

Objektno-orijentisano programiranje, Septembar 1

Matematički fakultet

Školska godina 2020/2021

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_OM_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**

Inicijalni asistenti: Ognjen - OM, Nevena - NC, Filip - FV, Denis - 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 35 poena.

1. [7 poena] Napraviti apstraktnu klasu `Nekretnina` koja se karakteriše poljima `opstina` (`String`, lokacija nekretnine), `kmOdCentra` (`double`, udaljenost nekretnine od centra u km), `kvadratura` (`double`, kvadratura nekretnine u m^2) i `cena` (`int`, cena nekretnine u evrima). Implementirati:

- konstruktor koji prima sve potrebne vrednosti i potrebne `get` metode
- `toString` metod koji vraća nisku sledećeg formata:

`opstina, kmOdCentra kilometara od centra, kvadratura m^2 , cena evra`

Klasa sadrži i apstraktan metod `double isplativostNekretnine()`.

2. [7 poena] Napraviti klasu `Stan` koja nasleđuje klasu `Nekretnina`. Klasa se dodatno karakteriše poljem `parkingMesto` (`boolean`) koji predstavlja indikator da li uz stan dolazi i parking mesto ili ne. Implementirati konstruktor, potrebne `get` metode, metod `toString` koji vraća nisku kao u test primeru (slika 2), kao i nasleđeni apstraktni metod tako da se isplativost nekretnine izračunava po sledećoj formuli:

$$\begin{cases} \text{cenaNekretnine}/(\text{kvadraturaNekretnine} + 10 * 0.8) & \text{kada uz stan dolazi i parking mesto} \\ \text{cenaNekretnine}/\text{kvadraturaNekretnine} & \text{inače} \end{cases}$$

Računa se da parking mesto ima $10m^2$ i da je cena kvadrata parking mesta vredí 80% od cene kvadrata stana.

3. [7 poena] Napraviti klasu `Kuca` koja nasleđuje klasu `Nekretnina`. Klasa se dodatno karakteriše poljem `povrsinaPoseda` (`double`, površina poseda na kome se kuća nalazi u arima). Implementirati konstruktor, potrebne `get` metode, metod `toString` koji vraća nisku kao u test primeru (slika 3), kao i nasleđeni apstraktni metod tako da se isplativost nekretnine izračunava po sledećoj formuli:

$$\text{cenaNekretnine}/(\text{kvadraturaNekretnine} + \text{povrsinaPoseda} * 100 * 0.7)$$

Računa se da je $1 \text{ ar} = 100 m^2$ i da površina poseda vredí 70% od cene kvadrata kuće.

4. [7 poena] Napraviti klasu `ParkingMesto` koja nasleđuje klasu `Nekretnina`. Klasa se dodatno karakteriše poljem `natkrivenost` (`boolean`) koje predstavlja indikator da li je parking mesto natkriveno, odnosno da li je u pitanju garažno mesto ili ne. Implementirati konstruktor, potrebne `get` metode, metod `toString` koji vraća nisku kao u test primeru (slika 4), kao i nasleđeni apstraktni metod tako da se isplativost nekretnine izračunava po sledećoj formuli:

$$\begin{cases} \text{cenaNekretnine}/(\text{kvadraturaNekretnine} * 1.2) & \text{kada je parking mesto natkriveno} \\ \text{cenaNekretnine}/\text{kvadraturaNekretnine} & \text{inače} \end{cases}$$

Računa se da kvadrat natkrivenog parking mesta vredí 20% više od kvadrata parking mesta na otvorenom.

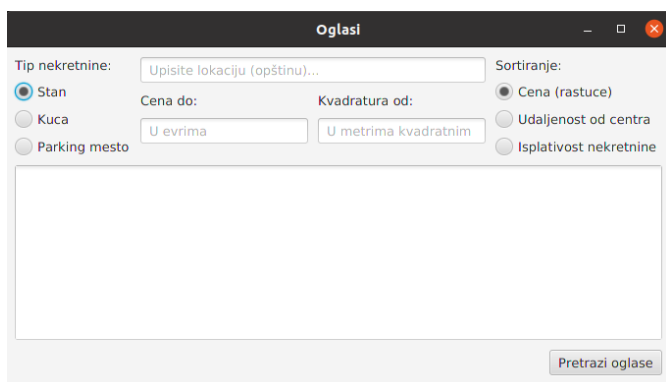
5. Napraviti klasu `PretragaOglasa` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Klasa sadrži polje `oglasa` (`List<Nekretnina>`).

- [10 poena] Obezbediti da aplikacija prilikom pokretanja izgleda kao na slikama. Za ispis početne poruke u `TextField` elemente može se koristiti metod `setPromptText`. Preporučena veličina prozora je 680×350 .
- [2 poena] Obezbediti da može biti selektovan samo jedan tip nekretnine za pretragu i da je prva opcija selektovana na početku.
- [2 poena] Obezbediti da može biti selektovan samo jedan kriterijum sortiranja oglasa i da je prva opcija selektovana na početku.

