

Računarske mreže 4R, Ispit - JUN1

18.06.2020.

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 u formatu **rm_rok_Ime_Prezime_miGGXXX** 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. Statistika protokola (15p) (za studente koji nisu radili projekat)

Napisati program koji vodi statistiku protokola u URL-ovima koji se nalaze u datotekama na fajl sistemu.

- U direktorijumu **urls** unutar direktorijuma **tests** na Desktop-u se nalaze datoteke koje sadrže spisak URL-ova (po jedan u svakoj liniji). Korišćenjem odgovarajućih tokova podataka prebrojati koliko linija sadrži svaka datoteka. Na standardni izlaz, za svaku datoteku, ispisati njenu apsolutnu putanju i broj linija kao u primeru ispod u levoj koloni. (2p)
- Za svaku datoteku pokrenuti zasebnu nit koja će obradivati i ispisivati statistiku za tu datoteku. (2p)
- Za svaku pročitanu liniju datoteke kreirati novi URL objekat koristeći **URL** klasu. Preskočiti sve linije koje ne predstavljaju validan URL. (1p)
- Za svaki validni URL proveriti protokol koji se koristi i pratiti ukupan broj pojavljivanja tog protokola u trenutnoj datoteci. Nakon što nit obradi datoteku, na standardni izlaz ispisati statistiku kao u primeru ispod u desnoj koloni. (4p)
- Postarati se da se ispisi svake niti na standardni izlaz ne prepliću. (4p)
- Postarati se da program ispravno obradjuje specijalnim slučajeve (npr. ako datoteka ne postoji na datoj putanji) i ispravno zatvoriti sve korišćene resurse u slučaju izuzetka. (2p)

```
/home/ispit/Desktop/tests/urls/1.txt 6      https   : 1
/home/ispit/Desktop/tests/urls/2.txt 4      - /home/ispit/Desktop/tests/urls/1.txt
/home/ispit/Desktop/tests/urls/3.txt 3      http    : 4
/home/ispit/Desktop/tests/urls/4.txt 5      https   : 2
                                          - /home/ispit/Desktop/tests/urls/2.txt
                                          ftp      : 3
                                          http     : 1
```

*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! —

2. Zbir brojeva (20p/12p)

Napraviti klijent-server TCP aplikaciju preko koje se računa zbir realnih brojeva.

- Napraviti Java klasu koja ima ulogu lokalnog TCP servera, koji osluškuje na portu 10000, koristeći Java **Socket** API. Za svakog povezanog klijenta, server pokreće zasebnu nit, preko koje se vrši komunikacija sa klijentom. (5p/3p)
- Napraviti Java klasu koja ima ulogu klijenta. Klijent formira konekciju sa lokanim serverom na portu 10000 i šalje serveru realne brojeve koje čita sa standardnog ulaza sve dok korisnik ne unese nisku KRAJ. Server ispisuje svaki realni broj koji dobije od klijenta na standardni izlaz. (5p/3p)
- Server za svakog klijenta računa zbir do tada poslatih realnih brojeva. Nakon što klijent pošalje sve realne brojeve, server klijentu vraća njihov zbir. (5p/3p)
- Klijent na standardni izlaz ispisuje realan broj koji mu je poslao server. (2p/1p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (3p/2p)

3. Površina kruga (25p/18p)

Napraviti klijent-server TCP aplikaciju preko koje se računaju površine krugova datih poluprečnika.

- Napraviti Java klasu koja ima ulogu lokalnog **neblokirajućeg** TCP servera koji osluškuje na portu 31415, koristeći Java **Channels** API. Nakon konekcije, klijent šalje serveru realne brojeve koji predstavljaju poluprečnike krugova. Kao odgovor, server šalje klijentu po jedan realan broj nakon svakog primljenog poluprečnika koji predstavlja površinu kruga sa primljenim poluprečnikom. Ukoliko server primi negativan broj od klijenta, to se tumači kao zahtev za prekid veze. (15p/10p)
- Napraviti Java klasu koja ima ulogu **blokirajućeg** TCP klijenta koristeći Java **Channels** API. Klijent formira konekciju sa lokanim serverom na portu 31415. Nakon formiranja konekcije, klijent serveru šalje jedan po jedan realan broj učitani sa standardnog ulaza i pre učitavanja narednog broja čeka odgovor od servera. Ispisati odgovore servera na standardni izlaz. (8p/6p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (2p)