

Računarske mreže, Ispit - SEP1 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. Dekompresovati arhivu na Desktop i ubaciti svoje podatke u ime pomenutog direktorijuma.

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. Selektivno kopiranje fajla (15p) (za studente koji nisu radili projekat)

- Napraviti Java aplikaciju koja koristeći odgovarajuće ulazne i izlazne tokove kopira sadržaj tekstualnog fajla sa imenom koje se unosi preko standardnog ulaza u fajl **timestamps.txt**. Postarati se da se u slučaju izuzetka prikaže odgovarajuća poruka (različita za različite tipove izuzetaka). (3p)
- Prekopirati samo one niske koje predstavljaju validne vremenske niske u formatu DD-MM-YYYY. Pretpostaviti da svi meseci imaju najviše 31 dan i da je godina veća od 2000 (npr. 02-12-2015). (6p)
- Koristiti baferisanje ulaznog i izlaznog toka zarad smanjenja broja IO operacija. (2p)
- Niske ispisati u fajl tako da po jedna niska bude u svakoj liniji. (2p)
- Podesiti kodne strane za oba fajla na UTF-8. (1p)
- Postarati se da se u slučaju izuzetka garantuje da su zatvoreni svi korišćeni resursi. (1p)

2. TCP Sockets - Aerodromi (20p/12p)

Napraviti osnovu za TCP klijent-server Java aplikaciju koja pruža informacije o odlaznim letovima sa aerodroma.

- U direktorijumu **aerodromi**, unutar direktorijuma **tests** na Desktop-u, nalaze se tekstualni fajlovi koji sadrže informacije o odlaznim letovima sa nekog aerodroma za taj dan. Ime fajla predstavlja ime grada u kome se aerodrom nalazi, a svaka linija fajla sadrži informacije o jednom leto i oblika je:
<JEDINSTVENA_ŠIFRA_LETA> <GRAD_SLETANJA> <VREME_POLETANJA> <VREME_SLETANJA>.
Na serverskoj strani keširati ove podatke kako se ne bi čitali iznova za svakog klijenta. (4p/3p)
- Napraviti Java klasu koja ima ulogu lokalnog TCP servera koji osluškuje na portu 12345. Svakom novom klijentu server šalje spisak svih gradova za čije aerodrome ima informacije o letovima. (4p/2p)
- Napraviti Java klasu koja ima ulogu lokalnog TCP klijenta. Nakon uspostavljanja konekcije sa serverom na portu 12345, klijent čeka spisak gradova od servera i ispisuje ih na standardni izlaz. Nakon toga, klijent dodatno šalje serveru ime grada, uneto sa standardnog ulaza, sa čijeg aerodroma želi da dobije informacije o odlaznim letovima, a zatim ponovo čeka odgovor od servera koji ispisuje na standardni izlaz i završava sa radom. (7p/4p)
- U zavisnosti od imena grada, server šalje keširane informacije iz odgovarajućeg fajla. (4p/2p)
- 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. Takođe, ukoliko se npr. preskoči implementacija servera, može se mock-ovati server putem **netcat-a**.*

Okrenite stranu!

3. Non-Blocking IO (25p/18p)

Napraviti klient-server Java aplikaciju koristeći TCP Sockets/Channels API.

- Napisati Java klasu koja ima ulogu **blokirajućeg** TCP klijenta koristeći Java Channels API. Klijent formira konekciju sa lokalnim serverom na portu 12345 i zatim šalje serveru 4 bajta jedan za drugim, učitani sa standardnog ulaza. (3p/2p)
- Napisati Java klasu koja ima ulogu lokalnog **neblokirajućeg** TCP servera, koji osluškuje na portu 12345, koristeći Java Channels API. Server čuva skriveni ceo broj. Prilikom obrađivanja klijenta server prihvata bajtove od viših ka nižim i kreira ceo broj (tipa `int`). Na primer, ukoliko klijent šalje redom bajtove `0xA0`, `0xB0`, `0xC0` i `0xD0`, server kreira broj `0xA0B0C0D0`. Nakon uspešnog prihvatanja bajtova, server klijentu vraća novi broj nastao primenom operacije XOR na prihvaćene bajtove i skrivenog broja. (6p/4p)
- Server slučajnim izborom generiše skriveni ceo broj prilikom pokretanja i ispisuje ga na standardni izlaz. Broj mora biti veći od 999 i prost. (10p/8p)
- Klijent ispisuje odgovor od servera na standardni izlaz. Klijent može više puta da učitava i šalje bajtove serveru dok ne prekine vezu. (3p/2p)
- Ukoliko klijent serveru pošalje broj bez ijednog bita postavljenog na 1, server klijentu vraća `-1` (1p/1p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (2p/1p)