**Thiết Kế From Đơn Giản**

public class MyMain {

public static void main(String[] args) {

new MyFrame().setVisible(true);

}

}

public class MyFrame extends JFrame implements MouseListener {

JLabel jlbSoA;

JLabel jlbSoB;

JTextField jtfSoA;

JTextField jtfSoB;

JLabel jlbKetQua;

JTextField jtfKetQua;

JButton jbtKetQua;

public MyFrame() {

initData();

initEvent();

initUI();

makeTable();

makeRadioButton\_Group();

makeComboBox();

initDataTable();

}

private void initData() {

//setSize(rộng, cao);

setSize(300, 500);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setTitle("Sang Đẹp Trai");

setLayout(null);

//jlbSoA = new JLabel("nội dung hiển thị");

jlbSoA = new JLabel("Số A");

//setBounds(cách trái, cách trên, rộng, cao);

jlbSoA.setBounds(50, 50, 50, 30);

jtfSoA = new JTextField();

jtfSoA.setBounds(100, 50, 100, 30);

jlbSoB = new JLabel("Số B");

jlbSoB.setBounds(50, 100, 50, 30);

jtfSoB = new JTextField();

jtfSoB.setBounds(100, 100, 100, 30);

jlbKetQua = new JLabel("Kết Quả");

jlbKetQua.setBounds(50, 150, 50, 30);

jtfKetQua = new JTextField("Kết Quả");

jtfKetQua.setBounds(100, 150, 100, 30);

//không cho nhập

jtfKetQua.setEditable(false);

jbtKetQua = new JButton("Kết Quả (A + B)");

//tên sữa dụng trong function mouseClicked

jbtKetQua.setName("ketQua");

jbtKetQua.setBounds(50, 200, 150, 30);

}

private void initUI() {

add(jlbSoA);

add(jtfSoA);

add(jlbSoB);

add(jtfSoB);

add(jlbKetQua);

add(jtfKetQua);

add(jbtKetQua);

}

private void initEvent() {

jbtKetQua.addMouseListener(this); //xử lý sự kiện trong function mouseClicked

//cách khác

jbtKetQua.addActionListener(new java.awt.event.ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

//xử lý sự kiện khi nhấn button kết quả

System.err.println("sang đẹp trai nhất");

}

});

}

@Override

public void mouseClicked(MouseEvent e) {

Component cpn = e.getComponent();

switch (cpn.getName()) {

//tên của button kết quả

case "ketQua": {

//xử lý sự kiện khi nhấn button kết quả

String soA = jtfSoA.getText().toString().trim();

String soB = jtfSoB.getText().toString().trim();

int ketQua = Integer.parseInt(soA) + Integer.parseInt(soB);

jtfKetQua.setText(ketQua + "");

break;

}

default: {

break;

}

}

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

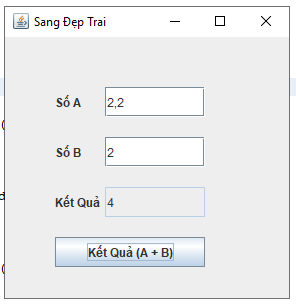
@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

**Một Số Component Khác**

1. ***Table***

private void makeTable() {

JTable jTable = new JTable();

//tạo thanh cuộn và bảng

JScrollPane jScrollPane = new JScrollPane();

jScrollPane.setBounds(50, 250, 200, 150);

jScrollPane.setViewportView(jTable);

add(jScrollPane);

//thêm dữ liệu vào bảng

String[] header = {"tiêu đề cột 1", "tiêu đề cột 2", "..."};

DefaultTableModel defaultTableModel = new DefaultTableModel(header, 0);

for (int i = 0; i < 10; i++) {

Object[] row = {"dữ liệu dòng 1 cột i", "dữ liệu dòng 2 cột i", "dữ liệu dòng ... cột i"};

defaultTableModel.addRow(row);

}

jTable.setModel(defaultTableModel);

//lấy số cột

int soCot = defaultTableModel.getColumnCount();

//lấy số dòng

int soDong = defaultTableModel.getRowCount();

//lấy dữ liệu

for (int i = 0; i < defaultTableModel.getRowCount(); i++) {

for (int j = 0; j < defaultTableModel.getColumnCount(); j++) {

if (defaultTableModel.getValueAt(i, j) == null) {

String dataDongICotJ = defaultTableModel.getValueAt(i, j).toString();

}

}

}

//sự kiện chọn dòng

jTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

try {

int dongDuocChon = jTable.getSelectedRow();

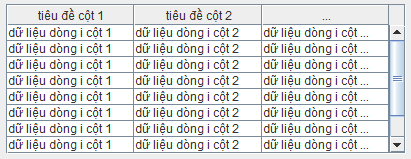
} catch (Exception e) {

}

}

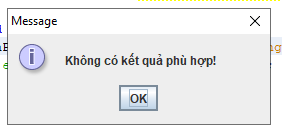
});

