



Programiranje II

Laboratorijske vježbe 8

1. Napraviti klasu **Narudzba**, čiji će objekti čuvati podatke o (jednoj) narudžbi napravljenoj u (našoj zamišljenoj) veletrgovini. Tipično, druge firme naručuju robu od naše veletrgovine i želimo zabilježiti sve što karakteriše tu narudžbu, a također i napraviti odgovarajuće funkcije članice koje će nam pomoći da radimo sa narudžbama (na primjer da ispišemo narudžbu, ili da saznamo ukupan iznos narudžbe). Podaci koje je potrebno sačuvati za jednu narudžbu su:

1. Datum narudžbe,
2. Komercijalista (ime i prezime radnika ko je napravio narudžbu),
3. Kupac (firma kojoj će se isporučiti roba). Želimo zabilježiti naziv kupca, adresu kupca, mjesto, državu, itd.
4. Stavke narudžbe. To je niz (vektor) koji se sastoji od jednog ili više elemenata: koji artikal je naručen, u kojoj količini i po kojoj cijeni.

(Ukoliko vam nije jasan koncept narudžbe, samo uzmite bilo kakav račun iz trgovine i dobro ga pogledajte i analizirajte - to bi trebalo da bude dovoljno).

Sve gore navedeno zapravo predstavlja varijable članice naše klase.

Za stavke narudžbe je potrebno napraviti posebnu klasu (**StavkaNarudzbe**), koja ima svoje varijable: naziv artikla, količina, cijena. Eventualno napravite konstruktor, gettere, settere. Po potrebi se mogu napraviti i nove klase za: komercijalistu, kupca, ili npr. adresu.

Sukcesivno ćemo dodavati nove mogućnosti, onako kako su pobrojane. Demonstrirajte funkcionalnost!

- a) Napravite konstruktor za narudžbu, na način da se po defaultu uzima današnji datum kao datum narudžbe. Normalno, kada se pravi narudžba, uvijek se zna za koga se pravi (kupac) i ko je unosi (komercijalista). Omogućite da se stavkama može pristupiti bez problema (**public**). Napravite gettere i settere za varijable članice, a pristup varijablama članicama (osim stavkama) sakrijte.
- b) Napravite funkciju članicu **validnaNarudzba** koja će vratiti tačno ukoliko narudžba ima podešen datum, (nepraznog) komercijalistu, (nepraznog) kupca i barem jednu stavku.
- c) Napravite funkciju članicu koja će vratiti iznos narudžbe. Iznos se računa kao suma iznosa svih stavki, a iznos jedne stavke je količina * cijena.
- d) Napravite funkciju članicu koja ispisuje narudžbu na ekran. Napravite i drugu varijantu funkcije koja ispisuje stavke ali ovaj put složene po nazivu.
- e) Napravite posebne funkcije koje vrše dodavanje nove stavke, brisanje postojeće ili izmjenu neke od stavki. Pri tome, sakrijte pristup nizu stavki, jer **public** pristup više nije potreban.
- f) Napravite funkciju članicu koja vraća datum isporuke. Datum isporuke se računa kao datum narudžbe + dodatnih 5 dana. (Za one koji žele više: možete "zakomplikovati" time što u tih 5 dana ne računate subote i nedjelje).
- g) Napravite da je datum isporuke varijabla članica. Dopunite konstruktor narudžbe. Napravite odgovarajuću validaciju za izmjenu datuma isporuke (logično: datum isporuke ne može biti "manji", odnosno ranije od datuma narudžbe, u tom slučaju javite exception).

2. (Koristi se prethodni zadatak - napredno/zadaća) Sada uradite različita unapređenja prethodne klase. Neke od ovih stvari ćemo raditi i na sljedećim vježbama, ali je dobro da se što prije sami upoznajte sa mogućostima jezika C++.

- a) Napravite da se narudžba može ispisivati na stream (preklopite operator <<).
- b) Napravite da se narudžbe mogu "sabirati", na način da se time dobije nova narudžba, koja sadrži stavke i iz jedne i iz druge narudžbe. Ukoliko se jedan te isti artikal pojavljuje u obje narudžbe, potrebno je onda sabrati količine i imati samo jednu stavku za dati artikal, ali samo pod uslovom da je cijena ista. U suprotnom, napravite exception.
- c) Preklopite operator [], čime dajete pristup odgovarajućoj stavki narudžbe (npr. ako je **n** objekat narudžbe, **n[3]** vraća četvrtu stavku narudžbe).
- d) Napravite da ne čuvamo niz stavki (niz objekata stavki) nego niz pokazivača na (objekte) stavke narudžbe.
- e) Dodavanje nove stavke treba da ima odgovarajuću validaciju: da li dodajemo artikal koji se već nalazi u narudžbi? Ukoliko je odgovor potvrđan, to je moguće samo ako je cijena ista, u suprotnom javite exception.
- f) Omogućite pretragu stavki po nazivu (slobodna implementacija). Unaprijedite time što ćete koristiti lambde.
- g) Napravite čitanje objekta narudžbe iz streama (preklopite operator >>).