

### Bài 9: Thiết kế giao diện như mô tả sau:

The screenshot shows a macOS-style window titled "btdTrinh-QuanLyThongTinSinhVien". The main title "QUẢN LÝ THÔNG TIN SINH VIÊN" is in red. The form contains the following fields and controls:

- MSSV:** A text input field.
- Qualitication:** A dropdown menu with "Graduate" selected.
- Name:** A text input field.
- Age:** A text input field.
- Mark:** A text input field.
- Address:** A large text area.
- Hoppy:** A group box containing three checkboxes: "Reading", "Singing", and "Dacing".
- Sex:** A group box containing two radio buttons: "Male" and "Female".
- Buttons:** "Validate" and "Reset" buttons at the bottom.

#### ✓ Kiểm lỗi nhập liệu trường hợp người dùng không nhập đủ thông tin

This screenshot shows the same application window as above, but with a validation error message dialog box displayed over the form. The dialog box is titled "Message" and contains the following text:

- Không được để trống MSSV;
- Chưa nhập Họ Tên sinh viên;
- Tuổi từ 18->35
- Điểm từ 0.0 -> 10.0
- Chưa chọn sở thích
- Chưa nhập địa chỉ

The "OK" button is at the bottom right of the dialog. In the background, the form fields for MSSV, Name, and Age are highlighted in red, indicating they are required and currently empty or invalid. The "Validate" button is also visible at the bottom of the window.

#### ✓ Trường hợp người dùng nhập đủ thông tin nhưng sai tiêu chí yêu cầu thì cũng cảnh báo lỗi nhập liệu

## HƯỚNG DẪN:

### 1. Viết class *Validator.java* để thực hiện kiểm lỗi

```
1 package baitapthuchanhGUI3.quanlythongtinsinhvien;
2 import java.awt.Color;
3 import javax.swing.JTextField;
4 /**
5  *
6  * @author btdtrinh
7  */
8 public class Validator {
9     //KIEM TRA RONG
10    public static boolean isEmpty(JTextField txtField, StringBuilder str,
11        String msg){...10 lines }
12    //KIEM TRA SO
13    public static boolean isNumber(JTextField txtField, StringBuilder str,
14        String msg, int type){...26 lines }
15
16    //KIEM TRA TUOI NHAP TU 18-35
17    public static boolean isNumberMinMax(JTextField txtField, StringBuilder str,
18        String msg, int type, double min, double max) {...27 lines }
19 }
```

Chi tiết từng hàm thành phần được mô tả như sau

```
9 //KIEM TRA RONG
10 public static boolean isEmpty(JTextField txtField, StringBuilder str,
11     String msg){
12     String strField = txtField.getText().trim();//Cắt khoản trắng 2 đầu
13     if(strField.equals( anObject:"")){
14         str.append( str:msg).append( str:"\n");
15         txtField.setBackground( bg:Color.red);
16         return false;
17     }
18     txtField.setBackground( bg:Color.WHITE);
19     return true;
20 }
```

```

21 //KIEM TRA SO
22 public static boolean isNumber(JTextField txtField, StringBuilder str,
23     String msg, int type){
24     if(!isEmpty(txtField, str, msg)){
25         return false;
26     }else{
27         try{
28             // Integer.parseInt(txtField.getText());
29             // txtField.setBackground(Color.WHITE);
30             switch(type){
31                 case 0: //so nguyen
32                     Integer.parseInt(s:txtField.getText());
33                     txtField.setBackground( bg:Color.WHITE);
34                     break;
35                 case 1: //so thuc
36                     Double.parseDouble(s:txtField.getText());
37                     txtField.setBackground( bg:Color.WHITE);
38                     break;
39             }
40         }catch(Exception e){
41             System.out.println("ERROR Validator isNumber: " +e.toString());
42             str.append( str:msg).append( str:"\n");
43             txtField.setBackground( bg:Color.red);
44             return false;
45         }
46     }
47     return true;
48 }

```

```

50 //KIEM TRA TUOI NHAP TU 18-35
51 public static boolean isNumberMinMax(JTextField txtField, StringBuilder str,
52     String msg, int type, double min, double max) {
53     //Kiem tra phai so hay ko
54     if(!isNumber(txtField, str, msg, type)){
55         return false;
56     }else{
57         switch(type){
58             case 0: //Integer
59                 int numberInt = Integer.parseInt( s:txtField.getText());
60                 if(numberInt < min || numberInt > max){
61                     str.append( str:msg).append( str:"\n");
62                     txtField.setBackground( bg:Color.red);
63                     return false;
64                 }
65                 break;
66             case 1: //Double
67                 double numberDouble = Double.parseDouble( s:txtField.getText());
68                 if(numberDouble < min || numberDouble > max){
69                     str.append( str:msg).append( str:"\n");
70                     txtField.setBackground( bg:Color.red);
71                     return false;
72                 }
73                 break;
74             }
75         }
76         txtField.setBackground( bg:Color.WHITE);
77         return true;
78     }
79 }

