

**SEMINARSKI RAD**

**PANEVROPSKI UNIVERZITET APEIRON  
FAKULTET INFORMACIONIH TEHNOLOGIJA**

**Redovne studije  
Smjer “Računarska bezbjednost i zaštita informacija”**

**Predmet  
OBJEKTNKO ORJENTISANO PROGRAMIRANJE  
(sa primjenom na programskom jeziku Java)**

**„Projekat – Biblioteka (unošenje, izmjena,  
brisanje i pretraga knjiga)”**

**(seminarski rad)**

**Predmetni profesor  
Prof. dr Saša Salapura**

**Student  
Milan Panić  
Index br. 77-19-R-ITB-S**

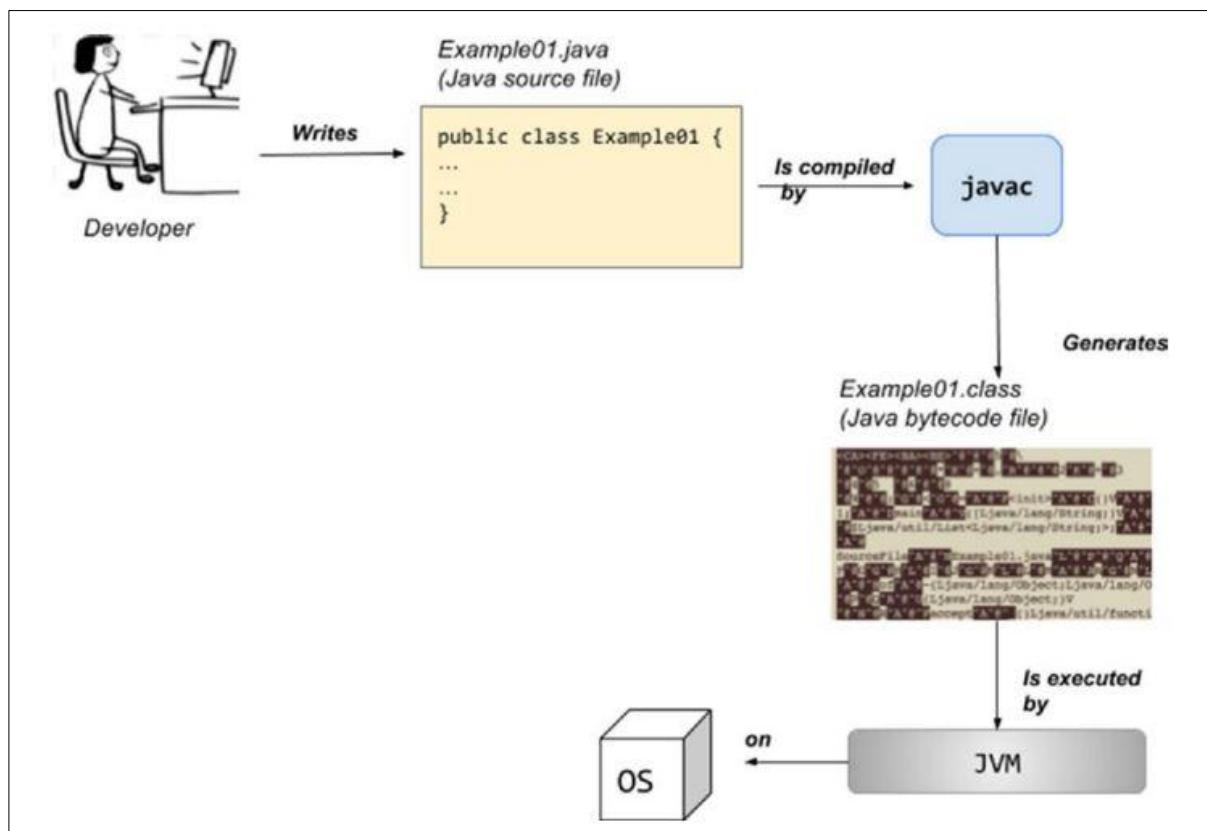
Banja Luka, novembar 2021.

