

Klase 2. deo - Zadaci

Istorija revizija

<i>Verzija</i>	<i>Autori</i>	<i>Datum</i>	<i>Status</i>	<i>Komentar</i>
1.00	Ivan Stojić	12.11.2021.	Inicijalna verzija	Alokacija memorije (automatska, statička, dinamička), specijalne metode (konstruktori i destruktori)

Sadržaj

Zadatak 1: Prodavnica	4
-----------------------------	---

Zadatak 1: Prodavnica

Napisati klasu **Prodavnica** koja će opisivati prodavnicu sa artiklima. Klasa treba da sadrži: polje **naziv** tipa **string**, polje **el** celobrojnog tipa **int** koje predstavlja broj mesta za artikle na rafovima prodavnice, polje **artikli** tipa pokazivač **Artikal**** (struktura **Artikal**), konstruktor sa dva parametra (**string**, **int**), destruktor, metodu **int brojMesta()const**, metodu **int brojArtikala()const**, metodu **string getNaziv()const**, metodu **Artikal** getArtikli()const**, metodu **bool dodajArtikal(Artikal)**, metodu **bool obrisiArtikal(int)**, i metodu **bool azurirajCenu(int, double)**, metodu **void sortiraj(VrstaSortiranja)**, metodu **const Artikal* vrstaArtikla(VrstaArtikla)const**.

Napisati slobodnu funkciju **void prikazi(const Prodavnica &)** koja za prosleđeni objekat tipa klase **Prodavnica** ispisuje njen naziv i sve artikle.

Napisati slobodnu funkciju **void prikazi(const Artikal &)** koja za prosleđeni objekat tipa strukture **Artikal** ispisuje njenu šifru, naziv, i cenu.

Napomene:

- Struktura **Artikal** treba da ima: član **sifra** celobrojnog tipa **int**, član **naziv** tipa **string**, član **cena** realnog tipa **double**
- Konstruktor sa parametrima sa sadržajem prvog argumenta koji predstavlja naziv prodavnice definiše polje **naziv**, a sa sadržajem drugog argumenta koji predstavlja broj mesta za artikle na rafovima prodavnice definiše polje **el**. Dinamički alocirati memoriju dužine **el** za niz elemenata tipa **Artikal*** na koji pokazuje pokazivač na pokazvač **artikli**, i sve elemente prve dimenzije (***artikli**) postaviti na **NULL**.
- Destruktor vrši oslobađanje dinamički alocirane memorije tako što prvo oslobađa dinamički alociranu memoriju druge dimenzije (prostor na kome se čuvaju artikli), a zatim oslobađa dinamički alociranu memoriju prve dimenzije pokazivača **artikli**.
- Metoda **int brojMesta()const** kao povratnu vrednost vraća koliko artikala može da stane na rafove prodavnice - polje **el**
- Metoda **int brojArtikala()const** kao povratnu vrednost vraća broj zauzetih mesta na rafovima (broj elemenata prve dimenzije pokazivača **artikli** različit od **NULL**)
- Metoda **Artikal** getArtikli()const** kao povratnu vrednost vraća pokazivač **artikli**

- Metoda **bool dodajArtikal(Artikal)** prosleđeni artikal dodaje na prvu slobodnu poziciju. Ako je dodavanje uspešno vraća **true**, u suprotnom **false**.
- Metoda **bool obrisiArtikal(int)** briše artikal čija šifra odgovara sadržaju prosleđenog argumenta tako što oslobađa dinamički alociranu memoriju na kojoj se nalazi. Ako je brisanje uspešno vraća **true**, u suprotnom **false**.
- Metoda **bool azurirajCenu(int, double)** ažurira cenu artikla sa sadržajem drugog argumenta ako je pronađen artikal čija šifra odgovara sadržaju prvog argumenta. Ako je ažuriranje cene uspešno vraća **true**, u suprotnom **false**.
- Metoda **void sortiraj(VrstaSortiranja)** vrši opadajuće/rastuće sortiranje artikala u prodavnici na osnovu prosleđene konstante enumeracije `VrstaSortiranja {OPADAJUCE, RASTUCE}`
- Metoda **const Artikal* vrstaArtikla(VrstaArtikla) const** kao povratnu vrednost svom pozivaocu treba da vrati najjeftiniji/najskuplji artikal u prodavnici na osnovu prosleđene konstante enumeracije `VrstaArtikla {NAJJEFTINIJI, NAJSKUPLJI}`.

```

C:\Users\Administrator\CodeBlocks\C++\bin\Debug\C++.exe
===== Mikro market=====
Stanje: 0 od 10
Artikli:
    Svi rafovi su prazni.

Dodavanje artikala...
===== Mikro market=====
Stanje: 3 od 10
Artikli:
    Sifra: 1
    Naziv: Hleb Sava
    Cena: 55

    Sifra: 2
    Naziv: Milka sa lesnikom 100gr
    Cena: 123.45

    Sifra: 3
    Naziv: CocaCola 1.5l
    Cena: 87.02

Brisanje artikla...
===== Mikro market=====
Stanje: 2 od 10
Artikli:
    Sifra: 1
    Naziv: Hleb Sava
    Cena: 55

    Sifra: 3
    Naziv: CocaCola 1.5l
    Cena: 87.02