```

## 2. Viết Form giao diện để thực hiện xử lý các yêu cầu

```

1 package baitapthuchanhGUI3.quanlythongtinsinhvien;
2 import javax.swing.JOptionPane;
3 /**
4  *
5  * @author btdtrinh
6  */
7 public class frmQuanLyThongTinSV extends javax.swing.JFrame {
8     /**
9      * Creates new form frmQuanLyThongTinSV
10     */
11     public frmQuanLyThongTinSV() {
12         initComponents();
13         rdMale.setSelected( b:true);
14         setLocationRelativeTo( c:null);
15     }
16     /** This method is called from within the constructor to initialize the form
21     @SuppressWarnings("unchecked")
22     Generated Code
271
272     public boolean validateForm() {...25 lines }
273     private void btnValidateActionPerformed(java.awt.event.ActionEvent evt) {...
331
332     private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {...15
347
348     /**
349     * @param args the command line arguments
350     */
351     public static void main(String args[]) {
352         /* Set the Nimbus look and feel */
353         Look and feel setting code (optional)
374
375         /* Create and display the form */
376         java.awt.EventQueue.invokeLater(new Runnable() {...5 lines });
381     }

```

*Chi tiết từng hàm thành phần được mô tả như sau:*

a) Hàm kiểm tra dữ liệu được nhập vào: `validateForm()`

```

272 public boolean validateForm(){
273     StringBuilder sb = new StringBuilder();
274     Validator.isEmpty( txtField:txtID, str:sb,
275         msg: "Không được để trống MSSV;");
276     Validator.isEmpty( txtField:txtName, str:sb,
277         msg: "Chưa nhập Họ Tên sinh viên;");
278     Validator.isNumberMinMax( txtField:txtAge, str:sb,
279         msg: "Tuổi từ 18->35", type:0, min:18, max:35);
280     Validator.isNumberMinMax( txtField:txtMark, str:sb,
281         msg: "Điểm từ 0.0 -> 10.0", type:1, min:1, max:10);
282
283     //BAY LOI, nếu ko được chọn thì nhắc lỗi
284     if(!rdMale.isSelected() && !rdFemale.isSelected()){
285         sb.append( str:"Chưa chọn giới tính\n");
286     }
287     if(!chkDancing.isSelected() && !chkReading.isSelected()
288         && !chkSinging.isSelected()){
289         sb.append( str:"Chưa chọn sở thích\n");
290     }
291
292     if(txtAddress.getText().isEmpty()){
293         sb.append( str:"Chưa nhập địa chỉ\n");
294     }
295     if(sb.length() > 0){
296         JOptionPane.showMessageDialog( parentComponent: this,
297             message: sb.toString());
298         return false;
299     }
300
301     return true;
302 }

```

*b) Hàm xử lý cho nút hành động btnValidate*

```

304 private void btnValidateActionPerformed(java.awt.event.ActionEvent evt) {
305     // TODO add your handling code here:
306     if(validateForm()){
307         StringBuilder str = new StringBuilder();
308         str.append( str:"ID: ");
309         str.append( str:txtID.getText());    str.append( str:"\n");
310         str.append( str:"Name: ");
311         str.append( str:txtName.getText());    str.append( str:"\n");
312         str.append( str:"Age: ");
313         str.append( str:txtAge.getText());    str.append( str:"\n");
314         str.append( str:"Giới tính: ");
315         if(rdFemale.isSelected()){
316             str.append( str:"Nữ"); str.append( str:"\n");
317         }
318         if(rdMale.isSelected()){
319             str.append( str:"Nam");    str.append( str:"\n");
320         }
321         str.append( str:"Sở thích: ");
322         if(chkDancing.isSelected()){
323             str.append( str:"Dancing");    str.append( str:"; ");
324         }
325         if(chkReading.isSelected()){
326             str.append( str:"Reading");    str.append( str:"; ");
327         }
328         if(chkSinging.isSelected()){
329             str.append( str:"Singing");    str.append( str:"\n");
330         }
331         str.append( str:"\nTrạng thái: ");
332         str.append( str:cbQualitication.getSelectedItem().toString());
333         str.append( str:"\n");
334         JOptionPane.showMessageDialog( parentComponent: this, message:str.toString());
335     }
336 }

```

a) *Hàm xử lý cho cho nút hành động btnReset*

```

339 private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
340     // TODO add your handling code here:
341     txtID.setText( t:"");
342     txtName.setText( t:"");
343     txtAge.setText( t:"");
344     txtMark.setText( t:"");
345     txtAddress.setText( t:"");
346     rdMale.setSelected( b:true); //Mac dinh
347
348     chkDancing.setSelected( b:false);
349     chkReading.setSelected( b:false);
350     chkSinging.setSelected( b:false);
351
352 }

```