

E-indeks

Drugi zadatak iz Jave

Cilj zadatka: U okviru ovog zadatka napravićemo aplikaciju za evidenciju studenata, predmeta koje slušaju i njihovih ocena tokom školske godine. Zadatak će moći da se realizuje u dve varijante, kao distribuirana klijent-server aplikacija ili kao „standalone“ aplikacija gde se sve aktivnosti implementiraju u okviru jednog procesa. Takođe, aplikacija se može realizovati kao konzolna aplikacija ili kao GUI aplikacija u obe varijante. Najviše poena će nositi distribuirana klijent-server GUI aplikacija, kod koje je *barem* jedna komponenta (klijent ili server) implementirana sa grafičkim korisničkim interfejsom.

Opis zadatka: U E-indeks aplikaciji postoje dve grupe korisnika: administratori i studenti. Administratori imaju mogućnost:

- dodavanja novih studenata, pri čemu svaki student ima svoje ime, prezime, broj indeksa, JMBG, predmete i ocene na svakom predmetu koji sluša, poene za svaku od kategorija svakog od predmeta, korisničko ime i lozinku. Dodatna ograničenja su:
 - Broj indeksa mora biti u formatu:
Smer delimiter godina gde je smer slovo **E** ili **e** praćen sa **1,2** ili **3**, delimiter / ili – a godina **2000 – 2023** (na primer E2-2015 ili e1/2019)
 - JMBG mora da ima 13 cifara gde prve dve predstavljaju dan, druge dve mesec a poslednje tri poslednje tri cifre godine rođenja
- dodavanje novih predmeta, pri čemu svaki predmet ima svoje ime i kategorije koje se ocenjuju (npr. T1-teorija 1, Z1-zadaci 1, T2-teorija 2 i Z2-zadaci 2). Kategorije ne moraju biti iste za svaki od predmeta i administrator mora da ima mogućnost da definiše kategorije za svaki od predmeta, koliko poena nosi svaka kategorija (ukupna suma poena mora biti 100p) i koliki je minimum poena neophodan da se osvoji u svakoj od kategorija, da bi se ispit smatrao položenim
- dodeljivanje svakom studentu predmeta koje on sluša, kao i ažuriranje poena osvojenih na predmetu. Pri tome, nije neophodno da se u istom trenutku unesu svi poeni osvojeni na svim kategorijama, a student je položio ispit ukoliko je sakupio 51 poen pri čemu je iz svake kategorije osvojio **barem minimum specificiranih poena**. Takođe, ocene se zaključuju na osnovu osvojenih poena (0-50 ocena 5, 51-60 ocena 6, 61-70 ocena 7, 71-80 ocena 8, 81-90 ocena 9 i 91-100 ocena 10)

Studenti mogu da:

- dobiju informacije o bodovnom stanju za svaku od kategorija svakog od predmeta koji slušaju

- dobiju informaciju koje predmete su položili a koje nisu (koje ocene imaju na kom predmetu)

Prilikom razvoja E-indeks aplikacije, potrebno je implementirati sledeće funkcionalnosti:

1. Podaci za logovanje svih korisnika aplikacije nalaze se u datoteci *users.txt* u formatu:

`username:password:role`

pri čemu su *username* i *password* korisničko ime i lozinka za svakog od korisnika (definisani od strane administratora), a *role* može biti **admin** ili **student**. Prilikom svakog dodavanja novog administratora ili studenta od strane postojećih administratora, neophodno je sadržaj datoteke *users.txt* ažurirati. Na početku rada, barem jedan administrator mora da bude definisan u datoteci *users.txt*, sa svojim korisničkim imenom, lozinkom i ulogom administratora.

2. Prilikom logovanja na sistem, korisnik unosi korisničko ime, lozinku i ulogu, a u slučaju da sva tri polja odgovaraju podacima iz datoteke *users.txt*, logovanje je uspešno i dalja funkcionalnost je definisana samom ulogom korisnika (da li je administrator ili student)

3. Kod distribuirane verzije E-indeks aplikacije, sva „logika“ je implementirana na serverskoj strani (autentifikacija/autorizacija korisnika prilikom logovanja, unos novih podataka administratora ili ispis postojećih kod administratora ili studenata) dok klijentska aplikacija samo šalje podatke ili upite i dobija nazad rezultate u slučaju upita

4. U slučaju implementacije distribuirane verzije E-indeks aplikacije, za svakog novo povezanog klijenta potrebno je implementirati zasebnu nit u okviru koje će server komunicirati sa tim konkretnim klijentom

5. U obe varijante aplikacije, mora postojati način da se terminira program, pri čemu će svi do tada uneti podaci biti sačuvani, tako da se prilikom narednog pokretanja aplikacije, nastavi izvršavanje kao da u međuvremenu ona nije bila prekidana. U slučaju distribuirane klijent-server E-indeks aplikacije, prilikom prekida, svi podaci se čuvaju i, kasnije, prilikom narednog pokretanja, rekonstruišu **od strane serverske aplikacije**

6. U slučaju da se implementira GUI varijanta aplikacije, sami odaberite korisnički interfejs, koje komponente koristite i na koji način

7. Sami odaberite koje klase (sa kojim atributima i metodama) ćete implementirati i na koji način ćete ih koristiti

Ocenjivanje (ukupno 25 poena):

- Logovanje na sistem, provera korisničkog imena, lozinke i uloge, kao i ažuriranje spiska sa *login* podacima za novo-dodate korisnike (datoteka *users.txt*) **3 poena**
- Omogućen prekid rada aplikacije, čuvanje do tada unetih podataka i mogućnost rekonstrukcije podataka nakon ponovnog pokretanja aplikacije **2 poena**

- implementacija administratorskog dela: omogućeno dodavanje studenata (sa svim podacima), provera validnih unosa podataka (za studente broj indeksa i JMBG), dodavanje predmeta (sa kategorijama, uz proveru da maksimalni osvojeni poeni svih kategorija budu 100) i dodeljivanje studenata na predmete - **7 poena**
- implementacija studentskog dela: nakon logovanja sa ulogom studenta, pregled osvojenih poena za željeni predmet i trenutnih ocena – **3 poena**
- implementacija distribuirane klijent/server E-indeks aplikacije **6 poena**
- implementacija GUI aplikacije, pri čemu u slučaju distribuirane aplikacije ne moraju obe komponente biti GUI, dovoljno je da bude samo jedna (npr. server konzolna, a klijent GUI)- **4 poena**