

```

package view;

import javax.swing.table.DefaultTableModel;
import dto.Subject;
import dao.Connect;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author Sérgio Felipe Starke
 */
public class Main extends javax.swing.JFrame {
    DefaultTableModel table = new DefaultTableModel();
    Connect dao = new Connect();
    Subject dto = new Subject();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    public Main() {
        initComponents();
        conn = dao.getConnection();
        setLocationRelativeTo(null);
        table = (DefaultTableModel) jTable.getModel();
        clean();
    }

    private void search_cpf(){//Busca pelo cpf e resultado aparece na jTable.
        try {
            ps = conn.prepareStatement(
                "select id,cpf,nome,sobrenome,email,fone,logradouro from pessoa where cpf like ?"
            );
            ps.setString(1, txtCpf.getText()+" %");
            rs = ps.executeQuery();

            ((DefaultTableModel)jTable.getModel()).setRowCount(0);//Limpa jTable.
            while(rs.next()){
                table.addRow(table.getRowCount(), new Object[] {rs.getString(1),
                    rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),
                    rs.getString(6),rs.getString(7)});
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private void search_nome(){//Busca pelo nome e resultado aparece na jTable.
        try {
            ps = conn.prepareStatement(
                "select id,cpf,nome,sobrenome,email,fone,logradouro from pessoa where nome like ?"
            );
            ps.setString(1, txtNome.getText()+" %");
            rs = ps.executeQuery();

            ((DefaultTableModel)jTable.getModel()).setRowCount(0);//Limpa jTable.
            while(rs.next()){
                table.addRow(table.getRowCount(), new Object[] {rs.getString(1),
                    rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),
                    rs.getString(6),rs.getString(7)});
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private void setText(){//Preenche os campos"txt" quando selecionado em jTable.
        int numero = jTable.getSelectedRow(); // "numero" da linha da jTable.
        //ATENÇÃO: Em jTable as colunas começa com o nº "0" CPF = 0/ Nome = 1...
        txtCpf.setText(jTable.getModel().getValueAt(numero, 0).toString());
        txtNome.setText(jTable.getModel().getValueAt(numero, 1).toString());
        txtSobrenome.setText(jTable.getModel().getValueAt(numero, 2).toString());
        txtEmail.setText(jTable.getModel().getValueAt(numero, 3).toString());
        txtFone.setText(jTable.getModel().getValueAt(numero, 4).toString());
        txtLogradouro.setText(jTable.getModel().getValueAt(numero, 5).toString());
        btnCreate.setEnabled(false);
    }

    private void clean(){
        txtId.setText(null);
        txtCpf.setText(null);
        txtNome.setText("");
        txtSobrenome.setText(null);
        txtEmail.setText("");
        txtFone.setText("");
        txtLogradouro.setText(null);
        search_cpf();
        btnCreate.setEnabled(true);
        btnUpdate.setEnabled(false);
        btnDelete.setEnabled(false);
    }

    private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
        GEN-FIRST:event_btnCreateActionPerformed
        if (((txtCpf.getText().equals("")) || (txtNome.getText().equals("")) ||
            (txtSobrenome.getText().equals("")) || (txtEmail.getText().equals("")) ||
            (txtLogradouro.getText().equals("")))) {
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios.");
        } else {
            String id, cpf, nome, sobrenome, email, fone, logradouro;
            cpf = txtCpf.getText();
            nome = txtNome.getText();
            sobrenome = txtSobrenome.getText();
            email = txtEmail.getText();
            fone = txtFone.getText();
            logradouro = txtLogradouro.getText();

            dto.setCpf(cpf);
            dto.setNome(nome);
            dto.setSobrenome(sobrenome);
            dto.setEmail(email);
            dto.setFone(fone);
            dto.setLogradouro(logradouro);

            dao.insert(dto);
            clean();
        }
        //GEN-LAST:event_btnCreateActionPerformed

        private void txtNomeKeyReleased(java.awt.event.KeyEvent evt) {
            GEN-FIRST:event_txtNomeKeyReleased
            // "btn"Função do mouse / txtNome / Eventos / Key / KeyReleased.
            search_nome();
        }
        //GEN-LAST:event_txtNomeKeyReleased

        private void txtCpfKeyReleased(java.awt.event.KeyEvent evt) {
            GEN-FIRST:event_txtCpfKeyReleased
            // "btn"Função do mouse / txtCpf / Eventos / Key / KeyReleased.
            search_cpf();
        }
        //GEN-LAST:event_txtCpfKeyReleased

        private void jTableMouseClicked(java.awt.event.MouseEvent evt) {
            GEN-FIRST:event_jTableMouseClicked
            // "btn"Função do mouse / jTable / Eventos / Mouse / MouseClicked.
