

U programskom jeziku JAVA potrebno je kreirati sledeće tipove:

Paket common

Apstraktni tip **MerljivUcinak** koji ima sledeće metode:

- **indeksKorisnosti** koja ima *celobrojni* povratni tip

Apstraktni tip **Uporediv** koji ima sledeće metode:

- **uporedi** koja vraća *celobrojni* tip i potencijalno baca izuzetak **Neuporedivo**

Paket izuzeci

Tip **NemaDovoljnoNovca** koji je izuzetak i ima konstruktor koji prihvata poruku i setuje je.

Tip **Neuporedivo** koji je izuzetak i ima konstruktor koji prihvata poruku i setuje je.

Tip **SportistaNijeOdgovarajuci** koji je izuzetak i ima konstruktor koji prihvata poruku i setuje je.

Paket osobe

Tip **Osoba** ima osobinu **Uporediv** i sledeća stanja i ponašanja:

- privatnu *String* promenljivu **ime**
- javni konstruktor koji prihvata ime i setuje ga
- javni metod **uporedi** koji vraća *celobrojnu* promenljivu i prihvata tip **Object**. Metod potencijalno baca izuzetak **Neuporedivo** u slučaju kada instanca osobe za koju se metod poziva i prosleđena instanca **Object** nemaju osobinu **merljivog učinka**. Ukoliko obe instance imaju osobinu **merljivog učinka** metod vraća 1 ukoliko je indeks korisnosti instance za koji se metod poziva veći od indeksa korisnosti prosleđenog objekta, -1 u suprotnom i 0 ukoliko su jednaki.
- getter za ime
- pregaziti metod **toString** tako da vraća ime osobe

Apstraktni tip **Sportista** koji je **Osoba**, ima osobinu **merljivog učinka** i ima sledeća stanja i ponašanja:

- privatnu promenljivu **menadzer** tipa *Menadzer*
- privatnu *celobrojnu* promenljivu **brojNaDresu**
- privatnu promenljivu **klub** tipa *Klub*
- javni konstruktor koji prihvata ime i brojNaDresu
- prepisanu apstraktnu metodu **indeksKorisnosti** iz *merljivog učinka*
- apstraktni metod **dodajStatistiku** koji prihvata tip *Statistika*
- settere za **menadzera** i **klub**
- javni metod **equals** koji vraća *boolean* promenljivu i prihvata *celobrojnu* promenljivu **brojNaDresu**. Ukoliko je prosleđeni broj na dresu isti kao kod sportiste za kojeg se metod poziva, vraća se **true** u suprotnom se vraća **false**.

Tip **Fudbaler** koji je **Sportista** i ima sledeća stanja i ponašanja:

- *celobrojne* promenljive **brojGolova** i **brojAsistencija**
- javni konstruktor koji prihvata **ime** i **brojNaDresu**

- javni pregaženi metod **indeksKorisnosti** koji vraća indeks korisnosti fudbalera kao **brojGolova * 10 + brojAsistencija * 5**
- javni pregaženi metod **dodajStatistiku** koji prihvata tip *Statistika* i na osnovu **enuma TIP_STATISTIKE** dodaje ostvareni učinak iz iste na **brojGolova** ili **brojAsistencija**

Tip **Kosarkas** koji je **Sportista** i ima sledeća stanja i ponašanja:

- *celobrojne* promenljive **brojKoseva** i **brojAsistencija**
- javni konstruktor koji prihvata **ime** i **brojNaDresu**
- javni pregaženi metod **indeksKorisnosti** koji vraća indeks korisnosti fudbalera kao **brojKoseva + brojAsistencija**
- javni pregaženi metod **dodajStatistiku** koji prihvata tip *Statistika* i na osnovu **enuma TIP_STATISTIKE** dodaje ostvareni učinak iz iste na **brojKoseva** ili **brojAsistencija**

Tip **Menadzer** koji je **Osoba** ima osobinu **merljivog učinka** i ima sledeća stanja i ponašanja:

- niz **sportisti** tipa *Sportista* (ovaj niz predstavlja sportiste koje menadžer zastupa)
- javni konstruktor koji prihvata ime i instancira niz sportista na 10 elemenata
- javni pregaženi metod **indeksKorisnosti** koji vraća sumu indeksa korisnosti svih sportista koje menadžer zastupa
- javni metod **potpisiUgovorSaKlubom** koji prihvata *Sportistu* i niz *klubova* sa kojima menadžer treba da potpiše ugovor. Ukoliko je potpisivanje uspešno vratiti **true** i setovati menadzera kod sportiste, u suprotnom vratiti **false**

