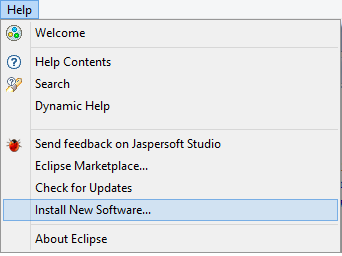
SETUP SWING TRONG ECLIPSE

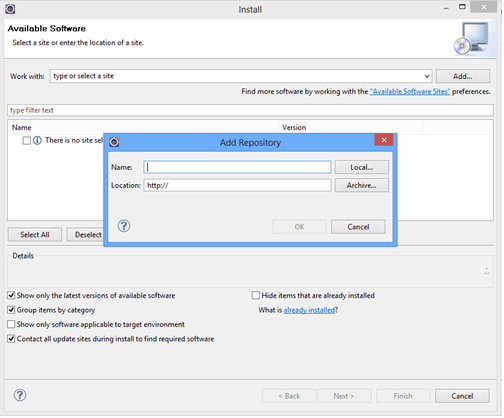
**Bước 1**: Khởi động Eclipse lên và vào phần Help => Install New Software.



**Bước 2**: Hộp thoại hiện lên, nhấn add ở phần Work With.

Name : điền tên bất kỳ

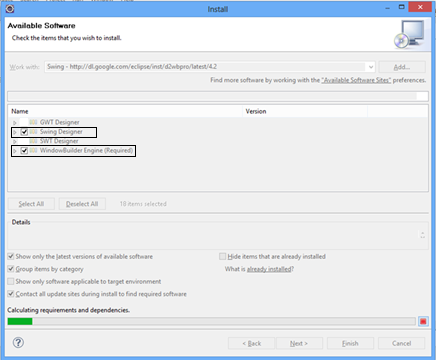
Location: copy link [**http://dl.google.com/eclipse/inst/d2wbpro/latest/4.2**](http://dl.google.com/eclipse/inst/d2wbpro/latest/4.2) và dán vào



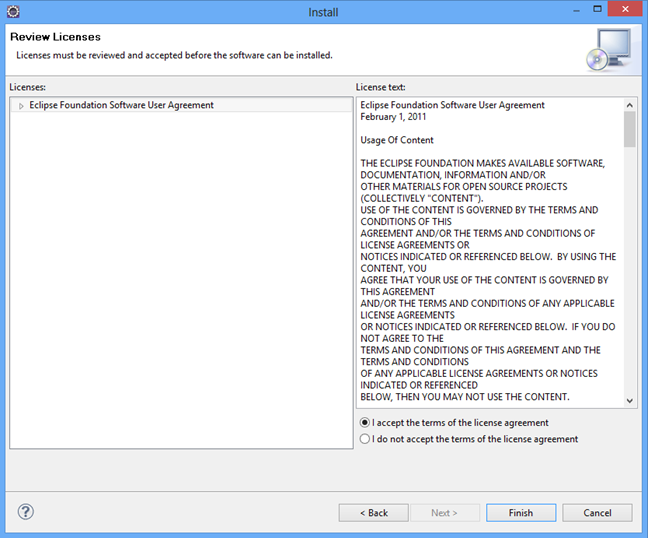
**Bước 3**: Chọn API :

* Swing Designer
* Window Builder (Required)

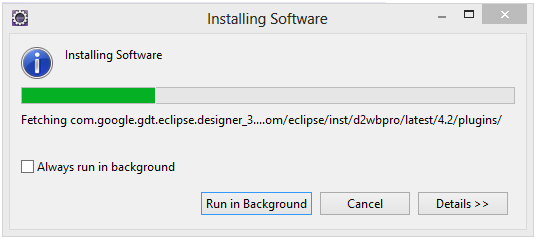
Sau đó nhấn next



**Bước 4**: Sau đó các bạn chuyển tiếp Next => Chọn I accept the terms of the license agreement

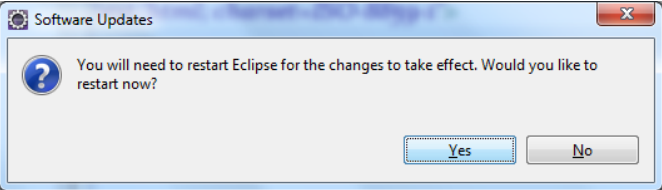


* Phần mềm sẽ tiếp tục cài đặt



* Nếu có cảnh báo nhấn OK

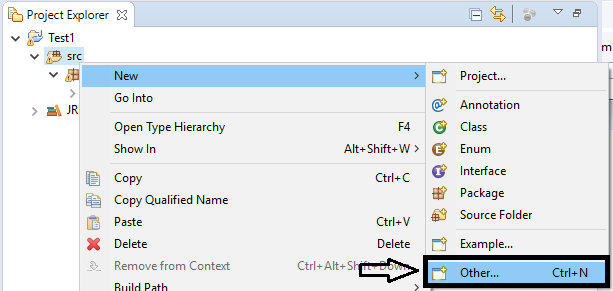
**Bước 5**: Sau khi cài đặt xong các bạn nhấn Yes để khởi động lại eclipse



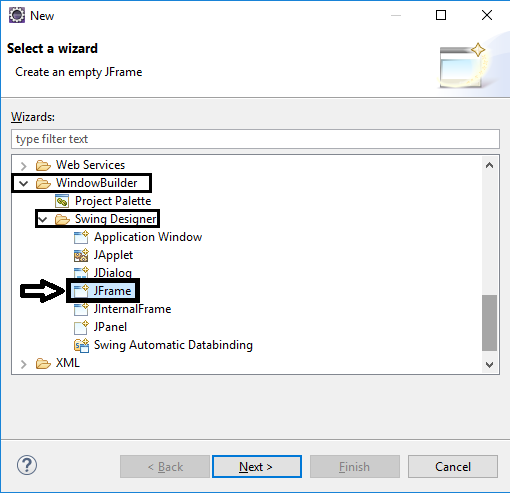
**Bước 6** : Tạo mới Project

1. New Project

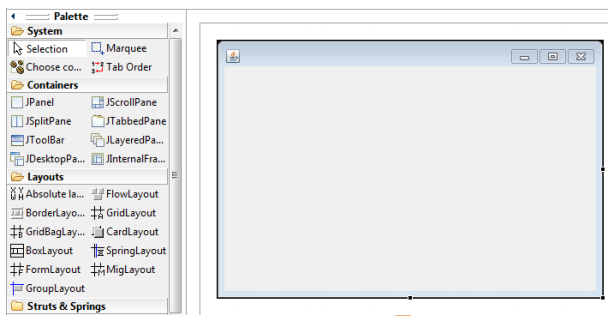
2. Right click thư mục src / New / Other



**Bước 7** : Chọn JFarme trong mục Swing Designer



 Và đây là cửa sổ Swing trong eclipse

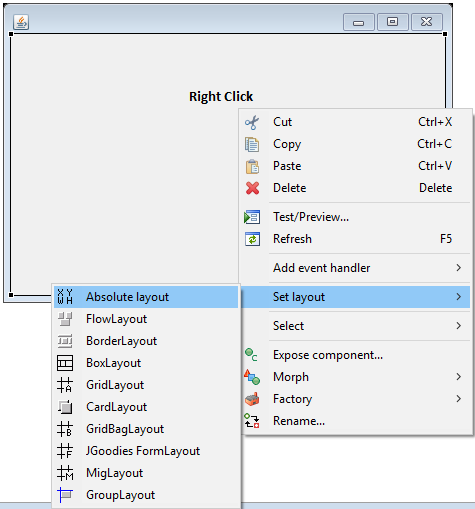


JAVA SWING - COMPONENTS (1)

# Lý thuyết

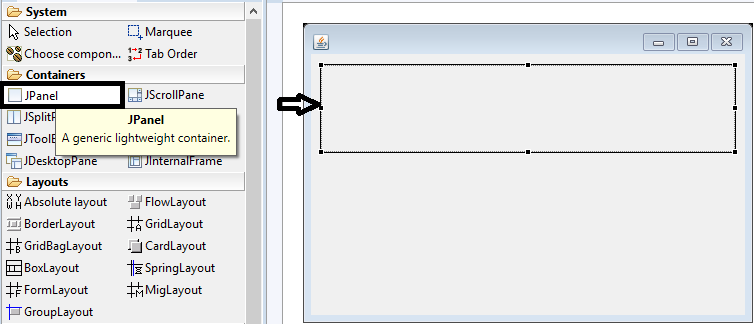
# Set Layout

Chọn Layout : Absolute layout để dễ dàng bố trí components nằm bất kỳ vị trí nào

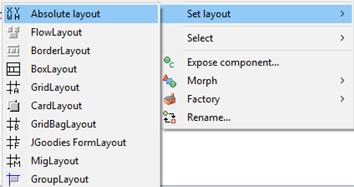


# JPanel

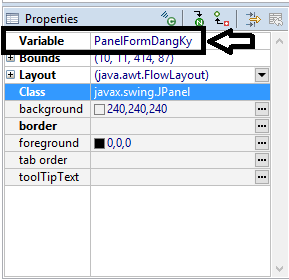
**Bước 1:** Thả component JPanel vào màn hình design



**Bước 2:** RightClick / Set Layout cho JPanel là Absolute layout

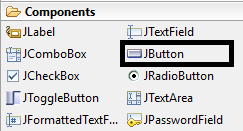


**Bước 3:** Đặt tên cho component



# JButton

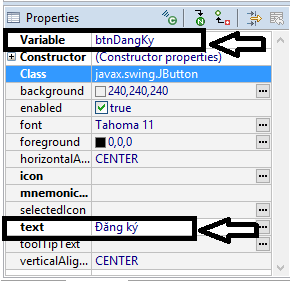
**Bước 1:** Thả component JButton vào màn hình design



