

Računarske mreže 2019, Kolokvijum, 4RA

08.04.2019.

1. Selektivno kopiranje fajla (10p)

- Napraviti Java aplikaciju koja koristeći odgovarajuće ulazne i izlazne tokove kopira sadržaj tekstualnog fajla sa imenom koje se unosi preko standardnog ulaza u fajl `names.txt`. Postarati se da se u slučaju izuzetka prikaže odgovarajuća poruka (različita za različite tipove izuzetaka). (2p)
- Prekopirati samo one niske koje predstavljaju validna vlastita imena i prezimena (počinju velikim slovom, sva ostala slova su mala i ne sadrže karaktere koji nisu slova) (npr. Petar Petrović). (2p)
- 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. (1p)
- 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. Matrično množenje (10p)

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. (1p)
- Kreirati posebnu klasu `MatrixMultiplicationException` i baciti izuzetak ovog tipa ukoliko matrice ne mogu da se množe. (1p)
- Označimo dimenzije matrica sa $n \times n$. Pokrenuti n^2 niti i postarati se da svaka računa jedan element rezultujuće matrice. Ispisati rezultujuću matricu na standardni izlaz. (4p)
- 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. (3p)
- Voditi računa o obradi izuzetaka - program ili nit ne sme da se zaustavi u slučaju izuzetka (npr. ukoliko se desi izuzetak prilikom računanja elemenata rezultujuće matrice, smatrati da je rezultat 0 i nastaviti sa radom). (1p)

3. URL Scanner (10p)

- Napraviti Java aplikaciju koja sa standardnog ulaza učitava URL-ove jedan po jedan u svakoj liniji. Formirati URL objekat za svaki učitani URL i ispisati na standardni izlaz poruku ukoliko URL nije validan. (2p)
- Ispisati protokol, `authority` i putanju iz URL-a koristeći URL klasu. Izlaz formatirati na sledeći način:
<KORIŠĆENI_PROTOKOL> <AUTHORITY> <PUTANJA_DO_RESURSA> npr.
ulaz: `http://www.matf.bg.ac.rs:3030/dir1/dir2/test.txt`
izlaz: `http www.matf.bg.ac.rs:3030 /dir1/dir2/test.txt` (3p)
- Ukoliko se unese IP adresa unutar URL-a dodatno uz informacije iznad ispisati informaciju o verziji IP adrese koja je uneta i ako je to IPv4 adresa ispisati njene bajtove, u formatu:
(v<VERZIJA_IP_ADRESE>) <KORIŠĆENI_PROTOKOL> <PUTANJA_DO_RESURSA> [<BAJTOVI_ADRESE>] npr.
ulaz: `http:///123.123.123.123:80/dir1/dir2/test.txt`
izlaz: (v4) `http /dir1/dir2/test.txt [123 123 123 123]`
ulaz: `sftp://2001:0db8:85a3:::8a2e:0370:7334/dir1/dir2/test.txt`
izlaz: (v6) `sftp /dir1/dir2/test.txt` (4p)
- Postarati se da u slučaju izuzetka aplikacija ispravno zatvori korišćene resurse. (1p)