

Objektno-orijentisano programiranje – Prvi kolokvijum

Zadatak : Selekcioni proces konkursa – termin 1 - grupa 1

Napraviti Java konzolnu aplikaciju za sprovođenje intervjua u selekcionom procesu kandidata na konkursu. Na konkurs se prijavljuju kandidati i prolaze kroz različite krugove selekcionog procesa, pri čemu ih u svakom krugu do kog prođu očekuje jedan intervju.

Prvi deo – Implementacija UML dijagrama klasa

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

- za sva polja dodati set i 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, metode kojih nema na dijagramu i implementacije ugrađenih interfejsa (na primer `java.lang.Comparable`)

2. Apstraktna klasa *Intervju* predstavlja intervju koji je deo selekcionog procesa na određenom konkursu. U svakom krugu će se sprovođiti intervju sa kandidatima. Dva intervjua su ista ako su u istom krugu na istom konkursu. Apstraktna metoda *izracunajBrojPoena(String odgovori)* služi da se izračuna broj poena za kandidata na osnovu njegovih odgovora (svaki konkretan intervju ima svoj način obračunavanja poena).

- (2.5p) *odaberiKandidateZaProlaz(int ogranicenje)* treba da vrati listu kandidata koji su odabrani da idu u naredni krug. Treba uzeti u obzir da li su kandidati odradili intervju i da li su osvojili minimalan broj poena za prolaz. Ukoliko je više takvih kandidata od zadatog ograničenja, treba odabrati one sa najvećim brojem poena, a ako više njih ima isti broj poena prednost se daje onima koji su ranije odradili intervju. Takođe, treba ažurirati status kandidata u zavisnosti od toga da li su prošli ili nisu prošli intervju.

3. Klasa *TehnickiIntervju* predstavlja vrstu intervjua sa određenim načinom bodovanja zadataka. Ova klasa sadrži poene predviđene za pitanja, gde su bodovi navedeni redom za svako pitanje.

- (0.5p) *definisiPoene(List<Integer> poeni)* treba da postavi prosleđenu listu za bodovanje uz proveru da li su za sve zadatke zadati poeni, ako nisu ispisati poruku.
- (1.5p) *izracunajBrojPoena(String odgovori)* vraća izračunati broj poena. Odgovori se prosleđuju na sledeći način "1-TACNO,2-TACNO,3-NETACNO" (redniBrojPitanja-da li je tačan odgovor ili ne odvojeni zarezom) što bi značilo da će poeni biti dobijeni za 1. i 2. pitanje na osnovu liste *poeniZaPitanja*. Očekivati da će broj odgovora biti jednak broju pitanja.

4. Klasa *IntervjuisanjeKandidata* namenjena je beleženju podataka o intervjuu za određenog kandidata koji ima učešće u njemu. Čuvaju se poeni, status koji može biti `NA_CEKANJU`, `PROSAO`, `NIJE_PROSAO` kao ishod na tom intervjuu za kandidata i datum rađenja intervjua ukoliko se kandidat odlučio da ga uradi. Na početku, svako intervjuisanje kandidata ima status `NA_CEKANJU`.

5. Klasa *Kandidat* predstavlja osobu koja se prijavljuje za neki konkurs. Za svakog kandidata se čuva email, ime, prezime, sposobnosti i rezultati koje je postigao na određenim intervjuima.

- (1.5p) *prijaviSeNaKonkurs(Konkurs konkurs)* treba da doda kandidata u listu kandidata za prosleđeni konkurs, ukoliko važe sledeći uslovi:
  - za konkurs su prijave otvorene
  - kandidat ima dovoljan broj sposobnosti koje se zahtevaju za taj konkurs

Ukoliko važe uslovi vratiti true, u suprotnom false.

- (3p) *odradiIntervju(Intervju intervju, LocalDate datum, String odgovori)* treba da zabeleži rezultate koje je kandidat postigao na prosleđeni datum za prosleđeni intervju.

Kandidat ne može da odradi intervju ukoliko nije planiran za taj intervju. Ukoliko je već odradio taj intervju ranije, ne treba mu omogućiti da ponovo radi. Ukoliko kandidat pokuša da uradi intervju nakon roka izrade intervju ne treba mu dozvoliti da odradi intervju. Za sve tri situacije ispisati odgovarajuću poruku, u suprotnom postaviti datum za intervjuisanje kandidata i postaviti poene nakon izračunavanja poena na osnovu njegovih odgovora.

6. Klasa *Konkurs* sadrži poziciju za koju se konkuriše, listu željenih sposobnosti kandidata koji se prijavljuju i koliki je minimalan broj sposobnosti koje moraju da se poklope da bi prihvatili kandidata. Na početku su prijave otvorene. Pamti se i trenutni broj kruga do koga se stiglo u procesu selekcije, na početku je to krug sa brojem 1.

