

**Objektno-orijentisano programiranje, JUN 2**  
*Matematički fakultet, školska godina 2019/2020*

**Napomena:** Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_NM_Peric_Pera_mi12082`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, paket takođe nazvati tako.

Kod **ne sme** imati sintakasnih grešaka niti izbacivanje `NullPointerException`-a.

Vreme za rad: **3 sata**

Inicijali: [2MNVA, 2RL1A, 2RL1B - BS], [2RL2A, 2RL2B - NM], [2I1A, 2I2A, 2MNVB - AB], [2I1B, 2I2B - DA]

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je (i ohrabrujemo Vas) dodati nove attribute, klase, metode, enume, interfejsu u slučaju da Vam olakšavaju implementaciju, i/ili smatrate da Vam poboljšavaju kvalitet koda i slično. Nekada će zahtevi u zadatku i zahtevati od Vas da dodate novi atribut ili slično.

Da bi se uspešno položio ispit potrebno je osvojiti **barem 25 poena**.

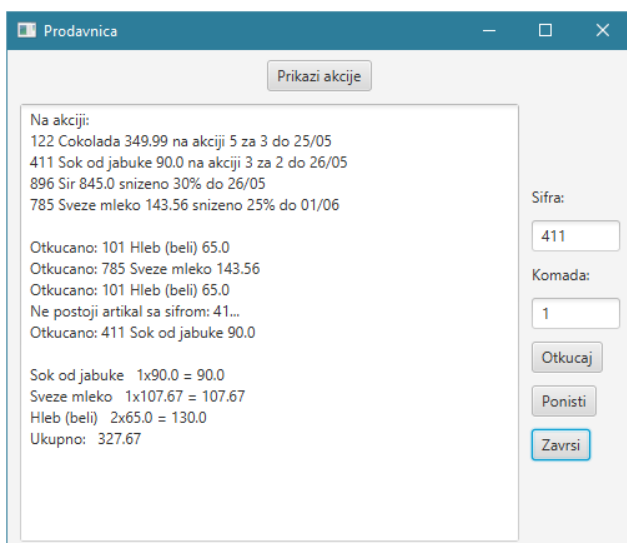
1. Napraviti apstraktnu klasu `Akcija` koja se karakteriše poljem `datumIsteka` (`String`, datum isteka akcije u formatu `dd/mm`). Implementirati konstruktor koji prima vrednost za datum i `get` metod za to polje. Klasa sadrži i apstraktan metod `double cenaPoKomadu(double cena, int otkucanoKomada)` koji izračunava cenu jednog komada na osnovu broja otkucanih artikala i regularne cene tog artikla.
2. Napraviti klasu `Popust` koja nasleđuje klasu `Akcija` i karakteriše se dodatnim poljem `procenat` (`int`, broj iz intervala `[1, 99]` - nije potrebno proveravati ispravnost vrednosti). Implementirati odgovarajući konstruktor i:
  - apstraktni metod koji računa cenu po komadu tako što redovnu cenu umanjuje za `procenat` posto;
  - metod `toString` koji vraća nisku koja sadrži informaciju o procentu popusta i datumu isteka akcije kao u primeru: `snizeno 30% do 26/05`.
3. Napraviti klasu `Gratis` koja nasleđuje klasu `Akcija` i karakteriše se poljima `potrebnoKomada` (`int`, broj komada koje je potrebno kupiti da bi se ostvarilo pravo na gratis komade) i `gratisKomada` (`int`, broj gratis komada koji se dobijaju za kupljenih `potrebnoKomada`). Implementirati odgovarajući konstruktor i:
  - apstraktni metod koji računa cenu po komadu tako što od ukupne sume za `otkucanoKomada` po regularnoj ceni oduzima cenu `gratisKomada`. Na primer, ako je artikal na akciji "5 za 3" i otkucano je 6 komada onda se plaća 4 komada, a 2 su gratis. Ako je na istoj akciji otkucano 4 komada onda se plaćaju 3 komada, a 1 je gratis.
  - metod `toString` koji vraća nisku koja sadrži informaciju o broju kupljenih, broju komada koje je potrebno kupiti da bi se ostvario popust i datumu isteka akcije kao u primeru: `na akciji 5 za 3 do 25/05`.
4. Napraviti klasu `Artikal` koja se karakteriše poljima `sifra` (`int`), `naziv` (`String`), `cena` (`double`) i `akcija` (`Akcija`). Implementirati:
  - konstruktor koji prima vrednost za sva polja;
  - konstruktor koji prima vrednosti za sva polja osim polja `akcija`, a `akciju` postavlja na `null`;
  - potrebne `get` metode;
  - metod `toString` koji vraća nisku koja sadrži informaciju o šifri, nazivu i ceni kao u primeru: `122 Cokolada 349.99`.
5. U klasi `Artikal` implementirati interfejs `Comparable<Artikal>` i neophodan metod. Dva artikla se porede tako što u sortiranom poretku prvo idu artikli koji su na nekoj akciji, a onda oni koji nisu na akciji. Ukoliko su oba artikla koja se porede na akciji, porediti ih prema datumu isteka akcije rastuće, a u slučaju istih datuma porediti po nazivu opadajuće. Ukoliko artikli koji se porede nisu na akciji porediti ih po nazivu opadajuće.
6. Napraviti klasu `Racun` koja se karakteriše poljem `stavke` (`Map<Artikal, Integer>`, vrednosti mape predstavljaju broj otkucanih komada). Implementirati metode:
  - `void dodajStavku(Artikal artikal, int komada)` - dodaje novu stavku na račun ako artikal nije na računu, odnosno uvećava broj otkucanih komada ako je artikal na računu (primer je dat na slici 1);
  - `Integer ukloniStavku(Artikal artikal)` - uklanja stavku računa i vraća broj otkucanih komada za tu stavku;
  - `String stampajStavku(Artikal artikal, int otkucanoKomada)` - vraća stringovnu reprezentaciju jedne stavke računa kao u primeru ispod. Cena po komadu je vrednost koju vraća metod `cenaPoKomadu()` za artikle koji su na akciji, odnosno puna cena artikla za one artikle koji nisu na akciji.  
$$\text{naziv otkucanoKomada} \times \text{cenaPoKomadu} = \text{otkucanoKomada} \times \text{cenaPoKomadu}$$
`Cokolada 2 x 349.99 = 699.98`