}



1. ***MessageDialog***

JOptionPane.showMessageDialog(null, "Không có kết quả phù hợp!");

****

1. ***RadioButton vs ButtonGroup***

private void makeRadioButton\_Group() {

ButtonGroup btg = new ButtonGroup();

JRadioButton jrbNam = new JRadioButton("Nam");

jrbNam.setBounds(50, 250, 100, 30);

JRadioButton jrbNu = new JRadioButton("Nữ");

jrbNu.setBounds(50, 280, 100, 30);

//tạo group cho 2 radiobutton

btg.add(jrbNam);

btg.add(jrbNu);

add(jrbNam);

add(jrbNu);

//xét mặc định 1 radiobutton được chọn

jrbNam.setSelected(true);

//sự kiện chọn button

jrbNam.addActionListener(new java.awt.event.ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.err.println("chọn nam");

}

});

jrbNu.addActionListener(new java.awt.event.ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.err.println("chọn nữ");

}

});

//kiểm tra radiobutton được chọn

if (jrbNam.isSelected()) {

System.err.println("button nam được chọn");

} else {

System.err.println("button nam không được chọn");

}

//set mặc định chọn button nữ bằng buttongroup

btg.setSelected(jrbNu.getModel(), true);

//kiểm tra group xem button nào đang được chọn

if (btg.getSelection().equals(jrbNam)) {

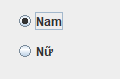
System.out.println("nam được chọn");

} else {

System.out.println("nữ được chọn");

}

}



1. ***ComboBox***

private void makeComboBox() {

JComboBox jComboBox = new JComboBox();

jComboBox.setBounds(50, 250, 100, 30);

add(jComboBox);

DefaultComboBoxModel defaultComboBoxModel;

defaultComboBoxModel = new DefaultComboBoxModel(new String[] { "sang", "đẹp", "trai"});

jComboBox.setModel(defaultComboBoxModel);

//xét mặc định chọn item "đẹp"

jComboBox.getModel().setSelectedItem("đẹp");

//thêm item vào combobox

jComboBox.addItem("nhất");

jComboBox.addItem("quả");

jComboBox.addItem("đất");

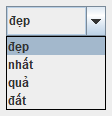
//xóa item

jComboBox.removeItem("trai");

//xóa item theo index

jComboBox.removeItemAt(0);

}



1. ***Đổ dữ liệu từ array lên table***

private void initDataTable() {

//tạo dữ liệu cho mảng

ArrayList<SinhVienDTO> arrayList = new ArrayList<>();

arrayList.add(new SinhVienDTO(1, "sang", true));

arrayList.add(new SinhVienDTO(2, "đẹp", false));

arrayList.add(new SinhVienDTO(3, "trai", true));

arrayList.add(new SinhVienDTO(4, "thanh", true));

arrayList.add(new SinhVienDTO(5, "lịch", false));

arrayList.add(new SinhVienDTO(6, "vô", true));

arrayList.add(new SinhVienDTO(7, "địch", true));

arrayList.add(new SinhVienDTO(8, "vũ", true));

arrayList.add(new SinhVienDTO(9, "trụ", true));

JTable jTable = new JTable();

//tạo thanh cuộn và bảng

JScrollPane jScrollPane = new JScrollPane();

jScrollPane.setBounds(50, 250, 200, 150);

jScrollPane.setViewportView(jTable);

add(jScrollPane);

//thêm dữ liệu vào bảng

String[] header = {"mã", "tên", "giới tính"};

DefaultTableModel defaultTableModel = new DefaultTableModel(header, 0);

for (int i = 0; i < arrayList.size(); i++) {

String gioTinh;

if (arrayList.get(i).getNam()) {

gioTinh = "nam";

} else {

gioTinh = "nữ";

}

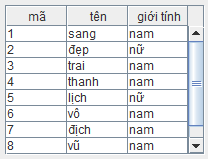
Object[] row = {arrayList.get(i).maSinhVien, arrayList.get(i).getTenSinhVien(), gioTinh};

defaultTableModel.addRow(row);

}

jTable.setModel(defaultTableModel);

}



1. ***Tắt frame, dialog...***

dispose();

**Mô Hình 3 Lớp**

1. ***Connection***

Mặc định đã có chỉ cần lấy ra dùng thôi kkk tên nó là: ConnectionDataBase

1. ***DAO***

public class SinhVienDAO {

public static ArrayList<SinhVienDTO> layDanhSachSinhVien() {

ArrayList<SinhVienDTO> arrayList = new ArrayList<>();

String sql = "select \* from sinhvien";

ConnectionDataBase connectionDataBase = new ConnectionDataBase();

try {

ResultSet resultSet = connectionDataBase.excuteQuery(sql);

while (resultSet.next()) {

SinhVienDTO sinhVienDTO = new SinhVienDTO();

sinhVienDTO.setMaSinhVien(resultSet.getInt("maSV"));

sinhVienDTO.setTenSinhVien(resultSet.getString("tenSV"));

sinhVienDTO.setNam(resultSet.getBoolean("nam"));