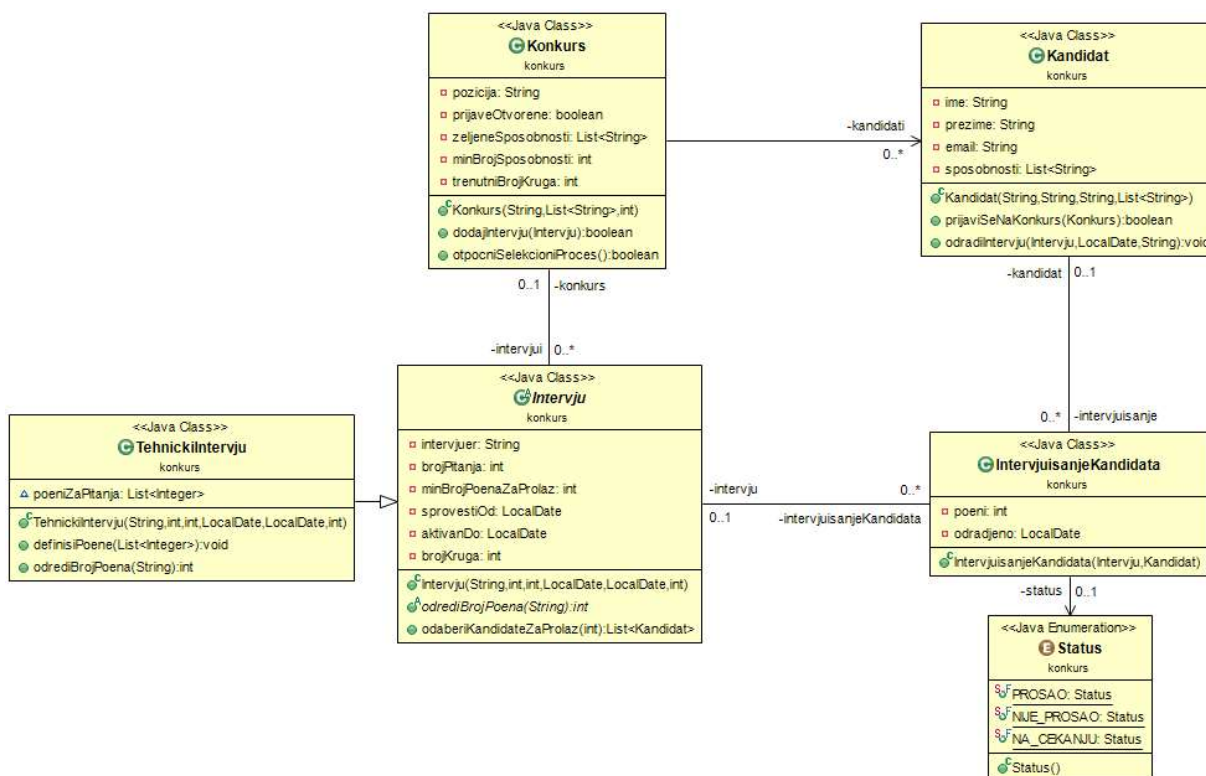
Dve instance klase *Konkurs* su iste ako su im iste pozicije.

- (1p) *dodajIntervju(Intervju intervju)* dodaje intervju u listu intervju, i intervjuu postavlja konkurs, ukoliko važe sledeći uslovi:

- intervju nije već dodan
- intervju koji se dodaje ima broj kruga za jedan veći od prethodnog intervju

Ukoliko je dodavanje uspešno vratiti true, u suprotnom false.

8. (2p) *otpocniSelekциониProces()* je metoda kojom kreće prvi selekциони krug. Ukoliko ne postoji nijedan intervju, treba ispisati poruku i vratiti false. Prilikom otpočinjanja, prijave se zatvaraju, i kandidatima konkursa se u listu intervjuisanja dodaje mogućnost učešća u intervjuu koji je u prvom krugu.



#### Drugi deo – proširenje modela i dodatne funkcionalnosti

9. (2.5p) Dodati još jedan tip intervjua - HrIntervju. Ovaj tip intervjua očekuje određene osobine od kandidata poput Upornosti, Timskog duha i slično, tako da treba omogućiti definisanje osobina koje se očekuju od kandidata na tom intervjuu. Odgovori koje kandidat daje za ovaj intervju se prosleđuju na sledeći način "Ambiciozan,Vredan,Uporan" (osobine odvojene zarezom). Za svaku osobinu koju je kandidat pokazao u svom odgovoru, a da je očekivana na tom intervjuu dodati po jedan poen.

10. (5p) U klasi *Konkurs* dodati metodu za slanje kandidata u naredni krug, ukoliko taj krug postoji. Treba omogućiti da se zada ograničenje u vidu broja kandidata koji mogu da prođu u naredni krug, na osnovu preformansi u trenutnom krugu.

11. (2.5p) U klasi *Konkurs* dodati ispis svih intervjua i rezultata kandidata koji su u njemu učestvovali. Ispisati intervjue redom, po broju kruga, i za svaki intervju rezultate kandidata sortirane po broju poena i datumu rada intervjua.

12. (3p) Dodati klasu *Test* koja sadrži main metodu. U njoj kreirati jedan konkurs i na njega prijaviti četiri kandidata uspešno i jednog bezuspešno. Dodati dva intervjua za konkurs, prvo tehnički, a onda i hr intervju. Zatim otpočeti selekcion proces. Svaki kandidat onda radi intervju i poslati određen broj najboljih u naredni krug. Ponoviti isto i za drugi intervju. Na kraju odraditi ispis u fajl, ispisati sve intervjue i intervjuisanja kandidata u okviru njih u proizvoljnom formatu.