

## PRIMER - SQLJ

### LEKARSKA ORDINACIJA

Posmatrani sistem je jedna lekarska ordinacija koja opslužuje pacijente koji se vode u evidenciji i u kojoj svaki od određenog broja lekara može da pruža sve usluge. Ordinacija radi tako što pacijenti prvo zakazuju termin u fiksnom trajanju od jednog sata. Tom prilikom, operater u ordinaciji prvo evidentira pacijenta ako već nije u evidenciji, a zatim određuje prvi raspoloživi termin, pri čemu tada nije poznato koji će ga lekar opslužiti. Svaki zakazani termin može biti ostvaren ili neostvaren (ako pacijent ne dođe). Prilikom svakog ostvarenog termina određeni lekar prvo utvrđuje koje će sve usluge obaviti a zatim pruža te usluge u vidu tretmana. Na kraju ostvarenog termina a po okončanju svih tretmana pacijentu se ispostavlja račun koji se plaća u celini odmah. Račun sadrži stavke od kojih svaka odgovara jednom tretmanu.

Šema relacione baze podataka je:

```
USLUGA (IDUsl, Naziv, Cena)
LEKAR (IDLek, PrezimeIme)
PACIJENT (IDPac, PrezimeIme, KontaktPodaci)
TERMIN (IDTer, Datum, Vreme, Ostvarenost, IDPac, IDLek)
TRETMAN (IDTer, RedBr, IDUsl)
RACUN (IDRac, Datum, IDTer, Status)
STAVKA (IDRac, RedBr, IDUsl)
```

Sastaviti Java metodu koja koristeći SQLJ implementira funkcionalnost ObradaTermina, koja se obavlja samo ukoliko je pacijent došao u svom zakazanom terminu. Pretpostaviti da se koristi podrazumevana konekcija.

```

public boolean ObradaTermina(int idPac, int idLek){

    //provera ispravnosti unetih podataka

    int idTermina;

    #SQL {SELECT IDTer
        INTO :idTermina
        FROM TERMIN T
        WHERE T.IDPac = :idPac
        AND T.IDLek = :idLek
        AND T.Ostvarenost = 'N'};

    //pretpostavka je da je dovoljno da postoji zakazan termin
    //mogla je i provera datuma za kada je zakazan termin

    if(idTermina == NULL) return false;

    //evidentiranje pocetka termina

    #SQL {UPDATE TERMIN
        SET Ostvarenost = 'D'
        WHERE IDTer = :idTermina};

    //utvrdjivanje potrebnih usluga

    int[] idPotrebnihUsluga = UcitajPotrebneUsluge();

    //pruzanje usluga

    int redBr = 0;

    for (int idUsluge : idPotrebnihUsluga){

        redBr++;

        #SQL {INSERT INTO TRETMAN VALUES (:idTermina, :redBr, :idUsluge);

    }

    //izrada racuna

    redBr = 0;
    idRacuna = 0;

    #SQL {SELECT MAX(IDRac) + 1
        INTO :idRacuna
        FROM RACUN};

    #SQL {INSERT INTO RACUN VALUES (:idRacuna, CURRENT_DATE, :idTermina, 'N');

    for (int idUsluge : idPotrebnihUsluga){

        redBr++;
        #SQL {INSERT INTO STAVKA VALUES (:idRacuna, :redBr, :idUsluge);

    }

    //naplata racuna

    #SQL {UPDATE RACUN SET Status = 'D' WHERE IDRac = :idRacuna};

    return true;

}

```