

Elektrotehnički fakultet, Sarajevo

Razvoj softvera

Izvještaj

Aplikacija za elektronske ispite i kvizove

Student: Elma Trako

Predmetni nastavnik: Doc. dr Vedran Ljubović

Sarajevo, 14. 07. 2021.

SADRŽAJ

1. Opis aplikacije.....	3
2. Osnovne ideje pri implementaciji.....	4
3. Implementacija.....	5
3.1. Predavanja prve i druge sedmice.....	5
3.2. Grafički korisnički interfejs.....	5
3.3. Objektno zasnovano programiranje u Javi.....	6
3.4. Dizajn korisničkog interfja.....	6
3.5. Java kolekcije.....	7
3.6. Unit testovi u Javi, testiranje JavaFX aplikacija koristeći TestFX.....	7
3.7. MVC s JavaFX.....	7
3.8. Rad s bazom podataka.....	7
3.9. Rad s datotekama.....	7
3.10. Maven.....	8
3.11. Napredne teme objektno-orjentisanog programiranja.....	8
3.12. Objektno orjentisani dizajn.....	8
3.13. Kvaliteta koda i refactoring.....	8
4. Prilike za unaprjeđivanje.....	9

1. Opis aplikacije

Aplikacija koja se opisuje u nastavku predstavlja ograničenu simulaciju aplikacije za online ispite i kvizove. Aplikacija sadrži osnovne funkcionalnosti i predstavlja jako dobru podlogu za unaprjeđivanje iste u budućnosti. Zbog šire upotrebe, korišten je engleski jezik. Mogućnost promjene jezika i na bosanski jezik je još uvijek u izradi. Na samom početku, imate mogućnost da učitajte file s pitanjima za kviz. Podržava maksimalno sedam pitanja, a primjer kako bi taj file trebao da izgleda – priložen je uz samu aplikaciju. Nakon što učitajte kviz, on počinje te na svako sljedeće pitanje prelazite klikom na dugme Next. Tokom izrade, imate obavještenje gdje ste i koliko Vam je ostalo do kraja kviza. Nakon završetka, pojavljuje se poruka koja Vas obavještava koliko ste osvojili bodova te imate mogućnost da se vratite na pogrešno odgovorena pitanja, da se vratite na početak ili da ponovo učitajte novi kviz. Pored ovih mogućnosti, menu Vam nudi opciju spašavanja rezultata. Klikom na ovu opciju, otvara Vam se login forma te nakon prijave imate uvid u bazu podataka, odnosno u podatke već unesenih osoba koje su radile kviz te možete dodati ili obrisati osobu.

Kako pisati pitanja u file koji ćemo učitati?

Primjer file-a koji ćete učitati će Vam biti priložen uz samu aplikaciju, ali bitno je naglasiti da je jako jednostavno kreirati isti. Sastoji se od pitanja i navedenih potencijalnih odgovora. Tačan odgovor uvijek pišete kao prvi u nizu, a prikaz istih će biti izmještan prilikom pokretanja kviza. Dakle, tačni odgovori osobi koja radi kviz neće biti prikazani uvijek kao prvi.

Kako zapravo radi spašavanje rezultata?

Spašavanje rezultata je dinamična opcija, odnosno može i ne mora imati ograničenja. Po defaultu, da biste pristupili bazi podataka i dobili mogućnost dodavanja ili brisanja osoba, potrebno Vam je korisničko ime i lozinka.

- username: student
- password: student2020

Međutim, možete ukloniti ograničenja pri čemu svaki student normalno može unositi svoje rezultate. Možete i razdvojiti profesor-student način prijave, sve u zavisnosti šta tačno trebate.