# SADRŽAJ

UVOD .....	3
1. Razlika između JDK, JRE i JVM .....	4
2. Korištene tehnologije za projekat .....	4
2.1    Baze i MySQL.....	6
3. Pokretanje XAMPP-a i pravljenje baze.....	7
4. Dizajn aplikacije .....	8
5. Kreiranje vizuelnih elemenata .....	8
6. Programski kôd i struktura aplikacije .....	9
6.1    Povezivanje na bazu .....	10
6.2    Učitavanje tabele i podataka .....	10
6.3    Dugme za dodavanje knjiga .....	11
6.4    Programski kôd za novi unos .....	11
6.5    Programski kôd za pretraživanje knjige u bazi .....	12
6.6    Programski kôd za izmjenu knjige u bazi .....	13
6.7    Programski kôd za brisanje knjige iz baze .....	14
ZAKLJUČAK .....	15
POPIS SLIKA .....	16

# UVOD

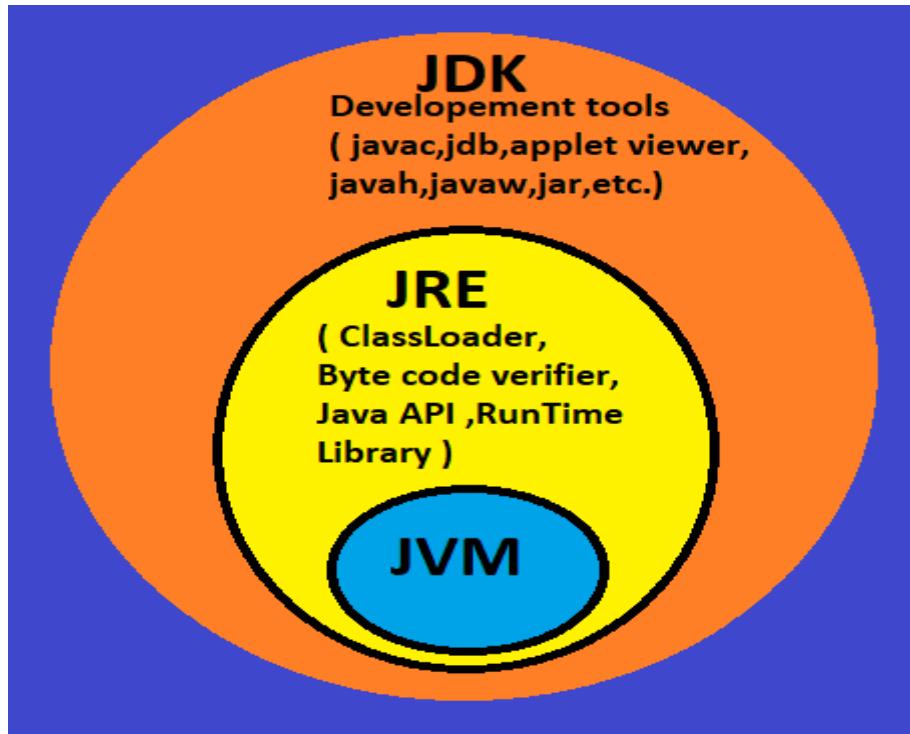
U ovom seminarskom radu će biti opisan projekat knjižara sa mogućnošću dodavanja, izmjena i brisanja knjiga iz baze. Aplikacija će biti obrađena u programskom jeziku Java, te će se povezivati na bazu kreiranu u MySql-u. Java je trenutno jedan od najuticajnijih programskih jezika. Java je programski jezik, razvijen od strane Sun Microsystems-a . Izuzetno je robusna i fleksibilna. Sve je počelo u 1990. kada je američka kompanija koja je vodila revoluciju u računaru industrija je odlučila okupiti svoje najbolje inženjere kako bi dizajnirali i razvili proizvod ,te da bi im omogućilo da postanu važan igrač na novom Internetu u svijetu. Među tim inženjerima bio je i **James Arthur Gosling**, kanadski informatičar koji je prepoznat kao „otac“ programskog jezika Java. Ono što je važno napomenuti jeste da je Java u potpunosti platformski nezavisna. Java je “*case sensitive*” što znači da je jezik koji je osjetljiv na velika i mala slova. Java je u potpunosti objektno orijentisana i za nju važi princip kisanja koda koji glasi **WORA – “Write Once Run Anywhere”**. Da bi se programiralo u javi potrebno je razvojno okruženje i JDK (eng. *Java Development Kit*), a koje tehnologije su se koristile za izradu ovog projekta će biti opisane u nastavku rada.



