

1. Napraviti niz od N niti koje broje od 1 do milijardu, tako što na svakih milion brojeva naprave pauzu od 2 milisekunde. Na konzoli, za svaku od niti, odštampati vrijeme koje je prošlo od trenutka kada je nit startovana do trenutka kada je završila, izraženo u milisekundama.
2. U programskom jeziku Java napisati klasu Brojac koja implementira nit koja inkrementira brojačku promjenljivu sve dok određeni fleg nije setovan. Nakon svakog inkrementiranja, nit spava između 10 i 100 milisekundi (generisati slučajno). Iz glavne niti kontrole pokrenuti implementiranu nit i nakon 5 sekundi setovati fleg kako bi se brojačka nit zaustavila. Nakon toga, odštampati vrijednost brojačke promjenljive.
3. U programskom jeziku Java napisati klase Generator i Tajmer. Klasa Generator implementira tred koji generiše slučajne prirodne brojeve manje od 1000 sve dok određeni fleg nije setovan. Tajmer sačeka 5 sekundi, a zatim setuje fleg za tred Generator. U glavnoj niti kontrole pokrenuti ove dvije niti i kada se one završe odštampati koliko je vremena proteklo od njihovog startovanja do tog trenutka.
4. Kreirati nit koja računa vrijednost sinusa ugla datog u stepenima, kao i nit koja računa proizvod dva data realna broja. Zatim, pomoću ovih niti, izračunati i odštampati površinu trougla ako su date dužine dviju stranica i ugao između njih.
5. Kreirati klasu MaxThread koja sadrži tri polja: array - niz realnih brojeva, start – prirodan broj, end – prirodan broj. MaxThread implementirati kao nit koja u datom nizu pronalazi najveći element u opsegu od elementa na indeksu start do indeksa end. U glavnoj niti kontrole kreirati niz od milion slučajno izgenerisanih realnih brojeva iz opsega $(-10^6, 10^6)$ i pronaći maksimum tog niza uz pomoć tri MaxThread-a. Potrebno je ravnomjerno podijeliti posao između tredova.
6. U programskom jeziku Java napisati program koji sortira dati niz cijelih brojeva tako što podijeli taj posao između barem 3 niti.
7. U programskom jeziku Java napisati klasu SkalarniProizvod, čija su polja dva niza cijelih brojeva jednake dužine i jedno cjelobrojno polje. Ova klasa implementira nit koja računa skalarni proizvod dva niza i upisuje ga u cjelobrojno polje. U glavnoj niti kontrole napraviti dvije cjelobrojne matrice A i B dimenzija $M \times K$ i $K \times N$, respektivno, a zatim napraviti i pokrenuti $M \times N$ niti za računanje skalarnog proizvoda. Svako od ovih niti predati kopiju jednog reda matrice A i kopiju jedne kolone matrice B, tako da se, nakon što niti izračunaju odgovarajuće skalarne proizvode, dobije vrijednost svake ćelije za matricu proizvoda A i B. Rezultate množenja upisati u rezultujuću matricu i odštampati je.
8. Porodica broji 4 člana koji imaju zajednički bankovni račun. Na računu se na početku nalazi 10 000 eura. Svaki član porodice nasumično izabere da li će podići ili uplatiti novac na račun. Suma koja se podiže ili uplaćuje se takođe bira slučajno (od 5 do 500 eura). Napisati program koji simulira opisani proces korišćenjem niti. Postarati se da račun ne dođe u nekonzistentno stanje. Štampati poruku kada neki od članova porodice izvrši transakciju. Potrebno je odštampati ime osobe, koju operaciju vrši (podizanje ili uplaćivanje novca) i u kolikom iznosu, kao i stanje na računu nakon izvršene transakcije. Svi vremenski intervali mogu biti uzeti proizvoljno. Obavezno napisati klasu iz koje se pokreće simulacija.

9. Tri studenta žive u istom stanu. Svaki student ima svoju sobu, a kuhinja je zajednička. Studenti dugo uče, zbog čega ogladne i poželevi sebi spremi obrok. Međutim, kuhinja je mala i u njoj u jednom trenutku može boraviti samo jedan student koji sprema obrok. Za to vrijeme, preostali studenti koji žele koristiti kuhinju moraju strpljivo čekati. U programskom jeziku Java napisati višenitni program koji simulira život studenata u ovom stanu. Život studenata protiče u učenju, pripremi obroka, jelu i spavanju. Trajanje svake od ovih aktivnosti možete proizvoljno definisati. Postarati se da ne dođe do deadlock-a. Štampati poruku kada student:
- uči,
 - sprema obrok,
 - jede,
 - spava.
10. Postoji jedan proizvođač i N potrošača koji dijele zajednički bafer kapaciteta B. Proizvođač ubacuje proizvod u bafer i to samo u slobodne slotove na kojima čekaju svih N potrošača. Svaki potrošač mora da primi proizvod tačno u onom redoslijedu u kojem su proizvedeni. Kada svaki potrošač uzme određeni proizvod on se uklanja iz bafera i taj slot se oslobađa. Napisati program koji simulira opisani sistem. Spriječiti blokiranje. Obavezno napisati klasu iz koje se pokreće simulacija. Štampati poruku kada proizvođač ubaci novi proizvod u bafer i kada neki potrošač uzme taj proizvod.
11. U programskom jeziku Java napisati klase Generator i Brojac. Klasa Generator implementira tred koji generiše million slučajnih prirodnih brojeva manjih od 1000. Klasa Brojac implementira tred koji broji koliko je savršenih brojeva generisao tred Generator. U glavnoj niti kontrole pokrenuti ove dvije niti i odštampati koliko je prostih brojeva generisano.
12. Kreirati dvije klase **A** i **B** koje predstavljaju dva treda. Tred **A** na standardnom izlazu treba da odštampa dvije poruke:

Tred A prva poruka.
Tred A druga poruka.

Tred B takođe na standardnom izlazu štampa sljedeće dvije poruke:

Tred B prva poruka.
Tred B druga poruka.

Napisati program tako što ćete osigurati da se prilikom svakog pokretanja programa poruke odštampaju u sljedećem redoslijedu:

Tred A prva poruka.
Tred B prva poruka.
Tred B druga poruka.
Tred A druga poruka.

Tredovi se moraju pokrenuti istovremeno.

13. U računarskoj sali postoji N računara i jedan štampač. Svaki od računara, nakon slučajno odabranog vremenskog intervala, šalje zahtjev za štampu određene riječi. Napisati program koji simulira opisani sistem. Obavezno napisati klasu iz koje se pokreće simulacija. Riječi koje računari šalju na štampu ispisivati u fajlu „stamp.txt“. Obezbijediti da dok se izvršava jedan zahtjev za štampu ostali zahtjevi čekaju.
14. U programskom jeziku Java napisati klasu koja predstavlja nit koja na standardnom izlazu štampa dvije poruke:

Došla sam na barijeru i čekam ostale!

Napuštam barijeru!

U glavnoj niti kontrole učitati prirodan broj $N > 3$ i napraviti i pokrenuti N ovakvih niti.

- a. Sinhronizovati kreirane niti tako da nijedna nit ne odštampa drugu poruku dok sve niti ne odštampaju prvu. U sinhronizaciji se ne smije oslanjati na funkciju sleep, koliko god bio dugačak interval uspavlivanja niti. Na primjer, za $N=4$, standardni izlaz treba da izgleda ovako:

Došla sam na barijeru i čekam ostale!

Došla sam na barijeru i čekam ostale!

Došla sam na barijeru i čekam ostale!

Došla sam na barijeru i čekam ostale!

Napuštam barijeru!

Napuštam barijeru!

Napuštam barijeru!

Napuštam barijeru!

- b. Proširiti prethodno rješenje tako da se nakon jednog ciklusa štampanja poruka niti opet vraćaju na štampanje prve poruke i opet se sinhronizuju kao u zadatku pod a.
15. Postoji jedan agent i tri nervozna pušača. Agent posjeduje rezerve tri neophodna predmeta za liječenje nervoze: papir, duvan i šibice. Jedan od pušača ima beskonačne zalihe papira, drugi duvana, a treći šibice. Agent počinje tako što dva različita predmeta stavlja na sto, jedan po jedan. Pušač kojem baš ta dva predmeta fale, uzima ih, zavija i pali cigaretu i uživa. Nakon toga obavještava agenta da je završio, a agent onda stavlja dva nova predmeta na sto itd.

16. N studenata, označenih brojevima od 1 do N, došli su da polažu ispit. Profesor im redom dijeli zadatke (od 1 do N). Profesoru je potrebno određeno vrijeme da dođe do svakog studenta i preda mu papir sa zadacima. Studenti se na ispitu moraju pridržavati sljedećih pravila. Niko ne smije početi sa radom dok svi ne dobiju zadatke. Takođe, niko ne smije napustiti ispit dok svi ne završe. Svakom studentu potrebno je različito vrijeme da uradi zadatke. U programskom jeziku Java simulirati opisanu situaciju pomoću niti. Studente i profesora predstaviti kao niti. Sinhronizacijom obezbijediti da se poštuju pravila ispita. Sve vremenske intervale uzeti proizvoljno ili slučajno generisati. Štampati sljedeće poruke:

- kada student čeka da dobije ispitne zadatke,
- kada student dobije ispitne zadatke,
- kada student krene da radi zadatke,
- kada student izađe sa ispita.

17. Data je matrica dimenzija $N \times N$ čija svaka ćelija predstavlja jedan organizam koji može biti živ ili mrtav. Organizmi mogu da komuniciraju samo sa svojim susjedima. Organizmi su susjedi ako se nalaze u ćelijama koje imaju zajedničku ivicu. Važe sljedeća pravila:

- Živ organizam koji ima manje od dva živa susjeda umire od usamljenosti.
- Živ organizam sa više od tri živa susjeda umire od prenatrpanosti.
- Živ organizam sa dva ili tri živa susjeda preživljava.
- Mrtav organizam sa tri živa susjeda oživljava.