//maSV, tenSV, nam: là tên cột trong bảng sinhvien có thể thay thế bằng index của cột 0,1,2

arrayList.add(sinhVienDTO);

}

} catch (SQLException ex) {

//in ra lỗi nếu có, có thể có hoặc k k quan trọng

connectionDataBase.displayError(ex);

} finally {

//đóng kết nối

connectionDataBase.close();

}

return arrayList;

}

public static void themSinhVien(SinhVienDTO sinhVienDTO) {

String sql = "insert into tacgia (maSV, tenSV, nam) values (?, ?, ?)";

ConnectionDataBase connectionDataBase = new ConnectionDataBase();

PreparedStatement preparedStatement;

try {

preparedStatement = connectionDataBase.connection.prepareStatement(sql);

preparedStatement.setInt(1, sinhVienDTO.getMaSinhVien());

preparedStatement.setString(2, sinhVienDTO.getTenSinhVien());

preparedStatement.setBoolean(3, sinhVienDTO.getNam());

preparedStatement.execute();

} catch (SQLException ex) {

connectionDataBase.displayError(ex);

} finally {

connectionDataBase.close();

}

}

public static void xoaSinhVienTheoMa(Integer maSinhVien) {

String query = "delete from tacgia where maSV=?";

ConnectionDataBase connectionDataBase = new ConnectionDataBase();

PreparedStatement preparedStatement;

try {

preparedStatement = connectionDataBase.connection.prepareStatement(query);

preparedStatement.setInt(1, maSinhVien);

preparedStatement.execute();

} catch (SQLException ex) {

connectionDataBase.displayError(ex);

} finally {

connectionDataBase.close();

}

}

public static void suaSinhVien(SinhVienDTO sinhVienDTO) {

String query = "UPDATE tacgia SET tenSV = ?, nam = ? WHERE maSV = ?";

ConnectionDataBase connectionDataBase = new ConnectionDataBase();

PreparedStatement preparedStatement;

try {

preparedStatement = connectionDataBase.connection.prepareStatement(query);

preparedStatement.setString(1, sinhVienDTO.getTenSinhVien());

preparedStatement.setBoolean(2, sinhVienDTO.getNam());

preparedStatement.setInt(3, sinhVienDTO.getMaSinhVien());

preparedStatement.execute();

} catch (SQLException ex) {

connectionDataBase.displayError(ex);

} finally {

connectionDataBase.close();

}

}

}

1. ***BUS***

public class SinhVienBUS {

public static ArrayList<SinhVienDTO> layDanhSachSinhVien() {

return SinhVienDAO.layDanhSachSinhVien();

}

public static void themSinhVien(SinhVienDTO sinhVienDTO) {

SinhVienDAO.themSinhVien(sinhVienDTO);

}

public static void xoaSinhVien(Integer maSinhVien) {

SinhVienDAO.xoaSinhVienTheoMa(maSinhVien);

}

public static void suaSinhVien(SinhVienDTO sinhVienDTO) {

SinhVienDAO.suaSinhVien(sinhVienDTO);

}

}

1. ***DTO***

public class SinhVienDTO {

Integer maSinhVien;

String tenSinhVien;

Boolean nam; //true là nam false là nữ

public SinhVienDTO(Integer maSinhVien, String tenSinhVien, Boolean nam) {

this.maSinhVien = maSinhVien;

this.tenSinhVien = tenSinhVien;

this.nam = nam;

}

public SinhVienDTO() {

}

public Integer getMaSinhVien() {

return maSinhVien;

}

public void setMaSinhVien(Integer maSinhVien) {

this.maSinhVien = maSinhVien;

}

public String getTenSinhVien() {

return tenSinhVien;

}

public void setTenSinhVien(String tenSinhVien) {

this.tenSinhVien = tenSinhVien;

}

public Boolean getNam() {

return nam;

}

public void setNam(Boolean nam) {

this.nam = nam;

}

}

1. ***GUI***

Từ from đơn giản ở trên vs các componet khác chúng ta đễ dàng xây dựng from theo yêu cầu!!!

public class SinhVienGUI {

public SinhVienGUI() {

initData();

initEvent();

initUI();

}

private void initData() {

ArrayList<SinhVienDTO> arrayList = new ArrayList<>();

arrayList = SinhVienBUS.layDanhSachSinhVien();

//đổ lên tabel

}

private void initEvent() {

//bắt sự kiên của componet và xử lý yêu cầu

SinhVienBUS.themSinhVien("Sinh viên cần thêm");

SinhVienBUS.xoaSinhVien("mã sinh viên kkkkkk");

SinhVienBUS.suaSinhVien("sinh viên cần sủakk");

}

private void initUI() {

}

}