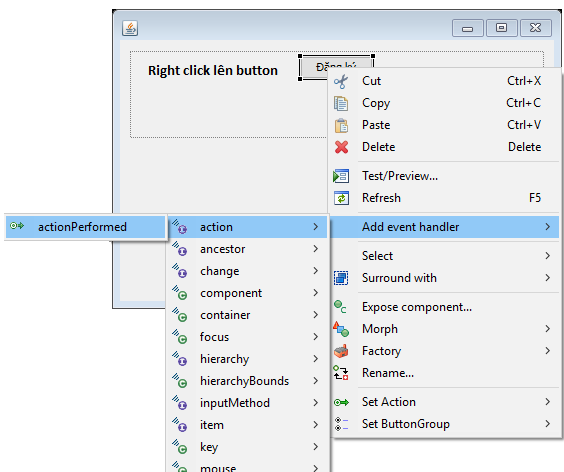
**Bước 2:** Thay đổi nội dung trên component

Variable: Tên của component

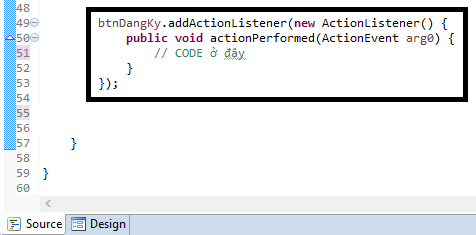
Text : Nội dung sẽ được thể hiện trên nút



**Bước 3:** Đặt sự kiện cho component



**Bước 4:** Bật sang tab Source sẽ thầy hiện ra phần code cho sự kiện khi tương tác vào button



**Bước 5** : Đưa vào đoạn code sau :

btnDangKy.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

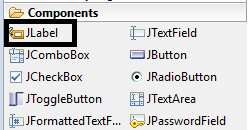
***JOptionPane.showMessageDialog(getParent(), "Goodmorning","Thông báo",JOptionPane.INFORMATION\_MESSAGE);***

}

});

# JLabel

**Bước 1:** Thả component JLabel vào màn hình design

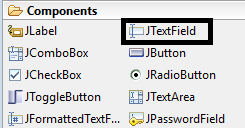


**Bước 2**: Gán nội dung vào cho Label

lblKq.setText("Bạn chưa điền họ tên ");

# JTextField

**Bước 1**: Thả component JTextFiled vào màn hình design



**Bước 2**: Cách lấy nội dung ô JTextFiled

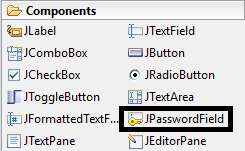
String hoten = txtHoten.getText();

Cách gán nội dung cho JtextFiled

txtHoten.setText(“Green Academy”);

# JPasswordField

**Bước 1**: Thả component JPasswordFiled vào màn hình design

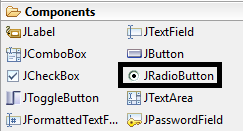


**Bước 2**: Cách lấy nội dung từ JPasswordField

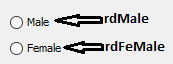
String pass = new String(txtPass.getPassword());

# JRadioButton

**Bước 1:** Thả component JRadioButton vào màn hình design

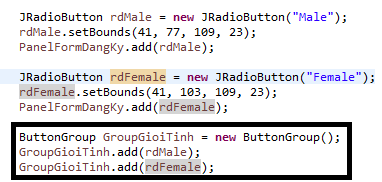


**Bước 2**: Đặt tên cho 2 components

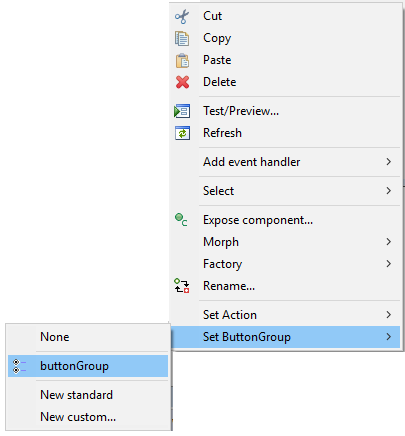


**Bước 3:** Để tạo nhóm cho radio button ( bật tắt thay phiên) ta thực hiện từng bước như sau:

**Cách 1** : Mở Tab Source thêm đoạn code sau :



**Cách 2**: Giữ Ctrl + Chọn tất cả radio button, Right Click / Set Button Group / button Group



**Bước 3:** Kiểm tra JRadioButton được chọn

if(! rdMale.isSelected())

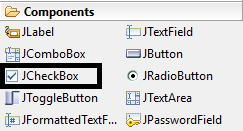
{

……………………

}

# JCheckBox

**Bước 1:** Thả component JCheckBox vào màn hình design



**Bước 2:** Kiểm tra JCheckbox được chọn

if( !(ckDocSach.isSelected())

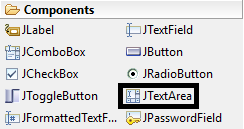
{

……………………

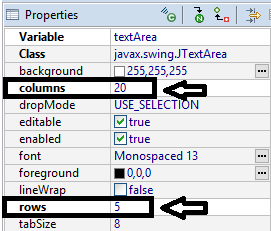
}

# JTextArea

**Bước 1**: Thả component JTextArea vào màn hình design

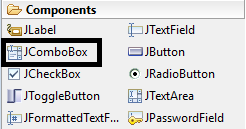


**Bước 2**: Chỉnh số lượng cột và dòng



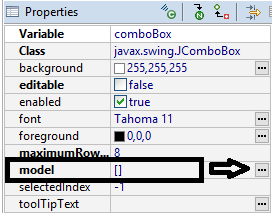
# JComboBox

**Bước 1**: Thả component JComboBox vào màn hình design

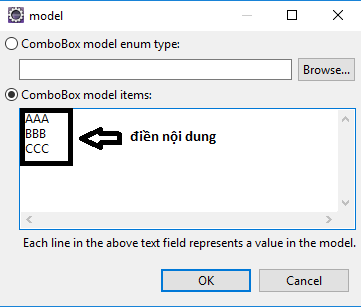


**Bước 2**: Đưa giá trị vào JCombobox

**1**. Nhấn vào nút […] ở phần model



**2.** Điền nội dung của từng dòng trong combobox



**3.** Run để xem kết quả như sau:

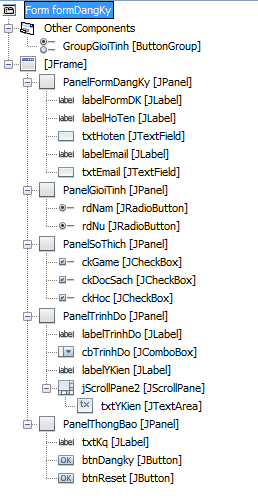


# Bài tập thực hành

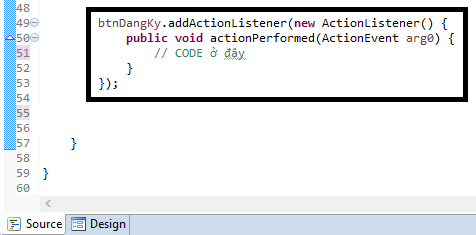
Thiết kế giao diện như sau và cài đặt báo lỗi khi user bỏ trống thông tin nào đó



**Bước 1**: Đặt tên các components theo hướng dẫn sau:



**Bước 2 :** Cài đặt sự kiện cho nút Đăng ký



**Bước 3 :** Cài đặt code bắt lỗi cho nút Đăng ký :

1. Kiểm tra họ tên

String hoten = txtHoten.getText();

if(hoten.equalsIgnoreCase(""))

{

txtKq.setText("Bạn chưa điền họ tên ");

txtHoten.requestFocus();

return;

}

1. Kiểm tra email không được bỏ trống

String email = txtEmail.getText();

if(email.equalsIgnoreCase(""))

{

txtKq.setText("Bạn chưa điền Email");

txtEmail.requestFocus();

return;

}

1. Kiểm tra email theo đúng chuẩn

String mau = "^[a-zA-Z0-9]+@[a-zA-Z]+(\\.[a-zA-Z]{2,4}){1,2}$";

Pattern p = Pattern.compile(mau);

Matcher m = p.matcher(email);

if(!m.matches())

{

txtKq.setText("Email ko đúng chuẩn");

txtEmail.requestFocus();

return;

}

1. Kiểm tra radio button phải được chọn

if( !(rdNam.isSelected()||rdNu.isSelected()) )

{

txtKq.setText("Bạn chưa chọn giới tính");

rdNam.requestFocus();

return;

}

1. Kiểm tra checkbox phải được chọn

if( !(ckDocSach.isSelected()||ckGame.isSelected()||ckHoc.isSelected())) {

txtKq.setText("Bạn chưa chọn sở thích");

ckGame.requestFocus();

return;

}

1. Kiểm tra combobox phải được chọn

int chon = cbTrinhDo.getSelectedIndex();

if(chon==0)

{

txtKq.setText("Bạn chưa chọn bằng cấp");

cbTrinhDo.requestFocus();

return;

}

**Bước 4:**

a. Cài đặt sự kiện cho nút Reset

b. Code cho sự kiện reset

txtHoten.setText("");

txtEmail.setText("");

GroupGioiTinh.clearSelection();

ckDocSach.setSelected(false);

ckGame.setSelected(false);

ckHoc.setSelected(false);

cbTrinhDo.setSelectedIndex(0);

txtYKien.setText("");

txtKq.setText("Mời bạn điền đầy đủ thông tin");

**Bước 5** : Nâng cao tính năng cho nút Đăng ký : Nếu kiểm tra các thông tin điền vào đều hợp lệ , tiến hành lưu thông tin (gồm : họ tên và email ) vào file.

