

Računarske mreže 4I, Ispit - JAN1

11.01.2020.

Pročitati sve zadatke **pažljivo** pre rada - sve što nije navedeno ne mora da se implementira!

Na **Desktop**-u se nalazi zip arhiva. Unutar arhive se nalazi direktorijum sa imenom `rm_rok_Ime_Prezime_miGGXXX` u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i zameniti svojim podacima.

Otvoriti IntelliJ IDEA, izabrati opciju **Open project** (ne **Import project**!) i otvoriti pomenuti direktorijum.

Sve kodove ostaviti unutar već kreiranih Java fajlova. **Kodovi koji se ne prevode se neće pregledati.**

Nepoštovanje formata ulaza/izlaza nosi kaznu od -10% poena na zadatku! Vreme za rad: **2.5h**.

1. TCP Sockets (15p)

Implementirati server, koji će imati ulogu da održava *in-memory* tabelu šahista i njihove trenutne rejtinge.

Tabela ima kolone: `id (int)`, `naziv (String)` i `elo (int)`.

- Napraviti Java aplikaciju koja ima ulogu klijenta. Povezati se na lokalni server na portu 1996 koristeći **Socket** klasu. Nakon formiranja konekcije, klijent može poslati više zahteva serveru (zahtevi se unose sa standardnog ulaza), sve dok mu ne pošalje `bye`. Odgovori servera na zahtev se ispisuju na standardni izlaz. Mogući zahtevi su (implementirati u ovoj stavci samo slanje od strane klijenta): (3p)
 - `sel id` (`id` je tipa `int`)
 - `ins naziv` (`naziv` je tipa `String`)
 - `upd id elo` (`id` i `elo` su tipa `int`)
- Napraviti Java aplikaciju koja ima ulogu servera. Pokrenuti lokalni server na portu 1996, koristeći **ServerSocket** klasu. Server za svakog primljenog klijenta pokreće zasebnu nit u kojoj će se taj klijent obraditi tako što se ispiše poruka o pristiglom klijentu kao u primeru ispod. (2p)
- Server pristigle zahteve obradjuje na sledeći način:
 - `sel id`: vraća naziv i elo šahiste sa datim identifikatorom `id` (2p)
 - `ins naziv`: ubacuje u tabelu šahistu sa datim imenom dodeljujući mu jedinstveni identifikator (sledeći slobodan ceo broj) i elo u vrednosti 1300 (to je najmanja moguća vrednost za elo) i vraća poruku o uspešnosti operacije (2p)
 - `upd id deltae`: vrši izmenu elo vrednosti šahiste sa identifikatorom `id` za `deltae` i vraća poruku o uspešnosti operacije (2p)
- Ukoliko bilo koji od ovih zahteva nije ispravno formiran ili nije naveden iznad, vratiti tekst kao u primerima ispod. (1p)
- Imajte u vidu da mogu da se dese konfliktne situacije (kao npr. da dva klijenta žele da promene elo istoj osobi). Obezbediti da se ovakvi zahtevi pravilno obrade. Takođe, obezbediti da u slučaju izuzetaka, resursi budu ispravno zatvoreni. (3p)

```
> ins Magnus Carlsen      > ins Fabiano Caruana     > ins Marko
ins je uspesno izvršen    ins je uspesno izvršen    ins je uspesno izvršen
> sel 1                   > sel 1                   > sel 1
Magnus Carlsen: 1300      Fabiano Caruana: 1300     Marko: 1300
> upd 1 30                > upd 1 1500              // drugi klijent: upd
upd je uspesno izvršen    upd je uspesno izvršen    > sel 1
> sel 1                   > sel 1                   Marko: 1400
Magnus Carlsen: 1330      Fabiano Caruana: 2800     > ins Marko
> upd 1 -10               > upd 1 -10000            ins je uspesno izvršen
upd je uspesno izvršen    upd je uspesno izvršen    // nije isti Marko
> sel 1                   > sel 1                   > sel 2
Magnus Carlsen: 1320      Fabiano Caruana: 1300     Marko: 1300
> bye                     > bye                     > bye
```

*Napomena: Ohrabrujemo studente da koriste **netcat** kako bi testirali delimične implementacije i otkrili greške pre vremena. Takođe, ukoliko se npr. preskoči implementacija servera, može se mock-ovati server putem **netcat-a**.*

2. UDP Sockets (15p)

Implementirati UDP server koji klijentima služi Fibonačijeve brojeve.

- Napraviti Java aplikaciju koja ima ulogu UDP servera. Slušati na portu 12345 i primiti datagrame od klijenata koristeći `DatagramPacket` klasu. Sadržaj svakog datagrama primljenog od klijenta je velicine 4B. Ispisati tekst **Stigao datagram!** kad god server primi validan datagram od nekog klijenta. (3p)
- Napraviti Java aplikaciju koja ima ulogu UDP klijenta. Poslati UDP datagram lokalnom serveru na portu 12345 koristeći `DatagramPacket` klasu. Sadržaj datagrama je jedan pozitivan ceo broj veličine 4B koji se unosi sa standardnog ulaza. (3p)
- Inicijalno, klijent šalje serveru datagram sa sadržajem koji predstavlja prirodni broj n (ne veći od 80). Nakon primanja datagrama, server klijentu šalje prvih n Fibonačijevih brojeva ¹, po jedan u svakom datagramu. Svaki poslati broj je veličine 8B. (4p)
- Primiti datagrame na klijentskoj strani, i za svaki primljeni datagram ispisati primljeni broj na standardni izlaz. (2p)
- Keširati Fibonačijeve brojeve na serverskoj strani (ne računati ih iznova za svakog klijenta). (2p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (1p)

¹Fibonačijevi brojevi se računaju po formuli: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ za $n > 1$