Slika 1. - Način izvršavaja Java kooda

## 1. RAZLIKA IZMEĐU JDK, JRE I JVM

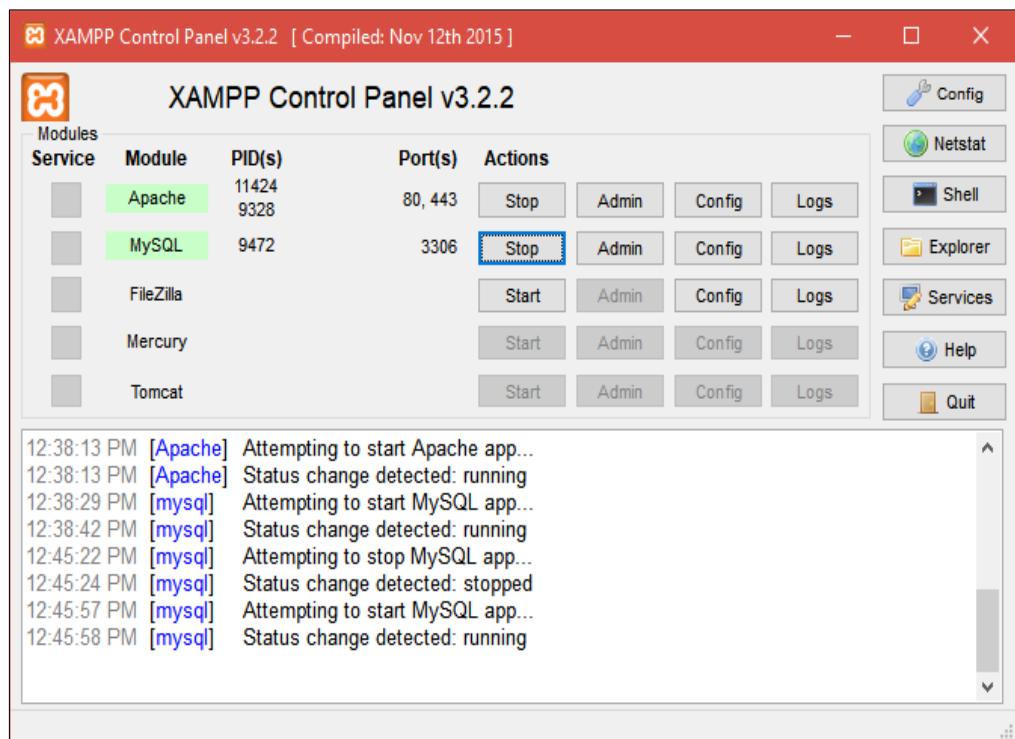
Kako bi smo bolje shvatili kako funkcioniše Java i od čega se sastoji potrebno je shvatiti razliku između ova tri termina. **JDK** je komplet za razvoj softvera, dok je **JRE** programski paket koji omogućava pokretanje Java programa, dok je **JVM** okruženje za izvršavanje bajt koda.



Slika 2. Razlika izmedu JDK, JRE i JVM

## 2. KORIŠTENE TEHNOLOGIJE ZA PROJEKAT

Aplikacija je razvijana u razvojnem okruženju **ECLIPSE 2021-03** kao i Java Development Kit (JDK verzija 15). Za GUI (*eng. Graphic User Interface*) je korišten Swing i WindowBuilder. Baza koja se koristila jeste na "lokalnom" računaru na **xampp-uo** kreirana u **MySQLu**.



