

OOP - K1 - PRIPREMA

Napisati klasu **Igrac** koja sadrži polja:

- **poeni** (tipa **int**)
- **brojPartija** (tipa **int**)

Za klasu **Igrac** implementirati:

- konstruktor sa podrazumevanim vrednostima za parametre - podrazumevana vrednost za poene je 0, podrazumevana vrednost za broj partija je 0
- konstruktor kopije
- metodu **void dodajPoene(int);** - dodaje igraču broj poena koji je prosleđen kao parametar
- metodu **void odigrajPartiju();** - povećava broj partija za 1
- metodu **void resetuj();** - postavlja broj odigranih partija i broj poena na 0
- metodu **void ispis() const;** - ispisuje sve informacije o igraču
- get metode za oba polja
- set metodu za polje poeni

Napisati klasu **Igra** koja modeluje proces ostvarivanja bodova prilikom igranja igrice na računaru.

Klasa **Igra** sadrži polja:

- **igrac** (tipa **Igrac**)
- **trenutnoStanje** (nabrojivog tipa sa mogućim vrednostima **NEPOZNAT_IGRAC**, **PROFIL_IGRACA**, **IGRA_RACUNAR**, **IGRA_PROTIVNIK**)

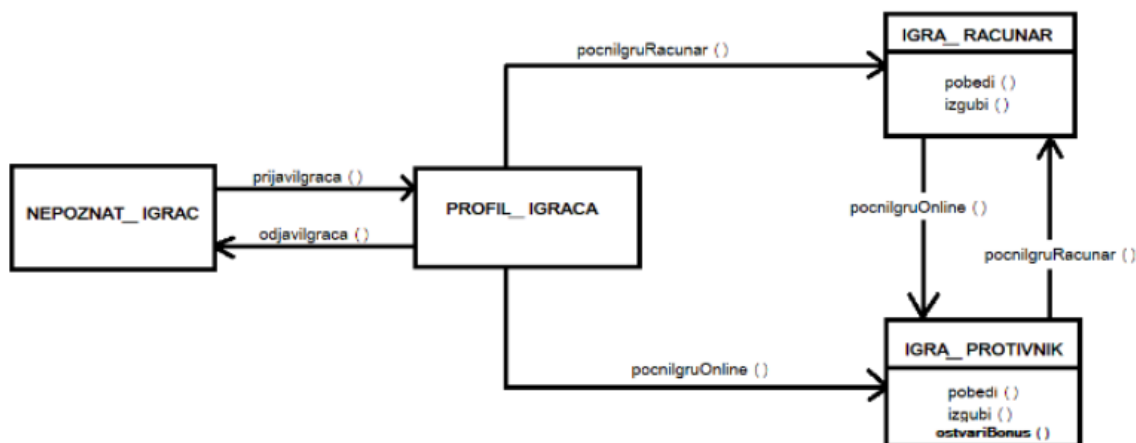
Nakon prijave, korisniku se prikazuje njegov profil i nakon toga može da bira da li želi da vežba igrajući protiv računara ili želi da igra protiv nekog protivnika online.

Svaki put kad igra protiv računara igrač može da dobije ili izgubi 1 poen, dok kad igra online dobija ili gubi po 2 poena.

Broj poena ne može pasti ispod 0 i ne može biti veći od 10.

Svaki put kada krene da igra igru, bilo protiv računara ili online, povećava se broj odigranih partija.

Broj odigranih partija ne može biti manji od 0 i ne može biti veći od 10.



Za klasu Igra implementirati:

- konstruktor bez parametara koji postavlja stanje na NEPOZNAT_IGRAC i igrača na igrača sa 0 poena i 0 partija
- konstruktor sa parametrima **Igra(int, int, StanjeIgre);** Prvi parametar za broj partija, drugi parametar za broj poena i treći parametar za stanje igre
- metodu **bool prijavilgraca();** Ukoliko su ispunjeni uslovi menja stanje igre. Kao povratnu informaciju vraća informaciju o uspešnosti promene stanja igre
- metodu **bool odjavilgraca();** Ukoliko su ispunjeni uslovi menja stanje i zatim resetuje i rezultate tog igrača (poene i broj partija). Kao povratnu informaciju vraća informaciju o uspešnosti promene stanja igre
- metodu **bool pocnilgruRacunar();** Ukoliko su ispunjeni uslovi i ukoliko je moguće povećava broj partija igrača i u tom slučaju menja stanje, u suprotnom vraća false
- metodu **bool pocnilgruOnline();** Ukoliko su ispunjeni uslovi i ukoliko je moguće povećava broj partija igrača i u tom slučaju menja stanje, u suprotnom vraća false
- metodu **bool pobedi();** Povećava igraču broj poena ako je to moguće
- metodu **bool izgubi();** Smanjuje broj poena ako je to moguće, u suprotnom vraća false
- metodu **bool ostvariBonus();** U slučaju da je igra u stanju IGRA_PROTIVNIK i da je ukupni broj poena koje je igrač ostvario do tada veći od 5, dodeljuje bonus poene igraču koji su slučajno broj između 1 i 3 i ispisuje mu poruku "Dobili ste dodatnih <slučajanBrojKojiJeDodat> poena". U slučaju da se predje maksimalan dozvoljeni broj poena nakon dodeljivanja bonusa, broj poena će biti postavljen na maksimalni mogući
- prijateljsku funkciju za ispis **void ispisilgru(const Igra&);** koja ispisuje sve informacije o igri uključujući i informacije o igraču

NAPOMENE:

- Obavezno pročitati SVE napomene
- Polja ne smeju biti public vidljivosti
- Napisati test program i u njemu testirati svaki implementirani konstruktor, metod i funkciju. Ono što nije testirano (u main() funkciji) se ne boduje
- Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše 1 bod
- Sve metode koje imaju bool povratnu vrednost vraćaju informaciju o uspešnosti metode u zavisnosti od uslova u zadatku i povratne vrednosti odgovarajuće metode
- Obezbediti da geteri ne mogu da promene vrednost polja
- Asistent za vreme testa daje informacije vezane isključivo za razumevanje teksta zadatka
- Pomoć: smanjivanje poena je moguće tako što se doda negativan broj poena
- Za generisanje slučajne vrednosti iskoristiti rand() funkciju koja se nalazi u C++ zaglavlju cstdlib
- Test traje 2 sata