

## Objektno-orientisano programiranje, Septembar 2

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 sintaksnih grešaka niti izbacivanje `NullPointerException`-a.

Vreme za rad: **3 sata**

Inicijalni asistenata: 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, 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.

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

- [4 poena] Definisati klasu `Tacka` koja se karakteriše poljima `x` (`double`, x-koordinata) i `y` (`double`, y-koordinata). Implementirati:
  - konstruktor koji prima sve potrebne vrednosti, konstruktor kopije i potrebne `get` metode
  - `toString` metod koji vraća nisku formata (`x`, `y`)
  - metod `double rastojanje(Tacka t)` koji računa euklidsko rastojanje između tačaka
- [2 poena] Definisati interfejs `Obim` koji sadrži metod `double obim()`.
- [2 poena] Definisati interfejs `Povrsina` koji sadrži metod `double povrsina()`.
- [14 poena] Definisati klasu `Mnogougao` koja implementira interfejse `Obim` i `Povrsina`. Klasa se karakteriše poljima `temena` (`Tacka[]`, lista temena) i `centar` (`Tacka`, geometrijski centar mnogougla). Implementirati:
  - konstruktor koji prima niz temena i centar
  - konstruktor koji prima niz koordinata temena mnogougla (`double[][]`) i centar
  - `get` metode koji vraćaju centar, niz temena, niz x-koordinata i niz y-koordinata temena mnogougla
  - `toString` metod koji vraća stringovsku reprezentaciju niza njegovih temena u formatu  $[T_0, \dots, T_{length-1}]$
  - metod `double[] duzineStranica()` koji izračunava dužine stranica mnogougla
  - metod `obim()` definisan interfejsom `Obim`
- [7 poena] Definisati klasu `Trougao` koja nasleđuje klasu `Mnogougao`. Implementirati:
  - konstruktor koji prima tri tačke A, B i C (temena trougla)
  - konstruktor koji prima niz koordinata temena trougla (`double[][]`)
  - Napomena:* Centar trougla se računa na osnovu koordinata datih temena kao težište trougla po sledećoj formuli:
$$\left( \frac{x_A + x_B + x_C}{3}, \frac{y_A + y_B + y_C}{3} \right)$$
  - metod `povrsina()` definisan interfejsom `Povrsina` (koristiti Heronov obrazac  $\sqrt{s(s-a)(s-b)(s-c)}$ )
  - metod `boolean jednakostranichni()` koji proverava da li je trougao jednakostranični (koristiti preciznost od 2 decimale prilikom poređenja dužina stranica)
- [7 poena] Definisati klasu `Pravougaonik` koja nasleđuje klasu `Mnogougao`. Implementirati:
  - konstruktor koji prima četiri tačke A, B, C i D (temena pravougaonika)
  - konstruktor koji prima niz koordinata temena pravougaonika (`double[][]`)
  - Napomena:* Centar pravougaonika se računa na osnovu koordinata datih temena kao presek dijagonala po sledećoj formuli:
$$\left( \frac{x_A + x_C}{2}, \frac{y_A + y_C}{2} \right)$$
  - metod `povrsina()` definisan interfejsom `Povrsina`
  - metod `boolean kvadrat()` koji proverava da li je pravougaonik kvadrat
- Definisati klasu `IscrtavanjeMnogouglova` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Klasa sadrži polje `mnogouglovi` (`Map<String, Mnogougao>`) u kojem će se čuvati podaci o mnogouglovima.

- [7 poena] Obezbediti da aplikacija prilikom pokretanja izgleda kao na slikama. Preporučena veličina prozora je  $520 \times 550$ . Za `Canvas` element preporučene dimenzije su  $500 \times 300$ .
- [6 poena] Implementirati pomoćni statički metod `void ucitaj(Map<String, Mnogougao> mnogouglovi)` koji učitava informacije o svim mnogouglovima iz datoteke `mnogouglovi.txt` i smešta ih u mapu `mnogouglovi`. *Napomena:* Pretpostaviti da nazivi trouglova uvek počinju slovom 'T', a nazivi pravougaonika slovom 'P'.
- [4 poena] Implementirati pomoćni statički generički metod `void ispisi(Map<K, V> mapa)` koji ispisuje sadržaj mape na standardni izlaz u formatu

