

Ôn

- String to Integer sử dụng `Integer.parseInt()`
- String to Integer sử dụng `Integer.valueOf()`
- Tương tự cho các kiểu dữ liệu khác...

Tạo CSDL trong XAMPP

use testjava;

Create Table NhanSu(

MANS varchar(5) not null Primary Key,

HOTEN varchar(40),

PHAI varchar(3),

THAMNIEN int,

MUCLUONG double (10,2));

use testjava;

insert into nhansu values ('10001','Trần Văn Tài', 'Nam', 10, 1200);

insert into nhansu values ('10002','Nguyễn Văn Đức', 'Nam', 13, 1500);

insert into nhansu values ('10003','Lê Thị Hiền', 'Nữ', 5, 900);

insert into nhansu values ('10004','Lê Văn Năm', 'Nam', 3, 600);

insert into nhansu values ('10005','Ngô Thị Bé Ba', 'Nữ', 6, 800);

insert into nhansu values ('10006','Phạm Văn Nhân', 'Nam', 5, 750);

...

use testjava;

SELECT * FROM nhansu;

MANS	HOTEN	PHAI	THAMNIEN	MUCLUONG
10001	Trần Văn Tài	Nam	10	1200.00
10002	Nguyễn Văn Đức	Nam	13	1500.00
10003	Lê Thị Hiền	Nữ	5	900.00
10004	Lê Văn Năm	Nam	3	600.00
10005	Ngô Thị Bé Ba	Nữ	6	800.00
10006	Phạm Văn Nhân	Nam	5	750.00

THIẾT KẾ ỨNG DỤNG THEO LAYER

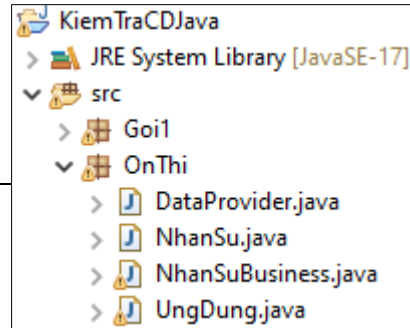
1/. Tạo Project, tạo Package

2/. Tạo Class **DataProvider**:

```
package OnThi;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DataProvider
{
    static Connection conn = null;
    public static Connection connectDB(String username, String password)
    { //phương thức kết nối với MySQL và trả về kết nối
        if (conn == null) {
            try
            {
                conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testjava",username,
password);
            }
            catch (SQLException e)
            {
                e.printStackTrace();
            }
        }
        return conn;
    }

    public static void disconnect()
    { //phương thức hủy kết nối
        if (conn != null) {
            try {
                conn.close();
                conn = null;
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```



3/. Tạo Class **NhanSu**:

```
package OnThi;
public class NhanSu
{
    private String maNS;
    private String hoTen;
    private String phai;
    private int thamNien;
    private double muLuong;
```

```
public NhanSu(String maNS, String hoTen, String phai, int thamNien,
double mucLuong)
{
    this.maNS = maNS; this.hoTen = hoTen; this.phai = phai;
    this.thamNien = thamNien; this.mucluong = mucLuong;
}
public NhanSu()
{}
// Viết các setter
public void setmaNS(String maNS)
{
    this.maNS = maNS;
}

public void sethoTen(String hoTen)
{
    this.hoTen = hoTen;
}

public void setphai(String phai)
{
    this.phai = phai;
}

public void setthamNien(int thamNien)
{
    this.thamNien = thamNien;
}

public void setmucLuong(double mucLuong)
{
    this.mucluong = mucLuong;
}
// viết các getter
public String getmaNS()
{
    return this.maNS;
}

public String gethoTen()
{
    return this.hoTen;
}

public String getphai()
{
    return this.phai;
}

public int getthamNien()
{
    return this.thamNien;
}

public double getmucLuong()
```

```

    {
        return this.muLuong;
    }
}

```

4/. Tạo Class NhanSuBusiness:

```

package OnThi;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
public class NhanSuBusiness
{
    static Connection conn = null;
    static List<NhanSu> ds;
    public static List<NhanSu> layDanhSach() throws SQLException
    {
        List<NhanSu> danhSach = new ArrayList();
        try {
            conn = DataProvider.connectDB("root", "");
            String sql = "select * from nhansu";
            Statement lenh = conn.createStatement();
            ResultSet ketqua = lenh.executeQuery(sql);
            NhanSu obj = null; //khai báo đối tượng nhân sự
            while (ketqua.next())
            {
                obj = new NhanSu(); // tạo đối tượng nhân sự mới
                obj.setmaNS(ketqua.getString("MANS"));
                obj.sethoTen(ketqua.getString("HOTEN"));
                obj.setphai(ketqua.getString("PHAI"));
                obj.setthamNien(ketqua.getInt("THAMNIEN"));
                obj.setmuLuong(ketqua.getDouble("MUCLUONG"));
                danhSach.add(obj); // thêm đối tượng vào danh sách
            }
        }
        catch (SQLException e)
        {
            System.out.print("Lỗi " + e.getMessage());
        }
        finally {
            DataProvider.disconnect();
        }
        return danhSach;
    }

    public static void addData(String maNS, String hoTen, String phai, int
    thamNien, double muLuong)
    {
        try {
            conn = DataProvider.connectDB("root", "");
            String sql = "insert into nhansu values ('" + maNS + "', '"
+ hoTen + "', '" + phai + "', '" + thamNien + "', '" + muLuong + "');";
            Statement stmt = conn.createStatement();
            int i = stmt.executeUpdate(sql);

```

```

        if (i>0)
            JOptionPane.showMessageDialog(null, "Record thêm
xong.");
        else
            JOptionPane.showMessageDialog(null, "Lỗi, Record
không được thêm!");
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
    }
    finally {
        DataProvider.disconnect();
    }
}

public static void deleteData(String maNS) throws SQLException
{
    conn = DataProvider.connectDB("root", "");
    String sql = "delete from nhansu where (MANS='" + maNS + "')";

    Statement stmt = conn.createStatement();
    int i = stmt.executeUpdate(sql);
    if (i>0)
        JOptionPane.showMessageDialog(null, "Record đã được
xóa.");
    else
        JOptionPane.showMessageDialog(null, "Lỗi, Record không xóa
được!");
}

public static void updateData(String maNS, String hoTen, String phai, int
thamNien, double mucLuong)
{
    try
    {
        conn = DataProvider.connectDB("root", "");
        String sql = "update nhansu set HOTEN='" + hoTen + "', PHAI='"
+ phai + "', THAMNIEN=" + thamNien + ", MUCLUONG=" + mucLuong
+ "where (MANS='" + maNS + "')";
        Statement stmt = conn.createStatement();
        int i = stmt.executeUpdate(sql);
        if (i>0)
            JOptionPane.showMessageDialog(null, "Record đã được
sửa.");
        else
            JOptionPane.showMessageDialog(null, "Lỗi, Record không
được sửa!");
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
    }
    finally {
        DataProvider.disconnect();
    }
}

