

Objektno-orijentisano programiranje – prvi kolokvijum

Zadatak: Aktivnosti učenika – termin 2

Napraviti Java konzolnu aplikaciju za evidenciju aktivnosti učenika srednje stručne škole u toku školske godine. Jedan učenik može imati proizvoljan broj učešća u različitim aktivnostima koje su unapred definisane. Jedna beleška o učešću u aktivnosti sadrži trajanje, tekstualni opis i broj osvojenih poena.

Prvi deo – Implementacija UML dijagrama klasa

1. **(6p)** Implementirati sve elemente na priloženom dijagramu uz sledeće napomene:

- za sva polja dodati get i set metode sa proverama koje se traže u zadatku,
- metode iz klase *Object*, *equals* i *toString* nadjačati prema proceni na mestima gde je to neophodno za realizaciju zahtevanih funkcionalnosti,
- proizvoljno se mogu dodati konstruktori, metode kojih nema na dijagramu i implementacije ugrađenih interfejsa (na primer *java.lang.Comparable*).

2. Klasa *Klasa* predstavlja grupu učenika sa zajedničkim razrednim starešinom. Svaka *Klasa* ima *Smer* (*PEKAR*, *LEKAR*, *APOTEKAR*), *razred*, *odeljenje* i *razrednog starešinu*. Dve klase su iste ako imaju isti smer, razred i odeljenje.

3. Klasa *Ucenik* predstavlja jedinstvenog učenika škole, koji ima *jmbg*, *ime*, *prezime* i *Klasu* kojoj pripada. Dva učenika su ista ako imaju isti jmbg.

4. Apstraktna klasa *Aktivnost* predstavlja proizvoljnu aktivnost kojom se učenici mogu baviti za ocenu. Svaka aktivnost ima *naziv*, *smer*, *minimalan* i *maksimalan broj poena* predviđen za tu aktivnost. Dve konkretne aktivnosti su iste ako imaju isti naziv i smer. Apstraktna metoda *izracunajOcenu* služi da se izračuna ocena na osnovu zadanog broja poena (svaka konkretna aktivnost ima svoj način obračuna ocene).

5. **(1p)** Klasa *RadUPekari* predstavlja vrstu *Aktivnosti* koja je predviđena za smer PEKAR. Minimum poena koji se može osvojiti za ovu aktivnost je 5, maksimum je 10, a ocena se računa kao ceo deo od (broj poena * 0.75 – 1.75).

6. **(1p)** Klasa *Vizita* predstavlja vrstu *Aktivnosti* koja je predviđena za smer LEKAR. Minimum poena predviđen za ovu aktivnost je 1, maksimum je 5, a ocena se računa kao ceo deo od (broj poena * 0.75 + 1.25).

7. Klasa *Beleska* predstavlja zapis o učešću nekog učenika u nekoj aktivnosti. Učešće u aktivnosti ima svoje *trajanje* (u minutima), *tekstualni opis* i *broj poena* koje je učenik osvojio.

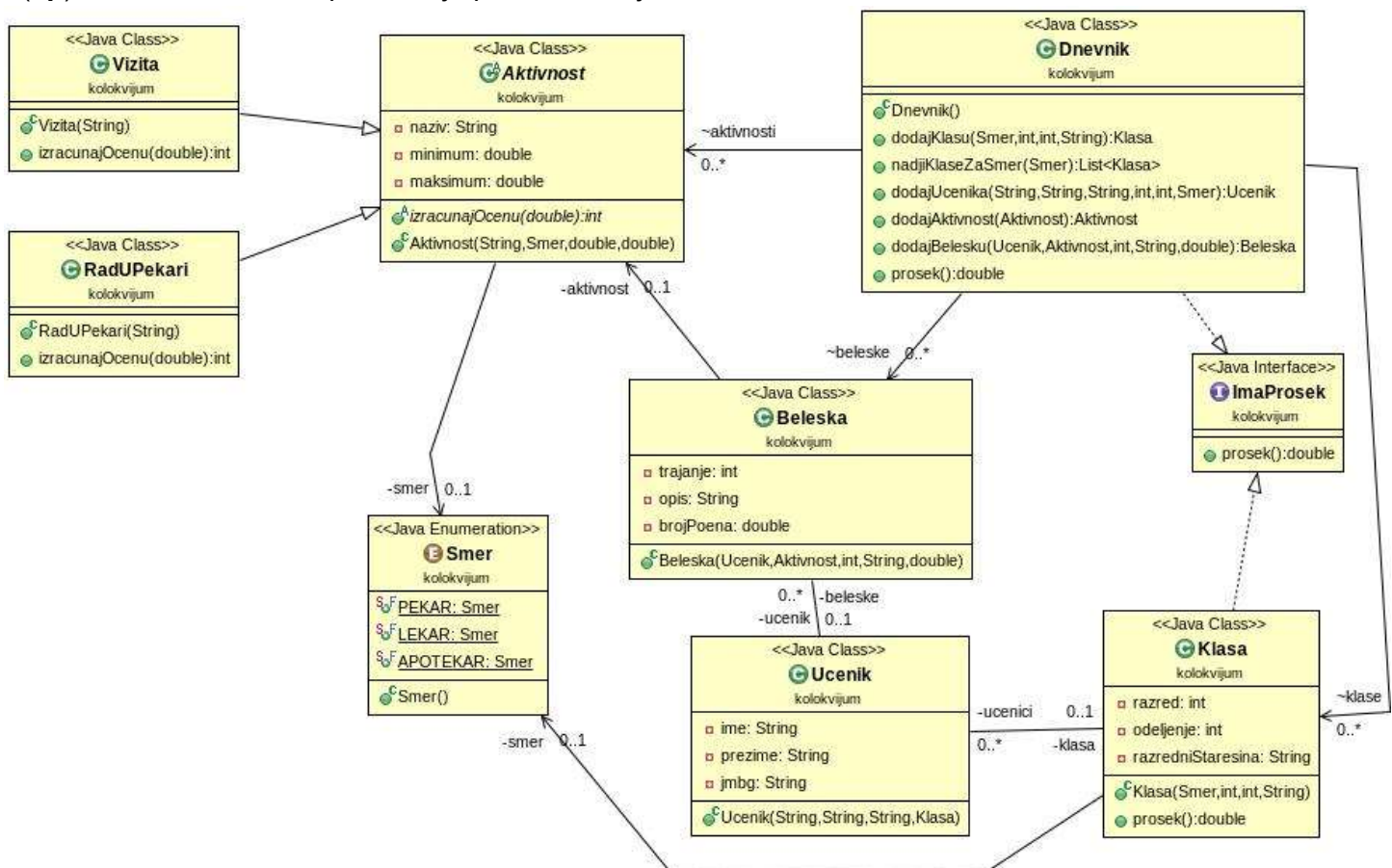
8. Klasa *Dnevnik* služi kao centralna evidencija svih *klasa*, *aktivnosti* i *beležaka*.

