

Računarske mreže, Vanredni kolokvijum

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_vkol_Ime_Prezime_miGGXXX** u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i preimenovati ga tako da ime odgovara podacima studenta. Otvoriti IntelliJ IDEA, izabrati opciju **Open 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: **2h**.

1. Tokovi podataka i niti (15p)

Napisati program koji ispisuje ukupan broj pojavljivanja zadatog HTML tag-a u svim HTML fajlovima sa zadatog spiska URL-ova.

- U datoteci **urls.txt** unutar direktorijuma **tests** na Desktop-u se nalazi spisak URL-ova (po jedan u svakoj liniji). Koristeći odgovarajuće **baferisane** ulazne tokove pročitati sadržaj pomenutog fajla i ispisati broj linija u tom fajlu. (2p)
- Za svaku pročitane liniju fajla **urls.txt** 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. Ukoliko je protokol **FILE** i ukoliko putanja vodi do HTML fajla (ekstenzija **.html**), kreirati zasebnu nit koja će otvoriti **baferisani** ulazni tok do tog fajla putem **URL** klase i pročitati sadržaj fajla (detalji obrade su u narednoj stavci). Kodnu stranu prilikom učitavanja postaviti na **UTF-8**. Ukoliko fajl na datoj putanji ne postoji, ispisati odgovarajuću poruku i ugasiti nit koja je pokrenuta da ga obradi. (5p)
- Pre parsiranja fajla **urls.txt**, sa standardnog ulaza učitati jednu nisku koja predstavlja naziv HTML tag-a. Prebrojati koliko se puta zadati tag pojavljuje u svim fajlovima iz prethodne stavke tako što će svaka nit prebrojati pojavljivanja za fajl koji joj je dodeljen. Ispisati ukupan broj na standardni izlaz (videti primere ispisa ispod teksta zadatka). Pritom, paziti na sinhronizaciju niti ukoliko se koristi deljeni brojač. (5p)
- Postarati se da program ispravno barata specijalnim slučajevima (npr. ako fajl ne postoji na datoj putanji) i ispravno zatvoriti sve korišćene resurse u slučaju izuzetka. (2p)

```
ulaz:  p
izlaz: lines:      29
      not found: /home/ispit/Desktop/tests/404.html
      result:     50

ulaz:  html
izlaz: lines:      29
      not found: /home/ispit/Desktop/tests/404.html
      result:     10

ulaz:  a
izlaz: lines:      29
      not found: /home/ispit/Desktop/tests/404.html
      result:      3
```

*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. 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)

[illegible]