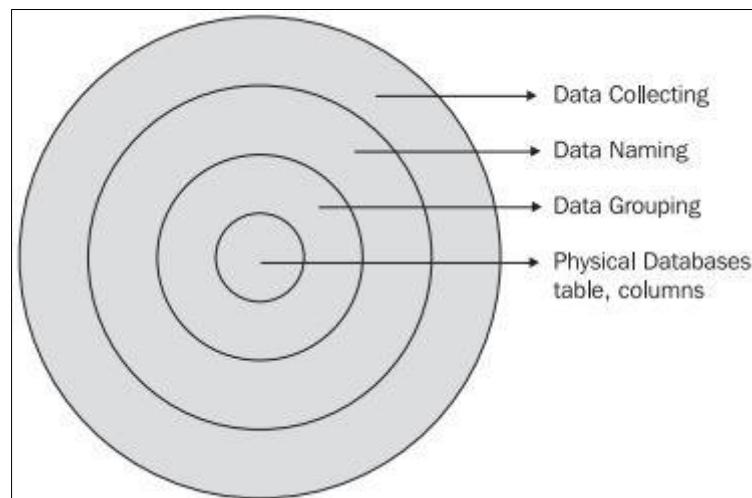
Slika 3. Izgled kontrolnog panela XAMPP-a



Slika 4. Eclipse razvojno okruženje

## 2.1 Baze i MySQL

MySQL, pokrenut 1995. godine, postao je najpopularnija baza otvorenog koda. Popularnost MySQL-a i phpMyAdmina dopustila je mnogima koji nisu IT stručnjaci za izradu dinamičkih web stranica s MySQL pozadinom. Vrlo je važno znati kako planirati strukturu podataka i kako je fizički implementirati koristeći MySQL-ov model.



Slika 5. - Manipulacija podacima



Slika 6. - MySQL logo

### 3. POKRETANJE XAMPP-A I PRAVLJENJE BAZE

Po pokretanju XAMPP-a, potrebno je ukucati localhost na web-browseru i prijaviti se preko PHPMyAdmina, kako bi smo kreirali bazu i dodali tabelu. S obzirom da je ovo samo jedan modul projekta kreiraćemo bazu pod nazivom knjižara i dodati jednu tabelu koja će se sastojati od kolona: **id, naziv, izdanje, dostupno i cijena**.

The screenshot shows the phpMyAdmin interface for MySQL. On the left, there's a sidebar with a tree view of databases: information\_schema, knjizara (which has a 'New' folder), mysql, performance\_schema, and sys. In the main area, the 'Databases' tab is selected. A 'Create database' button is visible. Below it, a search bar contains 'latin1\_swedish\_ci' and a 'Create' button. A table lists existing databases: information\_schema, knjizara, mysql, performance\_schema, and sys. Each entry includes its collation and a 'Check privileges' link. The total number of databases is 5.

Slika 7. - Kreiranje baze

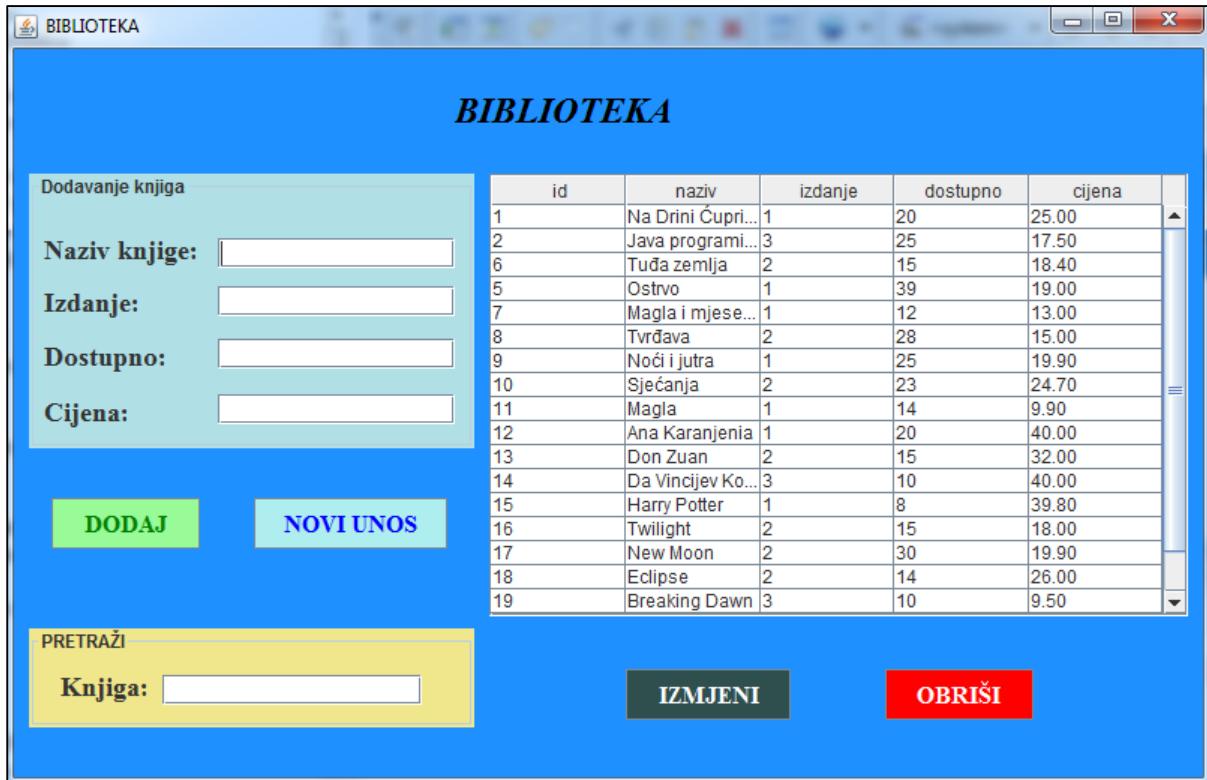
The screenshot shows the phpMyAdmin interface for the 'knjizara' database. The 'Table: knjige' tab is selected. The top navigation bar includes 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and a refresh icon. The main area displays the structure of the 'knjige' table with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>id</b> ↗	int(11)	utf8_unicode_ci		No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	<b>naziv</b>	varchar(250)	utf8_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	<b>izdanje</b>	int(11)	utf8_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	<b>dostupno</b>	int(11)	utf8_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	<b>cijena</b>	decimal(10,2)	utf8_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

