

9. Deveta laboratorijska vježba

9.1. JAVA FX – NAPREDNO KORIŠTENJE

Svrha laboratorijske vježbe je korištenje baze podataka umjesto datoteka od strane JavaFX aplikacije kroz operacije dohvaćanja i spremanja podataka. Funkcionalnosti aplikacije koje treba uključivati rješenje laboratorijske vježbe iste su onima koje su implementirane u sedmoj laboratorijskoj vježbi, samo što se za pohranu podataka koristiti baza podataka.

9.2. ZADATAK ZA PRIPREMU

Nastaviti razvoj aplikacije iz osme laboratorijske vježbe, oblikovati dodatne ekrane JavaFX grafičkog sučelja aplikacije i implementirati podršku za rad s bazom podataka prema koracima u nastavku.

1. Kopirati projekt iz osme laboratorijske vježbe i preimenovati ga u na način da mu se promijeni redni broj u „9“.
2. U „pom.xml“ dodati zavisnost o ispravnoj verziji drivera za H2 bazu podataka.
3. Kreirati datoteku „bazaPodataka.properties“ i u njoj dodati parove „ključ-vrijednost“ za URL baze podataka, korisničko ime i lozinku, npr.

```
#Osnovni podaci za spajanje na bazu podataka
bazaPodatakaUrl = jdbc:h2:tcp://localhost/~ /JDBC-TVZ-POSTAJE

#Podaci za pristup bazi podataka
korisnickoIme = student
lozinka = student
```

4. Kreirati bazu podataka prema podacima u datoteci „bazaPodataka.properties“ i u njoj izvršiti sljedeću skriptu koja će kreirati potrebne tablice i relacije:

```
CREATE SCHEMA POSTAJE;

CREATE TABLE POSTAJE.DRZAVA (
  ID INT NOT NULL GENERATED ALWAYS AS IDENTITY,
  NAZIV VARCHAR(50) NOT NULL,
  POVRSINA DOUBLE NOT NULL,
  PRIMARY KEY (ID)
);

INSERT INTO POSTAJE.DRZAVA (NAZIV, POVRSINA) VALUES
  ('Hrvatska', 56594),
  ('Slovenija', 20273);

CREATE TABLE POSTAJE.ZUPANIJA (
  ID INT NOT NULL GENERATED ALWAYS AS IDENTITY,
```

© Autor: Davor Lozić, struč. spec. ing. techn. inf.

```
NAZIV VARCHAR(50) NOT NULL,  
DRZAVA_ID INT NOT NULL,  
PRIMARY KEY (ID),  
FOREIGN KEY(DRZAVA_ID) REFERENCES POSTAJE.DRZAVA(ID)  
);  
  
INSERT INTO POSTAJE.ZUPANIJA (NAZIV, DRZAVA_ID) VALUES  
('Grad Zagreb', 1),  
('Ličko-senjska županija', 1),  
('Brodsko-posavska županija', 1),  
('Nova Gorica', 2),  
('Novo Mesto', 2);  
  
CREATE TABLE POSTAJE.MJESTO (  
ID INT NOT NULL GENERATED ALWAYS AS IDENTITY,  
NAZIV VARCHAR(50) NOT NULL,  
VRSTA VARCHAR(50) NOT NULL,  
ZUPANIJA_ID INT NOT NULL,  
PRIMARY KEY (ID),  
FOREIGN KEY(ZUPANIJA_ID) REFERENCES POSTAJE.ZUPANIJA(ID)  
);  
  
INSERT INTO POSTAJE.MJESTO (NAZIV, VRSTA, ZUPANIJA_ID) VALUES  
('Grad Zagreb', 'GRAD', 1),  
('Zavižan', 'OSTALO', 2),  
('Otočac', 'GRAD', 2),  
('Slavonski Brod', 'GRAD', 3);  
  
CREATE TABLE POSTAJE.MJERNA_POSTAJA (  
ID INT NOT NULL GENERATED ALWAYS AS IDENTITY,  
NAZIV VARCHAR(50) NOT NULL,  
MJESTO_ID INT NOT NULL,  
LAT DECIMAL(10, 8) NOT NULL,  
LNG DECIMAL(11, 8) NOT NULL,  
PRIMARY KEY (ID),  
FOREIGN KEY(MJESTO_ID) REFERENCES POSTAJE.MJESTO(ID)  
);  
  
INSERT INTO POSTAJE.MJERNA_POSTAJA (NAZIV, MJESTO_ID, LAT, LNG) VALUES  
('Maksimir', 1, 45.818557, 16.016666),  
('Zavižan', 1, 44.800000, 14.966667);  
  
CREATE TABLE POSTAJE.SENZOR (  
ID INT NOT NULL GENERATED ALWAYS AS IDENTITY,  
MJERNA_JEDINICA VARCHAR(10) NOT NULL,  
PRECIZNOST DOUBLE NOT NULL,  
VRIJEDNOST DECIMAL(7, 2),  
RAD_SENZORA VARCHAR(10),  
MJERNA_POSTAJA_ID INT NOT NULL,
```

```
PRIMARY KEY (ID),  
FOREIGN KEY (MJERNA_POSTAJA_ID) REFERENCES POSTAJE.MJERNA_POSTAJA (ID)  
);  
  
INSERT INTO POSTAJE.SENZOR (MJERNA_JEDINICA, PRECIZNOST, VRIJEDNOST, RAD_SENZORA,  
MJERNA_POSTAJA_ID) VALUES  
( 'g', 30, 20, 'STREAMING', 1),  
( 'kg', 5, 1, 'PING', 1);
```

