

Računarske mreže, Ispit - JAN2

04.02.2021.

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. Log-file parser (15p) (za studente koji nisu radili projekat)

Napraviti Java aplikaciju koja selektivno parsira log fajl.

- Napraviti Java aplikaciju koja prima putanju do regularnog fajla koji predstavlja log fajl i čita ga koristeći URL klasu i FILE protokol. Ispisati sadržaj fajla na standardni izlaz. (3p)
- U fajlu se nalaze linije u sledećem formatu:
[<DATUM_VREME>] [<IP_ADRESA>] [<URL_DO_RESURSA_NA_SERVERU>] npr.
[12.12.2010 17:41:00] [123.123.123.123] [http://test.com/path/secret.txt]
[12.12.2010 17:51:03] [2607:f0d0:1002:51::4] [https://test.com/path/secret.txt]
[15.12.2010 17:52:11] [123.23.2.33] [ftp://test.com/secret.txt]
[15.12.2010 17:52:21] [123.23.2.33] [FILE:///home/test/secret.txt]
Filtrirati pročitani sadržaj tako da se na standardni izlaz (umesto ispisivanja čitavog fajla) ispišu samo one linije u kojima je zahtev za resurs stigao preko HTTP ili HTTPS protokola. (3p)
- Format ispisa linije na standardni izlaz promeniti na:
v<VERZIJA_IP_ADRESE>: <KORIŠĆENI_PROTOKOL>: <PUTANJA_DO_RESURSA>
(zameniti tagove odgovarajućim informacijama) npr.
v4:http://path/secret.txt (2p)
- Ispisati samo one linije u kojima je vrednost niske koja predstavlja datum i vreme hronološki pre trenutnog datuma i vremena. (4p)
- Preskočiti linije u kojima je unutar URL-a naveden nepodrazumevani port za odgovarajući protokol. (2p)
- Postarati se da u slučaju izuzetka aplikacija ispravno zatvori korišćene resurse i tokove podataka. (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! —

2. UDP Sockets (20p/12p)

Napraviti osnovu za klijent-server Java aplikaciju koristeći *Java UDP API* koja vrši odgovarajuće kodiranje niske.

- Napraviti Java aplikaciju koja ima ulogu UDP klijenta. Poslati datagram lokalnom serveru na portu 12345 koji sadrži nisku (koja može da sadrži beline). Niska se učitava sa standardnog ulaza na klijentskoj strani. Primiti datagram koji predstavlja odgovor servera i rezultat ispisati na standardni izlaz. (8p/4p)
- Napraviti Java aplikaciju koja ima ulogu servera koji sluša na portu 12345. Kada dobije nisku od klijenta, tu nisku transformiše na sledeći način:
 - Ukoliko je karakter veliko slovo, postaje malo slovo i udvostručuje se.
 - Ukoliko je karakter malo slovo, postaje veliko slovo.
 - Ukoliko je karakter broj, postaje karakter '.' i udvostručuje se.

Primer:

niska: mReze Ispit 2

transformisana niska: MrrEZE iiSPIT ..

Poslati datagram pošiljaocu u kome se nalazi transformisana niska.

(10p/7p)

- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka.

(2p/1p)

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

Napraviti klijent-server Java aplikaciju koristeći *TCP Sockets/Channels API* preko koje se računa broj pogodaka u igri Loto.

- Napisati Java klasu koja ima ulogu **blokirajućeg** TCP klijenta koristeći Java Channels API. Klijent formira konekciju sa lokanim serverom na portu 12345 i zatim šalje serveru jednu Loto kombinaciju, to jest 7 celih brojeva iz intervala [1, 39], učitane sa standardnog ulaza. (8p/6p)
- Napisati Java klasu koja ima ulogu lokalnog **neblokirajućeg** TCP servera, koji osluškuje na portu 12345, koristeći Java Channels API. Server slučajnim izborom generiše jednu Loto kombinaciju. Nakon konekcije, klijent šalje serveru učitane Loto kombinaciju. Brojevi unutar kombinacije se ne smeju ponavljati. Kao odgovor, server šalje klijentu informaciju o broju pogodaka u serverski generisanoj kombinaciji. (15p/11p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka.

(2p/1p)