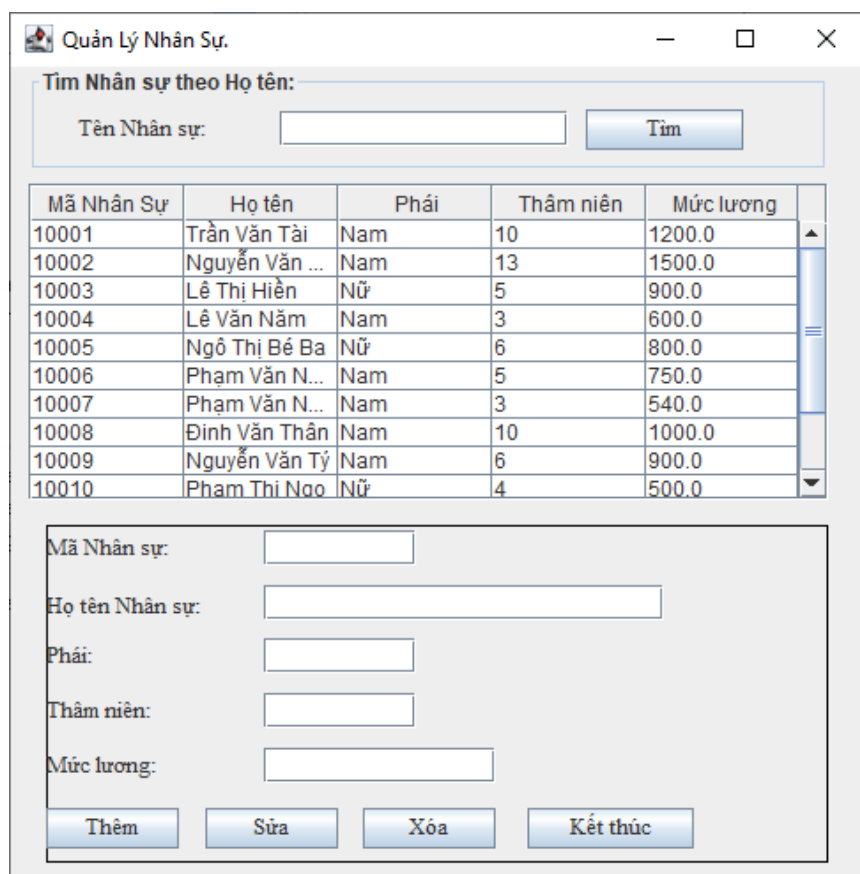
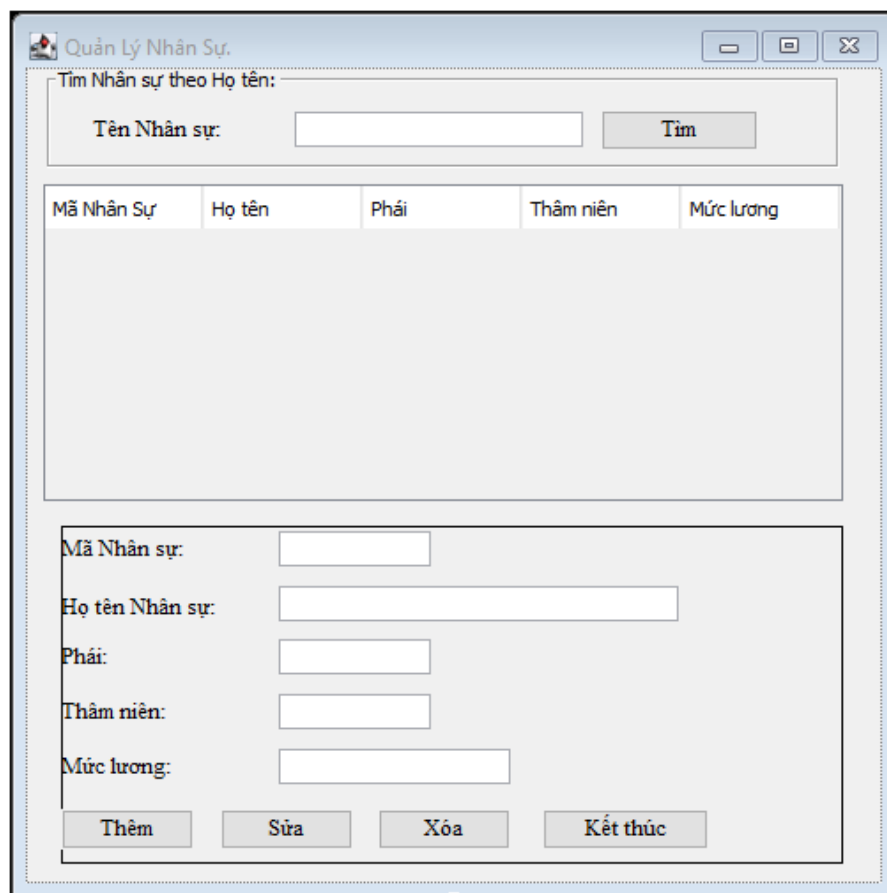
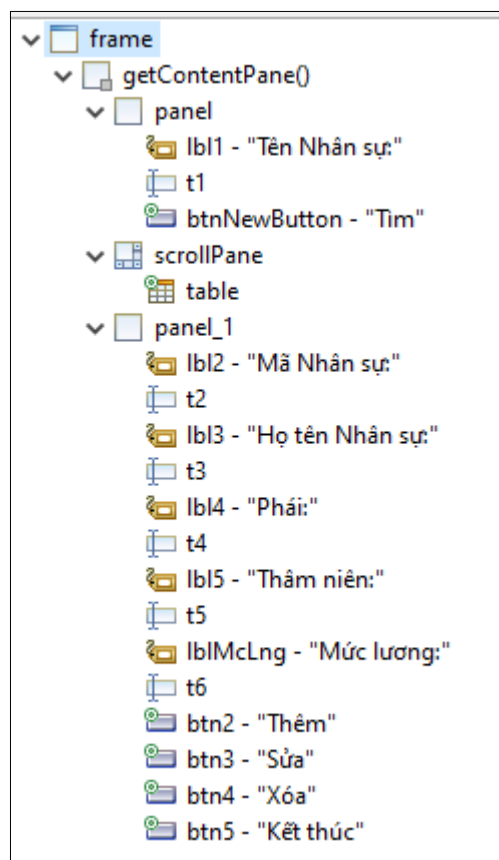
```

