Računarske mreže

05.06.2024.

Grupa I

Ime i prezime: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Broj indeksa: \_\_\_\_\_\_\_\_

Vrijeme za rad: 45 minuta

1. (10) U programskom jeziku Java napisati dvije klase Client i SensorsDataServer koje implementiraju dva procesa. SensorsDataServer opslužuje klijente vrijednostima očitavanja senzora. Protokol za komunikaciju ova dva procesa implementirati tako da proces Client otvara socket ka SensorsDataServer procesu, koji mu odmah po uspostavljanju konekcije pošalje posljednje očitavanje vrijednosti senzora. Vrijednost koju šalje server je realan broj koji možete slučajno generisati. Nakon što pošalje odgovor, server zatvara soket.
2. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu SensorsDataServer dodati K>1 nizova realnih brojeva dužine N>1 (samostalno odabrati vrijednosti K i N), koji predstavljaju očitavanja K različitih senzora u N vremenskih trenutaka. Protokol za komunikaciju izmijeniti tako da nakon što se konekcija otvori, klijent šalje poruku u kojoj navodi sa kojeg senzora potražuje očitavanja i u kom intervalu. Interval se definiše sa dva indeksa između 0 i N-1. Server zatim šalje potrebne podatke i zatvara soket.
3. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu SensorsDataServer dodati niz stringova koji predstavljaju korisnička imena registrovanih korisnika. Prethodno implementiran protokol izmijeniti tako da nakon otvaranja konekcije server prvo od klijenta traži da pošalje korisničko ime. Klijent učitava i šalje korisničko ime, koje server provjerava. Ukoliko se poslato korisničko ime nalazi u datom nizu, komunikacija se nastavlja kao u prethodnom zadatku, dok u suprotnom server klijentu šalje poruku da korisničko ime nije validno i zatvara soket.

Računarske mreže

05.06.2024.

Grupa II

Ime i prezime: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Broj indeksa: \_\_\_\_\_\_\_\_

Vrijeme za rad: 45 minuta

1. (10) U programskom jeziku Java napisati dvije klase Client i WeatherServer koje implementiraju dva procesa. WeatherServer opslužuje klijente informacijama o temperaturi vazduha. Protokol za komunikaciju ova dva procesa implementirati tako da proces Client otvara socket ka WeatherServer procesu, koji mu odmah po uspostavljanju konekcije pošalje trenutnu vrijednost temperature vazduha. Vrijednost temperature, koju šalje server, je realan broj koji možete slučajno generisati. Nakon što pošalje odgovor, server zatvara soket.
2. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu WeatherServer dodati tri niza realnih brojeva dužine N>1 (samostalno odabrati vrijednosti N), koji redom predstavljaju vrijednosti temperature, vlažnosti vazduha i brzine vjetra u N vremenskih trenutaka. Protokol za komunikaciju izmijeniti tako da nakon što se konekcija otvori, klijent šalje poruku u kojoj navodi koju od tri vrste podataka potražuje i u kom intervalu. Interval se definiše sa dva indeksa između 0 i N-1. Server zatim šalje potrebne podatke i zatvara soket.
3. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu WeatherServer dodati niz stringova koji predstavljaju korisnička imena registrovanih korisnika. Prethodno implementiran protokol izmijeniti tako da nakon otvaranja konekcije server prvo od klijenta traži da pošalje korisničko ime. Klijent učitava i šalje korisničko ime, koje server provjerava. Ukoliko se poslato korisničko ime nalazi u datom nizu, komunikacija se nastavlja kao u prethodnom zadatku, dok u suprotnom server klijentu šalje poruku da korisničko ime nije validno i zatvara soket.

Računarske mreže

05.06.2024.

Grupa III

Ime i prezime: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Broj indeksa: \_\_\_\_\_\_\_\_

Vrijeme za rad: 45 minuta

1. (10) U programskom jeziku Java napisati dvije klase Client i StockPriceServer koje implementiraju dva procesa. StockPriceServer opslužuje klijente informacijama o cijeni akcija jedne kompanije. Protokol za komunikaciju klijenta i servera implementirati tako da proces Client otvara socket ka StockPriceServer procesu, koji mu odmah po uspostavljanju konekcije pošalje trenutnu cijenu akcija kompanije. Cijena akcija koju šalje server je realan broj koji možete slučajno generisati. Nakon što pošalje odgovor, server zatvara soket.
2. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu StockPriceServer dodati K>1 nizova realnih brojeva dužine N>1 (samostalno odabrati vrijednosti K i N), koji predstavljaju cijene akcija K kompanija u N vremenskih trenutaka. Kompanije se identifikuju imenom. Protokol za komunikaciju izmijeniti tako da nakon što se konekcija otvori, klijent šalje poruku u kojoj navodi kompaniju čije cijene akcija potražuje, kao i željeni interval cijena. Interval se definiše sa dva indeksa između 0 i N-1. Server zatim šalje potrebne podatke i zatvara soket.
3. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu StockPriceServer dodati niz stringova koji predstavljaju korisnička imena registrovanih korisnika. Prethodno implementiran protokol izmijeniti tako da nakon otvaranja konekcije server prvo od klijenta traži da pošalje korisničko ime. Klijent učitava i šalje korisničko ime, koje server provjerava. Ukoliko se poslato korisničko ime nalazi u datom nizu, komunikacija se nastavlja kao u prethodnom zadatku, dok u suprotnom server klijentu šalje poruku da korisničko ime nije validno i zatvara soket.

Računarske mreže

05.06.2024.

Grupa IV

Ime i prezime: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Broj indeksa: \_\_\_\_\_\_\_\_

Vrijeme za rad: 45 minuta

1. (10) U programskom jeziku Java napisati dvije klase Client i ElectricityDataServer koje implementiraju dva procesa. ElectricityDataServer opslužuje klijente informacijama o potrošnji električne energije. Protokol za komunikaciju klijenta i servera implementirati tako da proces Client otvara socket ka ElectricityDataServer procesu, koji mu odmah po uspostavljanju konekcije pošalje trenutnu potrošnju električne energije. Vrijednost koju šalje server je realan broj koji možete slučajno generisati. Nakon što pošalje odgovor, server zatvara soket.
2. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu ElectricityDataServer dodati K>1 nizova realnih brojeva dužine N>1 (samostalno odabrati vrijednosti K i N), koji predstavljaju potrošnju električne energije za K gradova u N vremenskih trenutaka. Gradovi se identifikuju imenom. Protokol za komunikaciju izmijeniti tako da nakon što se konekcija otvori, klijent šalje poruku u kojoj navodi grad za koji potražuje informacije o potrošnji električne energije, kao i željeni interval. Interval se definiše sa dva indeksa između 0 i N-1. Server zatim šalje potrebne podatke i zatvara soket.
3. (10) Proširiti rješenje prethodnog zadatka tako što ćete u klasu ElectricityDataServer dodati niz stringova koji predstavljaju korisnička imena registrovanih korisnika. Prethodno implementiran protokol izmijeniti tako da nakon otvaranja konekcije server prvo od klijenta traži da pošalje korisničko ime. Klijent učitava i šalje korisničko ime, koje server provjerava. Ukoliko se poslato korisničko ime nalazi u datom nizu, komunikacija se nastavlja kao u prethodnom zadatku, dok u suprotnom server klijentu šalje poruku da korisničko ime nije validno i zatvara soket.