# Praktični ispit iz Programiranja baza podataka – Primer ispita

Zadaci se rade 180 minuta. Maksimalan broj poena je 50. Broj poena po zadacima je:

| Redni broj zadatka | 1 | 2 | 3 | **Ukupno** |
|---|---|---|---|---|
| Najviše poena | 50/3 | 50/3 | 50/3 | **50** |

## Pre početka rada

Za zadatke je potrebno koristiti tabelu `DA.PRAKSA` koja je kreirana u bazi podataka `STUD2020`. Nakon otpakivanja arhive `zadaci.zip`, u direktorijumu `Desktop/bp_PrezimeIme_alasNalog_grupa/resursi` možete pronaći datoteku `pripremaBaze.sql` koja sadrži SQL kod za rekreiranje ove tabele.

Tabela `DA.PRAKSA` sadrži informacije o kandidatima za studentsku praksu u nekoj kompaniji. Pored indeksa kandidata za praksu, tabela sadrži informacije o poenima ostvarenim tokom studija i poenima na testu za praksu.

## Informacije za čuvanje rešenja

### C/SQL zadatak

U direktorijumu `zadaci/` nalazi se poddirektorijum `c/` u kojem se nalazi datoteka `1.sqc`. U ovoj datoteci je neophodno da sačuvate rešenje C/SQL zadatka. Nakon pozicioniranja u ovaj direktorijum u terminalu, pozovite naredbu:

```
chmod +x prevodjenje
```

Da biste izvršili prevođenje, pozovite naredbu:

```
./prevodjenje 1 stud2020 student abcdef
```

### JDBC i Hibernate

1. Otvoriti IBM Data Studio. Savetuje se da odaberete podrazumevanu vrednost za radni direktorijum (*workspace*) prilikom otvaranja alata (tj. lokaciju **/home/ispit/IBM/rationalsdp/workspace**).
2. Iz glavnog menija odabrati *Window* → *Perspective* → *Open Perspective* → *Other* → *Java* → *OK*.
3. Uvesti Java projekat:
   (a) Klik na *File* → *Import...*.
   (b) Odaberite *General* → *Existing Projects into Workspace* → *Next*.
   (c) Kliknite na dugme *Browse* u liniji *Select root directory*:
      i. Iz dijaloga za pretragu datoteka odabrati:
      *Desktop* → **bp_PrezimeIme_alasNalog_grupa/zadaci/java**.
      ii. Odabrati dugme *OK*.
   (d) Nakon toga bi u tekstualnom polju ispod *Projects* trebalo da se prikaže projekat koji je automatski obeležen.
   (e) Osigurajte se da **NIJE** označena opcija *Copy projects into Workspace* u opcijama ispod tekstualnog polja.
   (f) Kliknite na dugme *Finish*.
   (g) Ovim ste uspešno dodali Java projekat sa podrškom za razvoj JDBC i Hibernate aplikacija. Sva rešenja koja budete pisali u ovom projektu će se čuvati u direktorijumu na Desktop-u, pa nije potrebno da posebno izvozite projekat na kraju rada.
4. Sada je potrebno da preimenujete pakete u Java projektu da odgovaraju Vašim podacima:
   (a) Proširite pogled Java projekta, pa zatim proširite pogled direktorijuma *src*.
   (b) Za svaki Java paket koji se ovde nalazi, uraditi sledeće:
      i. Desni klik na naziv paketa → *Refactor* → *Rename...*.
      ii. Izmeniti Vaše informacije: ime, prezime, alas nalog i grupa u kojoj polažete. Ostale informacije (reči hibernate, jdbc, bp, podvlake, itd.) ostati netaknute.
      iii. Osigurati se da je označena opcija *Update references*.
      iv. Kliknuti dugme *OK*.
      v. Klinuti dugme *Continue*.

# Zadaci*

*Ne podrazumevati da će redosled zadataka i raspored gradiva po zadacima odgovarati situaciji na ispitu.