Paket klub

Apstraktni tip **Klub** koji ima sledeća stanja i ponašanja:

- *String* promenljivu **nazivKluba**
- *double* promenljivu **budzet**
- niz **igraci** tipa *Sportista*
- javni konstruktor koji prihvata nazivKluba i budzet i setuje ih
- javni apstraktni metod **potpisiUgovor** koji vraća *boolean* tip, prihvata **sportistu** tipa *Sportista* i *double* promenljivu koja predstavlja **cenu**
- javni metod **dodajStatistikulgracu** koji prihvata **statistiku** tipa *Statistika*, pronalazi igrača na osnovu broja dresa iz statistike i kod njega je setuje
- javni metod **equals** koji prihvata *String* promenljivu **nazivKluba** i vraća true/false u zavisnosti od toga da li klub za koji se poziva metod ima isti naziv kao prosleđeni parametar ili ne
- pregaziti metod **toString** tako da vraća nazivKluba

Tip **KosarkaskiKlub** koji je **Klub** i ima sledeća stanja i ponašanja:

- javni konstruktor koji prihvata naziv kluba i budzet i setuje ih (košarkaški klub ima 5 igrača)
- javni pregaženi metod **potpisiUgovor** koji prihvata *Sportistu* i njegovu *cenu*. Metod potencijalno baca izuzetke **NemaDovoljnoNovca**, ukoliko budzet kluba nije dovoljan, **SpoustistaNijeOdgovarajući** ukoliko prosleđeni sportista nije košarkaš. Ako je sportista odgovarajući i klub ima novca, sačuvati sportistu u nizu sportista, umanjiti budžet i vratiti **true**.

Tip **FudbalskiKlub** koji je **Klub** i ima sledeća stanja i ponašanja:

- javni konstruktor koji prihvata naziv kluba i budžet i setuje ih (fudbalski klub ima 11 igrača)
- javni pregaženi metod **potpisiUgovor** koji prihvata **Sportistu** i njegovu **cenu**. Metod potencijalno baca izuzetke **NemaDovoljnoNovca**, ukoliko budžet kluba nije dovoljan, **SportistaNijeOdgovarajući** ukoliko prosleđeni sportista nije fudbaler. Ako je sportista odgovarajući i klub ima novca, sačuvati sportistu u nizu sportista, umanjiti budžet i vratiti **true**.

Paket turnir

Enum tip **TIP_STATISTIKE** koji ima sledeće vrednosti: *GOLOVI*, *ASISTENCIJE*, *KOSEVI*

Enum tip **TIP_SPORTA** koji ima sledeće vrednosti: *KOSARKA*, *FUDBAL*

Tip **Statistika** koji ima sledeća stanja i ponašanja:

- privatnu promenljivu **tipStatistike** tipa *TIP_STATISTIKE*
- privatnu *celobrojni* promenljivu **brojNaDresu** koja predstavlja broj sportiste za koga je statistika upisana
- privatnu *String* promenljivu **nazivKluba**, koja predstavlja klub za koji igrač nastupa
- privatnu *celobrojni* promenljivu **ostvareniUcinak** koja predstavlja ostvarenu statistiku
- Javni konstruktor koji prihvata sve promenljive i setuje ih
- gettere za sve promenljive

Tip **Utakmica** koji ima sledeća stanja i ponašanja:

- dve promenljive **domacin**, **gost** tipa *Klub*
- Javni konstruktor koji prihvata domaći i gostujući klub i setuje ih
- Javni metod **equals** koji prihvata *String* promenljivu **domacin** i *String* promenljivu **gost**. Ukoliko se nazivi prosleđenih parametara podudaraju sa nazivima odgovarajućih klubova vratiti **true**, u suprotnom vratiti **false**
- Javni metod **dodajStatistiku** koji prihvata promenljivu **statistika** tipa *Statistika* i na osnovu **naziva kluba** iz iste, setuje statistiku domaćem ili gostujućem klubu

Tip **Turnir** koji ima sledeća stanja i ponašanja:

- dva niza **kosarkaskeUtakmice** i **fudbalskeUtakmice** tipa *Utakmica*
- javni konstruktor koji instancira nizove utakmica na 10 elemenata
- javni metod **dodajUtakmicu** koja prihvata utakmicu i setuje je u odgovarajući niz.
- Javni metod **upisiStatistiku** koja prihvata promenljivu **tipSport** tipa *enum TIP_SPORTA*, *String* promenljive **domacin** i **gost** kao i promenljivu **statistika** tipa *Statistika* koju treba setovati. Metod na osnovu prosleđenih podataka poziva setovanje statistike za odgovarajuću utakmicu.

