1. Zadatak

Simulirati rad štampača na koji se dokumenti prosleđuju parcijalno, odnosno reč po reč. Korišćenjem **synchronized bloka** omogućiti da niti vrše štampu tako da se izbegne međusobno uplitanje teksta iz različitih dokumenata, niti.

1. Štampač je predstavljen klasom `Stampac` koja ima samo jednu metodu `stampaj` čiji je parametar tipa `String`. Ova metoda na ekranu ispisuje prosleđen tekst pomoću `System.out.print(...)`
2. `Dokument` je nit koja u svom konstruktoru prihvata instancu klase `Stampac`, dva parametra tipa `String` koji predstavljaju reči i još jedan parametar tipa String koji predstavlja naziv dokumenta. Tokom izvršavanja, nit prosleđuje najpre jednu reč instanci štampača, zatim pauzira pola sekunde i potom prosleđuje i drugu reč.
3. Ostale niti moraju da čekaju da trenutna nit završi sa štampanjem. Niti svakih 200 milisekundi proveravaju da li je moguće štampanje. Tokom svake provere na konzoli ispisati tekst "`Provera zauzeća štampača za dokument "+nazivDokumenta`.
4. U klasi `Stampanje` implementirati `main` metodu kojom se instancira jedan štampač i deset niti za štampanje. Svako štampanje instancirati uz navođenje reference na štampač i reči "`hello`" i "`world\n`". Pokrenuti sve niti za štampanje.

2. Zadatak

Banka rukuje transakcijama koje menjaju stanja računa. Na kraju, kada se sve transakcije završe, ispisuje se stanje svih računa. Postoji 4 računa sa po 3 transakcije. Sinhronizaciju izvesti korišćenjem **synchronized bloka.**

1. Račun je predstavljen klasom `Racun` i ima atribute `String imeVlasnika` i `double stanjeRacuna`.
2. Transakcija je nit i poseduje dve promenljive: referencu na račun (`Racun racun`) i promenu računa (`double promenaRacuna`).
3. Transakciji treba 1.5 sekundi da izračuna novu vrednost (`sleep`). Nakon toga menja stanje računa.
4. Instancirati 4 računa sa po 3 transakcije. Računima staviti 0 kao početno stanje, a transakcijama simulirati uplate čiji je iznos 1000.
5. Nakon što se sve transakcije završe, ispisati stanja svih računa: "Racun korisnika <imeVlasnika> je <stanjeRacuna>".

3. Zadatak

Implementirati aplikaciju koja simulira utovar paleta iz magacina na kamione pomoću viljškara.

Svaki viljuškar prenosi po jednu paletu. Sinhronizaciju izvesti korišćenjem **synchronized metoda.**

1. Klasa `Magacin` u svom konstruktoru prihvata celobrojnu vrednost ukupnog broja paleta koje se nalaze u magacinu. Ova klasa implementira metodu `uzmiPaletu()` koji vraća logičku vrednost `false` ukoliko u magacinu nema više paleta, a u suprotnom umanjuje njihovo brojno stanje za 1 i vraća logičku vrednost `true`.

2. Klasa `Kamion` u svom konstruktoru prihvata celobrojnu vrednost maksimalnog broja paleta koje kamion može nositi. Klasa poseduje atribut koji predstavlja ukupan broj utovarenih paleta i inicijalno se postavlja na nulu. Klasa implementira metodu `utovariPaletu()` koji vraća logičku vrednost `false` ukoliko u kamion više ne može stati paleta, dok u suprotnom uvećava njihovo brojno stanje za 1 i vraća logičku vrednosti `true`.

3. Napraviti nit za svaki viljuškar: `Viljuskar` koji u svom konstruktoru prihvata referencu na objekat klase `Magacin`, kao i na kolekciju instanci klase `Kamion` (koristiti `ArrayList`). Nit tokom svog izvršavanja ciklično pokušava preuzimanje palete iz magacina. Ako je preuzimanje palete uspešno, nit se uspava na pola sekunde (prevoženje palete do kamiona) i potom se pokušava utovar redom na neki od kamiona.

4. Klasa Utovar poseduje `main` metodu sa kojom se kreira instancu klase Magacin sa 100 paleta. Potom kreirati 10 instanci klase `Kamion` takvih da mogu poneti 10 paleta i smestiti ih u `ArrayList` kolekciju. Instancirati i 10 objekata klase `Viljuskar` i proslediti im reference na magacin i kolekciju kamiona. Pokrenuti niti svih viljuškara i sačekati da se završe. Na kraju ispisati broj utovarenih paleta za svaki od kamiona.