

Objektno orijentisano programiranje – kolokvijum

Zadatak: Dvoboj čudovišta

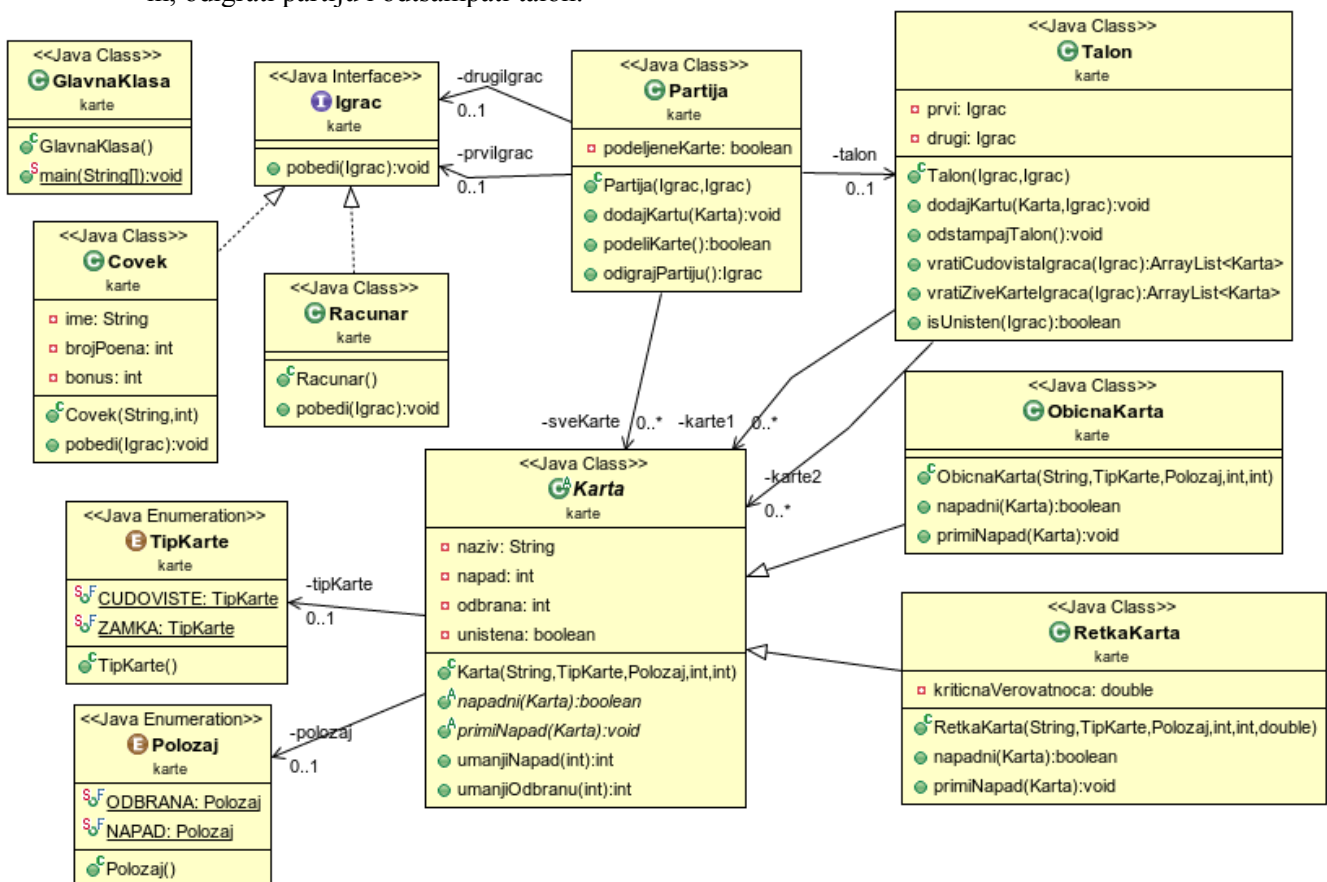
Napisati Java konzolnu aplikaciju kojom se simulira igra kartama Dvoboj čudovišta. U ovoj verziji postoje dve vrste karata – obične i retke. Svaku karakteriše tip karte – čudovište ili zamka i u skladu sa tipom one imaju različite efekte. Karte mogu biti u položaju za napad ili odbranu i u zavisnosti od položaja imaju različitu energiju. U jednom potezu, jedna karta može promeniti položaj ili, ako je karta čudovište, napasti (u tom slučaju automatski prelazi u položaj za napad i ostaje u njemu za taj potez). Karte zamke su strogo odbrambene i aktiviraju se kada ih protivničko čudovište napadne, posle čega bivaju uništene. U igri učestvuju dva igrača, od kojih svaki može biti čovek ili računar. Pobjednik je igrač koji prvi ostavi protivnika bez karata, ili ako obojica izgube sva čudovišta, onaj koji ima više preostalih poena na talonu.

