

## Objektno-orijentisano programiranje - prvi kolokvijum

Zadatak: Aktivnost studenata - termin 2

Napisati Java konzolnu aplikaciju za evidenciju aktivnosti studenata na fakultetu, aktivnosti su upisi i obnove godina, da bi se realizovale aktivnosti prethodno moraju biti zadovoljeni neki uslovi.

### Prvi deo - implementacija UML dijagrama klasa

1. (5p) implementirati sve elemente na priloženom dijagramu uz sledeće napomene:

- za sva polja dodati set/get 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 i metode kojih nema na dijagramu

2. Za aktivnost studenata se pamti datum kada je izvršena u vidu tri cela broja koja označavaju godinu, mesec i dan. Postoje dve vrste aktivnosti studenta, to su upis i obnova godine za koje se čuva godina koja se upisuje ili obnavlja

- (0.5p) metoda *getGodinaStudija()* vraća godinu upisa ili obnove,

- (3p) radi realizacije većine funkcionalnosti biće potrebno da aktivnosti studenta budu sortirane po datumu što treba realizovati implementacijom interfeja *java.lang.Comparable*.

3. Interfejs *Uslovljeno* predstavlja aktivnosti koje podrazumevaju da mora biti ispunjen neki uslov da bi se izvršile

- (3p) metoda *proveriUslov(Student)* u klasi *UpisGodina* proverava da li prosleđeni student zadovoljava uslov upisa, a to je broj osvojenih espb na osnovu položenih predmeta studanta, za upis u drugu godinu potrebno je 30, u treću 90, a u četvrtu 150 espb, za prvu godinu ne postoji uslov. Takođe da bi student upisao sledeću godinu, mora imati upisanu prethodnu (da bi se upisao u drugu godinu mora imati aktivnost u kojoj se upisao u prvu, odnosno poslednja aktivnost mora biti upis ili obnova prve godine)

- (1p) metoda *proveriUslov(Student)* u klasi *ObnovaGodine* proverava samo da li je student prethodno upisan u godinu koju obnavlja (ako mu je poslednja aktivnost upis ili obnova iste te godine).

4. Za studenta se, pored osnovnih podataka iz indeksa (broj, godina i studijski program) pamti kolekcija svih položenih predmeta i kolekcija aktivnosti.

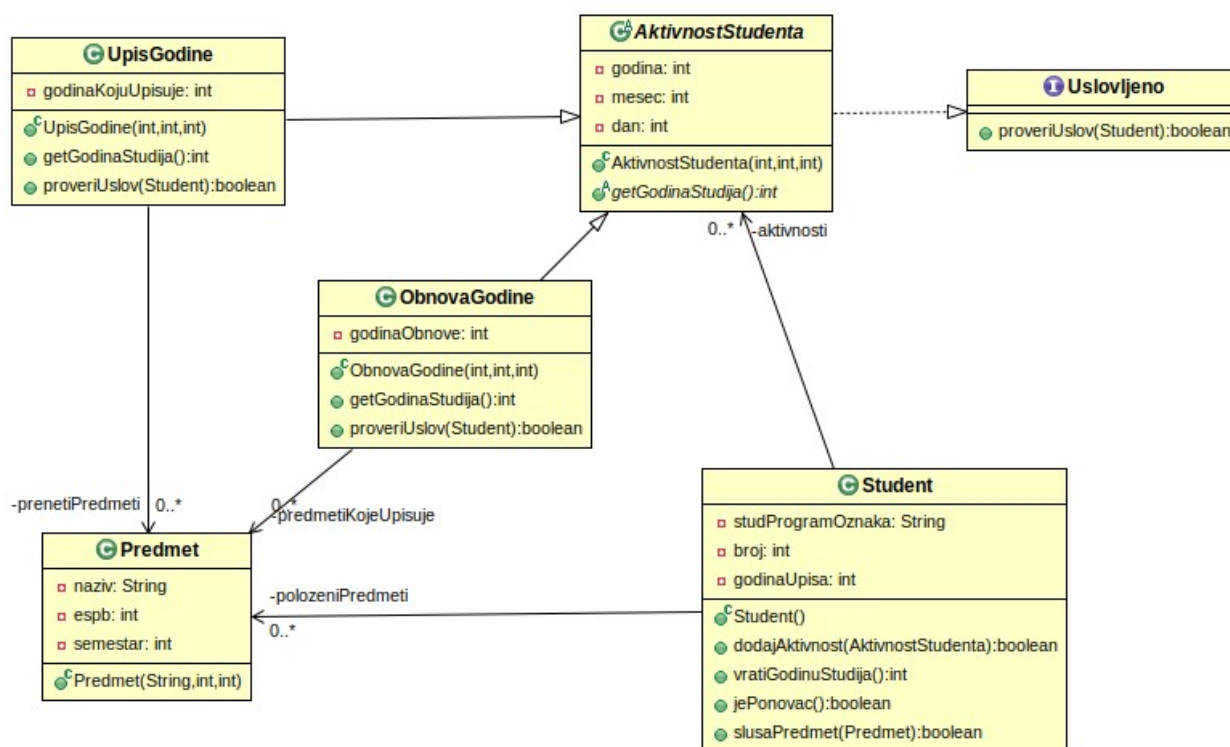
- (1p) metoda *vratiGodinuStudija()* u klasi *Student* treba da vrati poslednju upisanu ili obnovljenu godinu posmatrajući datume aktivnosti studenta,