            setText();
            btnUpdate.setEnabled(true);
            btnDelete.setEnabled(true);
        }
        //GEN-LAST:event_jTableMouseClicked

        private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
            GEN-FIRST:event_btnUpdateActionPerformed
            if (((txtCpf.getText().equals("")) || (txtNome.getText().equals("")) ||
                (txtSobrenome.getText().equals("")) || (txtEmail.getText().equals("")) ||
                (txtLogradouro.getText().equals("")) || (txtFone.getText().equals("")))) {
                JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios.");
            } else {
                dto.setId(Integer.parseInt(txtId.getText()));
                dto.setCpf(txtCpf.getText());
                dto.setNome(txtNome.getText());
                dto.setSobrenome(txtSobrenome.getText());
                dto.setEmail(txtEmail.getText());
                dto.setFone(txtFone.getText());
                dto.setLogradouro(txtLogradouro.getText());

                dao.update(dto);
                clean();
            }
            //GEN-LAST:event_btnUpdateActionPerformed

        private void btnCleanActionPerformed(java.awt.event.ActionEvent evt) {
            GEN-FIRST:event_btnCleanActionPerformed
            clean();
        }
        //GEN-LAST:event_btnCleanActionPerformed

        private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
            GEN-FIRST:event_btnDeleteActionPerformed
            int conf = JOptionPane.showConfirmDialog(null, "dta!-Excluir <strong>"+
                txtNome.getText()+"</strong> do cadastro?</strong>","ATENÇÃO",JOptionPane.YES_NO_OPTION);
            if (conf == JOptionPane.YES_OPTION) {
                String id = txtId.getText();

                dao.delete(Integer.parseInt(id));
                clean();
            }
            //GEN-LAST:event_btnDeleteActionPerformed

        /**
         * @param args the command line arguments
         */
        public static void main(String args[]) {
            //Set the Nimbus look and feel
            //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
            // If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
            // For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        }
    }

```

```

package dto;

public class Subject {
    private int id;
    private String cpf,nome,sobrenome,email,fone,logradouro;

    public Subject(){
    }

    public Subject(int id,String cpf, String nome, String sobrenome,
        String email, String fone, String logradouro){
        this.setId(id);
        this.cpf = cpf;
        this.nome = nome;
        this.sobrenome = sobrenome;
        this.email = email;
        this.setFone(fone);
        this.setLogradouro(logradouro);
    }

    @Override
    public String toString(){//Método não implementado. Apenas para conferência.
        return "ID="+this.getId()+" CPF: "+this.cpf+" Nome: "+this.nome+" "+
            "Sobrenome: "+this.sobrenome+" E-mail: "+this.getEmail()+" "+
            "Fone "+this.getFone()+"-VnLogradouro: "+this.getLogradouro()+"\n";
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getId() {
        return this.id;
    }

    public void setCpf(String cpf){
        this.cpf = cpf;
    }

    public String getCpf(){
        return this.cpf;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getNome() {
        return this.nome;
    }

    public void setSobrenome(String sobrenome) {
        this.sobrenome = sobrenome;
    }

    public String getSobrenome() {
        return this.sobrenome;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getFone() {
        return fone;
    }

    public void setFone(String fone) {
        this.fone = fone;
    }

    public String getLogradouro() {
        return logradouro;
    }

    public void setLogradouro(String logradouro) {
        this.logradouro = logradouro;
    }
}

package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import dto.Subject;
import java.util.ArrayList;

/**
 *
 * @author Sérgio Felipe Starke
 */
public class Connect {
    //1º Atribuir Classes, de conexão e de preparo de declaração.
    private Connection conn;
    private PreparedStatement ps;

    //2º Criar método de conexão com o Driver da biblioteca e localhost.
    //Este método em particular foi criado pensando no uso e jTable na classe Main da package view.
    public Connection getConnection(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/cadastro_march_2022","root",""
            );
        } catch (ClassNotFoundException | SQLException e) {
            JOptionPane.showMessageDialog(null, "ERRO NA CONEXÃO...");
            return null;
        }
    }

    //3º Criar métodos CreateReadUpdateDelete.
    public boolean insert(Subject get) {
        try {
            ps = conn.prepareStatement(
                "insert into pessoa(cpf,nome,sobrenome,email,fone,logradouro) values(?,?,?,?,?,?)"
            );
            ps.setString(1, get.getCpf());
            ps.setString(2, get.getNome());
            ps.setString(3, get.getSobrenome());
            ps.setString(4, get.getEmail());
            ps.setString(5, get.getFone());
            ps.setString(6, get.getLogradouro());
            ps.execute();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "ERRO AO CRIAR CONTACTO...<strong>Confira redundância no nº de CPF.");
            return false;
        }
    }

    public ArrayList<Subject> select() {
        //ATENÇÃO: Este método não será implementado. Não obstante,
        //permanece para simples conferência em sua lógica de sintaxe.
        ArrayList<Subject> list = new ArrayList();
        try {
            ps = conn.prepareStatement("select from pessoa");
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                Subject pessoa = new Subject(
                    rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),
                    rs.getString(5),rs.getString(6),rs.getString(7)
                );
                list.add(pessoa);
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "ERRO AO LISTAR CONTACTO...");
        }
        return list;
    }

    public boolean update(Subject get) {
        try {
            ps = conn.prepareStatement(
                "update pessoa set cpf = ?,nome = ?,sobrenome = ?,email = ?, "+
                "fone = ?,logradouro = ? where id = ?"
            );
            ps.setString(1, get.getCpf());
            ps.setString(2, get.getNome());
            ps.setString(3, get.getSobrenome());
            ps.setString(4, get.getEmail());
            ps.setString(5, get.getFone());
            ps.setString(6, get.getLogradouro());
            ps.setInt(7, get.getId());
            ps.execute();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "ERRO NA ATUALIZAÇÃO DO CONTACTO...");
            return false;
        }
    }

    public boolean delete(int numero){
        try {
            ps = conn.prepareStatement("delete from pessoa where id = ?");
            ps.setInt(1, numero);
            ps.execute();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "ERRO EM EXCLUIR CONTACTO...");
            return false;
        }
    }
}

```