

I parcijalni ispit

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Svojim potpisom student izjavljuje da je saglasan sa ovim sistemom bodovanja i da se rješenje postavljenog zadatka nalazi na serveru kako je objašnjeno u uputama za izradu ispita.

Zadatak:

Potrebno je implementirati Java projekat koji treba da sadrži sljedeće klase, sa navedenim atributima i metodama. Pored tih vi možete dodati i druge klase, metode i attribute po želji kako biste ispunili zadatak. Pri tome se pridržavati pravila vezanih za pisanje kvalitetnog koda obrađenih na predmetu, između ostalog svi atributi obavezno moraju biti privatni.

1. Klasu ili interfejs **Osoba** iz koje su izvedene klase ili interfejsi **Ucenik** i **Nastavnik**. Iz **Ucenik** su izvedene klase **Osnovac** i **Srednjoskolac**, a iz **Nastavnik** klase **Ucitelj**, **NastavnikOsnovneSkole** i **Profesor**.
2. Sve nabrojane klase sadrže (nasljeđuju) atribut **imePrezime** (jedan string), standardne settere i gettere za ovaj atribut te konstruktor koji prima string imePrezime.
3. Sve klase se trebaju moći ispisati na ekran koristeći System.out.println, npr. ovako:

```
Ucenik u = new Osnovac("Sara Sarac");  
System.out.println(u);
```
4. Ispis na ekran u slučaju odgovarajućih klasa treba izgledati ovako:
Za Osnovac: "Učenik osnovne škole Sara Sarac (12345), prosjek ocjena: 4.5"
Za Srednjoskolac: "Učenik srednje škole Sara Sarac (12345), prosjek ocjena: 3.7"
Za Ucitelj: "Učitelj Sara Sarac"
Za NastavnikOsnovneSkole: "Nastavnik osnovne škole Sara Sarac"
Za Profesor: "Profesor Sara Sarac"

Prosjek ocjena mora biti zaokružen na jednu decimalu.

5. Broj 12345 u zagradi je broj đачke knjižice. Sve klase izvedene iz Ucenik trebaju imati privatni atribut **brojKnjizice** tipa String, te odgovarajući getter i setter. Default vrijednost broja knjižice je prazan string "".
6. Klasa Ucenik treba imati privatni atribut **ocjene** koji je dat kao običan niz (a ne lista ili nešto slično!) tipa **int** sa fiksno 100 elemenata. Kod rješenja koja budu koristila nešto što nije niz, odgovarajući testovi neće biti priznati.
7. Također treba imati metodu **dodajOcjenu** tipa void koja dodaje jednu ocjenu u niz ocjena. Ako ocjena nije na intervalu [1,5] treba baciti izuzetak tipa **IllegalnaOcjena** sa tekstom poruke "Illegalna

ocjena 6" (gdje se umjesto broja 6 treba nalaziti ocjena koja je poslana metodi). U slučaju da se prekorači kapacitet niza treba baciti izuzetak tipa **IllegalArgumentException** sa tekstom poruke "Dosegnut maksimalan broj ocjena".

8. Konačno ove klase trebaju imati i metodu **prosjeck** koja vraća vrijednost tipa **double** koja predstavlja srednju vrijednost dodatih ocjena, ili broj 0 ako nije dodata niti jedna ocjena.

9. U klasi **Srednjoskolac** treba se nalaziti drugi niz koji predstavlja ocjene u srednjoj školi, kao i metode **dodajOcjenuSrednja** koja dodaje ocjenu u taj niz, i **prosjeckSrednja** koja vraća prosjek ocjena iz srednje škole. Metoda **prosjeck** klase **Srednjoskolac** treba vraćati prosjek ocjena za *zajedno za osnovnu i srednju školu* (posmatrajući ta dva niza kao jedan).

10. Program treba sadržavati i klasu **Skola** koja kao privatni atribut ima neku kolekciju osoba. Treba posjedovati metodu **dodajOsobu** koja stavlja novu osobu u tu kolekciju. Ako se škola proba ispisati na ekran metodom `System.out.println`, treba ispisati sve osobe koje su dodane u školu, svaku u zasebnom redu, onim redom kojim su dodavane.

11. Klasa **Skola** treba imati metodu **ucenici** koja vraća skup (**Set**) učenika koji sadrži sve učenike u školi, sortirane po prosječnoj ocjeni od veće ka manjoj. Skup treba biti tipa **Ucenik** a ne tipa **Osoba**!

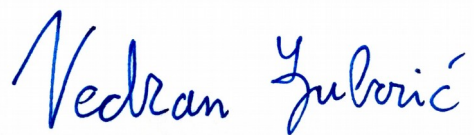
12. Dalje, treba imati metodu **filtriraj** koja prima *lambda funkciju* koja prima osobu a vraća boolean vrijednost. Metoda **filtriraj** vraća listu (**List**) svih osoba za koje je lambda funkcija vratila **true**.

13. Treba imati i metodu **topOsnovac** koja vraća listu svih Osnovaca koji imaju prosjek veći ili jednak 4. Metoda treba vraćati listu **Osnovaca**, a ne listu **Osoba**! Treba imati *dvije linije koda*, od čega je jedna poziv metode **filtriraj**. Za metode implementirane na neki drugi način neće biti priznati odgovarajući testovi.

14. Klase **Srednjoskolac** i **Ucitelj** (ali ne i ostale klase!) trebaju imati metodu **rodjendan** koja prima **datumRodjenja** (`LocalDate`) a vraća string "Sretan rođendan!" ako na datum izvršavanja programa pada rođendan osobe, u suprotnom vraća prazan string "".

15. Pored toga klasa **Skola** treba imati metodu **slavljenici** koja vraća **List<Slavljenik>** koja sadrži samo osobe koje ispunjavaju prethodni uslov.

Sarajevo, 19. 9. 2019

A handwritten signature in blue ink that reads "Vedran Zuborić". The signature is written in a cursive, flowing style.