

Objektno orijentisano programiranje – kolokvijum

18. 04. 2013.

Napomene:

- *Kolokvijum se radi dva sata*
- *Otvoriti Eclipse i napraviti nov projekat pod nazivom **ImePrezimeIndeks***
- *Projekat smestiti na **E:\<ime_prezime_indeks>***
- *U okviru projekta napraviti paket pod imenom **imePrezimeIndeks***
- *Zadatak se predaje snimanjem na fleš*
- *Projekat ne sme sadržati greške (mora se kompajlirati)*
- *Dodatna literatura, kao i gotovi projekti nisu dozvoljeni (Workspace mora biti prazan)*
- *Obavezno napisati klasu sa main metodom*

Zadatak: Olimpijske igre

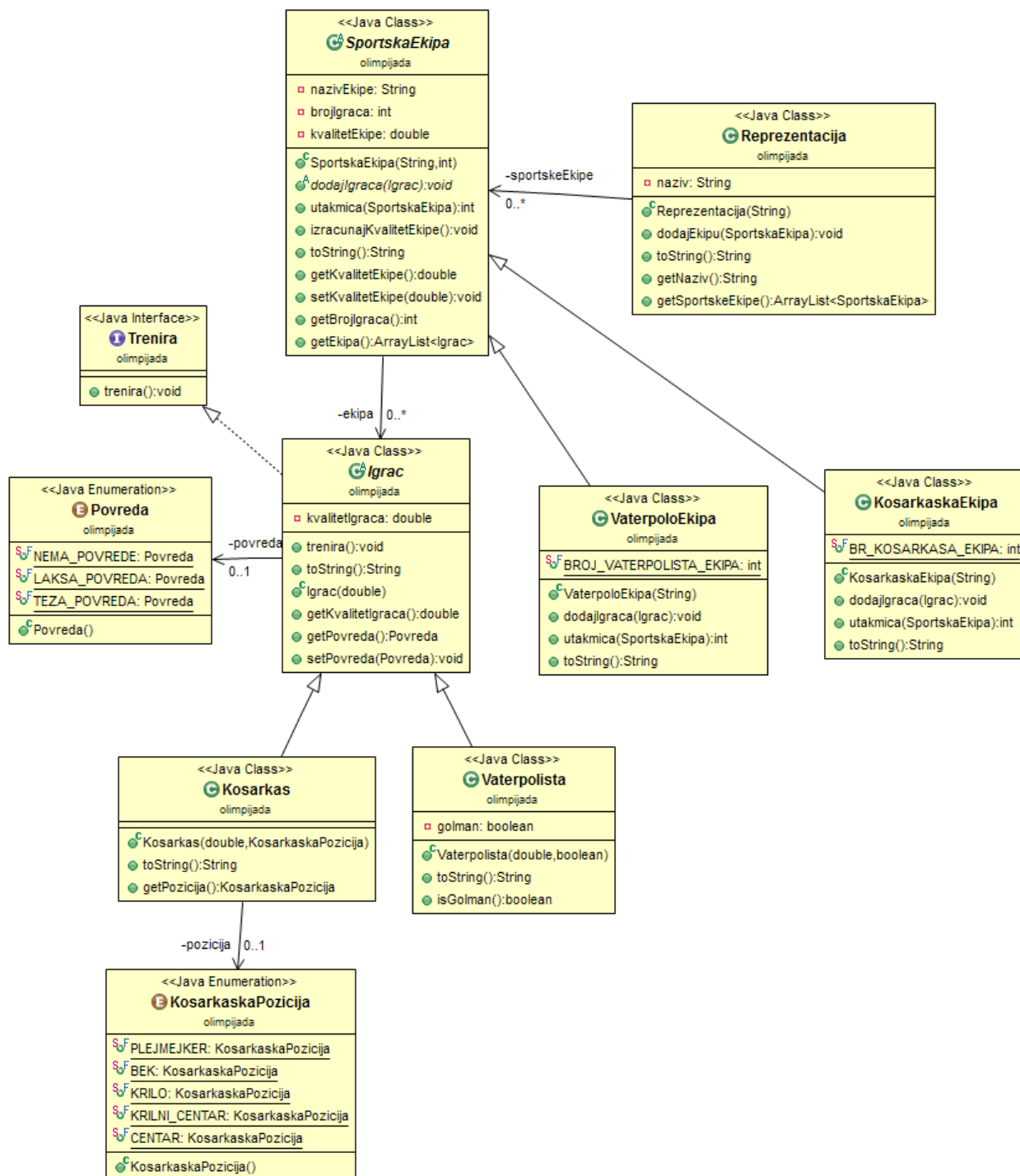
Napisati Java (konzolnu) aplikaciju za potrebe održavanja Olimpijskih igara u Srbiji. Na Olimpijskim igrama učestvuju reprezentacije, a svaka reprezentacija se sastoji od ekipa (za jedan sport ide jedna ekipa). Svaku ekipu čine sportisti, koje karakteriše kvalitet, kao i eventualna povreda. Igrači mogu trenirati i na taj način poboljšavati kvalitet, ali i rizikovati povredu. Individualni kvaliteti igrača neke ekipe direktno utiču na kvalitet cele ekipe.

Na Olimpijskim igrama u Srbiji, reprezentacije će se takmičiti u dva sporta – košarka i vaterpolo.

Sportske ekipe mogu odigravati utakmice, naravno ako pripadaju istom sportu (npr. dve vaterpolo ekipe). Ekipa koja je kvalitetnija ima većih šansi da pobedi u pojedinačnoj utakmici.

