

POPRAVNI I KOLOKVIJUM – OOP

Napisati na jeziku Java sledeće tipove, stanja i ponašanja:

Paket izuzeci

1. Tip **IgracVecPostojiException**
2. Tip **KapitenVezPostojiException**
3. Tip **NedovoljnoIgracaException**
4. Tip **NedovoljanBudzetException**
5. Tip **TransferNijeMogucException**

Paket ucesnici

Interfejs IVrednovanje

Definiše metodu:

- **dajVrednost()**
Vraća double vrednost učesnika koja.

Apstraktni tip **ClanTima** implementira **IVrednovanje**

Sadrži:

- String **ime**
- ceo broj **godine**

Metode:

- Kontruktor koji prima ime i godine
- geteri za sve promenljive
- **dajVrednost()** — apstraktna metoda za računanje vrednosti člana tima.
- **toString()** — vraća tekstualni opis člana tima u formatu **ime(godine)**.

Tip **Igrac** koji je **ClanTima**

Stanja:

- ceo broj koji predstavlja kolika je **vestina** igraca

- ceo broj koji predstavlja broj odigranih utakmica
- ceo broj koji predstavlja broj pobeda

Metode:

- konstruktor koji prima ime, godine i vestinu
- javni metod **dodajPobedu()** koji povecava broj Pobeda za 1.
- Javni metod **dodajUtakmicu()** koji evidentira da je igrač nastupio na utakmici (uvecava broj utakmica za 1).
- Javni metod **dajEfikasnost()** koji vraća realan broj koji predstavlja efikasnost igraca. Efikasnost se racuna kao odnos pobeda i odigranih utakmica uvecanih za 1 (npr ako igrac ima 0 pobeda efikasnost je 1, ukoliko igrac ima onoliko pobeda Koliko odigranih meceva, efikasnost je 2, ako je igrac pobedio 3 od 6 utakmica, efikasnost je 1.5)
- Javni metod **dajVrednost()** računa ukupnu vrednost igrača na osnovu proizvoda veštine i efikasnosti.

Tip Kapiten koji nasleđuje Igrac

Metoda:

- Konstruktor koji prima ime, godinu i vestinu
- **dajVrednost()** vraća povećanu vrednost igrača zbog liderske uloge kapitena za 10%.

Tip Trener koji je ClanTima

Stanje:

- ceo broj koji predstavlja godine **iskustva** trenera

Metoda:

- Konstruktor koji prima ime, godine i iskustvo
- Metodu **dajVrednost()** koja zavisi od broja godina iskustva (ako trener ima n godina iskustva, njegova vrednost je suma od 1 do n)

Paket timovi

Tip Tim

Sadrži:

- Javnu String promenljivu **naziv**
- Javnu realnu promenljivu **budžet**
- Javnu listu **igraca** koja predstavlja igrace tima
- Janu promenljivu **trenera**
- Celobrojne promeljive koje predstavljaju broj **pobeda** i broj **poraza** tima

Metode:

- Kontruktor koji prima naziv i budzet, ostale promenljive setovati na podrazumevane vrednosti
- Javni metod **daLiJeIgracUTimu(Igrac igrac)** koji vraca logicku promenljivu koja odredjuje da li je prosledjeni igrac u timu ili ne
- Javni metod **dodajIgraca(Igrac igrac)** koji dodaje igrača u tim, ako igrac već nije član tima. Ukoliko je igrac vec u timu metoda baca izuzetak IgracVecPostojiException. Ukoliko se dodaje Kapitan, metoda moze baciti izuzetak KapitenVecPostojiException ukoliko taj tim vec ima kapitena u timu.
- Javni metod **ukloniIgraca(Igrac igrac)** koji uklanja igrača iz tima ukoliko postoji.
- Javni metod **postaviTrenera(Trener trener)** koji postavlja trenera ekipe.
- Javni metod **ukloniTrenera()** koji uklanja trenera tima
- Javni metod **promeniBudzet(double iznos)** koji setuje prosledjeni budžet tima.
- Javni metod **evidentirajPobedu()** Uvećava broj pobeda i azurira statistiku igraca (odigrane utakmice, pobeđe).
- Javni metod **evidentirajPoraz()** Uvećava broj poraza i azurira statistiku igraca (odigrane utakmice).
- Javni metod **dajUkupnuSnaguTima()** računa ukupnu snagu sabiranjem vrednosti svih članova.
- Javni metod **dajNajboljegIgraca()** koji vraća igrača sa najvećom vrednošću u timu.

- **toString** metodu koja vraca tim u obliku **naziv(brojPobeda – brojPoraza)**

Tip **Transfer**

Sadrzi staticku metodu **transferujigraca**(Tim prviTim, Tim drugiTim, Igrac igrac)

Opis:

- proverava da li igrač postoji u prvomTimu, ukoliko nije u timu baca izuzetak TransferNijeMogucException
- proverava budžet drugog tima (cena transfera je vrednostilgraca), ukoliko budzet nije odgovarajuci baca izuzetak NedovoljanBudzetException
- uklanja igrača iz prvog tima,
- dodaje ga u novi tim,
- ažurira budžete (za cenu transfera),

*Paket **takmicenje***

Enum StatusUtkmice

- ZAKAZANA
- ODIGRANA
- OTKAZANA

Tip **Utkmica**

Sadrži:

- Celobrojni identifikator utakmice (automatski se dodeljuje, ne prosledjuje se konstruktoru)
- dva tima
- status utakmice (podrazumevana vrednost je ZAKAZANA)
- privatnu Referencu na pobednika utakmice

Metode:

- Javni metod **odigrajUtakmicu()** koji proverava broj igrača, generiše pobednika na osnovu snage timova i ažurira svu potrebnu statistiku. Ako oba tima imaju iste snage, rezultat je nerešen (pobednik je null). Da bi se utakmica odigrala oba tima moraju imati bar po 5 igrača, u suprotnom se baca izuzetak `NedovoljnoIgracaException`
- Javni metod **dajPobednika()** koji vraća tim koji je pobedio ili null ako je nerešeno.

Paket liga

Tip Liga

Sadrži:

- String promenljivu naziv koji predstavlja naziv lige
- listu timova
- listu utakmica

Metode:

- Javni metod **dodajTim(Tim tim)** koji dodaje tim u ligu.
- Javni metod **zakaziUtakmicu(Tim t1, Tim t2)** koji kreira novu utakmicu između dva tima.
- Javni metod **odigrajSezonu()** koji odigrava sve zakazane utakmice.
- Javni metod **tabela()** koji vraća sortiranu listu timova po broju pobeda

Default paket

Klasa Test

Potrebno je:

- napraviti nekoliko timova,
- dodati igrače i trenere,
- dodati kapitene
- izvršiti 2 transfera (Jedan uspešan I jedan neuspešan),
- Ispisati snage timova u ligi
- zakazati i odigrati utakmice,

- ispisati rezultate lige,
- Ispisati snage timova nakon odigranih utakmica
- Ispisati najboljeg igrača svakog tima