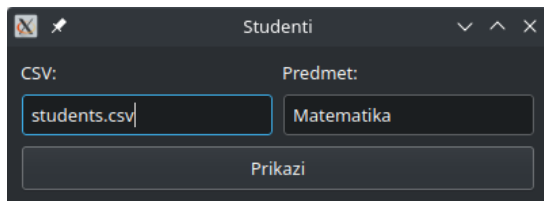
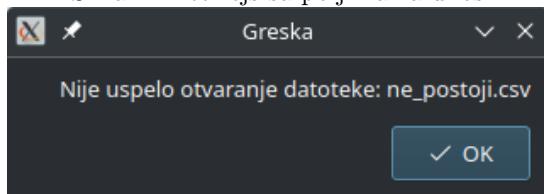


R smer, Dizajn Programskih Jezika, Septembar 2 2024

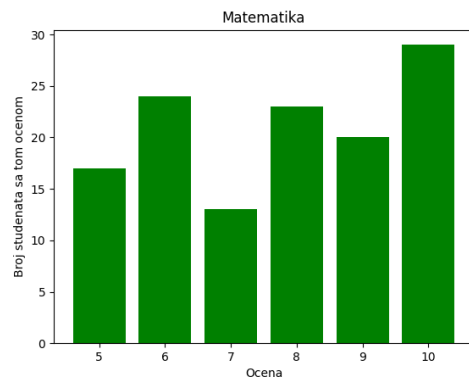
1. [Python] Napisati program koji omogućava korisniku da unese predmet i putanju do CSV fajla koji sadrži podatke o polaganjima predmeta studenata. Svako polaganje je predstavljeno poljima: indeks, predmet, ocena i opcioni dodatni komentar. Program treba da nacrtaj dijagram koji prikazuje za svaku ocenu broj studenata koji su položili sa tom ocenom, dok u naslovu grafika treba prikazati ime predmeta. Napraviti odgovarajući grafički interfejs sa poljem za unos imena predmeta i jedno polje za unos putanje do CSV fajla, kao i dugme za potvrdu unosa. Klikom na dugme prikazuje se dijagram sa traženim podacima. Ukoliko je otvaranje CSV datoteke neuspešno, prikazati odgovarajuću poruku.



Slika 1: Interfejs sa poljima za unos



Slika 2: Prikaz greške ukoliko se neuspešno otvori CSV datoteka



Slika 3: Prikaz informacija nakon klika na dugme "Prikazi"

2. [Go] Implementirati konkurentan program koji obradom n-dimenzionalnog vektora agregira sve brojeve unete sa standardnog ulaza. Korisnik unosi broj brojeva, zatim te brojeve. Iz svakog unetog broja se izdvajaju svi parovi uzastopnih cifara i od njih se grade dvocifreni brojevi. Konačan rezultat predstavlja razliku između najmanjeg i najvećeg dvocifrenog broja među svim unetim brojevima. Program treba da koristi zasebne gorutine za obradu svakog broja, pri čemu broj gorutina treba da odgovara broju dostupnih procesora. Nakon obrade svih brojeva, program treba da ispiše rezultat na standardni izlaz.

Napomena: *Zadatak se može rešiti sinhronizacijom niti preko katanca ili bez sinhronizacije, podelom poslova po nitima unapred. Pretpostaviti da su ulazni podaci ispravni.*

Primer 1	Primer 2	Primer 3	Primer 4	Primer 5
Ulaz: 1 123 Izlaz: 11	Ulaz: 3 152 243 364 Izlaz: 49	Ulaz: 5 12345 67890 98765 13579 24680 Izlaz: 86	Ulaz: 4 203 304 507 102 Izlaz: 40	Ulaz: 6 0 1 10 99 210 1000 Izlaz: 89

3. [Go] Parking garaža je mesto gde vozači dolaze kako bi parkirali svoja vozila. U ovoj garaži radi nekoliko parking službenika koji pomažu vozačima u pronalaženju slobodnih mesta. Garaža ima ograničen broj parking mesta koja predstavljaju ograničen resurs. Više vozila dolazi u garažu i svako vozilo mora čekati na slobodno mesto kako bi moglo da bude parkirano. Napisati program koji simulira rad ove garaže. Vozila dolaze nasumično i čekaju da budu parkirana. Kada sva vozila budu parkirana i napuste garažu, program se zaustavlja.

Garaža: Garaža ima fiksni broj parking mesta (10) koja su inicijalno sva slobodna. Svako mesto može biti zauzeto od strane jednog vozila u jednom trenutku. Nakon što vozilo napusti mesto, ono se ponovo oslobađa i može biti korišćeno od strane sledećeg vozila. Mehanizam čekanja za parking mesto realizovati korišćenjem uslovnog reda čekanja. Garaža od informacija ima ukupan broj mesta, trenutni broj slobodnih mesta i maksimalni kapacitet.