Igrači u košarkaškoj ekipi mogu igrati na različitim pozicijama (centar, plejmejker...), dok se kod vaterpolista razlikuju golmani od ostalih igrača. Svaka vaterpolo ekipa mora imati makar jednog golmana, dok svaka košarkaška ekipa mora imati makar jednog centra. Dodatno, broj igrača u košarkaškoj ekipi može biti najviše dvanaest, a u vaterpolo ekipi najviše trinaest.

Na slici je prikazan UML dijagram klasa koji predstavlja aplikaciju.



Metode i klase koje treba implementirati:

- **Povreda**
 - enum, sa vrednostima NEMA_POVREDE (ovo je podrazumevana vrednost za atribut *povreda* klase *Igrac*), LAKSA_POVREDA, TEZA_POVREDA
- **KosarkaskaPozicija**
 - enum, sa vrednostima PLEJMEJKER, CENTAR, BEK, KRILNI_CENTAR, KRILO
- **Trenira (interfejs)**
 - sadrži samo jednu metodu *trenira()*
- **Igrac (apstraktna klasa)**
 - *trenira* (iz interfejsa *Trenira*) – ako je povređen (bilo da se radi o lakšoj ili težoj povredi), ne raditi ništa; ako nije, uvećati kvalitet za 1 i odrediti sa verovatnoćom 30% da li će doći do povrede (ako dođe do povrede, onda sa verovatnoćom 50% odrediti da li se radi o težoj i lakšoj povredi)
 - *getKvalitetIgrača()* – ako nema povrede, vraća vrednost atributa *kvalitetIgrača*, a ako ima povrede onda vratiti 0, ako se radi o težoj povredi, odnosno polovinu vrednosti atributa *kvalitetIgrača*, ako se radi o lakšoj povredi
 - za ostale attribute uraditi getere i setere, kao i *toString* metod
- **Kosarkas – nasleđuje klasu Igrac**
 - dodatno ima atribut *pozicija* (*KosarkaskaPozicija*), kao i par geter-seter za taj atribut i metod *toString*
- **Vaterpolista – nasleđuje klasu Igrac**
 - dodatno ima *boolean* atribut *golman* (određuje da li je golman, ili ne), takođe geter za taj atribut i *toString* metod
- **SportskaEkipa (apstraktna klasa)**
 - sadrži apstraktnu metodu *dodajIgraca(Igrac igrac)*
 - metoda *izracunajKvalitetEkiye()* – računa kvalitet ekipe kao aritmetičku sredinu kvaliteta svih igrača (voditi računa da na kvalitet igrača utiče i eventualna povreda)
 - geter *getKvalitetEkiye()* – najpre računa kvalitet ekipe (pozivom metode *izracunajKvalitetEkiye()*), a potom vraća vrednost atributa *kvalitetEkiye*
 - metoda *utakmica(SportskaEkipa drugaEkipa)* – na slučajan način određuje koja će ekipa pobediti. Verovatnoću da će neka ekipa pobediti izračunati srazmerno razlici u kvalitetu ekipa (npr. ako jedna ekipa ima kvalitet 9, a druga 7, verovatnoća da će prva ekipa pobediti će biti 60%)

- geteri i seteri za ostale attribute, kao i *toString* metod
- **Klase KosarkaskaEkipa i VaterpoloEkipa – potklase klase SportskaEkipa**
 - *dodajIgraca(Igrac drugiIgrac)* metod – dodaje novog igrača u listu. Voditi računa da se u košarkašku ekipu dodaju samo košarkaši, a u vaterpolo ekipu samo vaterpolisti. Takođe, u košarkaškoj ekipi može biti najviše 12 košarkaša, a u vaterpolo ekipi najviše 13 vaterpolista
 - *utakmica(SportskaEkipa drugaEkipa)* – najpre proveriti da li je moguće da se održi utakmica: ako se radi o košarkaškoj utakmici, u obe ekipe mora biti najmanje po jedan centar, dok, u slučaju vaterpolo meča, obe ekipe moraju imati najmanje po jednog golmana, a onda, ako je sve ispravno, pozvati metod *utakmica* natklase (pomoću *super.utakmica(drugaEkipa)*). Metoda vraća 0, ako dva tima ne mogu odigrati utakmicu, 1, ako je pobednik domaćin, i 2, ako je pobednik gost
 - *toString* metod
- **Klasa Reprezentacija**
 - samo geteri i seteri i *toString* metod

Potrebno je napisati i klasu *Olimpijada* koja sadrži *main* metod i testira rad klasa. U okviru *main* metode uraditi sledeće:

- kreirati pet košarkaša i tri košarkaške ekipe i u dve košarkaške ekipe staviti po dva igrača, ali da obe imaju po jednog centra, dok u treću ekipu staviti samo jednog košarkaša, koji nije centar („neispravna“ ekipa)
- simulirati utakmicu između jedne „ispravne“ i jedne „neispravne“ ekipe
- proizvoljan broj puta trenirati igrače (prikazati kako napreduje njihov kvalitet i da li imaju povrede)
- simulirati utakmicu između dve „ispravne“ ekipe i prikazati rezultat
- napraviti dvojicu vaterpolista i dodati ih u vaterpolo ekipu
- napraviti dve reprezentacije („Srbija“ i „Španija“) i dodati im ekipe po želji i ispisati ih na konzolu

Napomena: metod *nextInt(int br)*, klase *Random* vraća slučajan ceo broj u intervalu od 0 do *br-1*, dok metod *nextBoolean()*, vraća slučajnu *boolean* vrednost (može biti korisno prilikom određivanja da li dolazi do povreda igrača ili ne, kao i kod odlučivanja ko je pobednik).