At the bottom, there are buttons for 'Check all', 'With selected:', 'Print', 'Propose table structure', 'Move columns', 'Normalize', and other database management options.

Slika 8. - Tabela knjige

## 4. DIZAJN APLIKACIJE

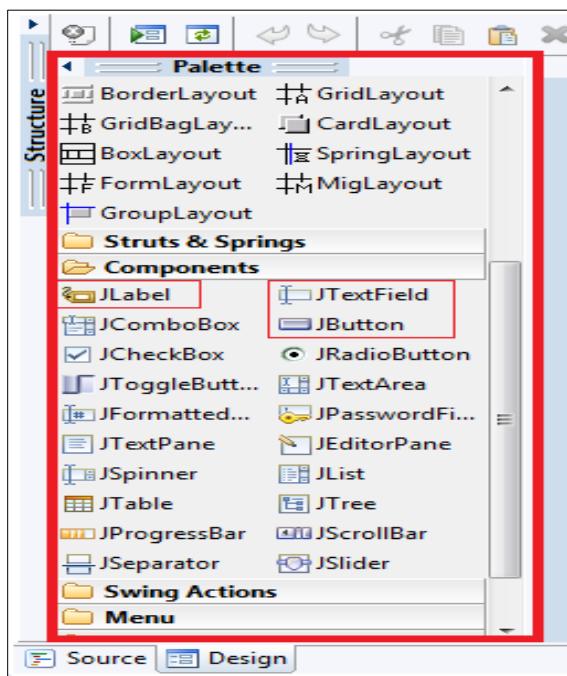


Slika 9. Izgled završne aplikacije

## 5. KREIRANJE VIZUELNIH ELEMENATA

Za kreiranje vizuelnih elemenata su korišteni:

- JFrame,
- JPanel,
- JLabel,
- JTextField,
- JTable i
- JButton.



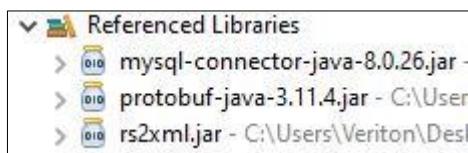
Sledeća slika prikazuje šta koji element predstavlja na aplikaciji. Crvenom i žutom bojom su označeni prethodno navedeni elementi.