## 1. zadatak

Napisati C/SQL aplikaciju koja prvo pita korisnika da li želi da obriše sve informacije iz tabele `DA.PRAKSA`. Ukoliko je odgovor potvrdan, **dinamičkom** SQL naredbom izvršiti brisanje podataka. Za ostatak aplikacije koristiti **statičke** SQL naredbe. Aplikacija zatim pronalazi sve studente kojima je ostalo manje od 60 ESPB do kraja studija, a koji se ne nalaze u tabeli `DA.PRAKSA`. Urediti slogove po broju preostalih poena rastuće. Za svakog studenta, aplikacija ispisuje naredne informacije na standardni izlaz: (1) indeks, (2) ime, (3) prezime, (4) broj preostalih ESPB i (5) prosek studenta pomnožen brojem 10. Aplikacija za tekućeg studenta pita korisnika da li želi da ga unese kao kandidata za praksu u tabelu `DA.PRAKSA`. Ukoliko korisnik potvrdi, onda aplikacija unosi kandidata u tabelu `DA.PRAKSA`, pri čemu se za poene za praksu postavlja vrednost -1 (tj. nijedan kandidat nije polagao test).

*Napomene: Obrada jednog studenta predstavlja jednu transakciju. Proveravati greške koje se javljaju prilikom izvršavanja aplikacije u višekorisničkom okruženju. Postaviti istek vremena za zahtevanje katanaca na 5 sekundi. Obezbediti da nijedna druga aplikacija ne može da pristupa redovima koje ova aplikacija obrađuje. Obavezno je obrađivanje grešaka.*

Primer interakcije korisnika sa programom je dat u nastavku. Tamo gde su navedene `...` znači da je izlaz iz programa skraćen radi povećanja čitljivosti.

```
Da li zelite da obrisete podatka o kandidatima za polaganje teksta? [d/n] d
    Pripremam tabelu za rad...
    Svi podaci su uspesno obrisani!
=====================================================
1. kandidat: Perica Divnic (20090073) - preostalo ESPB: 36
    Da li zelite da prijavite kandidata za polaganje testa za praksu? [d/n] d
    Uspesno ste prijavili kandidata!
=====================================================
2. kandidat: Darko Janjic (20100060) - preostalo ESPB: 36
    Da li zelite da prijavite kandidata za polaganje testa za praksu? [d/n] d
    Uspesno ste prijavili kandidata!
=====================================================
3. kandidat: Marko Lazic (20100080) - preostalo ESPB: 36
    Da li zelite da prijavite kandidata za polaganje testa za praksu? [d/n] n
=====================================================
...
```

## 2. zadatak

**Pre početka rada:** Uvesti Java projekat ukoliko to već nije učinjeno (uputsvo: klikni ovde). Preimenovati paket sa nazivom `jdbc_bp_PrezimeIme_alasNalog_grupa` (uputstvo: klikni ovde).

U klasi **Main.java** implementirati naredne metode. Nije dozvoljeno menjati potpise metoda; jedino je moguće *dodavati* izuzetke koje oni ispaljuju. Nije dozvoljeno implementirati opisane operacije van tela metoda (ali je moguće koristiti pomoćne metode).

(a) Napisati metod `private static void aPronadjiSveKandidate(Connection con) throws SQLException` koja na standardni izlaz ispisuje naredne informacije o kandidatima: (1) indeks, i (2) poeni tokom studija. Podatke ispisati samo za one kandidate koji nisu polagali test (tj. koji imaju -1 poen na testu). Podatke urediti po poenima ostvarenim tokom studija opadajuće.

*Napomene: Proveravati greške koje se javljaju prilikom izvršavanja aplikacije u višekorisničkom okruženju.*

(b) Napisati metod `private static void bUnesiPoeneSaTesta(Connection con, Scanner ulaz) throws SQLException` koji zahteva od korisnika da odabere jedan od indeksa koji je ispisan metodom pod (a). Nakon odabira indeksa, aplikacija vrši izmenu podataka za dati indeks metodom pod (c). Zatim, aplikacija pita da li korisnik želi da unese poene za još jednog studenta. Ukoliko je odgovor potvrdan, cela akcija se ponavlja. U suprotnom, metod se završava.