2. Osnovne ideje pri implementaciji

Pri implementaciji korištena je SQLite baza podataka jer je bila zadovoljavajuća za projekat ovih razmjera. Razvijene su osnovne funkcionalnosti za rad sa bazom podataka, odnosno iskorištena je za pamćenje studenata koji su radili kviz. Postoji posebna klasa koja sadrži metode i upite koji omogućavaju rad sa bazom podataka i koja je neovisna od ostatka, dakle, postignuto je odvajanje baze podataka od poslovne logike aplikacije te omogućava nadogradnju ili promjenu baze podataka bez da se cijela aplikacija mora mijenjati. Aplikacije je jako jednostavna, logična i sadrži osnovne funkcionalnosti. Sve forme su jako jednostavne za ispunjavanje i ukazuju Vam na greške što je omogućeno validacijom polja. Uz aplikaciju je class dijagram radi lakšeg razumijevanja aplikacije kao i ERD baze podataka.

3. Implementacija

U nastavku slijede opisi dijelova aplikacije prema predavanjima na predmetu Razvoj softvera.

3.1. Predavanja prve i druge sedmice

Projekat je rađen u razvojnom okruženju JetBrains IntelliJ IDEa. Korišten je programski jezik Java te GitHub kao server na kojem je glavna (centralna) verzija projekta. Commit je atomičan, a svaka commit message (poruka) objašnjava šta je upravo urađeno. Za pokretanje:

```
C:\Program Files\Java\javafx-sdk-15.0.1\lib
```

```
--module-path "C:\Program Files\Java\javafx-sdk-15.0.1\lib" --add -  
modules=javafx.controls,javafx.fxml
```

3.2. Grafički korisnički interfejs

Korištena je JavaFX biblioteka koja je namijenjena za web i mobilne aplikacije. Svaki FXML file je povezan sa Controller klasom koja sadrži handlera za događaje. Preciznije, povezanost je sljedeća:

- About.fxml i AboutController
- EditPerson.fxml i EditPersonController
- LoadView.fxml i LoadViewController
- Login.fxml i LoginController
- Menu.fxml i MenuController
- PersonList.fxml i PersonController
- QuizView.fxml i QuizViewController

Postoji još i Controller klasa koja je povezana s QuizModel klasom.

Kada su u pitanju dostupni rasporedi (layouts), korišteni su sljedeći: HBox, VBox, GridPane, BorderPane, AnchorPane. Izvršeno je stiliziranje pomoću CSS-a, tj. style.css file je po potrebi uključen u određene fxml file-ove. Izvršena je i organizacija koda za resurse što znači da je u kodu jedan folder označen kao folder za resurse (resources). Unutar ovog direktorija kreirani su poddirektoriji css, fxml i img. Omogućeno je upravljanje događajima i poljima pomoću `onAction` i `fx:id` atributa. Praćenje izmjene kontrola je postignuto pomoću `Listenera` i metode `initialize`. Pojedine klase u kojima je potrebno preuzimati kontrolu s forme posjeduju

properties binding i bidirectional binding. Također, većinski dio klasa preko metoda omogućava otvaranje novog prozora.

Aplikacija slijedi koncepte dobrog dizajna korisničkog interfejsa gdje su svi logički vezani elementi grupisani i na ekranu.

3.3. Objektno zasnovano programiranje u Javi

Svaka klasa je u zasebnom .java fajlu te za imenovanje istih je korištena CamelCase notacija.

Kompletan program je u istom paketu: ba.unsa.etf.rs. Klasa koja se koristi za pokretanje čitavog programa je Main klasa. Klase koje projekat sadrži su sljedeće:

- Main
- Controller
- Quiz
- QuizModel
- FileImport
- LoadViewController
- QuizViewController
- MenuController
- LoginController
- MyRuntimeException
- PersonController
- Person
- Student
- Professor
- PersonDAO
- AboutController
- EditPersonController

Klase posjeduju osnovne attribute, konstruktor i potrebne metode. Definisana je vlastita klasa izuzetaka, tj. klasa MyRuntimeException izvedena iz klase RuntimeException.

