OOP – POPRAVNI 2023/24
Niti

Napisati kod kojim se simulira komunikacija klijenata i servera preko reda za poruke.

Klijent:

- Ima svoj automatski generisan ID.
- Pri konstrukciji dobija red za poruke koji će koristiti.
- Mora prvo da se prijavi kao korisnik reda za poruke slanjem poruke sa tekstom "PRIJAVA".
- Generiše ograničen broj poruka namenjenih serveru (za ovu implemetaciju 3) i šalje ih na red za poruke.
- U tekst poruke upisuje podatke o sebi (Klijent #ID)
- Kada pošalje sve poruke, mora da se odjavi slanjem poruke sa tekstom "ODJAVA".

Server:

- Ima svoj automatski generisan ID.
- Pri konstrukciji dobija red za poruke koji će koristiti.
- U beskonačnoj petlji preuzima poruke sa reda i ispisuje poruku u formatu:
  Server ID primio : string reprezentacija poruke
  Ukoliko uhvati izuzetak tipa Stop, završava sa radom.

Poruka:

- Ima svoj ID.
- Ima tekst poruke.
- Definisanu string reprezentaciju u formatu
  Poruka ID: tekst poruke

Red za poruke:

- Zna koji broj klijenata ga koristi, tj. koliko je klijenata prijavljeno.
- Čuva poruke koje su klijenti poslali, a nisu preuzeti od strane servera.
- Ima ograničen kapacitet za čuvanje poruka (neka bude 3 za ovu implementaciju).
- Zna koliko je ukupno poruka primio od strane klijenata (ne samo koliko ih trenutno ima neobrađenih)
- Na zahtev klijenta za slanje poruke:
  - Ukoliko kapaciteti za čuvanje poruka nisu prevaziđeni, onda preuzima poruku i smeštaje u svoju internu kolekciju poruka.
  - Ukoliko su kapaciteti popunjeni, ostavlja klijenta da čeka.
- Na zahtev servera za preuzimanje poruke:
  - Ukoliko ima nepreuzetih poruka šalje najstariju poruku koju ima i skida je sa svoje evidencije.
  - Ukoliko poruka nema, a ima aktivnih klijenata, ostavlja server da čeka pristizanje poruke.
  - Ukoliko poruka nema i nema prijavljenih klijenata, generiše izuzetak tipa Stop, osim u slučaju da red nije primio ni jednu prijavu od strane klijenta.
    Da bi se izbegla situacija u kojoj je server pokrenut, a još uvek nema prijavljenih klijenata (red još nije korišćen za komunikaciju), red za poruke ima definisan bojač ukupnog broja poruka koje je primio od klijenata (kao indikator korišćenja), zna da ne treba da šalje poruku serveru za zaustavljanje i ostavlja server da čeka pristizanje poruke.

- U jednom trenutku red obrađuje samo jedan zahtev, bez obzira da li slanje i preuzimanje u pitanju.

BITNA NAPOMENA
Ostalo je nerazrešeno ko definiše ID poruke. Imate 3 opcije (Odaberite sami koji ćete implementirati):

- ID poruke dodeljuje sam klijent redom 1,2,3 - **6 poena**
- ID je jedinstven na na nivou reda za poruke – red za poruke ima svoj brojač čiju će vrednost da da klijentu na zahtev – **8 poena**
  Dakle, klijent prvo pita red koji je prvi slobodan ID, a onda generiše poruku koju će poslati na red.
- ID je jedinstven na nivou klase poruka – klasa Poruka ima svoj brojač, mora biti thread-safe i obezbediti da se u jednom trenutku može generisati samo jedan objekat (poruke koje se odnose na prijavu/odjavu klijenta možete, a ne morate brojati) – **10 poena**

Svi podaci koje tipovi sadrže imaju podrazumevanu vidljivost.

Napisati klase kojima se implementiraju tipovi sa navedenim ulogama (**ne zaboravite definisanje izuzetka Stop**). Red za poruke mora biti thread-safe klasa. Klijent i server moraju biti klase koje implementiraju intrefejs Runnable.
U main metodu testne klase generisati jedan red za poruke, tri klijenta koji šalju poruke, i dva servera.