* Tạo mới 1 class Student:

public class Student {

String hoten;

String email;

public Student() {

}

public Student(String hoten, String email) {

this.hoten = hoten;

this.email = email;

}

public String toString()

{

return String.format("%s %s",this.hoten,this.email);

}

b. Trong sự kiện của nút “Đăng ký”, bổ sung đoạn code sau:

Student a = new Student(txtHoten.getText(),txtEmail.getText());

try

{

FileWriter fr = new FileWriter("E://a.txt");

BufferedWriter br = new BufferedWriter(fr);

br.write(a.toString());

br.close();

}

catch(IOException e)

{

System.out.println(e.getMessage());

}

txtKq.setText("Chúc mừng đăng ký thành công");

Hoặc có thể hiện thông báo bằng bảng Thông báo như sau :

JOptionPane.showMessageDialog(getParent(),"Chúc mừng đăng ký thành công");

**Bước 6:** Kết thúc , kiểm tra nội dung file vừa được tạo

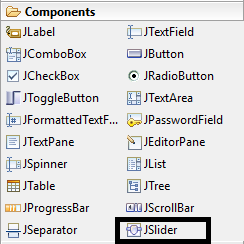
JAVA SWING - COMPONENTS ( 2 )

**BÀI TẬP 1 : Thiết kế 1 Slider như hình sau:**



**Hướng dẫn thực hiện**

**Bước 1** : Chọn component Jslider thả vào màn hình Design



**Bước 2** : Chỉnh Properties cho JSlider

|  |  |
| --- | --- |
|  | Giải thích:     * paintLables: Nhãn “ 0 10 20…100 ” * paintTicks: gạch chia nhỏ trên thanh Track * snapToTicks: Khi click chuột lên thanh track, thì pointer sẽ nhảy lên bao nhiêu đơn vị |

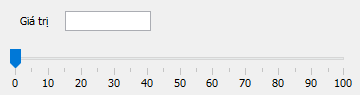
+ Nếu vẫn chưa thấy paintLables hiện lên thì ta thêm vào màn hình Source dòng code sau : ( nếu paintLables có hiện lên thì bỏ qua bước này)

slider.setLabelTable(slider.createStandardLabels(10));

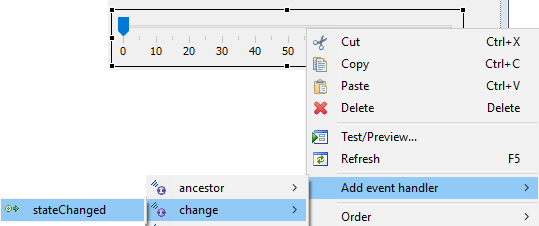
* ***Lưu ý đặt trên dòng :***

panel.add(slider);

**Bước 3**: Thiết kế thêm ô Jtextfield : thể hiện giá trị khi thay đổi thanh trượt của Slider



**Bước 4**: Thêm sự kiện khi thay đổi JSlider



**Bước 5 :** Bên màn hình Source, xuất hiện đoạn code:

slider.addChangeListener(new ChangeListener() {

public void stateChanged(ChangeEvent e) {

}

});

**Bước 6** : Thêm đoạn code in đậm bên dưới :

slider.addChangeListener(new ChangeListener() {

public void stateChanged(ChangeEvent e) {

**int gtri = slider.getValue();**

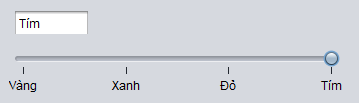
**txt1.setText(String.valueOf(gtri));**

}

});

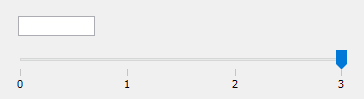
**Bước 7**: Run và thay đổi thanh trượt của slider để kiểm tra giá trị xuất hiện trên ô Textfield

**BÀI TẬP 2 : Thực hiện Slider thể hiện 4 mùa trong năm**



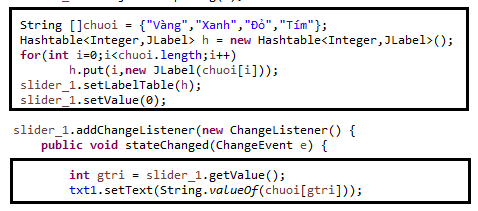
**Hướng dẫn thực hiện:**

**Bước 1**: Ban đầu thiết kế giao diện Slider như sau:

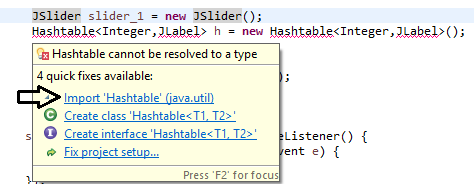


**Bước 2**: Thêm sự kiện stateChange cho Slider

Và thêm vào các dòng code trước và trong sự kiện như sau:



**Ghi chú** : Nếu dòng code HashTable và JLabel báo lỗi vì thiếu thư viện. Ta đưa chuột vào gần từ khóa HashTable sẽ hiện lên bảng gợi ý fix -> chọn dòng ‘Import Hashtable ( Java.util)’

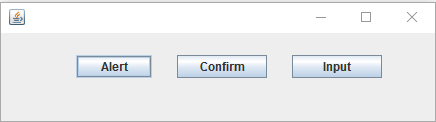


Tượng tự cho cách fix tại JLabel

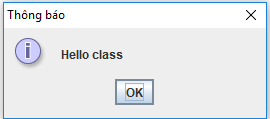
**Bước 3**: Run và kiểm tra kết quả trên ô Text khi thực hiện thay đổi thanh trượt của Slider

**BÀI TẬP 3:** **DIALOG**

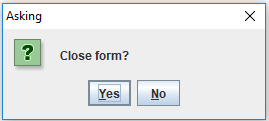
**Thiết kế form như sau :**



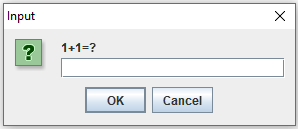
1. Nhấn nút Alert hiển thị thông báo



1. Nhấn nút Confirm :
   * + Nhấn nút Yes : đóng màn hình form ở trên
     + Nhấn nút No : hiện bảng alert “Hello”



1. Nhấn nút Input :
   * + Điền kết quả là 2 -> hiện bảng thông báo : “Kết quả đúng”
     + Điền kết quả khác 2 -> hiện bảng thông báo : “Kết quả sai”



**Hướng dẫn thực hiện:**

**Bước 1**: Tạo sự kiện cho nút Alert và đưa vào đoạn code như sau:

btnAlert.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

JOptionPane.showMessageDialog(getParent(), "Hello class","Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

});

**Bước 2**: Tạo sự kiện cho nút Confirm và đưa vào đoạn code như sau:

JButton btnCornfirm = new JButton("Confirm");

btnCornfirm.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

int chon = JOptionPane.showConfirmDialog(getParent(), "Close form?", "Asking",JOptionPane.ERROR\_MESSAGE);

if(chon==JOptionPane.YES\_OPTION)

System.exit(0);

if(chon==JOptionPane.NO\_OPTION)

JOptionPane.showMessageDialog(getParent(), "Hello");

}

});

**Bước 3**: Tạo sự kiện cho nút Input và đưa vào đoạn code như sau:

btnInput.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String noidung = JOptionPane.showInputDialog(getParent(),"1+1=?");

if(noidung.equals("2"))

JOptionPane.showMessageDialog(getParent(), "Kết quả đúng");

else

JOptionPane.showMessageDialog(getParent(), "Kết quả sai");

}

});

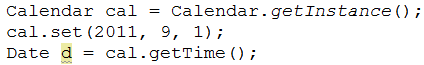
**BÀI TẬP 4: Viết chương trình hiển thị chuỗi thông tin thời gian (được chọn từ các control JSlider) dưới dạng thứ, ngày – tháng - năm**

|  |
| --- |
|  |

1. ***Tạo 3 JSliders biểu thị mốc thời gian năm, tháng, ngày như hình bên:***

* Năm: 1961 – năm hiện hành + 5
* Tháng : 0 🡪11 (Jan🡪Dec)
* Ngày: 1-31

1. ***Sử dụng lớp Calendar để tạo đối tượng Date ứng với 1 điểm thời gian cụ thể:***+ Ví dụ: để tạo đối tượng Date của ngày 1, tháng 10 năm 2011



1. ***Sử dụng lớp SimpleDateFormat để định dạng đối tượng Date :***

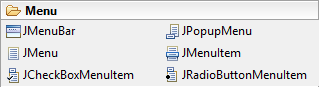


*Giải thích :* Với EEEE là tên của thứ trong tuần (vd Monday, Tuesday ...), MMMM là tên của tháng trong năm (vd January, Febuary...)

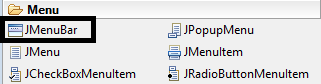
1. Để update chuỗi thông tin thời gian khi thay đổi giá trị của các slider year, month, day: lập trình sự kiện change của các slider này.

JAVA SWING - COMPONENTS ( 3 )

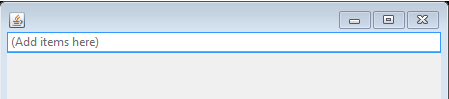
1. **Component Menu**



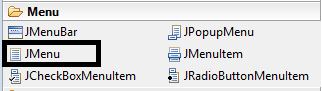
* 1. **JMenuBar**



Thả vào màn hình Design



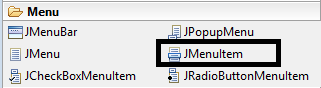
* 1. **JMenu**



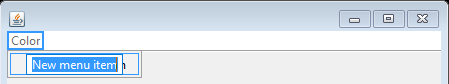
Thả vào màn hình Design



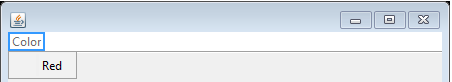
* 1. **JMenuItem**



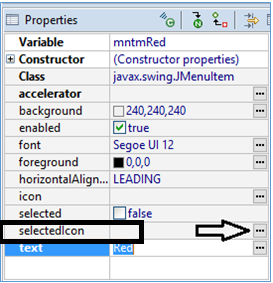
* 1. Thả vào màn hình Design và thay đổi text



*Ví dụ:*



* 1. Chọn icon hình cho Item



* 1. Tạo sự kiện cho MenuItem
* Right Click vào MenuItem/ Chọn



* Trong Source Code sẽ xuất hiện đoạn code sau :

mntmRed.addActionListener(new ActionListener() {

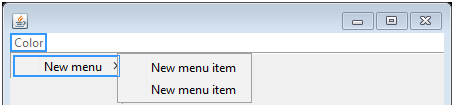
public void actionPerformed(ActionEvent e) {