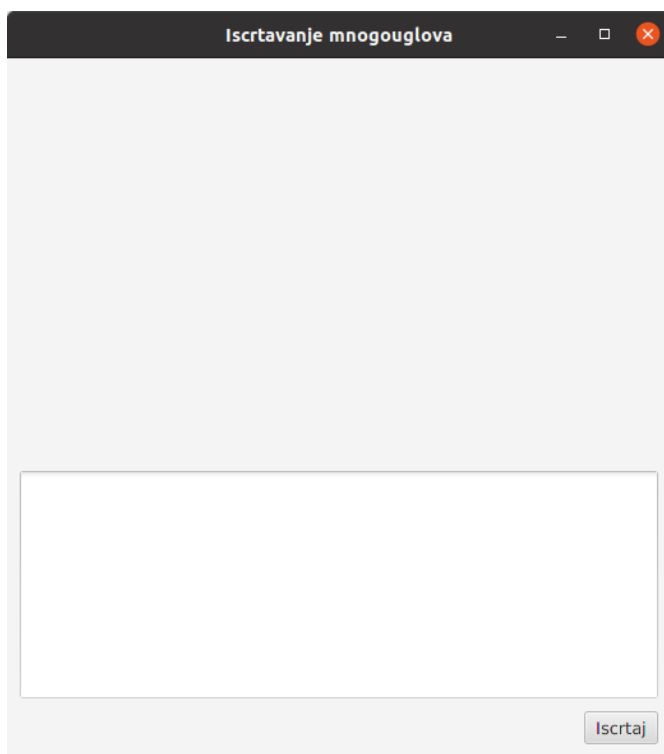
```
stringovskaReprezentacijaKljučaMape : stringovskaReprezentacijaVrednostiMape
```

i tako da svaki element bude u zasebnom redu.

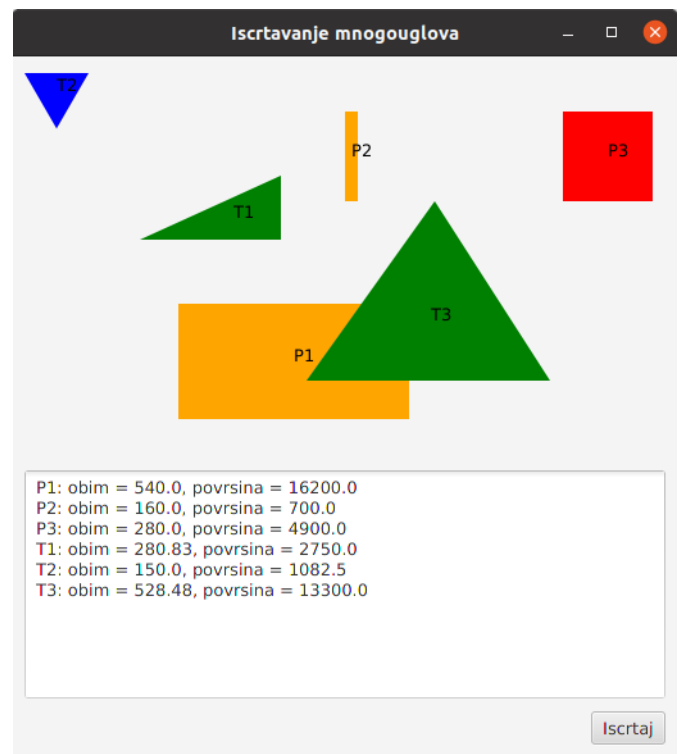
- [12 poena] Klikom na dugme 'Iscrtavanje' u `TextArea` element ispisati podatke o obimu i površini mnogouglova u odgovarajućem formatu (slika 2) i u `Canvas` element iscrtati mnogouglove, pri čemu se:
  - jednakostranični trouglovi popunjavaju plavom bojom, dok se nejednakostranični popunjavaju zelenom
  - kvadrati popunjavaju crvenom bojom, dok se pravougaonici koji nisu kvadrati popunjavaju narandžastom
 Svakom mnogouglu na koordinatama njegovog geometrijskog centra treba ispisati njegov naziv kao na slici 2. *Pomoć:* za iscrtavanje mnogouglova može se koristiti metod `fillPolygon`, a za ispisivanje naziva mnogouglova metod `fillText`.

Podaci o mnogouglovima se nalaze u datoteci `mnogouglovi.txt` u formatu `naziv x0, y0, ..., xn, yn`. Pretpostaviti da će sadržaj datoteke `mnogouglovi.txt` biti ispravan. Primer sadržaja datoteke:

```
T1 200 80 200 130 90 130
P1 300 180 300 270 120 270 120 180
P2 250 30 250 100 260 100 260 30
T2 50 0 0 0 25 43.3
P3 420 30 490 30 490 100 420 100
T3 220 240 410 240 320 100
```



Slika 1: Početni izgled aplikacije



Slika 2: Klik na dugme Iscrtaj