

[OOP] K3

Napisati apstraktnu klasu **Test** koja ima polje ***naziv*** (tipa String) i ***osvojeniPoeni*** (tipa int). U klasi implementirati:

- konstruktor bez parametara, postavlja naziv na prazan string, a osvojenePoene na 0
- konstruktor sa parametrima
- get i set metodu za polje osvojeniPoeni
- apstraktnu metodu *boolean položio()*

Napisati klasu **Kolokvijum** koja nasleđuje klasu Test. U klasi implementirati:

- konstruktor bez parametara
- konstruktor sa parametrima
- implementirati apstraktnu metodu *boolean položio()* tako da ako je osvojeno preko 12 poena vraća true, dok u suprotnom vraća false
- redefinisati toString metodu tako da ispisuje sve informacije o kolokvijumu u sledećem obliku:
----- KOLOKVIJUM -----
naziv: <naziv>
Osvojeno poena: <osvojeniPoeni>

Napisati interfejs **IPredmet** koji ima metode

- *boolean dodajTest(Test t);*
- *boolean položio();*
- *int izracunajPoene();*

Napisati klasu **Predmet** koja implementira interfejs IPredmet i ima polja naziv (tipa String), premeti (tipa ArrayList<Test>). Za klasu predmet implementirati:

- konstruktor bez parametara
- konstruktor sa parametrima
- get metodu za polje naziv
- implementirati metodu *boolean položio()* tako da u slučaju da su svi testovi u listi položeni i da je na svakom od testova osvojeno preko 55 poena predmet se smatra položenim i metoda vraća true, dok u svim ostalim slučajevima metoda vraća false.
- implementirati metodu *int izracunajPoene()* tako da kao povratnu vrednost vraća broj ukupno osvojenih poena na svim testovima.
- implementirati metodu *boolean dodajTest(Test t)* tako da dodaje test u listu testova pod uslovom da u listi već ne postoji test sa zadatim nazivom. U slučaju da u listi postoji takav test, metoda traži od korisnika da sa konzole unese novo ime za test. U slučaju da novo ime ne postoji u listi, dodaje test, dok u suprotnom metoda vraća false.
- Redefinisati toString metodu tako da ispisuje sve informacije o predmetu u sledećem obliku:

***** Predmet *****

Naziv: <NazivPredmeta>

Testovi:

----- KOLOKVIJUM -----

naziv: <nazivTesta>

Osvojeno poena: <osvojeniPoeni>

----- KOLOKVIJUM -----

naziv: <nazivTesta>

Osvojeno poena: <osvojeniPoeni>

U slučaju da je lista testova prazna ispisati odgovarajuću poruku.

Napisati klasu **Student** koja ima polja prezime (tipa String), indeks (tipa String, podrazumevamo da će biti testirano u formatu smerBrojIndeksa/ godinaUpisa, gde je smer uvek dva slova, a godina je četvorocifrena), predmeti (tipa HashMap<String, Predmet>). Za klasu implementirati:

- konstruktor bez parametara
- konstruktor sa parametrima
- metodu *boolean dodajPredmet(Predmet p)*, koja dodaje predmet u mapu, gde je ključ naziv predmeta, a vrednost predmet, pod uslovom da u mapi nema više od 10 predmeta i da se ne nalazi predmet sa istim nazivom
- metodu *boolean položioGodinu()* koja u slučaju da su svi predmeti u mapi položeni prazni mapu predmeta i kao povratnu vrednost vraća true, dok u suprotnom vraća false
- metodu *String formirajEMailAdresu()* koja kreira uns e-mail adresu u formatu: prva dva slova indeksa (podrazumeva se da su indeksi u formatu smerBrojIndeksa/ godinaUpisa) iza čega sledi tačka, prezime iza čega sledi tačka, broj indeksa - godina upisa @uns.ac.rs.

Na primer:

ra.Prezimenic.20-2015@uns.ac.rs

- metodu *int trajanjeStudija()*, koja računa broj godina koliko je student studirao, tako što od tekuće godine (radi jednostavnosti podrazumevati da je to 2022) oduzme broj godine upisa iz indeksa. Podrazumeva se da je godina upisa četvorocifreni broj koji se nalazi na kraju broja indeksa. U slučaju da se prilikom parsiranja vrednosti polja indeks desi izuzetak (NumberFormatException), ispisati poruku "Desio se izuzetak prilikom pokušaja parsiranja!".
- Preklopiti metodu toString tako da ispisuje sve informacije o studentu, uključujući i broj godina studiranja.

NAPOMENE:

1. Testirati svaki implementirani konstruktor i metod. Sve što nije testirano tretira se kao da nije ni implementirano i donosi 0 bodova.
2. Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše 1 bod.

3. Metode koje vraćaju bool kroz povratnu vrednost javljaju informaciju o tome da li je operacija uspešno izvršena.
4. Polja ne smeju biti public vidljivosti.
5. U slučaju da mislite da vam je za rešavanje zadatka potrebna još neka metoda, slobodno je implementirajte. Boduje se samo ono što je eksplicitno traženo u tekstu zadatka.