Slika 10. Korišteni elementi

## 6. PROGRAMSKI KÔD I STRUKTURA APLIKACIJE

Prije samog pisanja kooda potrebno je ubaciti i odgovarajuće biblioteke za povezivanje sa bazom podataka a to su: **mysql-connector-java-8.0.26.jar** i **protobuf-java-3.11.4.jar**. Pored ove biblioteke potrebno je ubaciti i još jednu biblioteku za "JTable" kako bi lakše maniupisali podacima i kako bi podaci uopšte bili prikazani na našoj aplikaciji, a ta biblioteka je: **rs2xml.jar**.



Slika 11. Korištene biblioteke

## 6.1 Povezivanje na bazu

Ispod se nalazi programski kood za povezivanje sa bazom:

```
//GLOBALNE PROMJENLJIVE ZA POVEZIVANJE SA BAZOM

Connection con;
PreparedStatement pst;
ResultSet rs;

//POVEZIVANJE NA BAZU

public void Connect()
{
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1/knjizara", "root","");
    }
    catch (ClassNotFoundException ex)
    {
        ex.printStackTrace();
    }
    catch (SQLException ex)
    {
        ex.printStackTrace();
    }
}
```

## 6.2 Učitavanje tabele i podataka

```
//UCITAVANJE TABELE I PODATAKA IZ BAZE

public void table_load()
{
    try
    {
        pst = con.prepareStatement("select * from knjige");
        rs = pst.executeQuery();
        table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
```

### 6.3 Dugme za dodavanje knjiga

```
// DUGME DODAJ KNJIGU

public void actionPerformed(ActionEvent e) {

    String nazivKnjige, izdanje, cijena, dostupno;

    nazivKnjige = txtNazivKnjige.getText();
    izdanje = txtIzdanje.getText();
    dostupno = txtDostupno.getText();
    cijena = txtCijena.getText();

    try {
        pst = con.prepareStatement("insert into
knjige(naziv, izdanje, dostupno, cijena)values(?, ?, ?, ?)");
        pst.setString(1, nazivKnjige);
        pst.setString(2, izdanje);
        pst.setString(3, dostupno);
        pst.setString(4, cijena);
        pst.executeUpdate();

        JOptionPane.showMessageDialog(null, "Knjiga
uspješno dodata!");

        table_load();

        txtNazivKnjige.setText("");
        txtIzdanje.setText("");
        txtDostupno.setText("");
        txtCijena.setText("");
        txtNazivKnjige.requestFocus();
    }

    catch (SQLException e1)
    {

        e1.printStackTrace();

    }
}
});
```

### 6.4 Programski kôd za novi unos

```
// DUGME "NOVI UNOS" (POSTAVLJANJE BLANK VRIJEDNOSTI)
public void actionPerformed(ActionEvent e) {
    txtNazivKnjige.setText("");
    txtIzdanje.setText("");
    txtDostupno.setText("");
    txtCijena.setText("");
    txtPretrazi.setText("");
    txtNazivKnjige.requestFocus();
}

});
```

## 6.5 Programski kôd za pretraživanje knjige u bazi

Nakon unosa knjige u polje traži se podudaranje sa knjigom koja postoji, odnosno ne postoji u bazi, kao što je prikazano na slici ispod. Podaci se ispišu u odgovarajućim text-boxovima.

id	naziv	izdanje	dostupno	cijena
1	Na Drini Čuprija	1	20	25.00
2	Java programi...	3	25	17.50
6	Tuđa zemlja	2	15	18.40
5	Ostrvo	1	39	19.00
7	Magla i mjesec...	1	12	13.00
8	Tvrđava	2	28	15.00
9	Noći i jutra	1	25	19.90
10	Sjećanja	2	23	24.70
11	Magla	1	12	9.90
12	Ana Karanjenja	1	20	40.00
13	Don Zuan	2	15	32.00

Slika 12. Primjer pretrage knjige

```
// PRETRAZIVANJE KNJIGE

@Override
public void keyReleased(KeyEvent e) {
    try {

        String unos = txtPretrazi.getText();

        pst = con.prepareStatement("SELECT * FROM knjige WHERE naziv = ?");
        pst.setString(1, unos);
        ResultSet rs = pst.executeQuery();

        if (rs.next() == true){

            String naziv = rs.getString(2);
            String izdanjeS = rs.getString(3);
            String dostupno = rs.getString(4);
            String cijena = rs.getString(5);

            txtNazivKnjige.setText(naziv);
            txtIzdanje.setText(izdanjeS);
            txtDostupno.setText(dostupno);
            txtCijena.setText(cijena);

        }
    } else
    {
        txtNazivKnjige.setText("");
        txtIzdanje.setText("");
    }
}
```

```

        txtDostupno.setText("");
        txtCijena.setText("");

    }

}

catch (SQLException ex) {
}

});

});
```