```

    }
}
}

```

5/. Tạo Class **UngDung** và thiết kế chương trình:



```

package OnThi;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import java.awt.*;
import java.awt.EventQueue;
import javax.swing.*;
import javax.swing.border.LineBorder;
import javax.swing.border.TitledBorder;
import javax.swing.table.DefaultTableModel;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class UngDung
{
    private JFrame frame;
    private JTextField t1;
    private static JTable table;
    private JTextField t2;
    private JTextField t3;
    private JTextField t4;
    private JTextField t5;
    private JTextField t6;
    private static DefaultTableModel model;
    static List<NhanSu> danhSach = new ArrayList();

    public static void showData() throws SQLException
    {
        try {
            danhSach = NhanSuBusiness.LayDanhSach();
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        table.setRowSorter(null);
        table.setModel(new DefaultTableModel());
        //Khai báo cột tiêu đề bảng
        String cotTieuDe[] = {"Mã Nhân Sự", "Họ tên", "Phái", "Thâm niên",
"Mức lương"};
        //Khai báo Model chứa dữ liệu
        model = new DefaultTableModel(cotTieuDe, 0);
        Object[] row;
        for(NhanSu ns: danhSach)
        {
            row = new Object[5];
            row[0] = ns.getmaNS();
            row[1] = ns.getthoTen();
            row[2] = ns.getphai();
            row[3] = ns.getthamNien();
            row[4] = ns.getmucLuong();
            model.addRow(row);
        }
    }
}

```

```

    }
    table.setModel(model);
}

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                UngDung window = new UngDung();
                window.frame.setVisible(true);
                showData();

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public UngDung() {
    initialize();
}

private void initialize() {
    frame = new JFrame("Quản Lý Nhân Sự.");
    frame.setBounds(100, 100, 500, 500);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    panel.setBorder(new TitledBorder(null, "T\u00ECm Nh\u00E0n s\u1EF1 theo H\u1EC7 t\u00E0n:", TitledBorder.LEADING, TitledBorder.TOP, null, null));
    panel.setBounds(10, 0, 452, 59);
    frame.getContentPane().add(panel);
    panel.setLayout(null);

