

## Programski jezici 2

– 15.06.2022. –

**1. (30)** Stanovnici Londona 1984. godine bili su podijeljeni su u tri grupe: radnici, nadzornici i prepravljači. Svakog stanovnika karakteriše ime, prezime i matični broj. Radnici, pored navedenih podataka, imaju još i platu. Radnici mogu da razmjenjuju poruke preko dijeljenog *buffer*-a poruka, u kom se poruke čuvaju kao objekat koji ima matični broj pošiljaoca, matični broj primaoca, vrijeme slanja i tekst poruke. Nadzornici imaju mogućnost pregleda poruka svakog od radnika i pretraživanje poruka po ključnim riječima, kao i slanje poruka radnicima i prepravljačima na *teleekran*, dok prepravljači imaju mogućnost pregleda poruka svakog od radnika i njihovo prepravljanje. *Teleekran* je posebna nit vezana za radnike i prepravljače koja prikazuje tekstualne poruke koje nadzornik šalje. Na teleekranu se prikazuju sva obavještenja (prikaz liste poruka, prikaz izmjena u porukama i sl.) Takođe, svaki radnik je zasebna nit.

Prepravljači vrše sljedeće: očitavaju sadržaj poruke i vrše zamjenu svih riječi iz prve kolone u tabeli prikazanoj ispod riječima iz druge kolone u tabeli, snimaju prepravljeni sadržaj u isti objekat i vraćaju objekat nazad u *buffer*. U tabeli mora biti najmanje 7 parova riječi za zamjenu. Nakon što se izvrši zamjena riječi, prepravljač šalje nadzorniku poruku o broju izvršenih izmjena preko teleekrana.

Riječ	Zamjena
ljubav	mržnja
sloboda	nadzor
neznanje	moć
nešto	neštodrugo
...	...

Nadzornici mogu da vrše pretragu po nizu ključnih riječi koje je potrebno pretražiti u tekstovima poruka za trenutni datum. Pretragom se vraća lista matičnih brojeva pošiljaoca poruka koji sadrže navedene riječi. Po prijemu matičnih brojeva, nadzornici šalju poruku radnicima sa odgovarajućim matičnim brojevima na teleekran. Poruka je sljedeća: **“Zbog neadekvatnog ponašanja, vaša plata će biti umanjena za x.”**, nakon čega se vrši i ažuriranje plate radnika - plata se umanjuje za x koji predstavlja slučajan broj u opsegu od 100 do 1000.

Simulacija “Big Brother” se sastoji od sljedećeg:

1. kreira se proizvoljno odabran broj radnika (minimalno dva),
2. kreira se jedan nadzornik i jedan prepravljač,
3. radnici obavljaju komunikaciju preko dijeljenog *buffer*-a (u *buffer* se smješta po 5 poruka od svakog radnika),
4. nadzornik i prepravljač obavljaju odgovarajuće poslove,
5. nadzornik može, u bilo kom trenutku, poslati poruku na teleekran; slanje poruke se inicira unosom ključne riječi poruka, nakon koje slijedi tekst poruke koju je potrebno poslati.

Koraci od 3 do 5 mogu se ponavljati neograničen broj puta. Simulacija može da se prekine unosom ključne riječi ZAUSTAVI, nakon čega se na ekranu prikazuje spisak svih radnika sa svim njihovim podacima i

vrijeme trajanja simulacije. Simulacija se pauzira unosom ključne riječi PAUZA, a nastavak izvršavanja se obavlja unosom ključne riječi NASTAVAK. Tokom pauze vrijeme trajanja simulacije se ne mjeri. Ukoliko pauza traje više od minute, simulacija se prekida (kao u slučaju unosa ključne riječi ZAUSTAVI).

**2. (20)** Neka su data tri tipa datoteka: predmet (čuva se naziv i identifikator predmeta), student (čuva se ime, prezime, broj indeksa i lista identifikatora predmeta) i profesor (čuva se ime, prezime, jmb i lista idenfitikatora predmeta). Napisati program koji iz određenog direktorijuma koji je dat kao argument komandne linije u realnom vremenu čita prethodno navedene tipove datoteka (primjer datoteka je dat na stranici predmeta), te za svaki novi predmet kreira novi direktorijum sa nazivom predmeta, a u njemu dalje direktorijume (studenti, profesori) sa studentima i profesorima koji su već dodani ili će se dodati kasnije. Pri tom, za svakog studenta ili profesora kreira se odvojen direktorijum (naziv broj indeksa ili jmb i zbog jednostavnosti neka budu cjelobrojnog tipa) u kojem se nalazi binarno serijalizovan objekat datog studenta ili profesora koji je kreiran na osnovu ulaznog fajla.

**3. (20)** Napisati generičku metodu *processData* koja kao argumente prima listu *Predicate* objekata koji su uslovi koji će biti testirani nad podacima, jedan *Consumer* objekat i neodređen broj listi podataka (više njih, ne lista listi) koje implementiraju interfejs *Data*. Interfejs *Data* ima metode *getType* i *getValue*. U metodi se pomoću *Predicate* uslova iz svake liste uzimaju samo oni elementi koji ispunjavaju sve proslijeđene uslove. Lista rezultata je sortirana od elementa koji ima najveću hashCode vrijednost ka objektima sa manjim vrijednostima. Dobijeni rezultati se obrađuju pomoću *Consumer* objekta. Napraviti klase koje implementiraju interfejs i imaju attribute: klasa *ExampleData1* (type, value, color) i *ExampleData2* (type, value, size). U *main* metodi potrebno je napraviti po jednu listu objekata za svaku klasu (sa 3 elementa), 3 uslova vezana za vrijednosti atributa i testirati generičku metodu (vrijednost *consumer-a* može biti ispis na konzolu).

**Vrijeme za rad: 180 minuta**