## 6.6 Programski kôd za izmjenu knjige u bazi

```

// DUGME IZMJENI KNJIGU
public void actionPerformed(ActionEvent e) {

    String nazivKnjige, izdanje, cijena, dostupno, unos;
    nazivKnjige = txtNazivKnjige.getText();
    izdanje = txtIzdanje.getText();
    dostupno = txtDostupno.getText();
    cijena = txtCijena.getText();
    unos = txtPretrazi.getText();

    try {
        pst = con.prepareStatement("UPDATE knjige SET
naziv=?, izdanje=?, dostupno=?, cijena=? WHERE naziv=?");

        pst.setString(1, nazivKnjige);
        pst.setString(2, izdanje);
        pst.setString(3, dostupno);
        pst.setString(4, cijena);
        pst.setString(5, unos);
        pst.executeUpdate();

        JOptionPane.showMessageDialog(null, "Knjiga
uspješno izmjenjena!");
        table_load();

        txtNazivKnjige.setText("");
        txtIzdanje.setText("");
        txtDostupno.setText("");
        txtCijena.setText("");
        txtNazivKnjige.requestFocus();
    }

    catch (SQLException e1)
    {

        e1.printStackTrace();
    }
}
});
```

## 6.7 Programski kôd za brisanje knjige iz baze

```
//DUGME ZA BRISANJE KNJIGE

public void actionPerformed(ActionEvent e) {

    String unos;

    unos = txtPretrazi.getText();

    try {
        pst = con.prepareStatement("DELETE FROM knjige
WHERE naziv=?");
        pst.setString(1, unos);

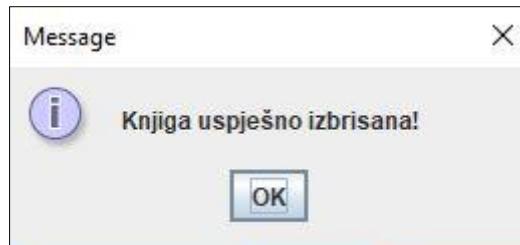
        pst.executeUpdate();
        JOptionPane.showMessageDialog(null, "Knjiga
uspješno izbrisana!");
        table_load();

        txtNazivKnjige.setText("");
        txtIzdanje.setText("");
        txtDostupno.setText("");
        txtCijena.setText("");
        txtPretrazi.setText("");
        txtNazivKnjige.requestFocus();
    }

    catch (SQLException e1)
    {

        e1.printStackTrace();
    }
}
});
```

Ako je knjiga uspješno obrisana bića ispisana poruka kao na slici ispod.



Slika 13. Poruka nakon uspješnog brisanja

## **ZAKLJUČAK**

Kroz ovaj seminarski rad je opisana aplikacija za dodavanje, izmjenu, brisanje i pretragu knjiga u bazi podataka. Takođe aplikacija predstavlja samo jedan moduо cijelog informacionog sistema za neku knjižaru, odnosno biblioteku.

## **POPIS SLIKA**

Slika 1. - Način izvršavaja Java kooda .....	3
Slika 2. Razlika između JDK, JRE i JVM .....	4
Slika 3. Izgled kontrolnog panela XAMPP-a .....	5
Slika 4. Eclipse razvojno okruženje .....	5
Slika 5. - Manipulacija podacima .....	6
Slika 6. - MySQL logo.....	6
Slika 7. - Kreiranje baze .....	7
Slika 8. - Tabela knjige .....	7
Slika 9. Izgled završne aplikacije.....	8
Slika 10. Korišteni elementi .....	9
Slika 11. Korištene biblioteke .....	9
Slika 12. Primjer pretrage knjige.....	12
Slika 13. Poruka nakon uspješnog brisanja .....	14