Parking službenici: Službenici pomažu vozačima da parkiraju vozila čim se oslobodi mesto. Svaki službenik može asistirati jednom vozaču u jednom trenutku. Ako nema slobodnih službenika, vozač čeka dok se neki ne oslobodi. Službeniku je potrebno između 3 i 6 sekundi da pomogne vozaču da

parkira (broj se bira nasumično). Potrebno je kreirati više službenika i inicijalno ih postaviti kao dostupne. Svakog službenika treba implementirati kao zasebnu gorutinu. Broj službenika se čita sa standardnog ulaza.

Vozila: Vozila dolaze u garažu nasumično. Kada stignu, čekaju dok se ne oslobodi parking mesto. Kada vozilo zauzme mesto, neki od slobodnih službenika pomaže u parkiranju. Nakon što su parkirana, vozila ostaju na mestu između 5 i 15 sekundi (broj se bira nasumično). Kada napuste parking, mesto se ponovo oslobađa. Potrebno je kreirati 30 vozila i inicijalno ih postaviti da čekaju na ulazu u garažu. Svako vozilo treba implementirati kao zasebnu gorutinu. Pri kreiranju vozila proslediti vreme parkiranja.

Modelovati opisani sistem rada garaže i prikazivati informacije kad god neko vozilo uđe u garažu, zauzme mesto, bude parkirano i napusti garažu.

4. [C++] Napisati šablonsku klasu *Trougao* koja predstavlja trougao definisan sa tri strane. Svi podaci su tipa T koji je parametar šablona. U klasi obezbediti:

- Konstruktor koji prima tri vrednosti tipa T koje predstavljaju stranice trougla (a, b, c).
- Metod **izracunajPovrsinu()** koji koristi Heronovu formulu za izračunavanje površine trougla. Formula je:

$$s = \frac{a + b + c}{2}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

gde je A površina trougla.

Ukoliko je trougao nevalidan (stranica je duža od zbira druge dve) vraća 0.

Napisati program koji učitava tri stranice trougla kao broj u pokretnom zarezu, kreira objekat klase *Trougao<float>*, poziva metodu za izračunavanje površine i prikazuje rezultat na standardnom izlazu.

Primer 1

```
Ulaz:
2.3 3.4 4.5
Izlaz:
3.8165
```

Primer 2

```
Ulaz:
7 14 4.5
Izlaz:
0.0
```

Primer 3

```
Ulaz:
12.2 14.4 4.5
Izlaz:
25.7287
```

Primer 4

```
Ulaz:
3 4 5
Izlaz:
6.0
```

5. [C++] Napisati program za upravljanje zalihama proizvoda u prodavnici.

- Napisati klasu *Proizvod* koja sadrži podatke o proizvodu: naziv (string), cena (broj u pokretnom zarezu) i količina na stanju (neoznačen ceo broj).
- Napisati klasu koja predstavlja funkcijski objekat *NajpovoljnijiProizvod* koja pri konstrukciji pamti maksimalni budžet, cenu i količinu proizvoda koji omogućava kupovinu najvećeg broja artikala. Pri funkcijskom pozivu ovog objekta prosleđuje se instanca klase *Proizvod* i proverava se da li možemo kupiti veći broj artikala od tog proizvoda unutar zadatog budžeta. Ako možemo kupiti isti broj artikala za više proizvoda, bira se proizvod sa nižom cenom. Ako je novi proizvod povoljniji prema ovim kriterijumima, ažuriraju se podaci o trenutnom najpovoljnijem proizvodu.
- Napisati šablon funkciju koja za argumente dobija proizvoljnu kolekciju objekata klase *Proizvod* kao i budžet i ispisuje naziv proizvoda kojeg se najviše može kupiti ili proizvod sa najnižom cenom. Ukoliko takav ne postoji ispisuje se adekvatna poruka.
- Sa standardnog ulaza se učitava ukupan broj proizvoda, podaci o njima i budžet, i ispisuje naziv najpovoljnijeg proizvoda prema kriterijumima.

Primer 1

```
Ulaz:
4
Mleko 100.50 50
Hleb 48.00 100
Sok 150.00 30
Jaja 50.00 150
200
Izlaz:
Hleb
```

Primer 2

```
Ulaz:
3
Cokolada 200.00 20
Kafa 120.00 10
Secer 160.00 40
2000
Izlaz:
Secer
```

Primer 3

```
Ulaz:
5
Maslac 250.00 15
Sir 250.00 10
Jogurt 200.00 20
Mleko 150.00 30
Hleb 120.00 50
750
Izlaz:
Hleb
```

Primer 4

```
Ulaz:
3
Ulje 300.00 25
Sir 300.00 30
Margarin 300.00 20
200
Izlaz:
Nema artikla koji se može kupiti!
```