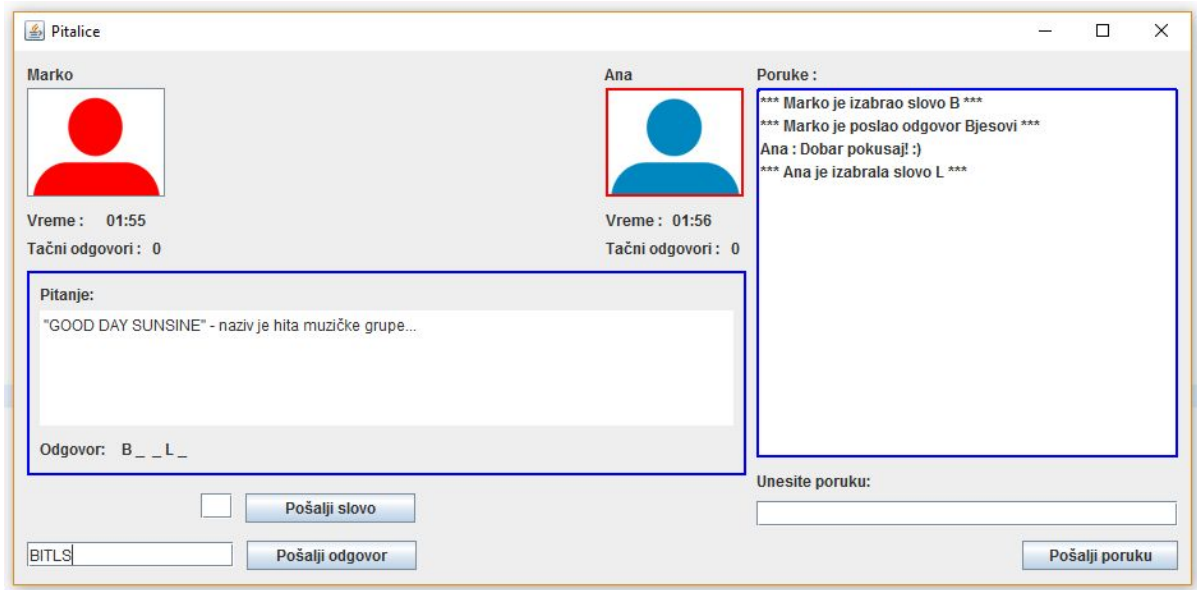


Institut za matematiku i informatiku
PSSS - I kolokvijum
28.04.2018.

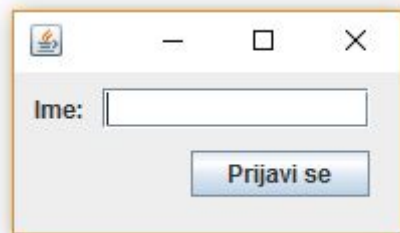
1. **(18 poena)** Napisati na programskom jeziku Java *klijent/server* aplikaciju za igru *Pitalice* (Slika 1).



Slika 1 - Igra *Pitalice*

Programsko rešenje mora imati sledeće ponašanje i karakteristike:

1. Server vodi celu logiku igre tako što čita pitanja iz fajla *pitanja.txt* i servira ih igračima.
2. Server kreira jednu igru za svaka dva igrača koja se pojave.
3. U jednom trenutku mogu biti samo dve aktivne igre.
4. Prilikom startovanja klijentske aplikacije prvo se pojavljuje prozor za prijavu:

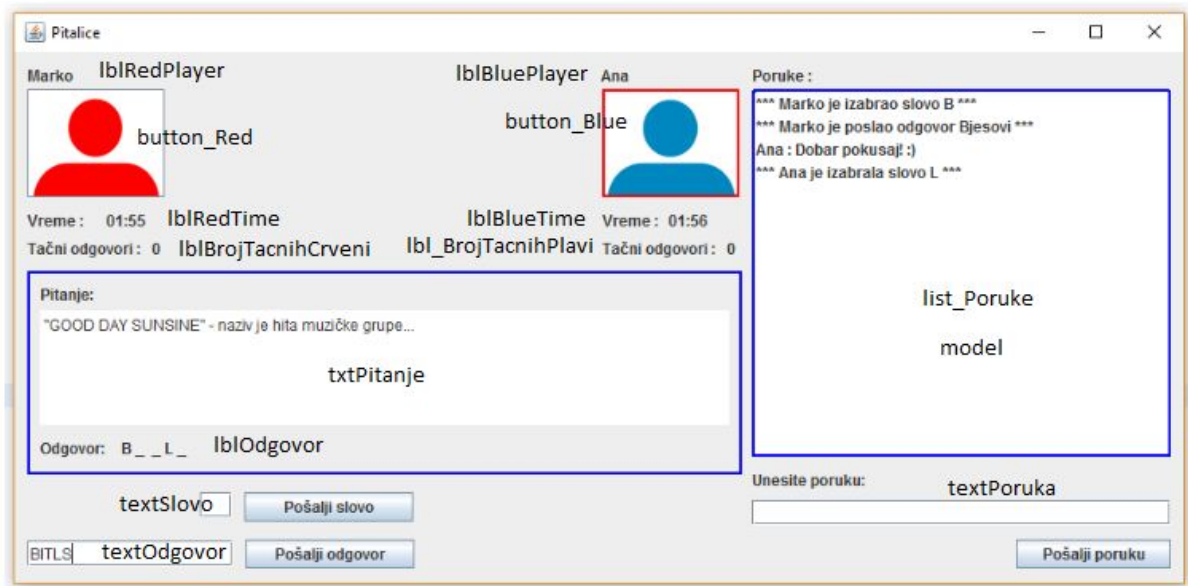


Ako je server zauzet (postoje dve aktivne igre), obavestiti klijenta i zatvoriti formu za prijavu. Ako server nije zauzet, klijentu se pojavljuje prozor kao na *Slici 1*.