    JLabel lbl1 = new JLabel("Tên Nhân sự:");
    lbl1.setBounds(28, 22, 104, 26);
    panel.add(lbl1);
    lbl1.setFont(new Font("Times New Roman", Font.PLAIN, 13));

    t1 = new JTextField();
    t1.setBounds(142, 25, 163, 20);
    panel.add(t1);
    t1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    t1.setColumns(10);

    JButton btnNewButton = new JButton("Tìm");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
            String value = t1.getText();
            for (int row = 0; row <= table.getRowCount() - 1; row++)
            {

```



```

        if (value.equalsIgnoreCase((String)
table.getValueAt(row, 1)))
        {
            table.scrollRectToVisible(table.getCellRect(row,
1, true));

            table.setRowSelectionInterval(row, row);
            for (int i = 0; i <= table.getColumnCount() - 1;
i++)
            {
                table.getColumnModel().getColumn(i);
                t1.setText(null); t1.requestFocus();
            }
        }
    }
});
btnNewButton.setBounds(315, 24, 89, 23);
panel.add(btnNewButton);
btnNewButton.setFont(new Font("Times New Roman", Font.PLAIN, 13));

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 66, 452, 179);
frame.getContentPane().add(scrollPane);

table = new JTable();
scrollPane.setViewportViewView(table);
table.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        int index = table.getSelectedRow();
        t2.setText(model.getValueAt(index, 0).toString());
        t3.setText(model.getValueAt(index, 1).toString());
        t4.setText(model.getValueAt(index, 2).toString());
        t5.setText(model.getValueAt(index, 3).toString());
        t6.setText(model.getValueAt(index, 4).toString());
    }
});

JPanel panel_1 = new JPanel();
panel_1.setAlignmentX(Component.RIGHT_ALIGNMENT);
panel_1.setBorder(new LineBorder(new Color(0, 0, 0)));
panel_1.setBounds(20, 259, 442, 191);
frame.getContentPane().add(panel_1);
panel_1.setLayout(null);

JLabel lbl2 = new JLabel("Mã Nhân sự:");
lbl2.setBounds(0, 0, 104, 26);
panel_1.add(lbl2);
lbl2.setFont(new Font("Times New Roman", Font.PLAIN, 13));

t2 = new JTextField();
t2.setBounds(123, 3, 86, 20);

```

```
panel_1.add(t2);
t2.setFont(new Font("Times New Roman", Font.PLAIN, 13));
t2.setColumns(10);

JLabel lbl3 = new JLabel("Họ tên Nhân sự:");
lbl3.setBounds(0, 33, 104, 26);
panel_1.add(lbl3);
lbl3.setFont(new Font("Times New Roman", Font.PLAIN, 13));

t3 = new JTextField();
t3.setBounds(123, 34, 226, 20);
panel_1.add(t3);
t3.setFont(new Font("Times New Roman", Font.PLAIN, 13));
t3.setColumns(10);

JLabel lbl4 = new JLabel("Phái:");
lbl4.setBounds(0, 61, 104, 26);
panel_1.add(lbl4);
lbl4.setFont(new Font("Times New Roman", Font.PLAIN, 13));

t4 = new JTextField();
t4.setBounds(123, 64, 86, 20);
panel_1.add(t4);
t4.setFont(new Font("Times New Roman", Font.PLAIN, 13));
t4.setColumns(10);

JLabel lbl5 = new JLabel("Thâm niên:");
lbl5.setBounds(0, 92, 104, 26);
panel_1.add(lbl5);
lbl5.setFont(new Font("Times New Roman", Font.PLAIN, 13));

t5 = new JTextField();
t5.setBounds(123, 95, 86, 20);
panel_1.add(t5);
t5.setFont(new Font("Times New Roman", Font.PLAIN, 13));
t5.setColumns(10);