Potrebno je implementirati sledeće klase i metode:

- (6) prepisati kompletan UML dijagram klase
- (1) Implementirati sve getere i setere za svaki atribut u svakoj klasi i implementirati toString metodu u svakoj klasi kojom se ispisuju svi atributi klase
- **Karta (abstract)**
 - (0.5) *umanjiNapad(int energija)* – umanjuje broj poena u napadu za datu energiju i setuje na 0 ako je novi broj poena manji od 0
 - (0.5) *umanjiOdbranu(int energija)* – umanjuje broj poena u odbrani za datu energiju i setuje na 0 ako je novi broj poena manji od 0
- **ObicnaKarta**
 - (0.5) *napadni(Karta protivnickaKarta)* – menja svoj položaj u NAPAD. Poziva metodu *primiNapad* na protivničkoj karti (u tom pozivu prosleđuje sebe). Vraća da li je protivnička karta uništena.
 - (1) *primiNapad(Karta protivnickaKarta)* – ako je karta zamka, protivnička karta mora da primi napad od nje, posle čega se zamka uništava. U suprotnom, u zavisnosti od svog položaja, umanjuje svoje poene za napad ili odbranu za poene napada protivničke karte; ako se broj poena smanjio na 0, karta je uništena.
- **RetkaKarta**
 - (1) *napadni(Karta protivnickaKarta)* – menja svoj položaj u NAPAD. Sa verovatnoćom *kriticnaVerovatnoca* samo za trenutni napad povećava se broj poena za 50000, i ako se to desi, posle napada se umanjuje za 50000. Poziva metodu *primiNapad* na protivničkoj karti (u tom pozivu prosleđuje sebe). Vraća da li je protivnička karta uništena.
 - (2) *primiNapad(Karta protivnickaKarta)* – ako je karta zamka, sa verovatnoćom *kriticnaVerovatnoca* za trenutni napad povećava se broj poena u napadu za 50000. Protivnička karta mora da primi napad od zamke, posle čega se zamka uništava. Ako karta nije zamka, ako je u položaju za napad, umanjuje svoje poene za napad za poene napada protivničke karte; ako se broj poena smanjio na 0, karta postaje uništena. Ako je u položaju za odbranu, umanjuje svoj broj poena odbrane samo sa verovatnoćom (1-*kriticnaVerovatnoca*), posle čega takođe može biti uništena.
- **Talon**
 - (1) *dodajKartu(Karta karta, Igrac igrac)* dodaje kartu u listu karata koje pripadaju datom igraču
 - (2) *odstampajTalon()* u tekstualnu datoteku *talon.txt* ispisuje "Prvi igrač: ", sve karte prvog igrača, "Drugi igrač: " i sve karte drugog igrača na talonu (bez obzira na to da li su uništene)
 - (1) *vратиCudovistaIgraca(Igrac i)* vraća sva neuništena čudovišta u posedu prosleđenog igrača
 - (1) *vратиZiveKarteIgraca(Igrac i)* vraća sve neuništene karte u posedu prosleđenog igrača
 - (0.5) *isUnisten(Igrac i)* vraća true ako i samo ako su sva čudovišta prosleđenog igrača uništena
- **Partija**
 - (0.5) *dodajKartu(Karta k)* dodaje kartu u špil za trenutnu partiju
 - (1) *podeliKarte()* ako ima više od 10 karata u špil, naizmenično dodeljuje po jednu nasumičnu

kartu svakom igraču (pomoću *talon.dodajKartu*) i izbacuje je iz špila. Vraća false ako na početku ima manje od 8 karata u špilu.

- (4) *odigrajPartiju()* simulira partiju i vraća pobednika. Ako karte nisu uspešno podeljene, vraća *null*. Dok neki od igrača nije uništen, naizmenično igraju prvi i drugi igrač tako što biraju nasumično jedno od svojih čudovišta (ako postoji) i sa verovatnoćom od 0.5 prebacuju ga u odbranu ili napadaju nasumično odabranu živu kartu protivnika. Ako je protivnički igrač uništen, pobednik je trenutni igrač (poziva se *igrac.pobedi(protvnickiIgrac)* i vraća se pobednik).
- **Racunar**
 - (0.5) *pobedi(Igrac protivnik)* – ispisuje poruku da je Računar pobedio
- **Covek**
 - (0.5) *pobedi(Igrac protivnik)* – povećava broj poena čoveka za 10. Ako je protivnik čovek, povećava broj poena čoveka za bonus protivnika. Ispisuje poruku da je čovek pobedio i broj poena.
- **GlavnaKlasa**
 - (0.5) *main(String args[])* - kreirati 8 karata (po dve obična obična i dva retka čudovišta i po dve obične i dve retke zamke), jednog čoveka, jedan računar, napraviti partiju, dodati sve karte, podeliti ih, odigrati partiju i odtšampati talon.



Dijagram klasa