5. Kreirati novi paket „hr.java.vjezbe.baza.podataka“ i u njemu klasu „BazaPodataka“.
6. Unutar klase „BazaPodataka“ kreirati privatnu metodu „spajanjeNaBazuPodataka“ koja ne prima nikakve parametre, a vraća objekt tipa „Connection“ koji predstavlja vezu na bazu podataka. Prilikom kreiranja veze potrebno je koristiti podatke iz datoteke „bazaPodataka.properties“. Osim metode za spajanje na bazu podataka potrebno je kreirati i privatnu metodu za zatvaranje veze s bazom podataka, npr. „zatvaranjeVezeNaBazuPodataka“, koja prima objekt tipa „Connection“ i zatvara vezu koju on predstavlja. Te metode će se morati pozivati prije i nakon izvršavanja upita nad bazom podataka.
7. U sve entitete dodati novu varijablu „id“ tipa „Integer“ te generirati „getter“ i „setter“ varijable.
8. Unutar klase „BazaPodataka“ za svaku komponentu napisati metodu koja služi za spremanje podataka u bazu. Primjer programskog koda za spremanje nove države u bazu podataka izgleda ovako:

```
public static void spremiDrzavu(Drzava drzava) throws SQLException,  
IOException {  
    Connection veza = spajanjeNaBazuPodataka();  
  
    veza.setAutoCommit(false);  
  
    try {  
        PreparedStatement insertDrzava = veza  
            .prepareStatement("INSERT INTO POSTAJE.DRZAVA  
(NAZIV, POVRSINA) VALUES (?, ?);");  
        insertDrzava.setString(1, drzava.getNaziv());  
        insertDrzava.setDouble(2, drzava.getPovrsina());  
        insertDrzava.executeUpdate();  
    }  
    catch (Throwable ex) {  
        veza.rollback();  
        throw ex;  
    }  
  
    zatvaranjeVezeNaBazuPodataka(veza);  
}
```

Osim spremanja u bazu podataka, svaki objekt mora biti vidljiv i u tablici na ekranu.

9. U klasi „BazaPodataka“ za svaku komponentu kreirati metodu za dohvat svih podataka iz tablice u bazi podataka i vratiti listu. Primjer metode koja dohvaća sve podatke o državama prikazan je u nastavku:

```
public static List<Drzava> dohvatiDrzave() throws SQLException,
IOException {

    Connection veza = spajanjeNaBazuPodataka();

    Statement statementDrzave = veza.createStatement();
    ResultSet resultSetDrzave =
statementDrzave.executeQuery("SELECT * FROM POSTAJE.DRZAVA");
    List< Drzava > listaDrzava = new ArrayList<>();

    while (resultSetDrzave.next()) {
        Integer drzavaId = resultSetDrzave.getInt("ID");
        String naziv = resultSetDrzave.getString("NAZIV");
        BigDecimal površina =
resultSetDrzave.getDouble("POVRŠINA");

        Drzava drzava = new Drzava(naziv, površina);
        drzava.setId(drzavaId);

        listaDrzava.add(drzava);
    }

    zatvaranjeVezeNaBazuPodataka(veza);

    return listaDrzava;
}
```

10. U klasu „BazaPodataka“ dodati metodu koja će služiti za spremanje podataka o konfiguraciji te tu metodu povezati s gumbom koji je do sada spremao podatke o datoteku.
11. Kod unosa podataka o entitetima koji uključuju i odabir jedne od predefiniраних vrijednosti (kao što je na primjer vrsta mjesta) potrebno je prikazivati korištenjem komponente „ComboBox“, a ne korištenjem „TextField“ polja.

NAPOMENE:

1. Svi detalji koji nisu definirati potrebno je proizvoljno implementirati.