

Prirodno-matematički fakultet Kragujevac
Institut za matematiku i informatiku
II kolokvijum iz Programiranja složenih softverskih sistema
26.1.2014.

1. **Tvrđenje:** Svaki paran broj veći od 2 može se predstaviti kao zbir dva prosta broja.
-Potrebno je proveriti da li tvrđenje važi za sve parne brojeve manje od **n** (zadaje se kao ulazni argument). Za potrebe rešenja potrebno je implementirati:
 - 1) Serverski *“remote”* objekat korišćenjem Java RMI API-a. Serverski objekat implementira metod ***List<Long> GetPrimeNumbers(Long number)***; koji za prosleđeni paran broj vraća listu od dva prosta broja koja zadovoljavaju tvrđenje. Ukoliko takvi brojevi ne postoje vratiti praznu listu. Potrebno je kreirati dve instance serverskog objekta i vezati ih na localhost-u na dva različita porta.
 - 2) Klijentski program koji za zadati broj **n** u argumentu za svaki paran koji se nalazi u intervalu (2, n) pokušava da pronađe dva prosta broja koja zadovoljavaju tvrđenje i ispisuje ih na standardnom izlazu u formatu **„*paran_broj: p1, p2*”**. Ukoliko prosti brojevi ne postoje štampa se **„*paran_broj: blank, blank*”**. Klijent mora koristiti oba *“remote”* objekta i pozivati njihove metode čim završe dokaz za prethodno poslat paran broj. Na kraju programa ispisati poruku, na standardnom izlazu, da li je tvrđenje tačno ili nije.
2. Kreirati program koji pokreće dve niti. Prva nit kreira MulticastSocket na adresi **228.5.6.7:5776** i setuje vreme važenja paketa na 1. Zatim u beskonačnoj petlji šalje poruku **„*Poruka prve niti.*”** na svakih random sekundi iz intervala [1, 4]. Druga nit se priključuje multicast grupi koja je kreirana od strane prve niti, prihvata sve poruke i štampa ih na standardnom izlazu u formatu **“*Stigla je poruka: poruka*”**. Potrebno je implementirati i mehanizam u kome će i prva i druga nit odlaziti u stanje **“*sleep*”** na svakih 10 primljenih poruka druge niti.
3. Nacrtati arhitekturu neblokirajućih soketa i napisati objašnjenje za svaki deo arhitekture.