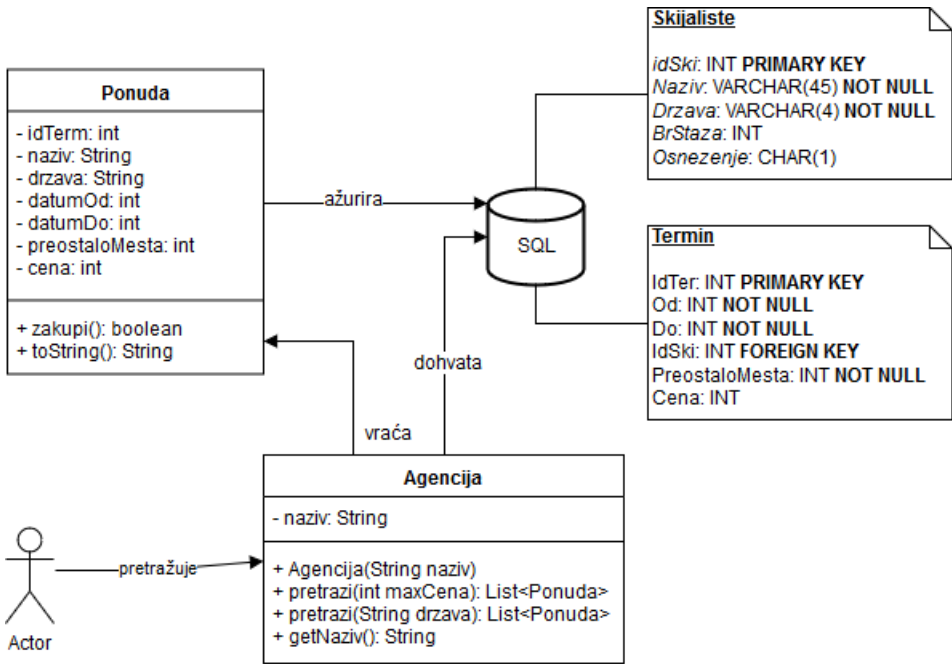


# Agencija

Napravite turističku agenciju koristeći bazu podatka nalik onoj iz prethodnog domaćeg. Dat je dijagram implementacije kako bi rešenje lakše vizuelizovali kao i detaljan tekst implementacije.



## Agencija

Agencija je opisana imenom i povezana je sa lokalnom bazom podataka. Agencija nudi mogućnost pretrage kreiranjem liste ponuda shodno rezultatu pretrage baze po parametru imena države ili po maksimalnoj ceni. Polje naziva može se dohvatiti.

## Ponuda

Ponuda predstavlja jedinstveni aranžman agencije dobijen spajanjem skijališta i termina iz baze. Opisana identifikacionim brojem termina, nazivom skijališta, državom u kojoj se nalazi skijalište, vremenskim intervalom održavanja, preostalim broj mesta i cenom. Termin ponude se može zakupiti pozivom metode `zakupi` koja vraća boolean vrednost u zavisnosti da li je zakup moguće izvršiti i `ažurira` vrednost u bazi ukoliko je izvršen zakup. Ponuda ima metodu `toString` koja vraća string u formatu `(idTerm) naziv(drzava) datumOd - datumDo / cena : preostaloMesta`.

Kod metoda za potrebe testiranja koda i dobijanja ispisa koji ne treba dirati osim za potrebe razrešavanja dodatnih izuzetaka ili drugačije veze sa bazom podataka:

```
import java.sql.SQLException;
import java.util.List;

public class Program {
    private static void simulacijaImpl(Agencija agencija, String pocetnaPoruka, List<Ponuda> ponuda)
    {
        System.out.println("-- Osoba ulazi u turističku agenciju " + agencija.getNaziv() + " --");
        System.out.println("Osoba: Dobar dan! " + pocetnaPoruka + " Šta imate u ponudi?");
        if(ponuda.size()==0) {
            System.err.println("Prodavac: Nažalost nemamo to u ponudi :(");
            System.out.println("Osoba: I vi ste mi neka agencija...!");
        }else {
            System.out.println("Prodavac: Ponuda za traženi upit je sledeća");
            for(int i=0;i<ponuda.size();i++)
                System.out.println("  " + ponuda.get(i));

            boolean zakupio = false;
            while(ponuda.size()>0 && !zakupio) {
                System.out.println("Osoba: Zakupiću termin -- " + ponuda.get(0));
                zakupio = ponuda.get(0).zakupi();
                if(!zakupio) {
                    System.err.println("Prodavac: Nažalost nemamo mesta u tom terminu :(");
                    ponuda.remove(0);
                }
            }
            if(zakupio) {
                System.out.println("Prodavac: Zakupljeno!");
                System.out.println("Osoba: Hvala! Nadam se da će provod biti odličan!");
            }else {
                System.out.println("Osoba: I vi ste mi neka agencija...!");
            }
        }
        System.out.println();
    }

    public static void simulacija(Agencija agencija, String drzava) throws SQLException {
        simulacijaImpl(agencija, String.format("%s je moja odabrana zemlja zimovanja!", drzava), agencija.pretrazi(drzava));
    }

    public static void simulacija(Agencija agencija, int maxCena) throws SQLException {
        simulacijaImpl(agencija, String.format("Zanimaju me ponude do %d€.", maxCena), agencija.pretrazi(maxCena));
    }
}
```

```

public static void main(String[] args) {
    try {
        Agencija agencija = new Agencija("ZimoTurs");

        simulacija(agencija, "Srbija");
        simulacija(agencija, 250);
        simulacija(agencija, "Svajcarska");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

SQL za generaciju baze:

```

DROP TABLE IF EXISTS Skijaliste;

CREATE TABLE Skijaliste (IdSki INTEGER PRIMARY KEY,
    Naziv VARCHAR(45) NOT NULL,
    Drzava VARCHAR(45) NOT NULL,
    BrStaza INTEGER,
    Osnezenje CHAR(1)) ;

INSERT INTO Skijaliste(IdSki,Drzava,Naziv,BrStaza,Osnezenje)
VALUES (1, 'Srbija', 'Kopaonik', 25, 'D'),
    (2, 'Srbija', 'Zlatibor', 4, 'N'),
    (3, 'Francuska', 'Les 2 Alpes', 96, 'D'),
    (4, 'Francuska', 'Paradiski', 253, 'D'),
    (5, 'Francuska', 'Chatel', 49, NULL),
    (6, 'Bugarska', 'Bansko', 18, 'D');

DROP TABLE IF EXISTS Termin;

CREATE TABLE Termin (IdTer INTEGER PRIMARY KEY,
    Od INTEGER NOT NULL,
    Do INTEGER NOT NULL,
    IdSki INTEGER,
    Cena INTEGER,
    PreostaloMesta INTEGER NOT NULL) ;

INSERT INTO Termin(IdTer,Od,Do,IdSki,Cena, PreostaloMesta)
VALUES (1, 20181229, 20190105, 1, 350, 4),
    (2, 20190105, 20190112, 1, 230, 0),
    (3, 20190126, 20190202, 6, 165, 3),
    (4, 20190125, 20190203, 3, 396, 2),
    (5, 20190201, 20190210, 3, 410, 5),
    (6, 20190308, 20190317, 3, 379 , 6);

```

Ispis koji se dobija:

-- Osoba ulazi u turističku agenciju ZimoTurs --

Osoba: Dobar dan! Srbija je moja odabrana zemlja zimovanja! Šta imate u ponudi?

Prodavac: Ponuda za traženi upit je sledeća

(2) Kopaonik(Srbija) 20190105 - 20190112 / 230€ : 0

(1) Kopaonik(Srbija) 20181229 - 20190105 / 350€ : 4

Osoba: Zakupiću termin -- (2) Kopaonik(Srbija) 20190105 - 20190112 / 230€ : 0

Prodavac: Nažalost nemamo mesta u tom terminu :(

Osoba: Zakupiću termin -- (1) Kopaonik(Srbija) 20181229 - 20190105 / 350€ : 4

Prodavac: Zakupljeno!

Osoba: Hvala! Nadam se da će provod biti odličan!

-- Osoba ulazi u turističku agenciju ZimoTurs --

Osoba: Dobar dan! Zanimaju me ponude do 250€. Šta imate u ponudi?

Prodavac: Ponuda za traženi upit je sledeća

(3) Bansko(Bugarska) 20190126 - 20190202 / 165€ : 3

(2) Kopaonik(Srbija) 20190105 - 20190112 / 230€ : 0

Osoba: Zakupiću termin -- (3) Bansko(Bugarska) 20190126 - 20190202 / 165€ : 3

Prodavac: Zakupljeno!

Osoba: Hvala! Nadam se da će provod biti odličan!

-- Osoba ulazi u turističku agenciju ZimoTurs --

Osoba: Dobar dan! Švajcarska je moja odabrana zemlja zimovanja! Šta imate u ponudi?

Prodavac: Nažalost nemamo to u ponudi :(

Osoba: I vi ste mi neka agencija...!