. . . . . . . } });

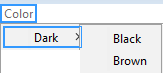
* 1. **Tạo Sub Menu**

**Bước 1** : Thả vào JMenu ( Main)

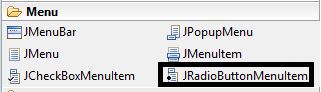
**Bước 2** : Thả các JMenuItem để làm menu item con



*Ví dụ:*



* 1. **JRadioButtonMenu**



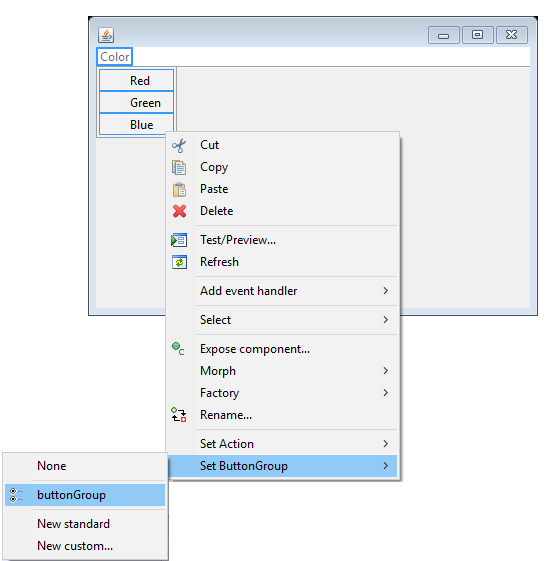
1. Chọn JRadioButtonMenuItem thả vào màn hình design



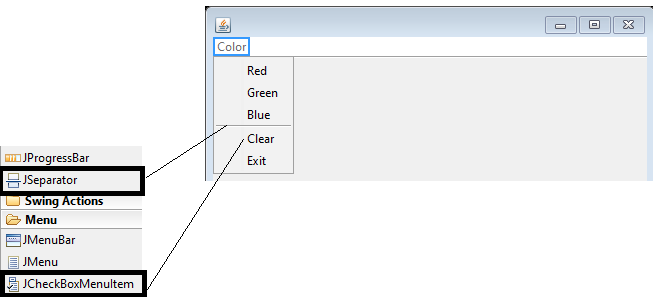
1. Nhóm các JRadioItem

+ Nhấn Ctrl + chọn 3 Item

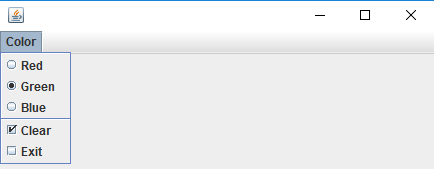
+ Right click / Set Button Group/ buttonGroup



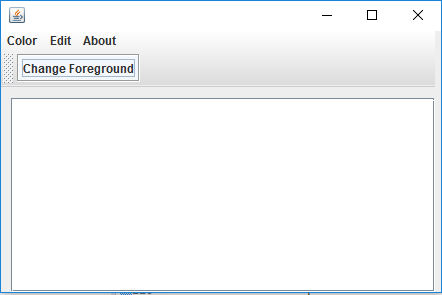
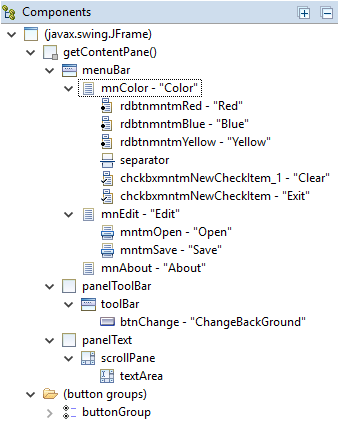
* 1. **JCheckBoxMenuItem**



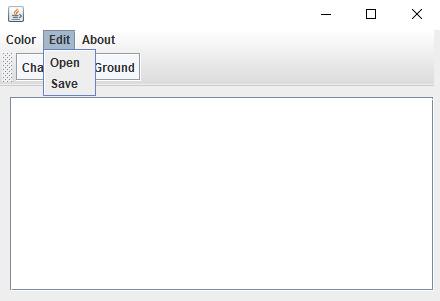
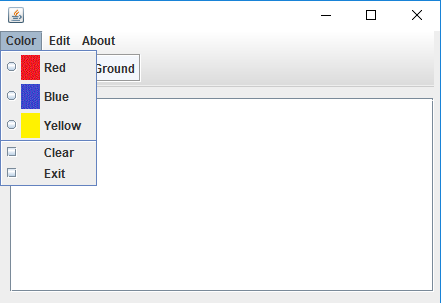
Chạy chương trình cho màn hình kết quả sau:



1. **Bài tập thực hành**

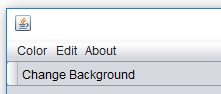
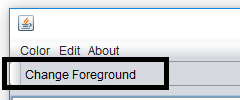


Nội dung chi tiết của các menu như sau:



**Mô tả bài toán** :

1. Nhấn vào nút Change Foreground , chữ sẽ chuyển thành Change BackGround và ngược lại.



1. Khi chọn Menu Color
   1. Red, Blue, Yellow thì ForeGround (màu chữ) hay Background (màu nền ) trong Text Area sẽ được chuyển đổi màu tương ứng ( Tùy vào Thanh Toolbar đang chọn là ForeGround hay BackGround)
   2. Clear : Đưa ứng dụng về trạng thái ban đầu
   3. Exit : Thoát chương trình
2. Chon Menu Edit
   1. Save : Cho phép save lại nội dung của Textarea vào 1 file (không cần lưu thông tin màu sắc)
   2. Open : Mở và load nội dung file vào ô Textarea.
3. Chọn Menu About

* Show Dialog bao gồm: thông tin tác giả và version của ứng dụng

**Hướng dẫn thực hiện :**

**Bước 1**: Chuyển khai báo JTextArea lên đầu

public class example3 extends JFrame {

private final ButtonGroup buttonGroup = new ButtonGroup();

**JTextArea textArea;**

**Bước 2** : Tạo sự kiện 1 lượt cho tất cả các menuItem : Red, Blue, Yellow, Clear, Exit, Save, Open, About, Nút Change ( trên thanh JToolbar)

**Bước 3 :** Khai báo biến flag ( dùng để nhận biết thay đổi màu nền hay màu chữ)

**int flag;**

public example3() {

**flag = 0;**

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 456, 300);

getContentPane().setLayout(null);

}

**Bước 4**: Đưa code vào sự kiện của nút Change

JButton btnChange = new JButton("Change Foreground");

**btnChange**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(flag==0)

{

flag = 1;

btnChange.setText("ChangeBackground");

}

else

{

flag = 0;

btnChange.setText("ChangeForeground");

}

buttonGroup.clearSelection();

}

});

**Bước 5** : Đưa code vào sự kiện của menu Red, Blue, Yellow

JRadioButtonMenuItem rdbtnmntmRed = new JRadioButtonMenuItem("Red");

**rdbtnmntmRed**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(flag==0)

textArea.setBackground(Color.RED);

else

textArea.setForeground(Color.RED);

}

});

JRadioButtonMenuItem rdbtnmntmBlue = new JRadioButtonMenuItem("Blue");

**rdbtnmntmBlue**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(flag==0)

textArea.setBackground(Color.BLUE);

else

textArea.setForeground(Color.BLUE);

}

});

JRadioButtonMenuItem rdbtnmntmYellow = new JRadioButtonMenuItem("Yellow");

**rdbtnmntmYellow**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(flag==0)

textArea.setBackground(Color.YELLOW);

else

textArea.setForeground(Color.YELLOW);

}

});

**Bước 6** : Đưa code vào sự kiện của menu Save

JMenuItem mntmSave = new JMenuItem("Save");

**mntmSave**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JFileChooser f = new JFileChooser();

int chon = f.showSaveDialog(getParent());

if(chon==f.APPROVE\_OPTION)

{

String filename = f.getSelectedFile().getAbsolutePath();

try

{

FileWriter fr = new FileWriter(filename);

BufferedWriter br = new BufferedWriter(fr);

String noidung = textArea.getText();

br.write(noidung);

br.close();

}

catch(IOException ex)

{

System.out.println(ex.getMessage());

}

}

} });

**Bước 7**: Đưa code vào sự kiện của menu Open

JMenuItem mntmOpen = new JMenuItem("Open");

**mntmOpen**.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JFileChooser f = new JFileChooser();

int r = f.showOpenDialog(getParent());

if(r == JFileChooser.APPROVE\_OPTION)

{

String filename = f.getSelectedFile().getAbsolutePath();

try

{

FileReader fr = new FileReader(filename);

BufferedReader br = new BufferedReader(fr);

String noidung = "";

while(br.ready())

{

noidung = noidung + br.readLine() + "\r\n";

}

textArea.setText(noidung);

br.close();

}

catch(IOException ex)

{

System.out.println(ex.getMessage());

}

}

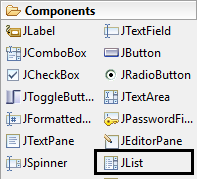
}

});

**Bước 8** : Sinh viên tự code các sự kiện cho các menu Clear, Exit, About

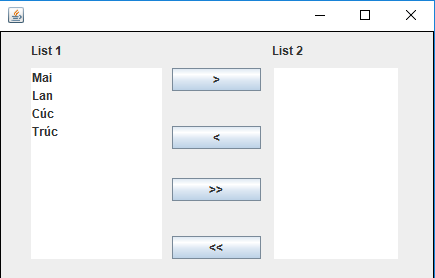
JAVA SWING - COMPONENTS ( 4 )

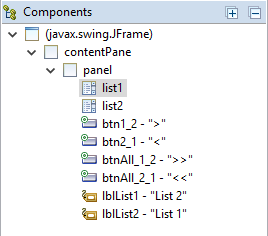
1. **Jlist**

****

**Bài tập 1:**

Thiết kế giao diện như sau:





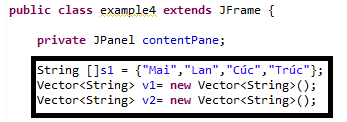
***Yêu cầu bài toán:***

1. Nút “>” “<” : để chuyển các dòng dữ liệu được chọn từ List này sang List kia và ngược lại.
2. Nút “>>” “<<” : để chuyển tất cả các dòng dữ liệu của List này sang list kia và ngược lại

**Hướng dẫn:**

**Bước 1:** Khai báo như sau:

* Vector v1: chứa nội dung của list1
* Vector v2: chứa nội dung của list2



**Bước 2:** Thêm dòng code **in đậm** như sau**:**

( chuyển vector v1 vào list1)

JList list1 = new JList();

list1.setBounds(31, 37, 131, 191);

panel.add(list1);

**v1.addAll(Arrays.asList(s1));**

**list1.setListData(v1);**

**Bước 3:** Code cho sự kiện của nút “>”

JButton btn1\_2 = new JButton(">");

btn1\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

**List chonnhieu = list1.getSelectedValuesList();**

**v2.addAll(chonnhieu);**

**v1.removeAll(chonnhieu);**

**list1.setListData(v1);**

**list2.setListData(v2);**

}

});

**Bước 4:** Code cho sự kiện của nút “<”

JButton btn2\_1 = new JButton("<");

btn2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

**List chonnhieu = list2.getSelectedValuesList();**

**v1.addAll(chonnhieu);**

**v2.removeAll(chonnhieu);**

**list1.setListData(v1);**

**list2.setListData(v2);**

**}**

**});**

**Bước 5:** Code cho sự kiện của nút “>>”

JButton btnAll\_1\_2 = new JButton(">>");

btnAll\_1\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

**v2.addAll(v1);**

**v1.removeAllElements();**

**list1.setListData(v1);**

**list2.setListData(v2);**

}

});

**Bước 6:** Code cho sự kiện của nút “<<”

JButton btnAll\_2\_1 = new JButton("<<");

btnAll\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

**v1.addAll(v2);**

**v2.removeAllElements();**

**list1.setListData(v1);**

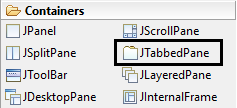
**list2.setListData(v2);**

}

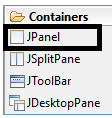
});

1. **JtabbPane**

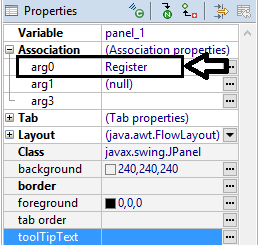
**Bước 1**: Chọn JTabbedPane thả vào màn hình design

****

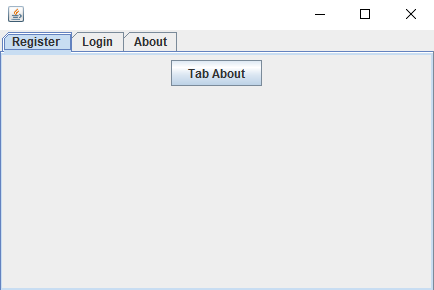
**Bước 2**: Muốn có bao nhiêu tab thì chọn bấy nhiêu JPanel thả vào màn hình JtabbedPane

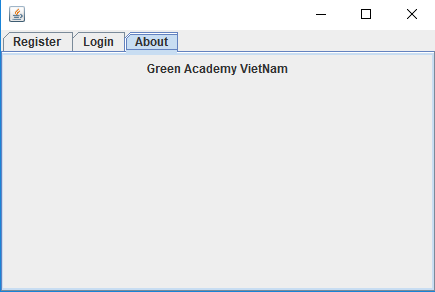


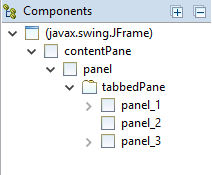
**Bước 3**: Đổi title cho tab

****

**Bài tập 2: Thiết kế màn hình gồm 3 Tab như sau:**

****

****

****

**Yêu cầu bài toán:**

* Khi nhấn **nút “Tab About”** trong Tab Register , thì Tab About sẽ được chọn load lên

**Hướng dẫn:**

JButton btnTabAbout = new JButton("Tab About");

btnTabAbout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

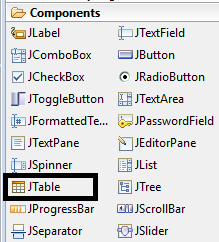
**tabbedPane.setSelectedIndex(2);**

}

});

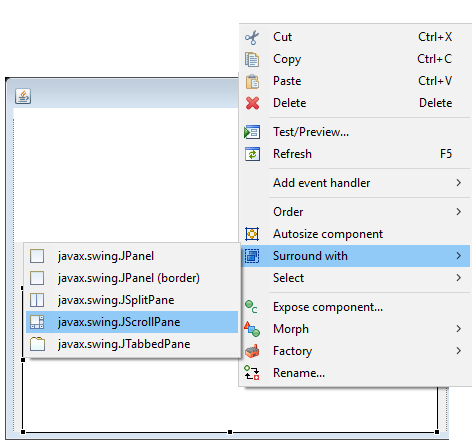
1. **JTable**

**Bước 1**: Chọn Jtable thả vào màn hình Design

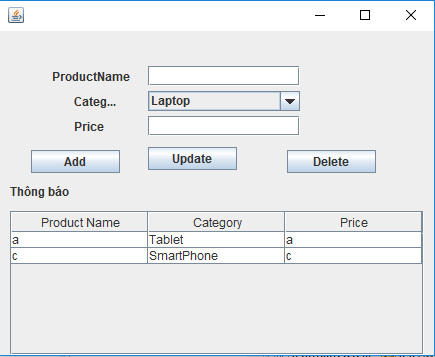


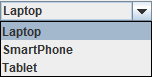
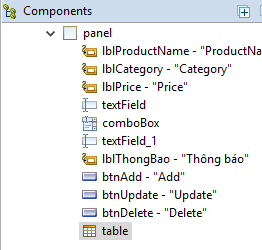
**Bước 2:** Right Click vào JTable vừa thả , chọn Surround with/ javax.swing.Scrollpane

( sẽ xuất hiện thanh scroll khi dữ liệu của bảng bị tràn màn hình )



**Bài tập 3: Thiết kế màn hình như sau:**





**Yêu cầu bài toán :**

1. **Nhấn nút “Add”**

* Dữ liệu ô nào bị bỏ trống hoặc không đúng chuẩn sẽ bào lỗi
* Dữ liệu đúng sẽ được đưa vào Table bên dưới và đồng thời lưu vào file

**2. Click chuột lên từng record của Table**

- Dữ liệu của dòng được chọn sẽ hiển thị ngược lại lên TextField và Combobox

**3**. **Nhấn nút “Update”**

- Dữ liệu trên các ô TextFiled và Combobox sẽ được cập nhật xuống Table ngay dòng đang được chọn, sau đó save nội dung mới vào file

**4**. **Nhấn nút “Delete”**

- Dữ liệu của dòng đang được chọn trên table sẽ được xóa khỏi danh sách, sau đó save nội dung mới vào file

**5. Load nội dung lên Table**

- Khi tắt và mở lại chương trình, thì dữ liệu trước đó của table sẽ được load lên trở lại , bằng cách thực hiện đọc file đả save lại ở trên

**Hướng dẫn:**

**Bước 1:** Khai báo DefaultTableModel để làm việc với model của Table

public class example7 extends JFrame {

private JPanel contentPane;

**DefaultTableModel model = new DefaultTableModel();**

**Bước 2:** Khai báo các tên cột của Table ( code thêm các dòng **in đậm** )

scrollPane.setViewportView(table);

**Object []columns = {"Product Name","Category","Price"};**

**model.setColumnIdentifiers(columns);**

**table.setModel(model);**

**Bước 3:** Code chức năng cho sự kiện nút Add( code thêm các dòng **in đậm** )

* Đưa dữ liệu các textfiled và combobox vào 1 row và add vào model

**Object []row = new Object[3];**

JButton btnAdd = new JButton("Add");

btnAdd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

**row[0] = txtProductName.getText();**

**row[1] = cbCategory.getSelectedItem();**

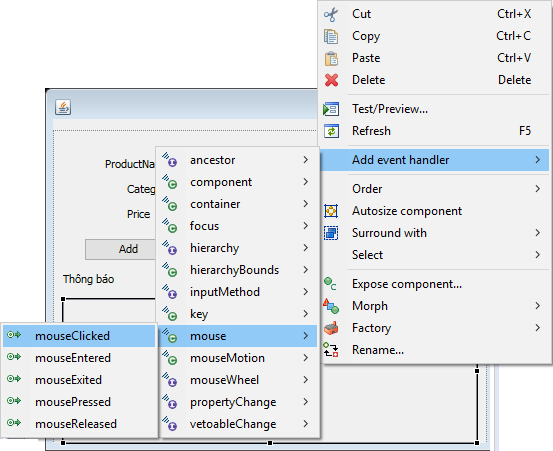
**row[2] = txtPrice.getText();**

**model.addRow(row);**

}

});

**Bước 4:** Tạo sự kiện click mouse lên Table



Code cho sự kiện Click chuột lên từng dòng dữ liệu của table ( code thêm các dòng **in đậm** )

table.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

**int i = table.getSelectedRow();**

**txtProductName.setText(model.getValueAt(i, 0).toString());**

**cbCategory.setSelectedItem(model.getValueAt(i, 1).toString());**

**txtPrice.setText(model.getValueAt(i, 2).toString());**

}

});

**Bước 5:** Code chức năng cho sự kiện nút Update( code thêm các dòng **in đậm** )

JButton btnUpdate = new JButton("Update");

btnUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

**int i = table.getSelectedRow();**

**model.setValueAt(txtProductName.getText(), i, 0);**

**model.setValueAt(cbCategory.getSelectedItem(), i, 1);**

**model.setValueAt(txtPrice.getText(), i, 2);**

}

});

**Bước 6:** Code chức năng cho sự kiện nút Delete( code thêm các dòng **in đậm** )

JButton btnDelete = new JButton("Delete");

