

**Objektno-orijentisano programiranje, , JUN 1, Grupa 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: IR: 2I1A — SS: 2I1B — VK: 2I2A, 2I2B

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. **(1 poen)** Napraviti interfejs `Takmicenje`.
2. **(2 poena)** Napraviti klasu `Igrica` kojom se opisuje jedna igrice koja je deo E-sport zajednice. Ova klasa treba da sadrži polje `String naziv` i `int godinaIzlaska`. Implementirati i:
  - Konstruktorkoji dobija vrednosti svih navedenih atributa
  - `get` metode za sve attribute
3. **(5 poena)** Napraviti klasu `ESport` kojom se opisuje jedan od aktuelnih E-sportova. Svaka instanca ove klase treba da sadrži polje koje opisuje igricu koju igraju igrači i ceo broj koji predstavlja broj aktivnih sportista. Implementirati konstruktorkoji dobija jednu instancu `Igrice` i jedan ceo broj koji predstavlja broj aktivnih registrovanih igrača. Klasa takođe treba da ima metode:
  - `double popularnost()` koji određuje popularnost određene igrice. Popularnost se računa kao broj igrača koji su godišnje postajali aktivni za određenu igru od njenog postanka. Kao trenutnu godinu računati 2021. Ako je ukupan broj igrača neke igrice 300000 a igrice je nastala 2013. godine, njena popularnost je 37500.
  - `String toString()` koji vraća `String` reprezentaciju `ESport`-a u formatu: `naziv: nazivIgrice, popularnost: popularnost`.
4. **(3 poena)** Napraviti apstraktnu klasu `Sport` koja karakteriše jedan sport. Od polja treba da sadrži jedan `String` koji predstavlja ime sporta, `boolean individualni` koji daje informaciju o tome da li je sport ekipni ili individualni. Definirati konstruktorkoji dobija vrednosti atributa klase, `get` metode za sve attribute i apstraktni metod `int ulaganje()` kojim se definiše koliko je godišnje potrebno uložiti kako bi neki takmičar bio uspešan u konkretnom sportu.
5. **(2 poena)** Napraviti klasu `Rekvizit` koja se karakteriše nazivom rekvizita koji je tipa `String` i cenom rekvizita koja je celobrojna vrednost. Definirati konstruktorkoji inicijalizuje sve attribute na vrednosti koje su mu prosledene, kao i `get` metode za sva polja ove klase.
6. **(5 poena)** Implementirati klasu `ZimskiSport` koja nasleđuje klasu `Sport` i koja ima atribut `rekviziti` tipa `List<Rekvizit>` koji čuva informacije o tome koji su sve rekviziti potrebni kako bi takmičar mogao da se takmiči u odgovarajućem zimskom sportu. Godišnje ulaganje koje se definiše za zimski sport se računa kao ukupna cena svih rekvizita koji su potrebni za takmičenje u tom sportu. Definirati i metod `String toString()` koji vraća naziv sporta, tip sporta, kao i nazive rekvizita sortirane po ceni rastuće (pogledati primer). Dodatno ispisuje i koliko je godišnje ulaganje u ovaj sport.
7. **(7 poena)** Implementirati klasu `CelosezonskiSport` koja nasleđuje klasu `Sport` i dodatno se karakteriše poljem `ulaganjeOprema` (`int`, koje predstavlja godišnje ulaganje u opremu), kao i poljem `ulaganjeTreneri` (`int`, koje predstavlja dodatno ulaganje u nutricionističke uloge, trenere itd). Godišnje ulaganje u sport se računa kao zbir ulaganja u trenere i opremu. Dodatno napisati i metod `String toString()` koji ispisuje naziv, tip sporta i godišnje ulaganje u njega (videti primer).
8. **(20 poena)** Napraviti generičku klasu `Takmicar<T>` (obebediti da `T` mora da bude tip koji implementira interfejs `Takmicenje`) koja opisuje sportistu koji može da se takmiči u nekom sportu ili E-sportu. Ova klasa treba da sadrži sledeće attribute:
  - `String ime`
  - `String prezime`
  - `int uspesnost` koji predstavlja broj osvojenih trofeja u karijeri takmičara
  - `T sport` koji daje informaciju o tome u kom se sportu takmičar takmiči

- `double osvojenaNagrada` vrednost nagrade koju je takmičar poslednju osvojio pobedom na nekom takmičenju

U ovoj klasi dodatno definisati metod `double nagrada(int ukupnaNagrada)` kojom se računa koliku nagradu dobija takmičar ukoliko pobedi na takmičenju. Takmičari koji se takmiče u individualnim sportovima ili E-sportovima dobijaju celokupnu nagradu za sebe, dok kod ekipnih sportova takmičar dobija samo 10% od ukupne nagrade, gde ostatak odlazi na ostale saigrače, klub, sponzore itd. Omogućiti sortiranje takmičara po uspešnosti opadajuće implementiranjem odgovarajućeg interfejsa. Za takmičara je potrebno napisati i metod `String toString()` koji ispisuje takmičara kao što je pokazano u primeru (poslednja vrednost predstavlja poslednju nagradu koju je takmičar osvojio na nekom takmičenju).

