

Računarske mreže 4I, Ispit - JAN2

24.01.2020.

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 sa imenom **rm_rok_Ime_Prezime_miGGXXX** u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i zameniti svojim podacima.

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: **2.5h**.

1. Daytime klijent-server (20p)

Implementirati serverski program koji povezanim klijentima na svakih 5 sekundi šalje trenutno vreme u formatu: **dd.mm.yyyy | hh:mm:ss** (videti primer poziva ispod).

- Napraviti Java aplikaciju koja ima ulogu servera. Pokrenuti **neblokirajući** lokalni server na portu 12345 koristeći Java Channels API. Kada do servera pristigne klijent, server ga prihvata i započinje obradu. Inicijalno implementirati da server klijentima šalje bilo kakvu nisku koja sadrži trenutno vreme, samo jednom. (3p)
- Postarati se da server klijentima periodično svakih 5 sekundi šalje trenutno vreme. Dakle, svaki klijent dobija novu nisku nakon 5 sekundi. Takođe, server ne raskida vezu do klijenta nakon jednog slanja već mu šalje poruke sve dok klijent ne prekine vezu. (4p)
- Postarati se da je niska koju server šalje klijentima u formatu: **dd.mm.yyyy | hh:mm:ss** (2p)
- Postarati se da server nastavi sa radom ukoliko neki klijent raskine vezu. (2p)
- Napraviti Java aplikaciju koja ima ulogu klijenta. Povezati se na lokalni server na portu 12345 koristeći **blokirajući** Java Channels API i čitati podatke primljene od servera u beskonačnoj petlji. Primljene podatke ispisivati na standardni izlaz bez izmena. (2p)
- Izmeniti klijenta tako da se primanje podataka od servera vrši u zasebnoj niti, dok program u glavnoj niti čeka na unos sa standardnog ulaza koji predstavlja "signal" za prekid rada klijentskog programa. Kada korisnik zatraži zaustavljanje klijenta, klijent se zaustavlja. (6p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka (za oba programa). (1p)

Klijent 1:	Klijent 2:	Server:
[start]		[start]
Connecting...		Accepted client
		Serving time...
24.01.2020 09:00:00	[start]	
	Connecting...	Accepted client
		Serving time...
24.01.2020 09:00:05	24.01.2020 09:00:05	
[end]		Client disconnected
		Serving time...
	24.01.2020 09:00:10	
	[end]	Client disconnected

*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**.*

2. Implementacija daytime protokol handler-a (10p)

Implementirati podršku za URL-ove koji koriste `daytime` protokol. Opis protokola je dat u prvom zadatku.

- Prilikom otvaranja konekcije, formirati vezu koristeći Socket API. Povezati se na server i port na osnovu URL-a i otvoriti ulazni tok do odgovora od strane servera. (5p)
- Ukoliko port nije naveden unutar URL-a, koristiti podrazumevani port isti kao u prvom zadatku. (1p)
- Predefinirati `getInputStream()` metod da vraća ulazni tok do odgovora od strane servera ukoliko je konekcija ostvarena, a `null` ako nije. (1p)
- Postarati se da je moguće bezbedno koristiti implementirani handler u višenitnom okruženju. (1p)
- Napisati jednostavan test - kreirati URL, otvoriti konekciju do resursa i preuzeti sadržaj (videti primere u nastavku). (2p)

```
URL:    daytime://localhost:12345
izlaz:  24.01.2020 | 09:00:00
        24.01.2020 | 09:00:05
        24.01.2020 | 09:00:10
```

```
URL:    daytime://localhost
izlaz:  24.01.2020 | 09:00:00
        24.01.2020 | 09:00:05
        24.01.2020 | 09:00:10
```

```
URL:    daytime://nonexistingserver:12345
izlaz:  Connection failed.
```

```
URL:    daytime://nonexistingserver:9999
izlaz:  Connection failed.
```