Računarske mreže, Ispit - Septembar0 2021

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!** Vreme za rad: **3h**.

1. **Matrično množenje (15p) (za studente koji nisu radili projekat)**
   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 sadržaje fajlova 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. **UDP Sockets (20p/12p)**

   - 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. (4p)
   - 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. (4p)
   - 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`. (5p)
   - 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). (4p)
   - Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (1p)

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!

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

3. **Površina pravougaonika (25p/18p)**
   Napraviti klijent-server TCP aplikaciju preko koje se računaju površine pravougaonika zadatih koordinatama gornjeg levog i donjeg desnog temena.

   - Napraviti Java klasu koja ima ulogu lokalnog **blokirajućeg** TCP klijenta koristeći Java Channels API. Klijent formira konekciju sa lokanim serverom na portu 54321. Nakon formiranja konekcije, klijent serveru u petlji šalje po 4 cela broja, učitana sa standardnog ulaza, koja predstavljaju koordinate gornjeg levog i donjeg desnog temena pravougaonika. Podrazumeva se da su stranice pravougaonika paralelne koordinatnim osama. Nakon slanja, klijent čeka odgovor od servera i ispisuje ga na standardni izlaz. (8p/6p)
   - Napraviti Java klasu koja ima ulogu lokalnog **neblokirajućeg** TCP servera, koji osluškuje na portu 54321 koristeći Java Channels API. Server kao odgovor na svaku iteraciju petlje šalje klijentu po jedan ceo broj koji predstavlja površinu zadatog pravougaonika. Ukoliko server primi četiri 0, to se tumači kao zahtev za prekid veze. (15p/10p)
   - Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (2p)