9. U projektu su date i 2 datoteke, *takmicari.txt* i *nagrade.txt*. Sadržaj datoteke *takmicari.txt* je u sledećem formatu:

- Oznaka sporta (E,Z,C) ostali podaci
  - Ako je oznaka sporta E, ostali podaci se definišu kao:
    - \* `brojAktivnihIgraca imeIgrice godinaIzlaskaIgrice imeTakmičara prezimeTakmičara uspešnostTakmičara`
  - Ako je oznaka sporta Z, ostali podaci se definišu kao:
    - \* `imeSporta tipSporta brojPotrebnihRekvizita imeRekvizita cenaRekvizita (n puta, gde je n broj rekvizita) imeTakmičara prezimeTakmičara uspešnostTakmičara`
  - Ako je oznaka sporta C, ostali podaci se definišu kao:
    - \* `imeSporta tipSporta troškoviOpreme troškoviZaTrenere imeTakmičara prezimeTakmičara uspešnostTakmičara`

Sadržaj datoteke *nagrade.txt* je u sledećem formatu:

- `ukupnaNagradaNaTakmičenju`

Za primere sadržaja oba fajla pogledati primer koji je dat. Svaka linije datoteke *nagrade.txt* odgovara istoj liniji u datoteci *takmicari.txt*.

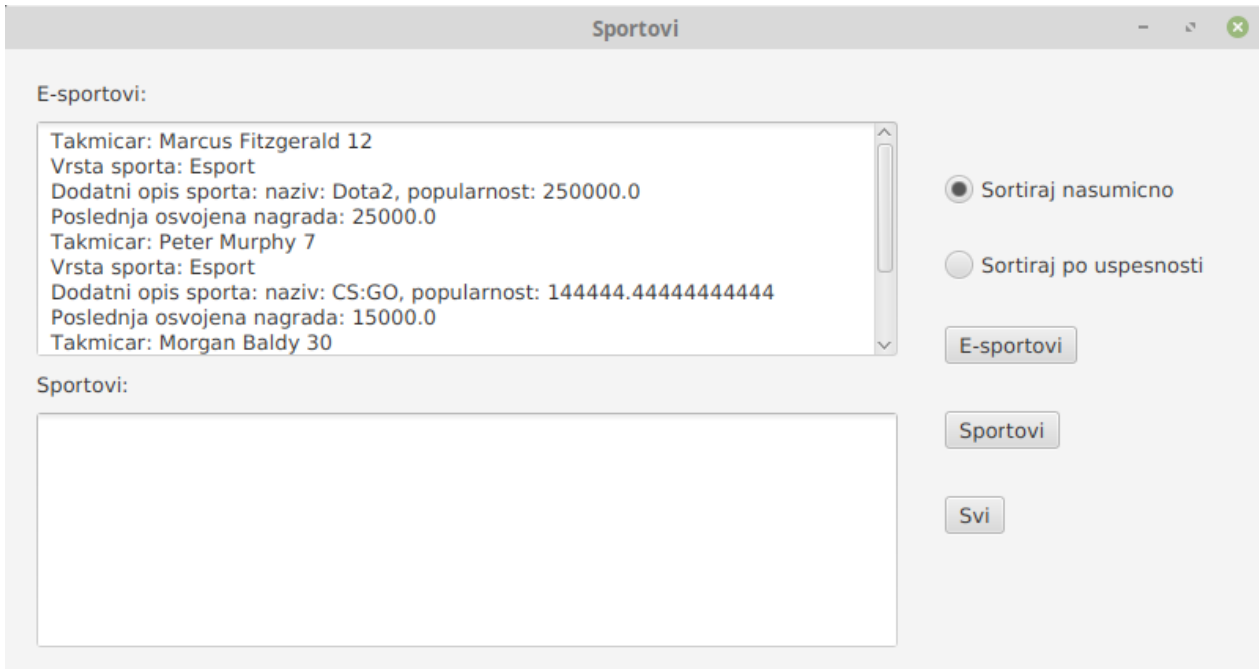
10. (15 poena) Implementirati klasu `Main` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Obezbediti da je uvek odabrano tačno jedno radio dugme kao i da je podrazumevano odabrano radio dugme koje ima vrednost `Sortiraj nasumicno`. Klikom na dugme `E-sportovi` se izlistavaju samo takmičari koji se takmiče u E-sportovima u polju koje je označeno ovom oznakom. Klikom na dugme `Sportovi` se izlistavaju samo takmičari koji se takmiče u celosezonskim sportovima u polju koje je označeno ovom oznakom. Klikom na dugme `Svi` se izlistavaju svi takmičari gde se E-sportisti smeštaju u odgovarajuće polje, a sportisti u drugo, takođe odgovarajuće polje. Ako je odabrano radio dugme `Sortiraj random` onda sportisti treba da budu prikazani u random poretku, a ako je označeno dugme `Sortiraj po uspesnoti`, sportisti treba da budu sortirani po uspešnosti opadajuće.

11. Primer sadržaja datoteke:

```
E 600000 Fortnite 2017 Morgan Baldy 30
Z skijanje individualni 3 skije 100 stapovi 70 naocare 30 Marius Cepes 47
E 2000000 Dota2 2013 Marcus Fitzgerald 12
C fudbal ekipni 2000 1000 Gianfranco Friuli 10
E 1300000 CS:GO 2012 Peter Murphy 7
C tenis individualni 1000 4000 Najbolji Ikada 86
```

12. Primer sadržaja datoteke:

```
10000
5000
25000
30000
15000
25000
```



Slika 1: Random sort E-sportovi



Slika 2: Sort po uspesnosti svi