- (1p) metoda *jePonovac()* treba da vrati *true* ako je poslednja aktivnost studenta obnova godine,

- (1p) metoda *dodajAktivnost(AktivnostStudenta)* prvo proverava da li je aktivnost moguća i ako jeste, dodaje je u kolekciju aktivnosti studenta i vraća *true*, ako nije moguća vraća *false*.

- (3p) metoda *slusaPredmet(Predmet)* vraća *true* ako student trenutno sluša prosleđeni predmet, a to može biti ako se predmet sluša u semestru koji pripada godini koju je student poslednju upisao, ako

ga je student preneo ili ga upisao u obnovljenu godinu, posmatra se poslednja aktivnost, predmeti se razlikuju po nazivu.



## Drugi deo - proširenje modela i dodatne funkcionalnosti

5. (6p) Dodati novu vrstu aktivnosti, a to je promena studijskog programa, za koju je uslov da student ima osvojenih 48 espb. Kada se dodaje ova vrsta aktivnosti studentu se menja oznaka studijskog programa, godina upisa i broj, i potrebno je osmisliti implementaciju kojom se ovo realizuje (hint: novi podaci se mogu upamtiti kao polja u klasi nove vrste aktivnosti). Takođe, prilikom promene studijskog programa student mora i da upiše sledeću godinu što znači da se prilikom promene studijskog programa dodaje još jedna aktivnost za studenta, za ovaj upis nema prenetih predmeta, a godina koja se upisuje se dobija na osnovu prethodno upisane godine iz liste aktivnosti.

6. (3p) Dodati funkcionalnost ispisa svih aktivnosti studenata na standardni izlaz i u fajl, aktivnosti treba da budu sortirane hronološki po datumu od najranije do najkasnije, ispisuje se datum, vrsta aktivnosti, godina koja se upisuje ili obnavlja ili studijski program na koji se prelazi i novi indeks.

7. (3p) Napraviti klasu sa main metodom u kojoj se kreira jedan student i dodaju mu se različite aktivnosti, prvo upis u prvu godinu, zatim upis u drugu, obnova druge i promena stud. programa. Kreirati određeni broj predmeta koje je student položio i koji se prenose u naredne godine. Testirati metode koje vraćaju godinu studija i informaciju da li je ponovac i da li sluša određeni predmet. Testirati ispis aktivnosti studenta na standardni izlaz i u fajl.