- `toString` - vraća nisku koja sadrži sve stavke u formatu kao u primeru i na kraju ukupnu sumu koju treba platiti (videti sliku 1).
7. Napraviti klasu `Kasa` koja se karakteriše statičkim poljem artikli (`List<Artikli>`, lista artikala koji se nalaze u prodavnici). Implementirati **statičke** funkcije:
- `void ucitajArtikle()` koja čita podatke o artiklima iz datoteke `artikli.txt` i smešta ih u listu `artikli`;
  - `Optional<Artikal> ocitajArtikal(int sifra)` koja vraća opcionu vrednost koja sadrži artikal iz liste artikala sa zadatom šifrom ako postoji. Ukoliko ne postoji artikal sa zadatom šifrom, vratiti praznu opcionu vrednost;
  - `List<Artikal> artikliNaAkciji()` koja vraća listu artikala koji su trenutno na nekoj akciji.
8. Napraviti klasu `Prodavnica` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Klasa sadrži jedan atribut `racun (Racun, račun koji se trenutno kuca)`.
9. U klasi `Prodavnica` implementirati sledeće događaje:
- Kada se aplikacija pokrene iz datoteke `artikli.txt` učitavaju se informacije o artiklima i smeštaju u listu `artikli` klase `Kasa`.
  - Klikom na dugme `Prikazi akcije` izdvajaju se artikli koji su na akciji i ispisuju se uređeno po datumu isteka akcije rastuće, a zatim po nazivu opadajuće.
  - Klikom na dugme `Otkucaj` čitaju se šifra i količina iz odgovarajućih `TextField` elemenata i dodaje se nova stavka na račun. Ispisati informacije o artiklu koji je otkucan. Ukoliko je neka od unetih vrednostih neispravna, ili ne postoji artikal sa odabranom šifrom ili je broj komada manji od jedan ispisati odgovarajuću poruku.
  - Klikom na dugme `Ponisti` čita se šifra iz odgovarajućeg `TextField` elementa i uklanja se odgovarajuća stavka sa računa. Ispisati naziv artikla koji je poništen. Ukoliko je uneta šifra neispravna ili na račun nije otkucan artikal sa odabranom šifrom ispisati odgovarajuću poruku.
  - Klikom na dugme `Završi` ispisuju se stavke računa, a zatim i ukupna suma koju je potrebno platiti. Ovom akcijom se završava kucanje trenutnog računa i započinje se novi račun. Ukoliko nijedna stavka nije otkucana ispisati odgovarajuću poruku.

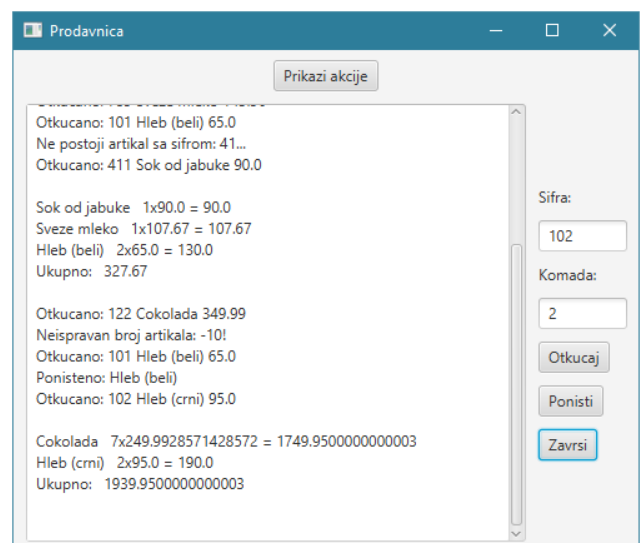
Sadržaj datoteke `artikli.txt` je oblika: `sifra, naziv, cena, akcija, datumIsteka`

Vrednosti `akcija` i `datumIsteka` postoje samo ako je artikal trenutno na akciji. Vrednost za `akcija` može biti oblika `popust%` ili `ukupnoKomada za potrebnoKomada`. Datum je oblika `dd/mm`. Pretpostaviti da je sadržaj datoteke ispravan i uvek u prethodno opisanom formatu. Primer sadržaja datoteke:

```
101, Hleb (beli), 65
102, Hleb (crni), 95
122, Cokolada, 349.99, 5 za 3, 25/05
312, Voda, 50.5
896, Sir, 845, 30%, 26/05
785, Sveze mleko, 143.56, 25%, 01/06
411, Sok od jabuke, 90, 3 za 2, 26/05
```



Slika 1: Prikaz akcija, otkucavanje artikala, neispravna šifra artikla, završetak jednog računa.



Slika 2: Otkucavanje i poništavanje artikala, neispravan broj artikala, završetak jednog računa.