

Objektno-orijentisano programiranje, Septembar 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, interfejse 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 sintakasnih grešaka niti izbacivanje `NullPointerException`-a.**

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

Zadatak je napraviti aplikaciju koja vrši predviđanje vrednosti neke realne ciljne promenljive od interesa pomoću tzv. *KNN prediktora* (eng. K-Nearest Neighbors).

- [0.5p] Definisati klasu `Vektor` koja se karakteriše poljima `elementi` (`ArrayList<Double>`) i `n` (`int`, veličina odnosno broj elemenata vektora). Implementirati:
 - [1p] podrazumevani konstruktor kao i konstruktor koji prihvata vrednost za polje `elementi`
 - [0.5p] metod `int velicina()` koji vraća veličinu vektora
 - [1p] metod `void dodajElement(Double e)` koji dodaje nov element na kraj vektora
 - [1p] metod `Double uzmiElement(int ind)` koji vraća element vektora koji se nalazi na indeksu `ind` ili izbacuje `IndexOutOfBoundsException` izuzetak sa odgovarajućom porukom o grešci ukoliko takav element ne postoji
 - [3p] metod `Vektor podvektor(int i, int j)` koji vraća podvektor - novi vektor koji sadrži elemente iz opsega `[i, j]`; ukoliko granične pozicije opsega nisu validne, metod treba da izbacuje `IllegalArgumentException` izuzetak sa odgovarajućom porukom o grešci
 - [0.5p] metod `toString` koji vraća nisku formata `[elem0, elem1, ..., elemn-1]`

- [1.5p] Definisati interfejs `Metrika` koji sadrži metod `double rastojanje(Vektor u, Vektor v)`.

- [3p] Definisati klasu `EuklidskaMetrika` koja implementira interfejs `Metrika` tako da se rastojanje vektora veličine `n` izračunava po sledećoj formuli:

$$\sqrt{\sum_{i=0}^{n-1} (u_i - v_i)^2}$$

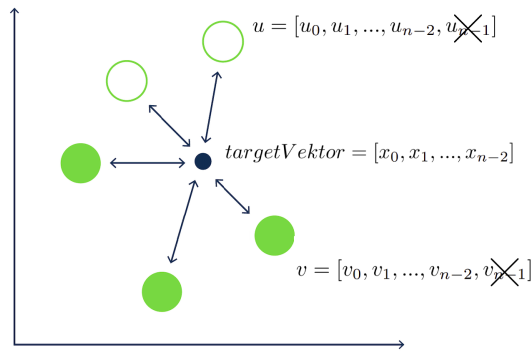
U slučaju da vektori nisu jednake veličine izbaciti `IllegalArgumentException` izuzetak sa odgovarajućom porukom o grešci.

- [3p] Definisati klasu `MenhetnMetrika` koja implementira interfejs `Metrika` tako da se rastojanje vektora veličine `n` izračunava po sledećoj formuli:

$$\sum_{i=0}^{n-1} |u_i - v_i|$$

U slučaju da vektori nisu jednake veličine izbaciti `IllegalArgumentException` izuzetak sa odgovarajućom porukom o grešci.

- [0.5p] Definisati klasu `VektorskiProstor` koja se karakteriše poljima `naziviKoordinata` (`ArrayList<String>`) i `vektori` (`ArrayList<Vektor>`). Implementirati:
 - [1p] podrazumevani konstruktor i potrebne `get` metode.
 - [6p] metod `void ucitajIzFajla(String filePath)` koji iz datoteke `filePath` učitava nazive koordinata i listu vektora kojima zatim inicijalizuje odgovarajuća polja klase, tj. objekta nad kojim je ovaj metod pozvan; primer sadržaja datoteke dat je na kraju teksta zadatka
 - [1.5p] metod `toString` koji vraća nisku koja u prvom redu sadrži nazive koordinata, a zatim i sve vektore iz liste, svaki u zasebnom redu



6. [9p] Definisati klasu `KomparatorVektora` koja implementira interfejs `Comparator<Vektor>` i karakteriše se poljima `targetVektor` (`Vektor`) i `metrika` (`Metrika`). Obezbediti konstruktor koji prihvata vrednosti za sva polja i implementirati metod koji vrši poređenje vektora prema rastojanju njihovih podvektora koji sadrže sve elemente osim poslednjeg od `targetVektor`-a u odnosu na zadatu metriku (videti sliku iznad). Pretpostaviti da je `targetVektor` uvek tačno za 1 manje veličine od datih vektora koje treba uporediti.
7. [10p] Definisati klasu `KNNPrediktor` koja se karakteriše poljima `vp` (`VektorskiProstor`), `metrika` (`Metrika`) i `k` (`int`). Implementirati konstruktor koji prihvata vrednosti za sva polja i metod `double predvidjanjeCiljnePromene(Vektor targetVektor)` koji vrši predikciju nepoznate koordinate `targetVektor`-a (što bi odgovaralo x_{n-1} prema oznakama na slici iznad) po sledećoj formuli:

$$x_{n-1} = \frac{1}{k} \sum_{i=1}^k v_{n-1}^i$$

gde su v^i , $i = 1, \dots, k$ odabranih k vektora iz vektorskog prostora `vp` koji su najbliži vektoru `targetVektor` u odnosu na prethodno definisani kriterijum poređenja vektora za datu metriku (polje `metrika`).

8. [10p] Definisati klasu `CenaNekretninaKNN` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama ispod.
- [4p] Klikom na dugme `Ucitaj bazu podataka` iz datoteke `nekretnine.txt` učitavaju se nazivi koordinata i lista vektora koji čine vektorski prostor na osnovu kojeg će se vršiti predviđanje cena novih nekretnina. Nakon učitavanja, onemogućiti ponovo klitanje na ovo dugme (slika 2). U slučaju neuspešnog učitavanja ispisati poruku o grešci na standardni izlaz za greške i prekinuti program.
 - [8p] Klikom na dugme `Predvidi cenu` prvo se čitaju vrednosti iz svih `TextField` elemenata kao i odabrani izbor metrike (`RadioButtons` elementi). Zatim se vrši predviđanje cene nekretnine pomoću `KNNPrediktor`-a i ispisuje dobijena vrednost plavom bojom u labeli pored dugmeta `Predvidi cenu` (slika 3). U slučaju da neki od unosa nije dozvoljenog formata ispisati odgovarajuću poruku crvenom bojom u labeli ispod dugmeta `Predvidi cenu` (slika 4).

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

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

```
kvadratura struktura spratnost udaljenost_od_centra cena
70 2.5 1 15.0 130000
80 3.0 5 12.5 145000
45 1.5 1 7.1 99000
105 4.0 6 14.7 220500
32 1.0 3 5.6 72000
56 2.0 2 17.2 116000
```



Slika 1: Početni izgled aplikacije



Slika 2: Učitavanje iz datoteke

KNN prediktor cena nekretnina

Kvadratura:

Struktura:

Spratnost:

Udaljenost od centra:

Izbor metrike:
 Euklidska metrika
 Menhetn metrika

Vrednost parametra k:

112400.00

Slika 3: Predviđanje cene nekretnina

KNN prediktor cena nekretnina

Kvadratura:

Struktura:

Spratnost:

Udaljenost od centra:

Izbor metrike:
 Euklidska metrika
 Menhetn metrika

Vrednost parametra k:

Nedozvoljen format brojcanih unosa

Slika 4: Neuspešno učitavanje unosa