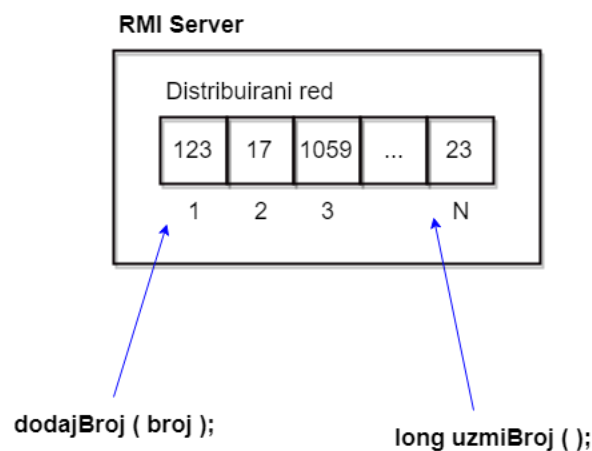
5. Igrači naizmenično pogađaju pitanja. U toku jednog pogađanja, igrač ima opciju da pokuša da pogodi jedno slovo (ne mora da je iskoristi), i da pošalje odgovor. Ako je odgovor tačan, šalje se novo pitanje, a u svakom slučaju je na potezu drugi igrač.
6. Igrač u toku pogađanja troši samo svoje vreme.
7. Slika igrača koji je na potezu ima crveni border (videti u kodu kako se postavlja border).
8. U slučaju da jedan igrač potroši svoje vreme, drugi igrač pogađa pitanja dok ne potroši svoje.

9. Svaka akcija igrača se prikazuje u listi poruka sa desne strane.
10. Igrači mogu razmenjivati poruke pomoću dogmeta "Pošalji poruku".
11. Na kraju igre ispisati u listi poruka ko je pobedio.
12. Kada se igra završi, omogućiti da se startuje nova igra, ukoliko su dve bile aktivne.
13. Aplikacija mora biti *thread-safe*.

Nazivi promenljivih:



2. (6 poena) Kreirati **klijent/server** aplikaciju koja implementira distribuirani red, pomoću *RMI* protokola (Slika 2).



Slika 2 - Distribuirani red

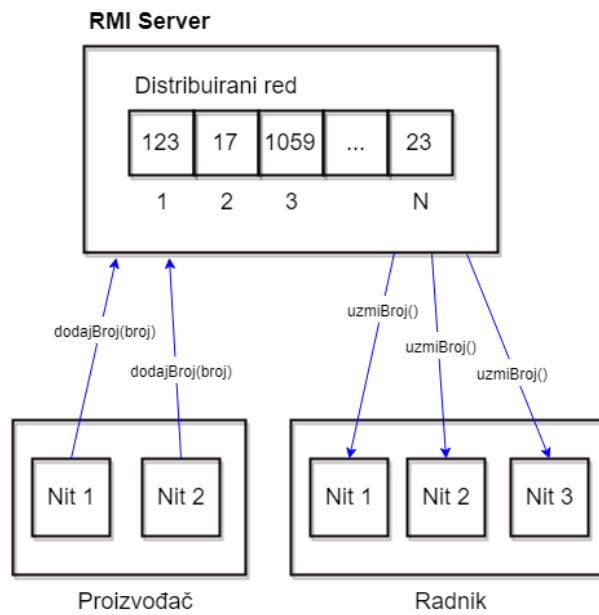
Serverski objekat ima sledeća stanja i ponašanje:

1. Privatni niz/listu dužine N , u kome se nalaze brojevi tipa **long**.
2. Metod **dodajBroj(long broj)** koji je dostupan *RMI* klijentima i pomoću koje oni mogu dodavati brojeve u niz/listu.

3. Metod ***long uzmiBroj()*** koji je dostupan *RMI* klijentima i pomoću kojeg oni mogu uzimati brojeve iz niza/liste.

Test aplikacija sadrži dve klijentske aplikacije (Slika 3):

1. **Proizvođač** - pokreće dve niti koje ukupno šalju 1000 brojeva koje biraju slučajnim izborom iz intervala [0, 1000000]
2. **Radnik** - pokreće tri niti koje uzimaju ukupno 1000 brojeva iz deljenog reda i za svaki broj proverava da li je prost. Ako je broj prost, ispisuju ga na standardni izlaz.



Slika 2 - Test aplikacija

Napomena: Aplikacija mora biti *thread-safe*.