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

**int i = table.getSelectedRow();**

**if(i>=0)**

**model.removeRow(i);**

}

});

**Bước 7:** Code ghi dữ liệu từ Table ra file

public void writefile()

{

try

{

FileWriter fr = new FileWriter("temp.txt");

BufferedWriter br = new BufferedWriter(fr);

String row;

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

{

row = model.getValueAt(i,0) +"\t" + model.getValueAt(i,1)+ "\t"+ model.getValueAt(i,2)+"\r\n";

br.write(row);

}

br.close();

}

catch(IOException ex)

{

System.out.println(ex.getMessage());

}

}

**Bước 8:** Lần lượt gọi hàm writefile() trong các sự kiện của nút Add, Update, Delete( code thêm các dòng **in đậm** )

btnAdd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

row[0] = txtProductName.getText();

row[1] = cbCategory.getSelectedItem();

row[2] = txtPrice.getText();

model.addRow(row);

**writefile();**

}

});

JButton btnUpdate = new JButton("Update");

btnUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int i = table.getSelectedRow();

model.setValueAt(txtProductName.getText(), i, 0);

model.setValueAt(cbCategory.getSelectedItem(), i, 1);

model.setValueAt(txtPrice.getText(), i, 2);

**writefile();**

}

});

JButton btnDelete = new JButton("Delete");

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int i = table.getSelectedRow();

if(i>=0)

model.removeRow(i);

**writefile();**

}

});

**Bước 9**: Code hàm đọc dữ liệu từ file đưa vào Table

public void readfile()

{

try

{

FileReader fr = new FileReader("temp.txt");

BufferedReader br = new BufferedReader(fr);

String str ;

Object []row = new Object[3];

while(br.ready())

{

str = br.readLine();

row = str.split("\t");

model.addRow(row);

}

br.close();

}

catch(IOException ex)

{

System.out.println(ex.getMessage());

}

}

**Bước 10:** Gọi hàm đọc file sau dòng code setModal( code thêm các dòng **in đậm**)

scrollPane.setViewportView(table);

Object []columns = {"Product Name","Category","Price"};

model.setColumnIdentifiers(columns);

table.setModel(model);

**readfile();**

JAVA SWING – JDBC ( 1 )

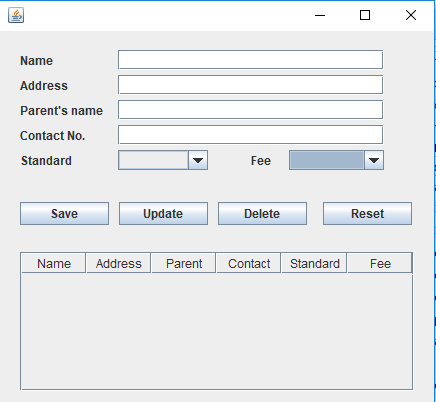
Kidszone School wants to store the information of students and their parents in their database. This will help them to better administer and control the new student registration process as well as manage already existing records of enrolled students.

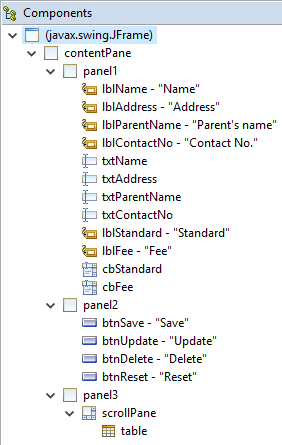
The School Management Board has chosen your company to provide a solution for the same. Consider yourself to be a part of the team that implements the solution for designing the application. Create a GUI based application that will allow the school staff to enter student details using Java Swing controls.

The form should allow entering ***student details***, such as ***name***, ***address***, ***phone number***, ***parents' name***, ***standard*** in which to be enrolled, and ***fees*** to be paid.

1. Information of fees and standard should be loaded from tables in database.
2. All the information should be stored in database tables when OK button is clicked.
3. The button Reset will blank out all text entry fields.
4. The button Cancel will abort any entry in the fields and the table.
5. Apply field level validations and display appropriate message while the staff enters data to prevent entry of junk data.

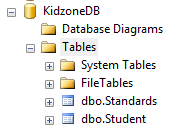
The interface of the student registration should resemble the following:





1. **THIẾT KẾ DATABASE**

**Bước 1:** Tạo database có tên là: KidzoneDB . Gồm 2 tables : Standards, Student



**Mô tả dữ liệu :**

**Standards**: Bảng chứa các tiêu chuẩn mà học sinh có thể chọn.

standard: mã tiêu chuẩn cũng là tên gọi, khóa chính của bảng

fee: học phí mà học sinh phải trả cho tiêu chuẩn này.

**Student** : Bảng chứa danh sách học sinh.

RegID : mã đăng ký, tăng tự động, khóa chính của bảng

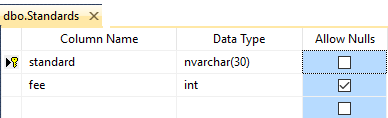
Name, Address, parentName: tên, địa chỉ, tên cha mẹ của học sinh

phone: số điện thọai liên lạc

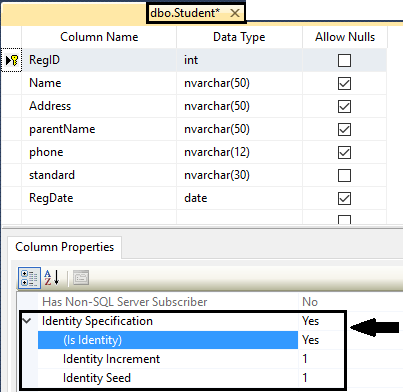
standard: tiêu chuẩn mà học sinh đã chọn. khóa ngoại tham chiếu đến bảng Standards.

RegDate : ngày đăng ký.

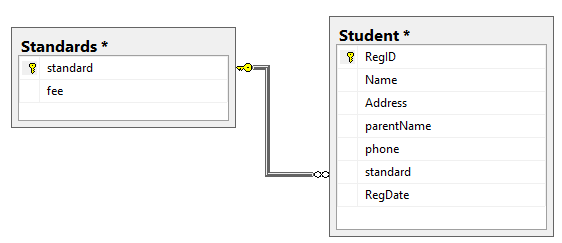
**Bảng Standard:**



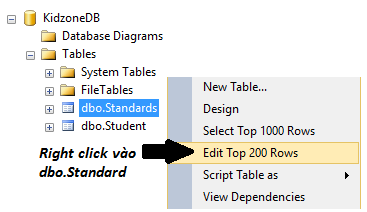
**Bảng Student:**

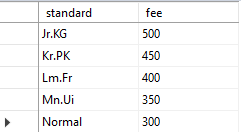


**Quan hệ giữa 2 bảng:**

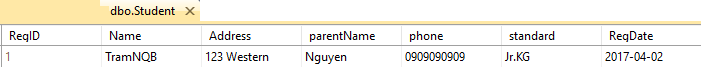


**Bước 2:** Input dữ liệu ban đầu cho bảng **Standards**:





**Bước 3 :** Tương tự , thêm 1 dòng dữ liệu vào bảng Student . Lưu ý : Dữ liệu tại cột RegID không điền ( sẽ tự động xuất hiện khi click sang dòng dữ liệu khác )



1. **HƯỚNG DẪN CODE GIAO DIỆN JFRAME**

**Bước 1:** Khai báo DefaultTableModel để làm việc với model của Table

public class Exercise extends JFrame {

private JPanel contentPane;

…………………

…………………

private JTable table;

DefaultTableModel model = new DefaultTableModel();

**Bước 2:** Khai báo các tên cột của Table ( code thêm các dòng **in đậm** )

scrollPane.setViewportView(table);

**Object []columns = {"RegID","Name","Address","Parent","Contact","Standard","RegDate"};**

**model.setColumnIdentifiers(columns);**

**table.setModel(model);**

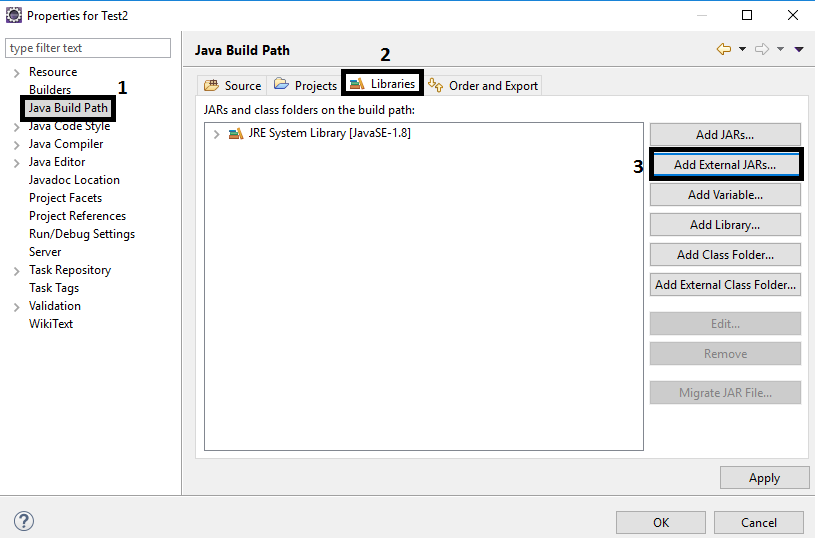
// ẩn cột đầu tiên ( RegID) không cho hiện trên table

**table.removeColumn(table.getColumnModel().getColumn(0));**

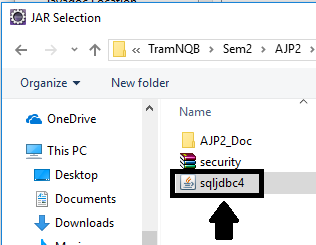
**Run để kiểm tra kết quả màn hình Jframe**

**Bước 3: Add driver sqljdbc để kết nối với database SQLserver**

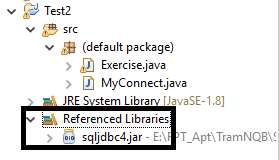
* Nhấn phải vào Project chọn Properties
* Bảng Properties hiện ra, chọn theo thứ tự 1,2,3



