

## PRIMER - SQLJ

### SISTEM MOBILNE TELEFONIJE

Sistem mobilne telefonije opslužuje svoje korisnike od kojih svaki može da ima jednu ili više SIM kartica u postpejd režimu (plaćanje je putem mesečnih računa). Obračun se vrši po sekundama razgovora odnosno broju znakova poruka. Svaki račun osim opštih podataka ima i stavke, od kojih svaka odgovara jednoj pruženoj usluzi.

Korisnicima su na raspolaganju usluge poziva za razgovor i slanja SMS poruka. Slanje SMS poruke uvek uspeva - poruka se zapisuje u bazi podataka i trajno čuva, a prosleđuje se odredištu čim je to moguće. Svaki poziv za razgovor se trajno evidentira i može da ima tri ishoda:

- Uspešno uspostavljena veza sa pozvanim brojem ; na kraju razgovora, evidentira se njegovo trajanje;
- Zauzetost pozvanog broja: automatski se generiše poruka o zauzeću koju će sistem poslati pozivaru čim pozvani prestane da bude zauzet;
- Nedostupnost pozvanog broja: automatski se geberiše poruka o dostupnosti koju će poslati pozivaru čim pozvani postane dostupan.

Automatski generisane poruke se raspoznaju po tome što je identifikacija pošiljaoca NULL. Aktor u ovom sistemu nije čovek nego uređaj - kontroler koji na početku poziva daje interfejsu odgovarajuće podatke o pozivu a posle toga, ako je razgovor obavljen, odgovarajuće dodatne podatke.

Tabele u bazi podataka su:

```
KORISNIK (IDKor, PrezimeIme, Adresa)
SIM_KARTICA (IDSIm, TelBroj, Status, IDKor)
USLUGA (IDUsl, IDSImOd, IDSImDo, DatumVremeOd, Status, Iznos)
POZIV (IDUsl, DatumVremeDo)
PORUKA (IDUsl, TekstPoruke)
CENA (IDCen, CenaPoziva, CenaPoruke, Datum)
RACUN (IDRac, ZaMesec, IDKor)
STAVKA (IDRac, RedBr, Iznos, IDUsl)
```

Sastaviti Java metodu koja koristeći SQLJ implementira funkcionalnost ObradaKrajaPoziva.

```

public boolean ObradaKrajaPoziva (int idUsluge,
                                Timestamp datumVremeDo){

    //odredjivanje pocetka poziva

    int idSimOd;
    Timestamp datumVremeOd;

    #SQL {SELECT IDSimOd, DatumVremeOd
          INTO :idSimOd, :datumVremeOd
          FROM USLUGA
          WHERE IDUsl = :idUsluge};

    //obracun iznosa usluge

    #SQL iterator CenaUsluga(int);
    CenaUsluga kursorCenaUsluga;

    #SQL kursorCenaUsluga = {SELECT CenaPoziva FROM CENA ORDER BY Datum DESC}

    int trenutnaCenaPoziva

    #SQL {FETCH :kursorCenaUsluga INTO :trenutnaCenaPoziva};

    int iznosUsluge = TrajanjePoziva(DatumVremeOd, DatumVremeDo) * trenutnaCenaPoziva;

    //evidentiranje usluge

    #SQL {UPDATE POZIV
          SET DatumVremeDo = :datumVremeDo
          WHERE IDUsl = :idUsluge};

    #SQL {UPDATE USLUGA
          SET Status = 'D', Iznos = :iznosUsluge
          WHERE IDUsl = :idUsluge};

    return true;
}

```