Kviz

Drugi zadatak iz Jave (školska 2023/2024)

Cilj zadatka:

U ovom zadatku je potrebno implementirati jednostavan kviz, u kojem takmičari dobijaju setove pitanja na koje odgovaraju, a koja su u formatu „zaokruži jedan tačan odgovor, od ponuđenog“. Za svakog takmičara je potrebno voditi evidenciju o broju pitanja na koje je sve davao odgovore, kao i o broju tačnih odgovora, na osnovu čega se učesnici rangiraju. 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 okviru kviza je potrebno implementirati dve grupe korisnika: administratori i takmičari.

Administratori imaju mogućnost:

- Dodavanja novih korisnika, odnosno administratora i takmičara. Svaki korisnik mora imati svoje korisničko ime, lozinku i ulogu. Dodatna ograničenja su:
  - Korisničko ime ne sme počinjati sa brojem i ne sme sadržati u sebi simbole (karaktere koji nisu ni slovo ni broj).
  - Lozinka mora sadržati bar jedno veliko slovo, bar jedno malo slovo, bar jedan broj i bar jedan simbol. Lozinka mora biti duža od 6 karaktera.
- Uklanjanja korisnika, odnosno administratora i takmičara.
- Menjanja seta pitanja na koje takmičari odgovaraju. Svi setovi pitanja se čitaju iz fajlova, njih ima više, a uvek je aktivan samo jedan set pitanja, koji administrator odredi. Svaki set sadrži 10 pitanja, plus još jedno bonus pitanje – za slučaj da takmičar zatraži zamenu pitanja. Svako pitanje sadrži 4 ponuđena odgovora, pod a), b), c) i d), od kojih je uvek **samo jedan** tačan.

Takmičari imaju mogućnost:

- Da zatraže set pitanja za odgovaranje. Nakon toga, iz trenutno aktivnog seta pitanja im se ispisuje pitanje po pitanje, na koja oni šalju odgovore. Na svaki trenutno aktivni set pitanja je moguće **samo jednom** davati odgovore, tako da kad takmičar prođe trenutno aktivni set pitanja, on ne može ponovo dobijati pitanja za odgovaranje, sve dok se ne postavi novi set pitanja.
- Da provere stanje na tabeli. Pošto se za svakog takmičara vodi evidencija o broju pitanja na koje su sve odgovarali, kao i o broju tačnih odgovora – ukoliko takmičar zatraži stanje na

tabeli, njemu se ispisuje lista svih postojećih takmičara, sortiranih po broju datih tačnih odgovora, uz dodatnu informaciju o ukupnom broju pređenih pitanja.

Primer ispisa liste:

```
1. Petar 15/20      ← 15 tačnih odgovora, od 20 pređenih pitanja
2. Milica 12/20     ← 12 tačnih odgovora, od 20 pređenih pitanja
3. Marko 10/30      ← 10 tačnih odgovora, od 30 pređenih pitanja
```

- Prilikom davanja odgovora, svaki takmičar ima pravo na 3 vrste pomoći. Svaka pomoć može biti iskorišćena **samo jednom** za dati set pitanja na koji takmičar odgovara. Kada se postavi novi set pitanja, takmičar može ponovo koristiti sve vrste pomoći. Ove pomoći su:
  - **Pola-pola** – odabirom ove pomoći, takmičaru se od 4 ponuđena odgovora, nasumično, odbacuju 2 netačna odgovora.
  - **Zamena pitanja** – odabirom ove pomoći, takmičaru se trenutno pitanje, na koje je odgovarao, zamenjuje sa novim pitanjem (tj. sa bonus pitanjem iz tog seta pitanja). Na primer, ako takmičar ne zna odgovor na 4. pitanje koje je dobio, ukoliko odabere ovu pomoć i da tačan odgovor na novo pitanje, računa se kao da je odgovorio tačno na 4. pitanje.
  - **Pomoć prijatelja** – ovu pomoć je moguće implementirati samo u varijanti klijent-server aplikacije. Odabirom ove pomoći, takmičar ima mogućnost slanja jedne poruke nekom od preostalih aktivnih takmičara, u kojoj može zatražiti pomoć/odgovor na pitanje za koje nije siguran. Takmičar, koji primi poruku, mora imati indikaciju od koga je ona stigla, a zatim, ukoliko on to želi, može uzvratiti poruku prvom takmičaru. Sadržaj poruka je proizvoljan. Za pomoć u implementaciji ove funkcionalnosti možete pogledati Chatroom primer iz materijala.

<u>Prilikom razvoja aplikacije, potrebno je implementirati sledeće funkcionalnosti:</u>

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 **contestant** (takmičar). Prilikom svakog dodavanja novog administratora ili takmičara 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.