```

```
C:\Users\Administrator\CodeBlocks\C++\bin\Debug\C++.exe

Artikal je dodat na novu pozicju
===== Mikro market=====
Stanje: 3 od 10
Artikli:
    Sifra: 1
    Naziv: Hleb Sava
    Cena: 55

    Sifra: 2
    Naziv: Milka sa lesnikom 100gr
    Cena: 123.45

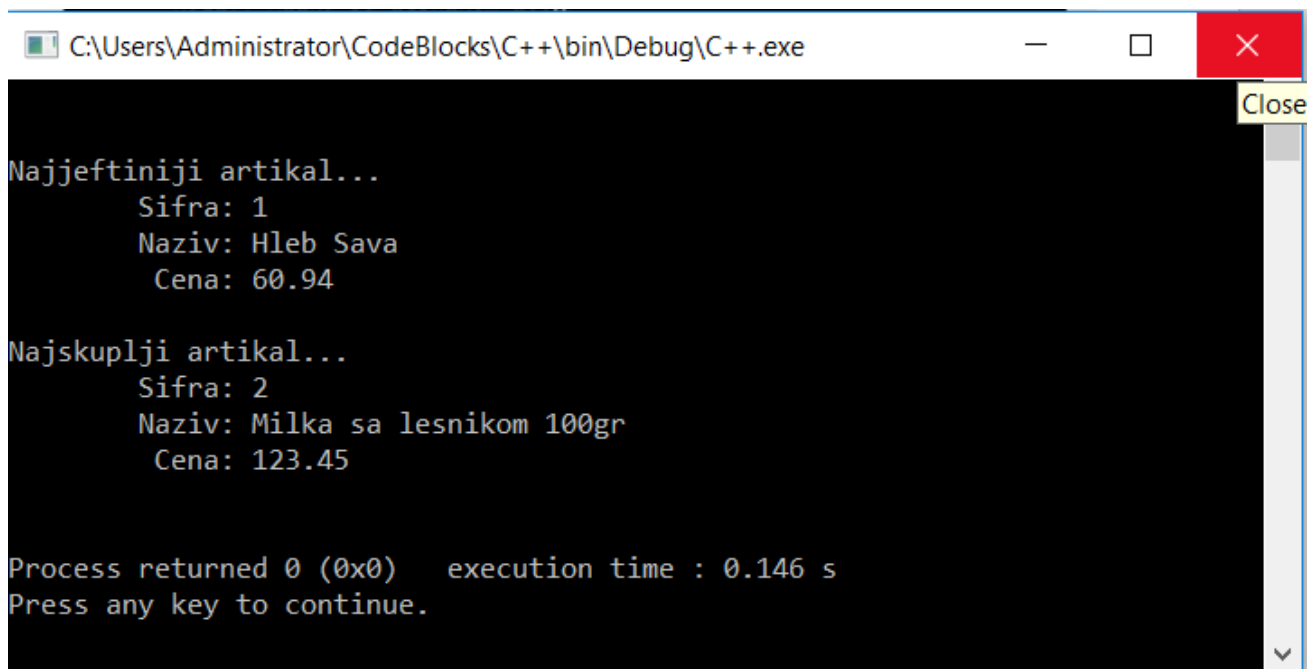
    Sifra: 3
    Naziv: CocaCola 1.5l
    Cena: 87.02

Azuriranje cene...
===== Mikro market=====
Stanje: 3 od 10
Artikli:
    Sifra: 1
    Naziv: Hleb Sava
    Cena: 60.94

    Sifra: 2
    Naziv: Milka sa lesnikom 100gr
    Cena: 123.45

    Sifra: 3
    Naziv: CocaCola 1.5l
    Cena: 87.02

Rastuce azuriranje cene...
===== Mikro market=====
Stanje: 3 od 10
Artikli:
    Sifra: 1
    Naziv: Hleb Sava
    Cena: 60.94
```



C:\Users\Administrator\CodeBlocks\C++\bin\Debug\C++.exe

```
Najjeftiniji artikal...  
  Sifra: 1  
  Naziv: Hleb Sava  
  Cena: 60.94  
  
Najskuplji artikal...  
  Sifra: 2  
  Naziv: Milka sa lesnikom 100gr  
  Cena: 123.45  
  
Process returned 0 (0x0)   execution time : 0.146 s  
Press any key to continue.
```