



NAPOMENE

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

Zadatak

Napisati klasu **Igrac** koja sadrži polja:

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

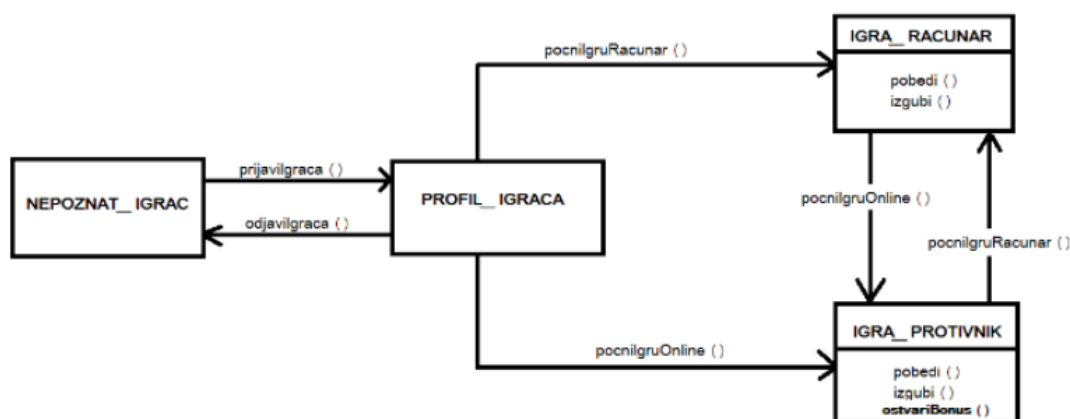
Za klasu **Igrac** implementirati:

- Konstruktor sa podrazumevanim vrednostima za parametre - podrazumevana vrednost za poene je 0, podrazumevana vrednost za broj partija je 0.
- 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 **`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. 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 povratni 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 povratni 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 otvoriBonus()**; U slučaju da je igra u stanju IGRA_PROTIVNIK i da je ukupni broj poena koje je igrač ostvario do tada veći od 5, dodaljuje bonus poene igraču koji su slučajni broj između 0 i 3 i ispisuje mu poruku "Dobili ste dodatnih »slučajniBrojKojiJeDodat« 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 ispisIgru(const Igra&)**; koja ispisuje sve informacije o igri uključujući i informacije o igraču.