- **(1p)** dodajKlasu(smer, razred, odeljenje, razredniStaresina) – kreira novi objekat tipa Klasa sa zadatim atributima i dodaje ga u svoju kolekciju ukoliko on u njoj trenutno ne postoji (po definiciji jednakosti Klasa). Ako je dodat, vraća novokreirani objekat, a ako nije, vraća objekat iz svoje kolekcije.
- **(1p)** nadjiKlaseZaSmer(smer) – vraća sve Klase iz svoje kolekcije koje odgovaraju smeru zatom u argumentu.

- **(2p)** dodajUcenika(ime, prezime, jmbg, razred, odeljenje, smer) – upisuje novog učenika u klasu sa datim razredom, smerom i odeljenjem, vodeći računa da isti učenik već nije upisan, te da klasa postoji u dnevniku. Ukoliko klasa ne postoji, prvo treba dodati novu klasu. Vraća upisanog učenika ako su ispunjeni uslovi za upis ili null ako nisu.
- **(1p)** dodajAktivnost(aktivnost) – dodaje aktivnost u kolekciju aktivnosti ukoliko ista već nije u njoj. Ako je dodata, vraća istu, a ako aktivnost već postoji, vraća postojeću iz kolekcije.
- **(2p)** dodajBelesku(učenik, aktivnost, trajanje, opis, brojPoena) – kreira i dodaje novu belešku o učešću učenika u aktivnosti. Dodaje se beleška učeniku i u dnevnik. Učenik može učestvovati samo u aktivnosti predviđenoj za njegov smer. Trajanje aktivnosti ne može biti kraće od 5 minuta. Ako su ispunjeni uslovi, vraća se kreirana beleška. Ako nije ispunjen neki od uslova, ispisuje se odgovarajuća poruka o grešci na standardni izlaz i vraća se null.

9. Interfejs *ImaProsek* obezbeđuje metodu za računanje proseka.

- **(1p)** Prosek klase *Dnevnik* predstavlja prosečno trajanje svih beležaka o aktivnostima.
- **(1p)** Prosek klase *Klasa* predstavlja prosečan broj beleški za učenika.



Drugi deo – proširenje modela i dodatne funkcionalnosti

10. **(2p)** Dodati klasu za praksu u apoteci, koja predstavlja tip aktivnosti predviđen za smer APOTEKAR. Minimum poena za ovu aktivnost je 0, maksimum 100, a ocena se definiše tako što se za sve poene do 52 daje ocena 1, a onda na svakih sledećih 12 poena po jedna ocena više (npr. u intervalu (52, 64] je 2, u intervalu (88, 100] je 5).

11. **(2p)** Proširiti klasu *Beleska* poljem *datum*, koje odgovara datumu obavljene aktivnosti. Za datum se može koristiti ugrađena klasa *LocalDate* ili klasa koju ćete sami napraviti. Dodati odgovarajuće getere, setere, proširiti konstruktor i ostale metode po potrebi.

12. (6p) U dnevnik dodati obračun prosečne ocene svih aktivnosti učenika, ako učenik nema aktivnosti vratiti 0. Dodati i ispis aktivnosti učenika u fajl, ispisuju se prvo ime i prezime učenika, a zatim aktivnosti sortirane rastuće po datumu, svaka aktivnost u novom redu u sledećem obliku:

Dan.Mesec.Godina – Naziv aktivnosti– BrojPoena (Ocena)“.

Na kraju ispisati ukupan broj poena i prosečnu ocenu, a ako učenik nije imao nijednu aktivnost, ispisati poruku „Nema zabelezenih aktivnosti“.

13. (3p) Dodati klasu *Test* koja sadrži *main* metodu. U njoj kreirati jedan dnevnik i u njega dodati po 3 klase i 3 aktivnosti za svaki smer, po 3 učenika za svaku klasu, po 3 beleške za jednog učenika. Testirati rad svih metoda u dnevniku i napraviti ispis aktivnosti .