

Računarske mreže (modul Računarstvo i informatika) - praktični ispit -

Tehničke napomene

Na radnoj površini nalazi se šifrovana zip arhiva sa nazivom `mreze.r.jan.zip`. Unutar te arhive nalazi se direktorijum sa nazivom `rnr_jan_ImePrezime_mrGGXXX` (gde `mrGGXXX` predstavlja korisničko ime Vaše Alas mejl adrese). U ovom direktorijumu nalazi se validan IntelliJ projekat (koji predstavlja Vaš rad) sa paketima `domain_analyzer`, `rock_paper_scissors` i `caesar_cipher`. Ovaj direktorijum izvući iz arhive na Desktop i preimenovati ga u skladu sa Vašim podacima. Otvoriti IntelliJ IDEA, izabrati opciju `Open project` (ne `Import project`!) i otvoriti pomenuti direktorijum. **Kodovi koji se ne prevode se neće pregledati.** Vreme za izradu ispita je **3 sata**.

Srećan rad!

Ispitni zadatak 1: Domain analyzer (15 poena)

Data je datoteka `domains.txt` koji sadrži listu različitih domena (u jednoj liniji se nalazi jedan domen, primer fajla je dat u prilogu). Implementirati Java aplikaciju koja čita sadržaj fajla `domains.txt` i određuje različite IP adrese na koju upućuju domeni iz datog fajla. Informacije o tim IP adresama, zajedno sa spiskom domena koji upućuju na svaku od IP adresa, aplikacija upisuje u datoteku `ip.with.domains.txt`.

Primer ulazne (`domains.txt`) i izlazne (`ip.with.domains.txt`) datoteke:

`domains.txt`

```
www.matf.bg.ac.rs  
www.mg.edu.rs  
www.math.rs
```

`ip.with.domains.txt`

```
147.91.66.73  
- Domains:  
--- www.matf.bg.ac.rs  
--- www.math.rs  
  
147.91.16.156  
- Domains:  
--- www.mg.edu.rs
```

Naredni zadatak se nalazi na sledećoj strani!

Ispitni zadatak 2: Rock-paper-scissors (20 poena)

Rock-paper-scissors (na srpskom *papir-kamen-makaze*) je igra između dva igrača koji nezavisno jedan od drugog biraju jednu od tri reči: **rock** (na srpskom *kamen*), **paper** (na srpskom *papir*) ili **scissors** (na srpskom *makaze*). Ukoliko igrači izaberu različite reči smatra se da:

- **rock** pobeđuje **scissors**;
- **scissors** pobeđuje **paper**;
- **paper** pobeđuje **rock**.

Ukoliko igrači izaberu iste reči smatra se da je rezultat nerešen.

Implementirati igru **rock-paper-scissors** kao *client-server* aplikaciju korišćenjem Java **Datagram** API-ja (klase **DatagramSocket** i **DatagramPacket**):

- Nakon pokretanja klijentske aplikacije, unosi se ime klijenta. Nakon toga, klijent se konektuje na server slanjem paketa serveru koji sadrži ime klijenta.
- Server čeka na konekciju klijenta i nakon konektovanja pokreće igru. Nakon pokretanja igre, server nasumično bira svoj potez, nakon čega šalje zahtev klijentu da odigra svoj potez. Nakon što klijent odigra svoj potez, server određuje rezultat igre i tu informaciju šalje klijentu. Nakon toga, klijentska aplikacija završava sa radom, a server nastavlja da prima konekcije od klijenata. *Napomena: može se pretpostaviti da će unos biti ispravan kao i da u toku opsluživanja jednog klijenta ne dolaze zahtevi od drugih klijenata.*
- Primer rada klijentske aplikacije:

```
Enter your name: Petar
Petar, what is your move: rock
Server move: scissors. You win!
```

Server pokrenuti na portu 5555. Server prestaje sa radom prosledjivanjem SIGINT ([CTRL+C]) signala (odnosno, nije potrebno posebno implementirati zaustavljanje serverskog dela aplikacije). **Obavezno je pravilno zatvoriti i osloboditi sve korišćene resurse.**

Naredni zadatak se nalazi na sledećoj strani!

Ispitni zadatak 3: Caesar cipher (25 poena)

Caesar cipher (na srpskom *Cezarova šifra*) je jednostavan algoritam za šifrovanje teksta, u kom se svako slovo u tekstu pomera unapred u abecedi za unapred definisani broj mesta **shift** (na srpskom *pomak*). Na primer, uz pomak od 3:

- slovo **a** postaje **d**;
- slovo **b** postaje **e**;
- ...
- slovo **z** postaje **c** (pomera se ciklično).

Pomak se primenjuje isključivo na mala slova engleske abecede (**a-z**), dok se ostali karakteri (razmaci, interpunkcija, brojevi itd.) ne menjaju.

Implementirati klijentsko-serversku aplikaciju korišćenjem Java **TCP Socket** API-ja (klase **ServerSocket** i **Socket**), koja omogućava šifrovanje i dešifrovanje teksta Cezarovom šifrom, po sledećim pravilima:

- Klijent se nakon pokretanja povezuje na server i unosi jednu od sledećih komandi:
 - **ENC <shift> <text>** – server vraća tekst šifrovan Cezarovom šifrom;
 - **DEC <shift> <text>** – server vraća dešifrovani tekst;
 - **EXIT** – zatvara konekciju i završava program.
- Server prihvata konekcije i za svakog klijenta pokreće *zasebnu nit* koja obradjuje zahteve klijenta i vraća rezultat dokle god klijent ne pošalje komandu **EXIT**. Nakon toga se konekcija zatvara i nit se prekida, dok glavna nit servera nastavlja da prihvata nove konekcije.
- Ako je uneta komanda neispravna (npr. pogrešan format, nedovoljan broj argumenata, pomak nije broj itd.), server odgovara sa:

`Invalid command format.`

- Primer rada klijentske aplikacije:

```
Enter command: ENC 3 hello world
Result: kloor zruog
```

```
Enter command: DEC 3 kloor zruog
Result: hello world
```

```
Enter command: EXIT
Connection closed.
```

Server pokrenuti na portu 5555. Server prestaje sa radom prosledjivanjem **SIGINT** (**[CTRL+C]**) signala (odnosno, nije potrebno posebno implementirati zaustavljanje serverskog dela aplikacije). **Obavezno je pravilno zatvoriti i osloboditi sve korišćene resurse.**