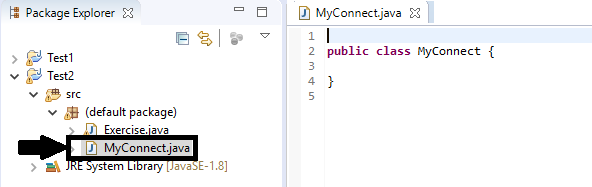
Chọn đường dẫn đến driver để trên ổ cứng của máy



Driver đã được đưa vào project như hình bên dưới:



**Bước 4: Tạo class MyConnect.java**



Thực hiện code kết nối database SQL server

public class MyConnect {

String servername ;

String port ;

String db\_name ;

String db\_user;

String db\_pass;

public MyConnect() {

servername = "127.0.0.1";

port = "1433";

db\_name = "KidzoneDB";

db\_user="sa";

db\_pass="abc123"; // password kết nối vào MS SQL

}

public MyConnect(String servername, String port, String db\_name, String db\_user, String db\_pass) {

this.servername = servername;

this.port = port;

this.db\_name = db\_name;

this.db\_user = db\_user;

this.db\_pass = db\_pass;

}

public Connection getcn()

{

Connection cn = null;

try

{

//Database url

String db\_url = "jdbc:sqlserver://"+ servername +":"+ port +";databasename = " + db\_name;

//Load JDBC

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

//Ketnoivoi database

cn = DriverManager.getConnection(db\_url, db\_user, db\_pass);

}

catch (ClassNotFoundException | SQLException ex)

{

System.out.println(ex.getMessage());

}

return cn;

}

} // Ngoặc đóng class

**Bước 5: Khai báo đối tượng Connection bên file Jframe**

DefaultTableModel model = new DefaultTableModel();

Connection conn;

**Bước 6: Trong hàm dựng Jframe gọi hàm getcn() kiểm tra kết nối đến database**

public Exercise() {

conn = new MyConnect().getcn();

if(conn==null)

{

JOptionPane.showMessageDialog(getParent(),"Kết nối Database thất bại");

return;

}

**Run chương trình để kiểm tra kết quả**

**Bước 7 : Hàm** loadComboBox() : thực hiện đổ dữ liệu từ bảng Standards trong database lên combobox Standard và combobox Free

Vector v1 = new Vector();

Vector v2 = new Vector();

public void loadComboBox()

{

try

{

String sql = "Select \* from Standards";

PreparedStatement ps = conn.prepareStatement(sql);

ResultSet rs = ps.executeQuery();

while(rs.next())

{

v1.addElement(rs.getString(1));

v2.addElement(rs.getString(2));

}

//Đổ dữ liệu từ vector lên combobox

cbStandard.setModel(new DefaultComboBoxModel(v1));

cbFee.setModel(new DefaultComboBoxModel(v2));

//Đóng kết nối

rs.close();

ps.close();

conn.close();

}

catch(SQLException e)

{

System.out.println(e.getMessage());

}

}

**Bước 8 :** Gọi hàm loadComboBox() trong hàm dựng của JFrame **( Code theo chữ in đậm )**

public Exercise() {

conn = new MyConnect().getcn();

if(conn==null)

{

JOptionPane.showMessageDialog(getParent(), "Kết nối Database thất bại");

return;

}

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 410);

contentPane = new JPanel();

………………

………………

**loadComboBox();**

}

Kết quả :



**Bước 9:** Cài sự kiện cho từng combobox: để dữ liệu được chọn tại 2 combobox sẽ tương ứng với nhau

cbStandard = new JComboBox();

cbStandard.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

cbFee.setSelectedIndex(cbStandard.getSelectedIndex());

}

});

cbFee = new JComboBox();

cbFee.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cbStandard.setSelectedIndex(cbFee.getSelectedIndex());

}

});

**Bước 11:** Hàm loadTable() : Đổ dữ liệu từ bảng Student trong database lên table của JFrame

public void loadTable()

{

conn = new MyConnect().getcn();

Object []row = new Object[7];

try

{

String sql = "select \* from Student";

PreparedStatement ps = conn.prepareStatement(sql);

ResultSet rs = ps.executeQuery();

model.setRowCount(0);

while(rs.next())

{

row[0] = rs.getString(1);

row[1] = rs.getString(2);

row[2] = rs.getString(3);

row[3] = rs.getString(4);

row[4] = rs.getString(5);

row[5] = rs.getString(6);

row[6] = rs.getString(7);

model.addRow(row);

}

rs.close();

ps.close();

conn.close();

}

catch(SQLException e)

{

System.out.println(e.getMessage());

} (e.getMessage());

}

**Bước 12**: Gọi hàm loadTable () trong hàm dựng của Jframe

public Exercise() {

conn = new MyConnect().getcn();

if(conn==null)

{

JOptionPane.showMessageDialog(getParent(), "Kết nối Database thất bại");

return;

}

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

………………

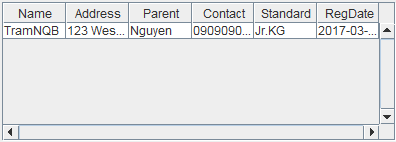
………………

loadComboBox();

**loadTable()**

}

Kết quả :



**Bước 13: Viết hàm valid() kiểm tra dữ liệu nhập**

* Các tên, địa chỉ: buộc nhập 🡪 Độ dài chuỗi nhập phải >0
* Số điện thoại: 7.. 12 ký số
* Nếu dữ liệu không hợp lệ thì ứng dụng sẽ xuất thông báo phù hợp.



Hàm valid() sẽ return true nếu check tất cả điều kiện đúng, ngược lại return false.

**Bước 14: Code cho sự kiện của nút Save**

JButton btnSave = new JButton("Save");

btnSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// kiểm tra nếu điều kiện sai sẽ không thực hiện tiếp được

if(!valid())

return;

String name = txtName.getText();

String address = txtAddress.getText();

String parentName = txtParentName.getText();

String phone = txtContactNo.getText();

String standard = (String) cbStandard.getSelectedItem();

Calendar c = Calendar.getInstance();

SimpleDateFormat sf = new SimpleDateFormat("MM/dd/yyyy");

String regDate = sf.format(c.getTime());

conn =new MyConnect().getcn();

try

{

String sql = "insert into Student values(?,?,?,?,?,?)";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setString(1,name);

ps.setString(2,address);

ps.setString(3,parentName);

ps.setString(4,phone);

ps.setString(5,standard);

ps.setString(6,regDate);

int kq = ps.executeUpdate();

if(kq==0)

JOptionPane.showMessageDialog(getParent(),"Insert thất bại");

else

JOptionPane.showMessageDialog(getParent(),"Insert thành công");

}

catch(SQLException ex)

{

System.out.println(ex.getMessage());

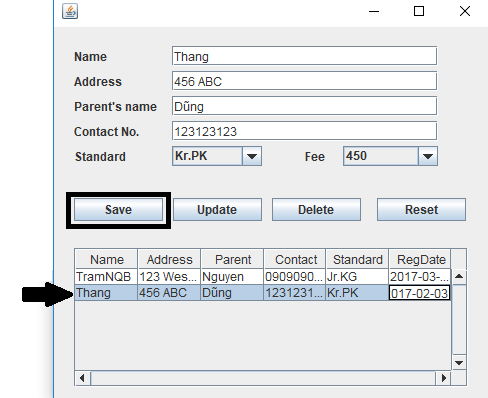
}

loadTable();

}

});

Kết quả:



**Bước 15:** Khai báo biến **regID** để giữ RegID của dòng dữ liệu đang được chọn **( Code phần chữ in đậm )**

public class Exercise extends JFrame {

private JPanel contentPane;

private JTextField txtName;

………

………

**String regID;**

Code nội dung cho sự kiện click mouse vào từng dòng trên table : dữ liệu tại dòng được chọn sẽ hiển thị ngược lại trên form

table.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent arg0) {

int i = table.getSelectedRow();

regID = model.getValueAt(i, 0).toString();

String name = model.getValueAt(i, 1).toString();

String address = model.getValueAt(i, 2).toString();

String parent = model.getValueAt(i, 3).toString();

String contact = model.getValueAt(i, 4).toString();

String standard= model.getValueAt(i, 5).toString();

txtName.setText(name);

txtAddress.setText(address);

txtContactNo.setText(contact);

txtParentName.setText(parent);

cbStandard.setSelectedItem(standard);

}

});

**Bước 16: Code cho sự kiện của nút Update**

JButton btnUpdate = new JButton("Update");

btnUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(!valid())

return;

String name = txtName.getText();

String address = txtAddress.getText();

String parentName = txtParentName.getText();

String phone = txtContactNo.getText();

String standard = (String) cbStandard.getSelectedItem();

conn = new MyConnect().getcn();

try

{

String sql = "update Student set Name=?,Address=?,parentName=?,phone=?,standard=? where RegID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setString(1,name);

ps.setString(2,address);

ps.setString(3,parentName);

ps.setString(4,phone);

ps.setString(5,standard);

ps.setString(6,regID);// regID của dòng được chọn

int kq = ps.executeUpdate();

if(kq==0)

JOptionPane.showMessageDialog(getParent(),"Update thất bại");

else

JOptionPane.showMessageDialog(getParent(),"Update thành công");

}

catch(SQLException ex)

{

System.out.println(ex.getMessage());

}

loadTable();

}

});

**Bước 17 : Code cho sự kiện của nút Delete**

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

conn = new MyConnect().getcn();

try

{

String sql = "delete from Student where RegID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setString(1,regID);

int kq = ps.executeUpdate();

if(kq==0)

JOptionPane.showMessageDialog(getParent(),"Delete thất bại");

else

JOptionPane.showMessageDialog(getParent(),"Delete thành công");

}

catch(SQLException ex)

{

System.out.println(ex.getMessage());

}

loadTable();

}

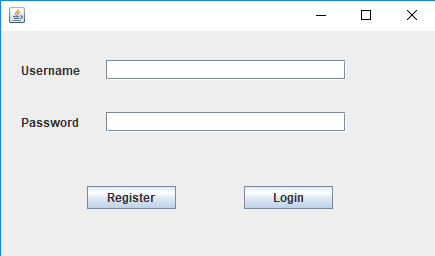
});

**Bước 18 : Nâng cao chức năng Delete :** Trước khi thực hiện delete database , chương trình sẽ hiện bảng Confirm xác nhận lại “Bạn có thực sự muốn xóa dữ liệu hay không?” , Nếu người dùng nhấn “Yes” thì thực hiện thao tác xóa.