Default paket

U nastavku su prikazane klase koje je potrebno instancirati i potrebno je dodati sledeće stvari:

1. Menadžeri potpisuju ugovor sa svakim trećim sportistom, tako što prvi menadžer počne od prvog sportiste, drugi od drugog, treći od trećeg. Cena svakog igrača je 1000.
2. Ispisati string reprezentacije osoba sortirane po indeksu korisnosti u rastućem poretku.
3. Pronaći najboljeg košarkaša i fudbalera
4. Pronaći menadžera koji ima najveći indeks korisnosti.

```
Osoba sportisti[] = new Sportista[18];
```

```

sportisti[0] = new Kosarkas("Milos Teodosic", 4);
sportisti[1] = new Kosarkas("Nemanja Bjelica", 5);
sportisti[2] = new Kosarkas("Bogdan Bogdanovic", 6);
sportisti[3] = new Kosarkas("Majk Cirbes", 7);
sportisti[4] = new Kosarkas("Stefan Jovic", 8);
sportisti[5] = new Kosarkas("Marko Guduric", 9);
sportisti[6] = new Kosarkas("Nikola Kalinic", 10);
sportisti[7] = new Kosarkas("Luka Mitrovic", 11);
sportisti[8] = new Kosarkas("Nikola Milutinov", 12);
sportisti[9] = new Fudbaler("Sasa Ilic", 4);
sportisti[10] = new Fudbaler("Marko Perovic", 5);
sportisti[11] = new Fudbaler("Filip Stojkovic", 6);
sportisti[12] = new Fudbaler("Dejan Savicevic", 7);
sportisti[13] = new Fudbaler("Dejan Stankovic", 8);
sportisti[14] = new Fudbaler("Dejan Govedarica", 9);
sportisti[15] = new Fudbaler("Piksi Stojkovic", 10);
sportisti[16] = new Fudbaler("Peter Ceh", 11);
sportisti[17] = new Fudbaler("Ivica Kralj", 12);

Klub klubovi[] = new Klub[6];
klubovi[0] = new FudbalskiKlub("FK Crvena Zvezda", 3000);
klubovi[1] = new FudbalskiKlub("FK Partizan", 3000);
klubovi[2] = new FudbalskiKlub("FK Radnicki", 3000);
klubovi[3] = new KosarkasiKlub("KK Crvena Zvezda", 3000);
klubovi[4] = new KosarkasiKlub("KK Partizan", 3000);
klubovi[5] = new KosarkasiKlub("KK Mega Vizura", 3000);

```

```

Menadzer menadzeri[] = new Menadzer[3];
menadzeri[0] = new Menadzer("Menadzer 1");
menadzeri[1] = new Menadzer("Menadzer 2");
menadzeri[2] = new Menadzer("Menadzer 3");

```

//TO DO 1
// Menadžeri pokušavaju da potpišu ugovor sa svakim trećim sportistom, tako što prvi menadžer počne od prvog sportiste, drugi od drugog, treći od trećeg.

```

Turnir turnir = new Turnir();

```

```

// Primer dodavanja
turnir.dodajUtakmicu(new Utakmica(klubovi[0], klubovi[1]));
turnir.dodajUtakmicu(new Utakmica(klubovi[0], klubovi[2]));
turnir.dodajUtakmicu(new Utakmica(klubovi[1], klubovi[2]));
turnir.dodajUtakmicu(new Utakmica(klubovi[3], klubovi[4]));
turnir.dodajUtakmicu(new Utakmica(klubovi[4], klubovi[5]));
turnir.dodajUtakmicu(new Utakmica(klubovi[3], klubovi[5]));

```

//TO DO 2 - Sortiranje sportista po indeksu korisnosti
//TO DO 3 - Pronaći najboljeg košarkaša i fudbalera
//TO DO 4 - Pronaći menadžera koji ima najveći indeks korisnosti.