

Objektno-orijentisano programiranje, Jun 1

Matematički fakultet

Školska godina 2022/2023

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_OM_Peric_Pera_mr21082`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom.

Vreme za rad: **3 sata**

Inicijalni asistenata: 2MNV & 2RL1 → BJ, 2RL2 → NC

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

Da bi se uspešno položio ispit potrebno je osvojiti barem 35 poena.

Zadatak je napraviti aplikaciju koja predstavlja aukcijsku kuću, i dozvoljava korisniku da kupuje artikle.

- [3p] Napraviti i nabrojivi tip `Kvalitet` čije su vrednosti kategorije kvaliteta umetničkih dela: `LOS`, `OSREDNJI` i `VRHUNSKI`. Tip sadrži polje `celobrojnaVred` (`int`) sa vrednostima 1, 2 ili 3 koje redom odgovaraju navedenim kategorijama kvaliteta. Implementirati konstruktor, `get` metod za polje `celobrojnaVred` i statički metod `Kvalitet.createFromInt(int celobrojnaVred)` koji na osnovu celobrojne vrednosti `celobrojnaVred` vraća odgovarajuću vrednost nabrojivog tipa, a u slučaju prosleđene nedozvoljene vrednosti izbacuje izuzetak `IllegalArgumentException` sa odgovarajućom porukom o grešci.
- [3p] Napraviti apstraktnu klasu `UmetnickoDelo` koja se karakteriše poljima `ID` (`String`), `naziv` (`String`), `imeAutora` (`String`) i `kvalitet` (`Kvalitet`). Obezbediti konstruktor koji prihvata vrednosti za sva polja, kao i potrebne `get` metode. Definirati apstraktni metod `int getCena()` koji treba da vraća procenjenu vrednost umetničkog dela. Implementirati i metod `toString` koji vraća nisku formata `[ID] kvalitet : naziv, imeAutora`.
- [4.5p] Napraviti klasu `Knjiga` koja je nasleđuje klasu `UmetnickoDelo`. Klasa se dodatno karakteriše poljima `izdavac` (`String`), `godinaIzdanja` (`int`) i `brojStrana` (`int`). Obezbediti konstruktor koji uzima vrednosti za sva polja, kao i potrebne `get` metode. Implementirati nasleđeni apstraktni metod tako da se cena knjige računa prema formuli

$$\text{celobrojnaVredKvaliteta} * \text{brojStrana}$$

Implementirati i metod `toString` koji koristi `toString` od nadklase, i na to nadovezuje stringovske reprezentacije dodatnih polja kojima se karakterisu knjige kao u test primeru (slika 2).

- [4.5p] Napraviti klasu `Skulptura` koja je nasleđuje klasu `UmetnickoDelo`. Klasa se dodatno karakteriše poljima `godinaStvaranja` (`int`) i `tezina` (`int`). Obezbediti konstruktor koji uzima vrednosti za sva polja, kao i potrebne `get` metode. Implementirati nasleđeni apstraktni metod tako da se cena skulpture računa prema formuli

$$\text{celobrojnaVredKvaliteta} * \text{godinaStvaranja} + \text{tezina}/100$$

Implementirati i metod `toString` koji koristi `toString` od nadklase, i na to nadovezuje stringovske reprezentacije dodatnih polja kojima se karakterisu skulpture kao u test primeru (slika 2).

- [7p] Napraviti klasu `KomparatorUmetnickihDela` koja implementira interfejs `Comparator<UmetnickoDelo>` i tako da definiše sledeći kriterijum poređenja:
 - u sortiranom redosledu trebaju prvo da idu knjige pa skulpture
 - knjige i skulpture se među sobom porede prvenstveno prema kvalitetu tako da prvo idu `VRHUNSKI`, pa `OSREDNJI`, i na kraju `LOSI`), a ako su istog kvaliteta onda prema godini izdanja, odnosno godini stvaranja rastuće
- Napraviti klasu `AukcijskaKuca` koja se karakteriše poljima `umetnickaDelaNaProdaju` (`List<UmetnickoDelo>`) i `zarada` (`int`). Implementirati:
 - [2p] podrazumevani konstruktor i potrebne `get` metode
 - [1p] metod `void dodajUmetnickoDelo(UmetnickoDelo ud)` koji dodaje dato umetnicko delo u listu za prodaju
 - [7p] metod `void ucitajUmetnickaDelaIzFajla(String filePath)` koji iz datoteke `filePath` učitava podatke o umetničkim delima i dodaje ih u listu za prodaju; pretpostaviti da je svaka linija datoteke jednog od sledeća dva formata:
 - `IDknjige, naziv, imeAutora, kategorija, izdavac, godinaIzdanja, brojStrana`
 - `IDskulpture, naziv, imeAutora, kategorija, godinaStvaranja, tezina`pri čemu prvi karakter `ID`-a ukazuje na to da li se radi o knjizi ili skulpturi (primer sadržaja datoteke je dat na kraju teksta zadatka)

