

Računarske mreže, Ispit - JUN2 2021

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 u formatu **rm_rok_Ime_Prezime_mXGGXXX** u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i ubaciti svoje podatke u ime. 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!

1. Threads/URL(17p) (za studente koji nisu radili projekat)

U direktorijumu **urls**, unutar direktorijuma **tests** na Desktopu, nalaze se datoteke koje sadrže spisak URL-ova (po jedan u svakoj liniji) i brojeva.

- Za svaku datoteku pokrenuti zasebnu nit koja će je obrađivati i ispisivati statistiku za tu datoteku. (4p)
- Za svaku pročitane liniju datoteke kreirati novi URL objekat koristeći URL klasu. Preskočiti sve linije koje ne predstavljaju validan URL. (2p)
- Za validni URL ispisati protokol i putanju iz URL-a koristeći URL klasu. Izlaz formatirati na sledeći način: PUTANJA_DO_RESURSA KORIŠĆENI_PROTOKOL (5p)

```
ulaz: 3 http://www.matf.bg.ac.rs:3030/dir1/dir2/test.txt
izlaz: /dir1/dir2/test.txt http
```

- Postarati se da se ispisi niti na standardni izlaz ne prepliću. (3p)
- Početak svake linije u datotekama je broj koji označava dužinu linije obrade (u daljem tekstu n). Ukoliko je ime hosta unutar URL-a navedeno putem IP adrese, dodatno uz informacije iznad ispisati linije obrade (matrica $n \times n$ poput matrice u primeru) i informaciju o verziji IP adrese, kao u primeru ispod. (3p)

```
ulaz: 6 http://123.123.123.123:80/dir1/dir2/test.txt
izlaz:
>=====
=>=====
==>=====
===>=====
====>=====
=====>
(v4) /dir1/dir2/test.txt http
ulaz: 4 sftp://2001:0db8:85a3:::8a2e:0370:7334/dir1/dir2/test.txt
izlaz:
>===
=>===
==>===
===>
(v6) /dir1/dir2/test.txt sftp
```

*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**.*

— Okrenite stranu! —

2. Kviz (28p/20p)

Napraviti TCP klijent-server aplikaciju preko koje se korisnici takmiče u kvizu. Server ima ulogu sudije u kvizu i vodi evidenciju o poenima takmičara (klijenata).

- Napraviti Java klasu koja ima ulogu lokalnog TCP servera (koristeći *Java Sockets API*) koji osluškuje na portu 12321. Pri pokretanju, server sa standardnog ulaza učitava nisku koja predstavlja putanju do direktorijuma u kome se nalaze fajlovi sa pitanjima za kviz. Naziv fajla predstavlja naziv oblasti iz koje su pitanja u njemu, a svaki fajl ima format kao u primeru ispod. (4p/3p)
- Napraviti Java klasu koja ima ulogu TCP klijenta (koristeći *Java Sockets API*). Klijent formira konekciju sa lokanim serverom na portu 12321. Nakon prihvatanja svakog novog klijenta, server kreira novu nit koja će preuzeti dalju komunikaciju sa njim. Nakon uspostavljanja konekcije, klijent šalje serveru nisku koja predstavlja ime tog klijenta, učitano sa standardnog ulaza, a zatim od servera dobija nazive oblasti iz kojih može da se takmiči. (6p/5p)
- Pošto klijent odabere oblast iz koje želi da dobija pitanja, započinje kviz. Server šalje klijentu pitanje po pitanje iz odabrane oblasti i čeka 5 sekundi da pročita odgovor od klijenta. (2p/2p)
- U slučaju da takmičar (klijent):
 - (a) zakasni da pošalje odgovor — server mu ne menja rezultat i odgovara porukom:
Niste stigli da odgovorite na vreme.
 - (b) pogrešno odgovori — server umanjuje rezultat tog klijenta i odgovara porukom:
Netačan odgovor. Izgubili ste 1 poen.
 - (c) tačno odgovori — server mu dodaje na trenutni rezultat broj poena koje nosi to pitanje i obaveštava ga o tome porukom:
Tačan odgovor. Osvojili ste <broj poena>.
 - (d) pošalje kao odgovor opciju: Ne znam — server mu ne menja rezultat i odgovara porukom:
Niste znali tačan odgovor.(7p/5p)
- Server sve vreme vodi evidenciju o najbolja 3 rezultata iz svake oblasti. (3p/2p)
- Nakon potrošenih svih pitanja u fajlu, server šalje klijentu poruku Kviz je završen!. Dodatno, ukoliko je rezultat među najbolja 3 rezultata do sada iz oblasti, šalje mu i poruku Medju najbolja tri ste rezultata iz ove oblasti do sada! i ažurira evidenciju o najboljim takmičarima. (3p/2p)
- Ukoliko klijent odustane pre kraja kviza i ne odgovori na sva pitanja, smatrati da je odustao i njegove poene ne računati za ukupan plasman. Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (2p/1p)

```
ulaz: /home/ispit/Desktop/Kviz/Geografija.txt
```

Sadržaj fajla:

- 1.Koji je glavni grad Holandije? Amsterdam 3
- 2.Koje je najveće po površini ostrvo na svetu? Grenland 5
- 3.Koja je najsevernija prestonica na kopnu Evrope? Helsinki 7
- 4.Koja je najduža reka na svetu? Nil 5

*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**.*

— Okrenite stranu! —

3. Protocol handlers (15p/10p)

Implementirati podršku za URL-ove koji koriste `quiz` protokol. Opis protokola je dat u prethodnom 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/3p)
- Omogućiti slanje upita pomoću parametra `oblast` iz URL-a, npr. za upit , kompletan URL bi bio:

```
quiz://localhost:1337?oblast=Geografija
```

Server šalje nazad pitanje koja klijent ispisuje kao u primeru ispod. (5p/3p)

- Ukoliko port nije naveden unutar URL-a, iskoristiti predefinisani podrazumevani port isti kao u prethodnom zadatku. (1p/1p)
- Predefinisati `getInputStream()` metod da vraća ulazni tok do odgovora od strane servera ukoliko je konekcija ostvarena, a `null` ako nije. (1p/1p)
- Postarati se da je moguće bezbedno koristiti implementirani handler u višenitnom okruženju. (1p/1p)
- Napisati jednostavan test - kreirati URL, otvoriti konekciju do resursa i ispisati sve podatke koje server pošalje. (2p/1p)

```
URL:    quiz://localhost
izlaz:  Nije uneta oblast.
```

```
URL:    quiz://localhost:12345?oblast=Geografija
izlaz:  Koji je glavni grad Holandije?
```

```
URL:    quiz://localhost:7337?oblast=Geografija
izlaz:  Neuspela konekcija.
```

```
URL:    quiz://localhost?oblast=x
izlaz:  Nepostojeca oblast.
```