JLabel lblMcLng = new JLabel("Mức lương:");
lblMcLng.setBounds(0, 123, 104, 26);
panel_1.add(lblMcLng);
lblMcLng.setFont(new Font("Times New Roman", Font.PLAIN, 13));

t6 = new JTextField();
t6.setBounds(123, 126, 131, 20);
panel_1.add(t6);
t6.setFont(new Font("Times New Roman", Font.PLAIN, 13));
t6.setColumns(10);

JButton btn2 = new JButton("Thêm"); //nút thêm
btn2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        String maNV = t2.getText();
        String hoTen = t3.getText();
        String phai = t4.getText();
```

```

        int thamNien = Integer.valueOf(t5.getText());
        double mucLuong = Double.valueOf(t6.getText());
        if (maNV == null || maNV.isEmpty() || maNV.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập Mã Nhân
sự");

            t2.requestFocus();
            return;
        }
        if (hoTen == null || hoTen.isEmpty() ||
hoTen.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập Họ tên");
            t3.requestFocus();
            return;
        }
        if (phai == null || phai.isEmpty() || phai.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập phái");
            t4.requestFocus();
            return;
        }
        if (thamNien==0)
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập thâm niên");
            t5.requestFocus();
            return;
        }
        if (mucLuong==0.0)
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập mức lương");
            t5.requestFocus();
            return;
        }
        NhanSuBusiness.addData(maNV, hoTen, phai, thamNien,
mucLuong);

        try {
            showData();
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
});
btn2.setBounds(0, 160, 75, 23);
panel_1.add(btn2);
btn2.setFont(new Font("Times New Roman", Font.PLAIN, 13));

JButton btn3 = new JButton("Sửa"); //nút sửa
btn3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        String maNV = t2.getText();
        String hoTen = t3.getText();
        String phai = t4.getText();

```

```

        int thamNien = Integer.valueOf(t5.getText());
        double mucLuong = Double.valueOf(t6.getText());
        if (maNV == null || maNV.isEmpty() || maNV.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập Mã Nhân
sự");

            t2.requestFocus();
            return;
        }
        if (hoTen == null || hoTen.isEmpty() ||
hoTen.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập Họ tên");
            t3.requestFocus();
            return;
        }
        if (phai == null || phai.isEmpty() || phai.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập phái");
            t4.requestFocus();
            return;
        }
        if (thamNien==0)
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập thâm niên");
            t5.requestFocus();
            return;
        }
        if (mucLuong==0.0)
        {
            JOptionPane.showMessageDialog(null, "Hãy nhập mức lương");
            t5.requestFocus();
            return;
        }
        NhanSuBusiness.updateData(maNV, hoTen, phai, thamNien,
mucLuong);

        try {
            showData();
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
});
btn3.setBounds(90, 160, 75, 23);
panel_1.add(btn3);
btn3.setFont(new Font("Times New Roman", Font.PLAIN, 13));

JButton btn4 = new JButton("Xóa"); //nút xóa
btn4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        String maNS = t2.getText();
        if (!t2.getText().isEmpty())
        {

```

```

        int result = JOptionPane.showConfirmDialog(null, "Có
        chắc xóa không?", "Xóa Record này.", JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE);
        if (result == 0)
            try {
                NhanSuBusiness.deleteData(maNS);
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
    }

    try {
        showData();
    } catch (SQLException e1) {
        e1.printStackTrace();
    }
}

});
btn4.setBounds(179, 160, 75, 23);
panel_1.add(btn4);
btn4.setFont(new Font("Times New Roman", Font.PLAIN, 13));

JButton btn5 = new JButton("Kết thúc"); //nút kết thúc
btn5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        int response = JOptionPane.showConfirmDialog(null,
        "Bạn có chắc chắn muốn tiếp tục không?",
        "Xác nhận kết thúc chương trình.",
        JOptionPane.YES_NO_OPTION);
        if (response==0)
            System.exit(0);
    }
});
btn5.setBounds(272, 160, 94, 23);
panel_1.add(btn5);
btn5.setFont(new Font("Times New Roman", Font.PLAIN, 13));

}
}

```

-----Hết-----