3.4. Dizajn korisničkog interfejsa

Početni, a ujedno i glavni prozor aplikacija sadrži glavni meni koji se sastoji od tri stavke:

- File: Restart quiz – vraća kviz na početak, Open new quiz – otvara novi kviz, Save result – mogućnost spašavanja rezultata kviza, Quit - izlaz
- Edit: Copy - kopiraj, Paste - zalijepi, Language – promjena jezika
- Help: About – pomoć za rad

Aplikacija također posjeduje i pomoćni meni u stavki Save result, koji se sastoji od također tri stavke:

- File: Exit - izlaz
- Edit: Add – dodavanje nove osobe, Delete – brisanje postojeće osobe
- Help: About – pomoć za rad

Prisutne su i meni-kratice tastature koje su navedene pored svake stavke. Elementi koji su još zastupljeni su: Label, ImageView, ButtonBar, Button, TextField, PasswordField, TableView, ProgressBar, Spinner, DatePicker. Spinner i DatePicker su kontrole koje su omogućile da se ne dozvoli pogrešan unos, tj. Spinner za numeričke vrijednosti i DatePicker za unos datuma. Pored ovoga, izvršena je i validacija forme pomoću metode changed kao i napravljenih funkcija.

3.5. Java kolekcije

Zastupljena je upotreba osnovnih Java kolekcija, nizova i kolekcije ArrayList() koje su bile dovoljne za potrebe ovog projekta.

3.6. Unit testovi u Javi, testiranje JavaFX aplikacija koristeći TestFX

Aplikacija je pokrivena testovima.

org.testfx:testfx-junit5:4.0.15-alpha

3.7. MVC sa JavaFX

Aplikacija sadrži JavaBeans podatkovnu klasu Person, koja posjeduje atribute, settere, gettere i konstruktore te slijedi sva ostala pravila poput toString metode, povezivanja itd.

3.8. Rad sa bazom podataka

Korištena je SQLite baza podataka „person.db“. Sastoji se od dvije tabele „person“ i „login“. Kolone su tipa text, integer ili date. Nakon što korisnik završi kviz, ima mogućnost spašavanja

rezultata kviza i upravo taj dio je povezan s bazom – novi korisnik kojeg dodajemo ostaje spašen u bazi. Povezana je s klasom PersonDAO koja se sastoji od upita i metoda koje omogućavaju rad poput getInstance, deleteInstance, celarAll i ostalih metoda koje omogućavaju povezanost s bazom kao i dodavanje ili brisanje osoba.

3.9. Rad sa datotekama

Ova tema je povezana s klasom FileImport. Aplikacija omogućava da učitajte file s pitanjima koja će se prikazati na kvizu, a primjer izgleda iste će biti priložen uz ovaj izvještaj. Klasa FileImport omogućava to učitavanje file-a s diska i bitno je napomenuti da program podržava učitavanje maksimalno 7 pitanja. Klasa se sastoji od metode koja prikazuje dijalog za učitavanje i selektovanje file-a te od metode koja čita iz njega.

3.10. Maven

Korišten je Maven alat za automatsku igradnju koda. Uveden je sqlite-jdbc dependencies kao i onaj za omogućavanje pokretanje testova.

org.xerial:sqlite-jdbc:3.21.0

3.11. Napredne teme objektno-orjentisanog programiranja

Klase Professor i Student nasljeđuju klasu Person. U klasama se odvaja javni dio od privatnog dijela klase.

3.12. Objektno-orjentisani dizajn

Koncepti objektno-orjentisanog dizajna su većinski ispunjeni. U nastavku će biti prikazan dijagram klasa koji prikazuje sve klase, njihove atribute i metode, kao i međusobne veze klasa. Što se tiče općih principa, uglavnom su ispunjeni. Nema ponavljanja koda i korišten je najmanji broj klasa sa kojima je moguće realizovati sistem.

3.13. Kvaliteta koda i refactoring

Metode i klase nisu predugačke. Komentara nema jer nisu potrebni – kod je dovoljno razumljiv. Svaka klasa posjeduje metode koje su vezane za njihovu namjenu.

4. Prilike za unaprjeđivanje

Aplikacija je jako dobra podloga za unaprjeđivanje i širenje mogućnosti iste. Moguće je povećati broj pitanja koja možemo učitati za izradu kviza, dodati neke nove attribute poput timera koji će biti prikazan tokom izrade kviza, koristiti drugu bazu podataka koja će omogućiti rad s velikim brojem podataka, omogućiti bolji nadzor kada student upisuje svoje bodove (nešto u vidu automatske provjere).