   Ovo skladištenje podataka možete realizovati na dva načina: **sa enkripcijom ili bez nje**. U prvom varijanti, sadržaj *users.txt* datoteke mora biti **sve vreme** enkriptovan. U drugoj varijanti, sadržaj *users.txt* datoteke ne mora da se enkriptuje i sa njim onda radite kao i dosad kad ste čitali/upisivali u fajlove. Verzija bez enkripcije nosi manje bodova (pogledajte Ocenjivanje, na kraju teksta zadatka).

Za realizaciju enkripcije, možete obezbediti fiksan tajni ključ za enkripciju/dekripciju, kako biste mogli, prilikom svakog pokretanja aplikacije, da dešifrujete prethodno sačuvan enkriptovan fajl, jer se sve vreme onda koristi isti ključ. To možete realizovati, na primer, tako što, prilikom generisanja tajnog ključa, koristite fiksan *seed* („seme“) za generisanje i time ćete uvek dobijati generisan isti tajni ključ:

```
SecureRandom securerandom = new SecureRandom();
String seed = "OvdeStaviteSeedPoIzboru";
securerandom.setSeed(seed.getBytes());
...
```

Pošto je potrebno da *users.txt* fajl već inicijalno ima bar jednog admin korisnika, takav enkriptovan fajl možete zasebno generisati, pa koristiti nadalje. U prilogu zadatka imate enkriptovan *users.txt* fajl, sa jednim admin korisnikom čiji je username „admin“ i šifra „admin“ (dakle, admin:admin:admin). Fajl je enkriptovan koristeći simetrični AES, blok cipher algoritam, sa fiksnim inicijalizacionim vektorom od 16 bita **čiji su elementi redom brojevi od 1 do 16**, i sa ključem dužine 128 bita za čije generisanje je korišćen seed „**RSZEOS2024**“. Ako ne želite sami da generišete početni enkriptovani fajl, možete koristiti taj priloženi, sa istim postupkom enkripcije/dekripcije (pogledajte i primere enkripcije iz materijala).

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 takmičar).
3. Kod distribuirane verzije aplikacije, sva „logika“ je implementirana na serverskoj strani (autentifikacija/autorizacija korisnika prilikom logovanja, zamena seta pitanja, prosleđivanje poruka između takmičara i njihovo rankiranje) dok klijentska aplikacija samo šalje podatke ili upite/zahteve i dobija nazad rezultate u slučaju upita ili realizaciju određene funkcije.
4. U slučaju implementacije distribuirane verzije aplikacije, za svakog novog 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 i postignuti rezultati takmičara biti sačuvani, tako da se prilikom narednog pokretanja aplikacije, nastavi izvršavanje kao da u međuvremenu ona nije bila prekidana. U slučaju distribuirane klijent-server 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. U slučaju realizacije aplikacije bez GUI, svu funkcionalnost možete realizovati kroz jednostavne meni opcije. Na primer, prilikom pokretanja aplikacije se može prikazati meni:

```
-----------------------------
1. Uloguj se
2. Izlaz
-----------------------------
```

Ukoliko se uloguje administrator, njemu se može ponuditi sledeći meni:

```
------------------------------
1. Izmeni aktivni set pitanja
2. Unos učesnika
3. Brisanje učesnika
4. Izloguj se
------------------------------
```

Ukoliko se u prvom meniju ulogovao takmičar, njemu se može ponuditi sledeći meni:

```
------------------------------
1. Zatraži pitanja
2. Proveri stanje na tabeli
3. Izloguj se
------------------------------
```

Odabirom opcije 1, takmičaru se zatim ispisuje pitanje po pitanje, iz aktivnog seta pitanja. Pošto svako pitanje ima ponuđene odgovore pod a), b), c) i d), slanjem odgovarajućeg slova takmičar može davati odgovor, a slanjem broja 1, 2 ili 3 (pola-pola, zamena pitanja ili pomoć prijatelja), može tražiti neku od gore spomenutih pomoći.

8. Sami odaberite koje klase (sa kojim atributima i metodama) ćete implementirati i na koji način ćete ih koristiti.
9. Uz tekst zadatka se nalaze .txt fajlovi za setove pitanja, koje možete koristiti da lakše realizujete aplikaciju. U svakom pitanju, odgovor pod d) predstavlja tačan odgovor – **obezbedite da se takmičarima ne prikazuje isti redosled odgovora, kako ne bi uočili ovu pravilnost**.

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*) – **2 poena**
- Enkripcija i dekripcija sadržaja *users.txt* fajla – **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 takmičara (sa svim podacima), omogućeno brisanje takmičara i promena seta pitanja – **2 poena**
- implementacija takmičarskog dela: čitanje pitanja iz fajlova, prikazivanje pitanje po pitanje, davanje odgovora i validacija odgovora, omogućeno traženje pomoći prilikom odgovaranja, omogućeno zahtevanje rang liste takmičara, sortiranje i prikaz rang liste – **6 poena**
- implementacija distribuirane klijent/server 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**