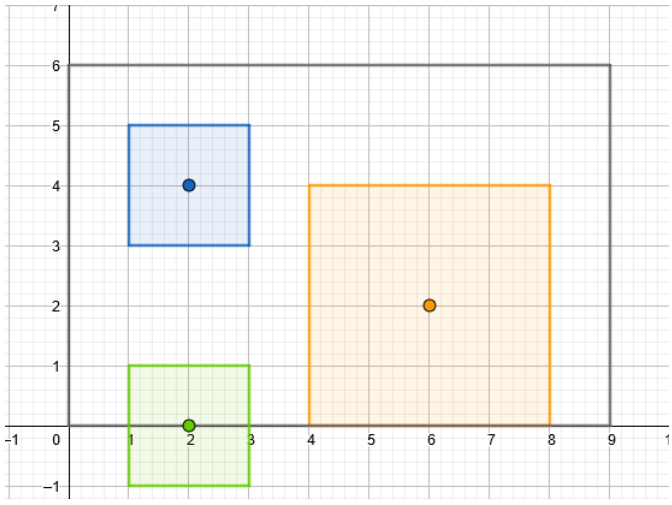
Računarske mreže, Ispit - SEP1 2022

Pročitati sve zadatke **pažljivo** pre rada - sve što nije navedeno ne mora da se implementira!
Na `Desktop`-u se nalazi zip arhiva. Unutar arhive se nalazi direktorijum u formatu `rm_rok_Ime_Prezime_mXGGXXX` u kome se nalazi validan IntelliJ projekat. Izvući direktorijum iz arhive na Desktop i ubaciti svoje podatke u ime. Otvoriti IntelliJ IDEA, izabrati opciju `Open project` (ne `Import project`!) i otvoriti pomenuti direktorijum. Sve kodove ostaviti unutar već kreiranih Java fajlova. **Kodovi koji se ne prevode se neće pregledati. Nepoštovanje formata ulaza/izlaza nosi kaznu od -10% poena na zadatku!**

1. **Magične matrice (20p)**
   Napisati Java aplikaciju koja proverava da li su zadate matrice *magične*. Kvadratna matrica je magična ako su sume elemenata u svim vrstama, svim kolonama i na glavnoj i sporednoj dijagonali jednake. U direktorijumu `tests` na `Desktop-u` se nalaze datoteke koje sadrže matrice — prvi broj predstavlja broj vrsta kvadratne matrice, nakon čega slede elementi matrice. Pretpostaviti da su dimenzije matrice ispravne i da su svi elementi matrice različiti.

   - Rekurzivno obići direktorijum `tests` i za svaku datoteku pokrenuti zasebnu nit koja je pretražuje. (4p)
   - Za svaku datoteku ispisati njen naziv i da li je zadata matrica magična ili ne. (6p)
   - Postarati se da se ispisi svake niti na standardni izlaz ne prepliću. (5p)
   - Ispisati koliko je ukupno pronađenih magičnih matrica (paziti na sinhronizaciju niti ukoliko se koristi deljeni brojač!). (3p)
   - Postarati se da program ispravno obrađuje specijalne slučajeve i ispravno zatvara sve korišćene resurse. (2p)

2. **Non-Blocking IO (25p)**
   Za potrebe skeniranja terena odlučeno je da se kreira *hab* (server) koji će prikupljati podatke koji se dobijaju od *skenera* (klijenata). Svaki skener se odlikuje svojom pozicijom (x, y) i može da pokrije odgovarajuću oblast radijusa r. Cilj je pokriti čitav teren skenerima. Teren se posmatra kao mreža jediničnih kvadrata dužine m i širine n (videti sliku ispod), dok je (x, y) jedno teme mreže. Skener pokriva kvadrat veličine $2r$ sa centrom u (x, y). m, n, x, y i r su nenegativni celi brojevi.



Slika 1: Teren dimenzija 9×6 sa tri postavljena skenera. Pokrivenost u ovom slučaju: $\frac{22}{9 \times 6} \approx 0.407 = 40.7\%$

- Napraviti Java aplikaciju koja ima ulogu skenera. Povezati se na lokalni server na portu 7337 koristeći **blokirajući Java Channels API**. Nakon formiranja konekcije, klijent serveru šalje brojeve (x, y) i r koje operater skenera unosi sa standardnog ulaza. Nakon slanja, klijent ispisuje odgovore od servera sve dok server ne prekine vezu. (4p)
- Napraviti Java aplikaciju koja ima ulogu centralnog haba. Pokrenuti lokalni server na portu 7337, koristeći **neblokirajući Java Channels API**. Prilikom pokretanja, operater haba unosi veličinu terena — brojeve m i n. Hab zatim opslužuje skenere. Nakon uspostavljanja veze, hab prima podatke od skenera — (x, y) i r (pozicija skenera mora biti unutar granica terena, u protivnom se raskida veza sa tim skenerom). U međuvremenu, na svakih 5 sekundi, hab svakom od skenera šalje broj drugih skenera cija se pokrivena teritorija preseca sa pozicijom trenutnog skenera. (9p)
- Server salje indikator svim skenerima ukoliko procenat pokrivenosti terena dostigne 100%. (2p)
- Obezbediti da u slučaju izuzetaka, svi resursi budu ispravno zatvoreni i da se ukupna pokrivenost terena eventualno promeni ukoliko je neki skener prekinuo vezu! (3p)

3. **UDP Sockets (15p)**

Napisati Java aplikaciju koja računa zbir svih prirodnih brojeva u zadatom opsegu.

- Napisati Java klasu koja ima ulogu UDP klijenta. Sa standardnog ulaza učitati dva prirodna broja. Poslati UDP datagram koji sadrži dva učitana broja lokalnom serveru na portu 12345 koristeći `DatagramPacket` klasu. (3p)
- Napisati Java klasu koja ima ulogu UDP servera. Slušati na portu 12345 i primati datagrame od klijenata koristeći `DatagramPacket` klasu. Sadržaj svakog datagrama su dva prirodna broja. (3p)
- Server očitava sadržaj datagrama i računa zbir svih brojeva u opsegu [N1, N2] (inkluzivno). (3p)
- Postarati se da je operacija računanja zbira složenosti O(1). (2p)
- U slučaju nevalidnog opsega (negativni brojevi ili prvi broj veći od drugog), poslati klijentu poruku "Nevalidan opseg!". (2p)
- Postarati se da su svi resursi ispravno zatvoreni u slučaju izuzetka. (2p)

Napomena: Ohrabrujemo studente da koriste `netcat` *kako bi testirali delimične implementacije i otkrili greške pre vremena. Takodje, ukoliko se npr. preskoči implementacija servera, može se mock-ovati server putem* `netcat`*-a.*

Okrenite stranu!