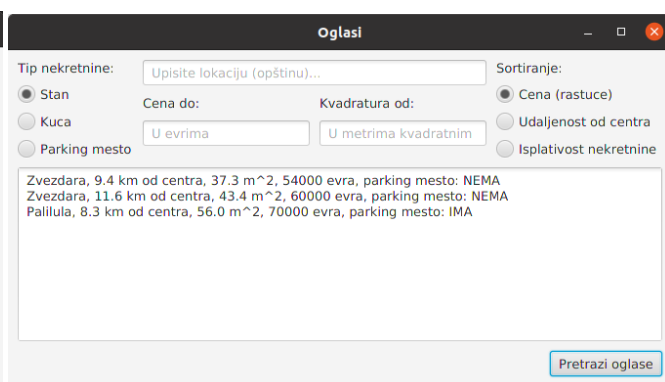
- [7 poena] Implementirati pomoćni metod `void ucitaj()` koji učitava informacije o svim oglasima (nekretninama) iz datoteke `oglas.txt` i smešta ih u listu `oglas`, a zatim ispisuje sadržaj liste na standardni izlaz tako da svaki element bude u zasebnom redu.
- [3x2.5 + 3x2 + 2.5 poena] Klikom na dugme 'Pretraži oglase' u `TextArea` element ispisati podatke o svim oglasima (nekretninama) koje zadovoljavaju zadate kriterijume pretrage za tip nekretnine, lokaciju, cenu i kvadraturu (*napomena*: ne moraju svi parametri pretrage biti zadati). Rezultate pretrage ispisati u poretku koji odgovara odabranom kriterijumu sortiranja. Ukoliko nema oglasa koji zadovoljavaju zadate kriterijume pretrage ispisati odgovarajuću poruku (slika 6).

Pretpostaviti da će sadržaj datoteke `oglas.txt` biti ispravan. Primer sadržaja datoteke:

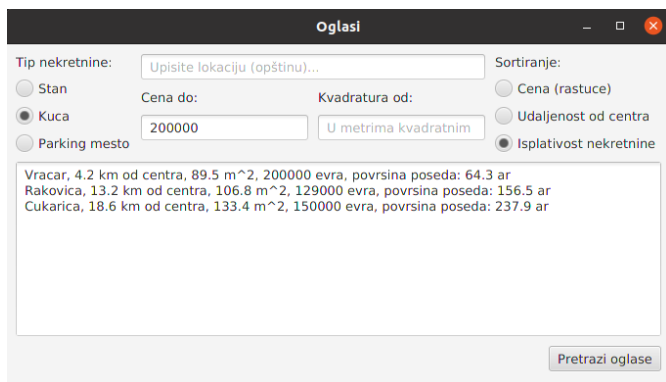
```
STAN Zvezdara 9.4 37.3 54000 NE
PARKING_MESTO Vozdovac 24.1 12.0 9000 DA
STAN Palilula 8.3 56.0 70000 DA
KUCA Rakovica 13.2 106.8 129000 156.5
PARKING_MESTO Cukarica 17.2 11.8 6500 NE
KUCA Cukarica 18.6 133.4 150000 237.9
STAN Zvezdara 11.6 43.4 60000 NE
KUCA Vracar 4.2 89.5 200000 64.3
```



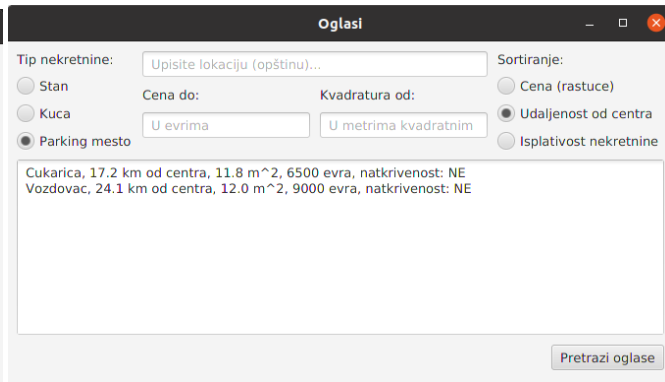
Slika 1: Početni izgled aplikacije



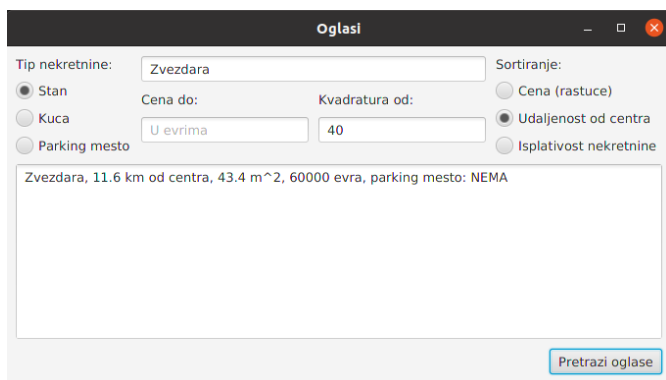
Slika 2: Pretraga stanova, sortiranje po ceni (rastuće)



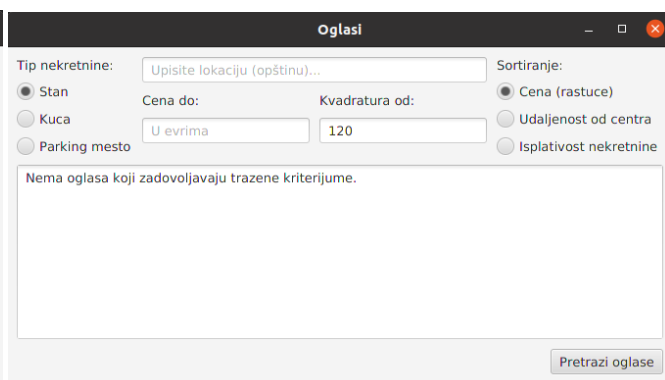
Slika 3: Pretraga kuća, sortiranje po isplatiosti



Slika 4: Pretraga stanova na lokaciji Zvezdara, sortiranje po udaljenosti od centra



Slika 5: Pretraga parking mesta, sortiranje po udaljenosti od centra



Slika 6: Prikaz poruke