U programskom jeziku Java implementirati program koji simulira opisanu situaciju i sprječava blokiranje. Sve organizme implementirati kao posebne niti. Na početku, na slučajan način raspodijeliti žive i mrtve organizme u matrici. Štampati poruku svaki put kada se stanje nekog organizma promijeni, kao i čitavu matricu na kraju simulacije. Simulaciju vršiti proizvoljan broj sekundi. Sve vremenske intervale uzeti proizvoljno.

18. Grupa od N putnika odlučila je da posjeti luna park. Putnici se naizmjenično šetaju po luna parku i voze na rolerkosteru. Rolerkoster može da primi najviše K putnika, pri čemu je $K \ll N$. Vožnja toboganom može da počne samo kada se skupilo tačno K putnika. U programskom jeziku Java napisati program koji simulira opisani sistem korišćenjem niti. Potrebno je spriječiti da dođe do nevalidne situacije i blokiranja. Obavezno napisati klasu iz koje se pokreće simulacija. Štampati poruku kada se putnik šeta po luna parku, kada putnik čeka da se ukrca na rolerkoster, kada se putnik ukrca na rolerkoster, kada počne vožnja, kada se vožnja završi i kada se putnik iskrca sa rolerkosteru. Sve vremenske intervale (trajanje vožnje, pauza između vožnji, trajanje šetnje) uzeti proizvoljno.

19. Most sa jednom saobraćajnom trakom povezuje dva grada. Automobili iz oba grada koriste taj most kako bi prešli iz jednog u drugi grad. Most može postati blokiran ako se automobili iz oba pravca istovremeno nađu na mostu. Da bi se riješile ovaj problem, gradske vlasti su postavile semafore (sa crvenim i zelenim svjetlom) na oba kraja mosta. Kada je na jednom semaforu upaljeno crveno svjetlo, na drugom je upaljeno zeleno. Ukoliko je upaljeno crveno svjetlo na semaforu, automobili ne

mogu preći most čak i ako sa druge strane nema vozila. Svijetla se mijenjaju nakon određenog vremena. Nakon što automobil pređe iz jednog u drugi grad, on tamo ostaje neko vrijeme, a zatim se vraća nazad i tako u krug. Implementirati algoritam koji simulira opisanu situaciju korišćenjem niti i sprječava da most postane blokiran. Štampati poruke kada neki automobil boravi u gradu, kada čeka na mostu i kada prelazi most. Svi vremenski intervali mogu biti uzeti proizvoljno. Obavezno napisati klasu iz koje se pokreće simulacija.

20. U jednom gradu u živi N osoba. Grad ima M autobuskih stanica. Nažalost, postoji samo jedan autobus koji može prevoziti K putnika ($K \leq N$) od stanice 1 do stanice M i nazad. Građani se nalaze na proizvoljnim stanicama na kojima čekaju autobus da ih poveze do određene stanice, nakon čega se oni određeno vrijeme šetaju po tom dijelu grada. Kada autobus dođe na neku stanicu, najprije putnici koji su izrazili želju da izađu na toj stanici to i učine. Nakon njihovog izlaska, svi putnici koji su čekali na ulazak uđu u autobus, ali samo ako ima mjesta i ukoliko autobus ide ka stanici do koje oni žele da odu. U suprotnom, putnici ostaju da čekaju na stanici. Napisati program u programskom jeziku Java koji simulira opisanu situaciju. Štampati poruku kada autobus dođe na određenu stanicu, kada neki od građana dođe na stanicu, uđe ili izađe iz autobusa.
21. U frizerskom salonu radi jedan frizer. Pored stolice koju on koristi za rad, u salonu postoji dodatnih N stolica za čekanje. Ukoliko mušterija dođe u salon i ako su sve stolice su zauzete, on odustaje od šišanja i odlazi iz salona. Ukoliko je neka od stolica za čekanje je slobodna, on sijeda na stolicu i čeka da dođe na red za šišanje. Frizer mušterije opslužuje u redu u kojem su došle. On odmara kada god za to ima priliku. Ukoliko nema mušterija koje čekaju, frizer će otići da odmara. U programskom jeziku Java napisati program koji simulira opisani sistem pomoću niti. Potrebno je spriječiti da dođe do nevalidne situacije i blokiranja. Obavezno napisati klasu iz koje se pokreće simulacija. Štampati sljedeće poruke:
- kada mušterija uđe u salon i koju odluku donosi, da li čeka red ili odlazi,
 - kada mušterija dođe na red za šišanje,
 - kada šišanje počne,
 - kada se šišanje završi,
 - kada mušterija napusti salon,
 - kada frizer odmara.