Univerzitet u Beogradu
Matematički fakultet

Datum: 30.08.2025.
Ispitni rok: Januar 2

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

Tehničke napomene

Na radnoj površini nalazi se šifrovana zip arhiva sa nazivom `mreze.r.feb.zip`. Unutar te arhive nalazi se direktorijum sa nazivom `rmr_feb_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_reachability`, `guess_the_animal` i `shopping_cart`. 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 reachability + statistika (10 poena)

Implementirati Java aplikaciju koja učitava listu domena iz fajla `domains.txt` i proverava njihovu dostupnost korišćenjem metode `InetAddress.isReachable(...)`, sa timeout-om od 3000ms. Aplikacija treba da izračuna koliko ukupno domena je dostupno, koliko nije, i koji domeni su najbrži po vremenu odziva (`System.currentTimeMillis()` metoda pre i posle provere dostupnosti domena se može iskoristiti za vreme odziva).

U fajl `reachable_stats.txt` upisati sledeće:

- Ukupan broj dostupnih i nedostupnih domena
- Lista dostupnih domena sa vremenom odziva u ms
- Top 3 najbrža domena po odzivu

Obavezno je pravilno zatvoriti i osloboditi sve korišćene resurse.

`domains.txt`

```
google.com
yahoo.com
openai.com
nonexistentdomain12345.com
facebook.com
```

`reachable_stats.txt`

```
Ukupan broj dostupnih domena: 4
Ukupan broj nedostupnih domena: 1

Dostupni domeni sa vremenom odziva:
facebook.com - 39 ms
google.com - 42 ms
openai.com - 58 ms
yahoo.com - 65 ms

Top 3 najbrža domena:
1. facebook.com - 39 ms
2. google.com - 42 ms
3. openai.com - 58 ms
```

Naredni zadatak se nalazi na sledećoj strani!

Ispitni zadatak 2: Guess the Animal (30 poena)

Napisati TCP client-server aplikaciju u kojoj klijent pokušava da pogodi životinju koju je server zamislio. Klijent šalje reč, a server vraća:

- `"Too early in alphabet"` ako je pogodak leksikografski pre reči koju je server zamislio
- `"Too late in alphabet"` ako je posle
- `"Correct!"` ako je tačna

Server bira jednu životinju iz datoteke *animals.txt*. Svaki klijent ima svoju sesiju pogodaka i za svakog klijenta se formira nit koja ga obrađuje. Poređenje se vrši metodom *String.compareTo(...)*.

Primer ulazne datoteke:

```
animals.txt:
ant
cat
dog
elephant
lion
tiger
zebra
```

Primer rada klijentske aplikacije:

```
Welcome to server for guessing the animal!
Available animals are: [ant, cat, dog, elephant, lion, tiger, zebra]
Enter your name:
Petar
Petar, guess the animal I chose:
Your guess: cat
Server: Too early in alphabet
Your guess: zebra
Server: Too late in alphabet
Your guess: lion
Server: Correct!
```

Server pokrenuti na portu `5555`. Server prestaje sa radom prosleđivanjem `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: Shopping cart (25 poena)

Shopping cart (na srpskom *korpa za kupovinu*) je aplikacija koja omogućava klijentima da dodaju i uklanjaju proizvode iz svoje korpe i izvrše plaćanje. Server vodi evidenciju o sadržaju korpe za svakog klijenta posebno.

Implementirati aplikaciju **shopping cart** 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 koji sadrži ime klijenta.
- Server po prijemu imena klijenta šalje klijentu spisak dostupnih namirnica i njihovih cena. Može se pretpostaviti da prodavnica ima neograničene količine svake namirnice.
- Nakon prijave, klijent serveru može da šalje sledeće komande:
  - `ADD <item>` — dodaje navedenu namirnicu u korpu;
  - `REMOVE <item>` — uklanja navedenu namirnicu iz korpe;
  - `VIEW` — prikazuje sadržaj korpe i ukupnu cenu;
  - `PAY` — završava kupovinu, server vraća ukupan iznos i prazni korpu tog klijenta.
- Server vodi evidenciju posebno za svakog klijenta (na osnovu imena) o tome šta se nalazi u njegovoj korpi i koliki je iznos.
- Server mora biti implementiran tako da u petlji stalno prima `DatagramPacket`-e i opslužuje više klijenata.
- Primer rada klijentske aplikacije:

```
Enter your name: Marko
Welcome Marko!
Available groceries:
- bread -- 1$
- milk -- 1.5$
- apple -- 0.7$
- eggs -- 2$

> ADD bread
Added to cart: bread
> ADD milk
Added to cart: milk
> VIEW
Cart: [bread, milk]  Total: 2.5$
> PAY
Total: 2.5$
```

- U slučaju neispravnog unosa (kao na primer unošenje namirnice koja ne postoji u prodavnici, uklanjanje namirnice koja se ne nalazi u korpi) server klijentu vraća poruku o grešci.

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