**Bước 19:** Sinh viên tự code cho sự kiện của nút Reset : thực hiện reset toàn bộ form input dữ liệu

JAVA SWING – JDBC ( 2 )

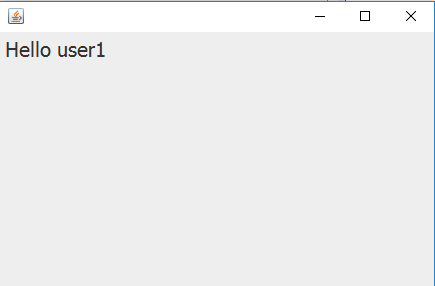
**BÀI TẬP 1 : Thiết kề và thực hiện các chức năng như màn hình sau:**



**1**. **Chức năng Register** : thực hiện tạo mới 1 account dùng để login ( Thực hiện mã hóa cho password)

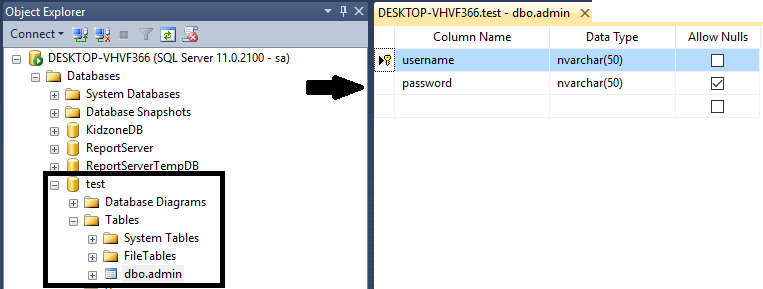
**2.** **Chức năng Login**: Kiểm tra account login (username và password ) phải trùng khớp với account đã lưu trữ trong database.

- Khi thực hiện login bằng username : **user1** => thành công . Màn hình (JFrame) Login sẽ ẩn đi và bật lên màn hình (JFrame) mới có chữ Hello user1 ( Tức username vừa login thành công )



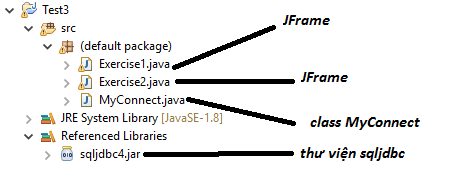
1. **THIẾT KẾ DATABASE**

Tạo database có tên **test** . Gồm 1 table : **admin** ( bảng admin không có dữ liệu)



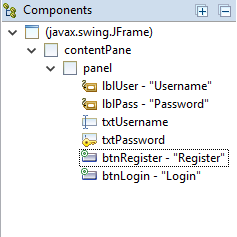
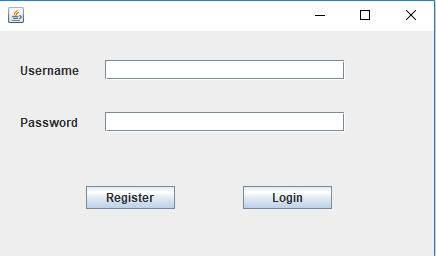
1. **HƯỚNG DẪN CODE GIAO DIỆN JFRAME**

**Bước 1: Tạo project như sau:**

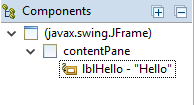
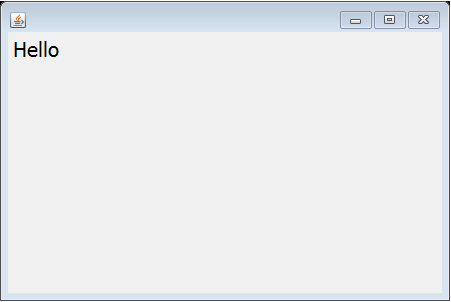
****

**Bước 2: Giao diện**

1. **JFrame Exercise1.java**

****

1. **JFrame Exercise2.java**

****

**Bước 3: Code cho nút Register**

1. Thực hiện chức năng insert account mới ( gồm username và password ) vào table admin trong database

btnRegister.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

try {

cn = new MyConnect().getcn();

if(cn==null)

{

JOptionPane.showMessageDialog(getParent(),"Kết nối thất bại");

return;

}

**//Kiểm tra user có tồn tại trước đó chưa ( không được trùng username)**

String user = txtUsername.getText();

String sql = "select \* from admin where username like ?";

PreparedStatement ps = cn.prepareCall(sql);

ps.setString(1,user);

ResultSet rs = ps.executeQuery();

if(rs.next())

{

JOptionPane.showMessageDialog(getParent(), "Username đã tồn tại");

return;

}

**//Lấy password**

String strpass= new String(txtPassword.getPassword());

**//Mã hóa password bằng thuật toán SHA-1**

MessageDigest md = MessageDigest.getInstance("SHA-1");

byte []bytepass = md.digest(strpass.getBytes());

**// Insert account mới (Username, Password) vào database**

sql = "insert into admin values(?,?)";

ps = cn.prepareStatement(sql);

ps.setString(1,user);

ps.setString(2, new String(bytepass));//convert mảng byte về String

int kq = ps.executeUpdate();

if(kq==0)

JOptionPane.showMessageDialog(getParent(),"Đăng ký thất bại");

else

JOptionPane.showMessageDialog(getParent(),"Đăng ký thành công");

**//Đóng tất cả kết nối**

ps.close();

cn.close();

} catch (NoSuchAlgorithmException ex) {

System.out.print(ex.getMessage());

} catch (SQLException ex) {

System.out.print(ex.getMessage());

}

}

});

1. **Run chương trình để kiểm tra chức năng Register**

**Bước 4: Code cho nút Login**

* Khi thực hiện Login thành công, JFrame Exercise2 sẽ hiện lên với dòng chữ “Hello User1” ( nếu User1 là username dùng để login)

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

cn = new MyConnect().getcn();

if(cn==null)

{

JOptionPane.showMessageDialog(getParent(),"Login thất bại");

return;

}

try

{

String user = txtUsername.getText();

String strpass= new String(txtPassword.getPassword());

MessageDigest md = MessageDigest.getInstance("SHA-1");

byte[] bytepass = md.digest(strpass.getBytes());

//**Sau khi mã hóa, chuyển ngược array byte về String để chuẩn bị so sánh với chuỗi String password lấy từ database**

String pass = new String(bytepass);

**//Lấy data theo username vừa login**

String sql ="select \* from admin where username like ?";

PreparedStatement ps = cn.prepareStatement(sql);

ps.setString(1,user);

ResultSet rs = ps.executeQuery();

if(rs.next())

{

if(pass.equals(rs.getString(2)))

{

JOptionPane.showMessageDialog(getParent(),"Login thành công");

**//Tạo đối tượng từ JFrame Exercise2=> để điều khiển mở JFrame Exercise2 lên**

Exercise2 ex2 = new Exercise2(user);

ex2.setVisible(true);

dispose();// lệnh dispose(); để ẩn JFrame hiện hành

}

else

JOptionPane.showMessageDialog(getParent(),"Login thất bại");

}

else

JOptionPane.showMessageDialog(getParent(),"Username ko tồn tại");

}// Đóng ngoặc của try

catch(SQLException e)

{

System.out.println(e.getMessage());

} catch (NoSuchAlgorithmException ex) {

System.out.println(ex.getMessage());

}

}

});

**Chờ thực hiện xong bước 5 , mới run chương trình để kiểm tra chức năng Login**

**Bước 5 : Thực hiện code bên JFrame Exercise2**

1. Khóa 2 dòng lệnh **( in đậm**) trong void main() của **JFrame Exercise2**

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

// Khóa 2 dòng lệnh dưới đây

**//Exercise2 frame = new Exercise2();**

**//frame.setVisible(true);**

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

1. Khai báo biến cho class Exercise2 **( thêm dòng in đậm )**

public class Exercise2 extends JFrame {

private JPanel contentPane;

**private String username;**

. . . . . .

. . . . . .

}

1. Truyền thêm biến vào hàm dựng của JFrame Exercise2 **( thêm code in đậm )**

public Exercise2(**String userlogin**) {

**this.username = userlogin;**

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5,

. . . . . .

. . . . . .

}

1. Gắn Hello User vào Label **( thêm dòng in đậm )**

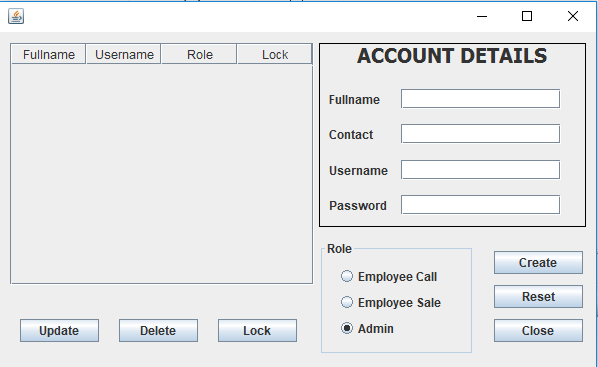
lblHello.setFont(new Font("Tahoma", Font.PLAIN, 20));

contentPane.add(lblHello, BorderLayout.NORTH);

**lblHello.setText(lblHello.getText()+ " " + username);**

**Bước 6: Run chương trình để kiểm tra chức năng Login**

**BÀI TẬP 2**

****

**Sinh viên tự thực hiện :**

1. Phân tích và thiết kế database
2. Code toàn bộ các tính năng của form