Računarske mreže, Ispit - SEP2 2022

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 u formatu `rm_rok_Ime_Prezime_mXGGXXX` u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i ubaciti svoje podatke u ime. 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!**

1. **Matrično množenje (15p)**
   Napraviti Java aplikaciju koja koristeći niti množi dve kvadratne matrice.

   - Kao ulaz u program se daje putanja do tekstualnih fajlova u kojima se nalaze kvadratne matrice koje je potrebno pomnožiti - po jedna u svakom fajlu. Učitati i ispisati matrice na standardni izlaz. (3p)
   - Ukoliko matrice ne mogu da se množe, ispisati poruku i prekinuti izvršavanje programa. (1p)
   - Označimo dimenzije matrica sa $n \times n$. Pokrenuti n^2 niti i postarati se da svaka nit računa jedan element rezultujuće matrice. Ispisati rezultujuću matricu na standardni izlaz. (5p)
   - Računati zbir elemenata rezultujuće matrice tokom rada svake niti. Kada svaka nit izračuna svoju vrednost, ažurira globalni zbir. Postarati se da nema trke za podacima - obezbediti kritičnu sekciju proizvoljnim mehanizmom. (4p)
   - Postarati se da program ispravno obrađuje specijalne slučajeve (npr. ako fajl ne postoji na datoj putanji) i ispravno zatvoriti sve korišćene resurse u slučaju izuzetka. (2p)

2. **NBIO (25p)**
   Bliži se kraj ispitnog roka, napravi klient-server Java aplikaciju koristeći TCP Sockets/Channels API koja ce pomoći studentima da izračunaju koliko stranica dnevno moraju da nauče kako bi uspešno položili ispite.

   - Napisati Java klasu koja ima ulogu **blokirajućeg** TCP klijenta koristeći `Java Channels API`. Klijent formira konekciju sa lokalnim serverom na portu `12345`. Nakon formiranja konekcije, klijent šalje serveru zahteve, red po red, sve do unosa niske `"exit"`. Zahtev čine, razdvojeni razmakom, datum od kog student planira da počne da uči, datum održavanja ispita, broj strana koje student treba da nauči do tada (datumi se unose u formatu `dan/mesec/godina`). Ukoliko je datum početka izostavljen postavlja se na današnji datum. Pre učitavanja narednog reda potrebno je sačekati odgovor servera i ispisati ga na standardni izlaz. Odgovor servera predstavlja broj strana koje student treba dnevno da nauči kako bi položio taj ispit. (8p)
   - Napisati Java klasu koja ima ulogu **neblokirajućeg** TCP servera koji osluškuje na portu `12345` koristeći `Java Channels API`. Server prima zahteve od klijenata i nakon svakog primljenog zahteva vraća odgovor. Ukoliko je broj preostalih dana do ispita manji od 7 dodatno obavestiti studenta o tome. Ukoliko su uneti datumi takvi da je datum početka veći ili jednak od datuma održavanja ispita obavestiti studenta da neće položiti ispit u ovom roku (13p)
   - Postarati se da svi resursi budu pravilno zatvoreni. (4p)

   Primer rada:

   ```
   ulaz: 12/09/2022 23/09/2022 220      izlaz: 20

   ulaz: 20/09/2022 23/09/2022 636      izlaz: 212, ostalo je manje od 7 dana

   ulaz: 10/09/2022 200                 izlaz: Vidimo se u narednom roku.
   ```

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

Okrenite stranu!

3. **UDP (15p)**
Napisati Java aplikaciju koja dohvata informacije o studentu na osnovu njegovog indeksa.

- Napisati Java klasu koja ima ulogu UDP klijenta. Poslati inicijalni datagram koji sadrži indeks u formatu `XXX/GGGG` i jednu od komandi `prosek` ili `ime` lokalnom serveru koji osluškuje na portu 12321. Primiti datagram od servera koji sadži informaciju koja je zatražena. Podatke u datagramu kodirati proizvoljno. (4p)
- Napisati Java klasu koja ima ulogu servera koji osluškuje na portu 12321.
  - Iz tekstualnog fajla `studenti.txt` učitati informacije o studentima i keširati ih na serveru. Moguće je koristiti proizvoljnu kolekciju podataka. (2p)
  - Primiti datagram od klijenta, isparsirati komandu, i vratiti datagram sa zatraženom informacijom o studentu. (4p)
  - U slučaju nepostojećeg indeksa, ili nevalidne komande, vratiti klijentu poruku "Zahtev nije validan". (2p)
- Postarati se da su svi resursi pravilno zatvoreni u slučaju izuzetka. (3p)

Primer izvršavanja:

```
klijent: 002/2018 ime                    server: Pera Peric

klijent: 100/2018 prosek                 server: 6.3

klijent: 101/2018 prosek                 server: Zahtev nije validan!

klijent: 100/2018 email                  server: Zahtev nije validan!
```