obrisati (što se koristi u testovima).

Kao i sljedeće metode kojih nema u tutorijalu 9:

- Grad nadjiGrad(String nazivGrada)
- void obrisiGrad(Grad grad)
- ArrayList<Drzava> drzave()

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.
 - Metoda resetujBazu() se poziva iz testova kako bi baza podataka bila regenerisana.
- **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.

Primjer JSON datoteke

```
[
  {
    "gradovi": [
      {
        "brojStanovnika":280200,
        "naziv":"Graz"
      },
      {
        "brojStanovnika":1899055,
        "glavni":true,
        "naziv":"Beč"
      }
    ],
    "naziv":"Austrija",
```

```
        "uredjenje":2
    }
]
```

Osnovni element JSON datoteke je niz država. Elementi tog niza su objekti za koje se zna da predstavljaju države. Obavezni atributi ovog objekta su **naziv** (string koji predstavlja naziv države), **uredjenje** (cijeli broj čije značenje određujete sami, a ako niste uradili zadatak 2 možete ga ignorisati) i **gradovi** (niz objekata tipa Grad).

Objekti tipa Grad pak imaju obavezne attribute **naziv** i **brojStanovnika**. Pored toga, postoji neobavezni atribut **glavni** tipa boolean, koji ako ima vrijednost true, onda je dati grad glavni grad svoje države. Svaka država treba imati tačno jedan glavni grad.