

Implementacija transakcione obrade korišćenjem JDBC API-a.

Rešenje:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.sql.DataSource;
import org.apache.commons.dbcp.BasicDataSource;

public class DBAccessExample {

    protected static void logException(Throwable e) { e.printStackTrace(); }

    public static class Student {
        String ime, index;

        public Student(String ime, String index) {
            super();
            this.ime = ime;
            this.index = index;
        }
    }

    public static void printStudents(ResultSet rs) throws SQLException {
        while (rs.next()) {
            System.out.println(rs.getString("ime") + " " + rs.getString("index"));
        }
    }

    public static void insertStudent(Student student, Connection con)
        throws SQLException {
        PreparedStatement stmt = null;
        try {
            stmt = con.prepareStatement(
                "INSERT INTO student (ime, prezime, indeks) " +
                "VALUES (?, ?, ?)");
            stmt.setString(1, student.ime);
            stmt.setString(2, "");
            stmt.setString(3, student.index);
            stmt.executeUpdate();
        }
        finally {
            if (stmt != null)
                try { stmt.close(); }
                catch (SQLException e) { logException(e); }
        }
    }
}
```

```

public static void insertStudents(List<Student> students)
throws SQLException {
    Connection con = null;
    try {
        con = ds.getConnection();
        con.setAutoCommit(false);
        for (Student s : students) {
            insertStudent(s, con);
        }
        con.commit();
    }
    finally {
        if (con != null)
            try { con.close(); } catch (SQLException e) { logException(e); }
    }
}

private static DataSource ds;

public static void main(String[] args) throws SQLException {
    ds = setupDataSource("");
    {
        List<Student> students = new ArrayList<Student>();

        students.add(new Student("Pera", "2009/0001"));
        students.add(new Student("Zika", "2009/0002"));
        students.add(new Student("Mika", "2009/0003"));

        insertStudents(students);
    }
    shutdownDataSource(ds);
}

public static DataSource setupDataSource(String connectURI) {
    BasicDataSource ds = new BasicDataSource();
    //ds.setDriverClassName("oracle.jdbc.driver.OracleDriver");
    ds.setDriverClassName("com.mysql.jdbc.Driver");
    ds.setUsername("epos_user");
    ds.setPassword("epos.123");
    ds.setUrl("jdbc:mysql://localhost:3306/iep");
    return ds;
}

public static void printDataSourceStats(DataSource ds) {
    BasicDataSource bds = (BasicDataSource)ds;
    System.out.println("NumActive: " + bds.getNumActive());
    System.out.println("NumIdle: " + bds.getNumIdle());
}

public static void shutdownDataSource(DataSource ds) throws SQLException {
    BasicDataSource bds = (BasicDataSource)ds;
    bds.close();
}
}

```