*Napomene: Obrada podataka za jednog studenta predstavlja jednu transakciju.*

(c) Napisati metod `private static void cUnesiPoeneNaTestuZaStudenta(Connection con, Integer indeks, Scanner ulaz) throws SQLException` koji za studenta čiji se indeks prosleđuje metodu pita korisnika da unese poene na testu za praksu. Nakon unošenja poena, aplikacija vrši izmenu podataka u tabeli `DA.PRAKSA` tako što studentu sa datim indeksom postavlja poene na testu.

*Napomene: Proveravati greške koje se javljaju prilikom izvršavanja aplikacije u višekorisničkom okruženju.*

Napisati Java/SQL aplikaciju u kojoj se SQL naredbe izvršavaju dinamički (JDBC) koja metodom pod (a) ispisuje podatke. Zatim, aplikacija metodom pod (b) obrađuje informacije o kandidatima, sve dok korisnik ne završi obradu.

*Napomene: Postaviti istek vremena za zahtevanje katanaca na 5 sekundi na nivou celokupne aplikacije. Omogućiti da nijedna druga aplikacija ne može da menja podatke sve dok ova aplikacija ne završi sa transakcijama, ali može da unosi nove podatke za obradu. Obavezno je obrađivanje grešaka.*

Primer interakcije korisnika sa programom je dat u nastavku. Tamo gde su navedene ... znači da je izlaz iz programa skraćen radi povećanja čitljivosti.

```
=========================================================
20100084 97.72727
20100062 95.652176
20100058 91.73913
20100060 91.666664
20100055 89.565216
20100036 78.63636
20100121 74.545456
20100201 73.181816
20090073 68.75
=========================================================
Odaberite narednog kandidata iz liste:
20100084
Unesite poene sa testa za kandidata sa indeksom: 20100084
91.25

Nastaviti dalje? [da/ne]
da

Odaberite narednog kandidata iz liste:
20090073
Unesite poene sa testa za kandidata sa indeksom: 20090073
84.75

Nastaviti dalje? [da/ne]
ne
```

## 3. zadatak

Napisati Java aplikaciju koja korišćenjem biblioteke Hibernate priprema finalni spisak kandidata. Za svaki studijski program čiji naziv počinje na karakter koji korisnik unese sa standardnog ulaza ispisuje zaglavlje sa narednim informacijama: (1) naziv, (2) identifikator, i (3) broj ESPB potrebnih da se taj studijski program završi. Aplikacija zatim pronalazi studente koji su kandidati za praksu na tom studijskom programu, i za svakog kandidata:

- Ako je kandidat polagao test za praksu, izdvojiti: (1) ime, (2) prezime studenta, (3) poene na osnovu studiranja, (4) poene koje je student dobio na testu za praksu i (5) ukupne poene. Urediti spisak po ukupnim poenima opadajuće.
- Ako kandidat nije polagao test za praksu, obrisati podatke o tom kandidatu.

Obavezno je koristiti podatke iz tabela `DA.PRAKSA`, `DA.DOSIJE` i `DA.STUDIJSKIPROGRAM`.

*Napomene: Obraditi ceo zahtev zadatka u jednoj transakciji. Objektno-relaciono preslikavanje uraditi isključivo Java anotacijama. Za sve tabele koje se koriste u rešenju zadatka, a koje imaju ograničenja stranih ključeva jedne ka drugima, potrebno je implementirati odgovarajuća dvosmerna preslikavanja tih stranih ključeva. Nije dozvoljeno korišćenje SQL upita. Obavezno je obrađivanje grešaka.*

Primer interakcije korisnika sa programom je dat u nastavku.

```
Unesite karakter za pretragu studijskih programa:
s

Statistika, aktuarska i finansijska matematika (208) -- bodovi: 240
```

Statistika, aktuarska i finansijska matematika (238) -- bodovi: 60

Studijski program spec. studija (2) -- bodovi: 60
    Miljana Minic 78.63636 -1.0
    Bojan Savkovic 95.652176 -1.0
    Ivan Rakic 74.545456 -1.0
    Marko Telebakovic 73.181816 -1.0
    Perica Divnic 68.75 84.75
    Milan Todovic 89.565216 -1.0
    Darko Janjic 91.666664 -1.0
    Jelena Dumanjic 97.72727 91.25
    Milan Tomasevic 91.73913 -1.0