- [7p] metod `boolean licitacija(String ID, int ponudjenaCena)` koji najpre pronalazi umetničko za koje se licitira a zatim proverava da li je ponuđena cena dovoljno velika u odnosu na procenjenu vrednost umetničkog dela, i ako je licitacija uspešna ažurira zaradu aukcijske kuće (razlika ponuđene cene i vrednosti umetničkog dela), izbacuje to umetničko delo iz liste za prodaju i vraća indikator da je licitacija uspešna; u suprotnom samo vratiti indikator neuspešne licitacije
- [3p] metod `toString` koji sortira listu umetničkih dela za prodaju i formira nisku tako svako umetničko delo bude u zasebnom redu (slika 2)

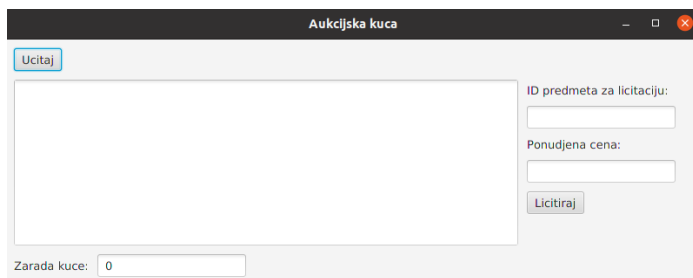
7. [11p] Napraviti klasu `Main` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Klasa sadrži jedno statičko polje `aukcijaskaKuca` (`AukcijaskaKuca`).

- [4p] Kada korisnik klikne dugme `Ucitaj`, potrebno je učitati iz datoteke `za_prodaju.txt` umetnička dela za prodaju u aukcijskoj kući i ispisati informacije njima u `TextArea` element na levoj strani. Nakon učitavanja, onemogućiti ponovo klitanje na ovo dugme (slika 2).
- [8p] Klikom na dugme `Licitiraj` čita se ID umetničkog dela koje korisnik želi da licitira iz gornjeg `TextField` elementa, kao i njegovu ponudu iz donjeg `TextField` elementa na desnoj strani. U labeli pored dugmeta `Licitiraj` se ispisuje odgovarajuća poruka o uspešnosti licitacije (slike 3, 4). U slučaju da je licitacija uspešna treba ažurirati ispis ponuđenih umetničkih dela za prodaju, kao i tekuće stanje zarade aukcijske kuće (`TextField` element na levojs strani).

Obraditi sve potencijalne izuzetke (`NumberFormatException`, `IOException`, `IllegalArgumentException` itd).

Primer sadržaja datoteke `za_prodaju.txt`:

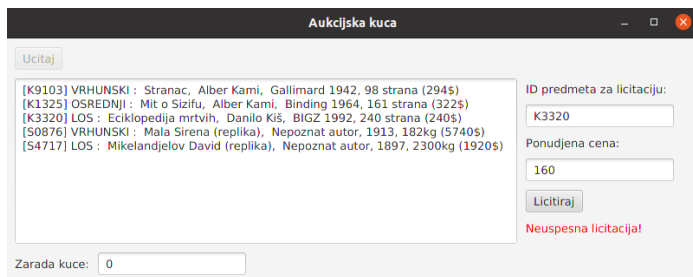
```
K1325, Mit o Sizifu, Alber Kami, 2, Binding, 1964, 161
S0876, Mala Sirena (replika), Nepoznat autor, 3, 1913, 182
K9103, Stranac, Alber Kami, 3, Gallimard, 1942, 98
S4717, Mikelandjelov David (replika), Nepoznat autor, 1, 1897, 2300
K3320, Eciklopedija mrtvih, Danilo Kiš, 1, BIGZ, 1992, 240
```



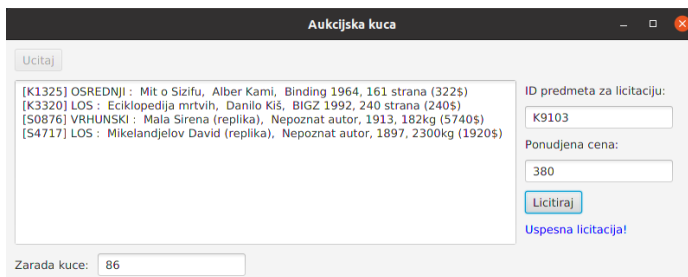
Slika 1: Početni izgled aplikacije



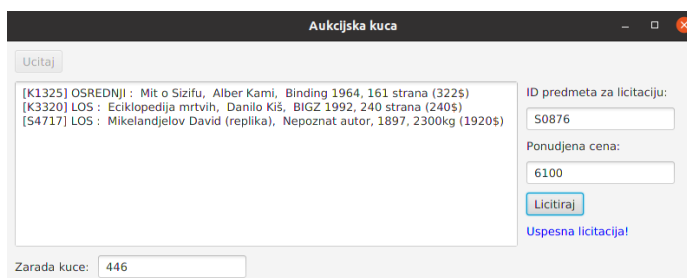
Slika 2: Učitavanje



Slika 3: Neuspešna licitacija



Slika 4: Uspešna licitacija



Slika 5: Uspešna licitacija