

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

btdTrinh-QuanLyThongTinSinhVien

QUẢN LÝ THÔNG TIN SINH VIEN

MSSV: Qualification:

Name:

Age: Mark:

Address:

Sex: Male Female

Hobby: Reading Singing Dacing

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

btdTrinh-QuanLyThongTinSinhVien

QUẢN LÝ THÔNG TIN SINH VIEN

MSSV:

Name:

Age:

Address:

Sex: Male Female

Message

i 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

Reading Singing Dacing

- ✓ 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
20}
```

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(); //Cat khoan trang 2 dau
13     if(strField.equals("")){
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             ////
29             Integer.parseInt(txtField.getText());
30             txtField.setBackground(Color.WHITE);
31             switch(type){
32                 case 0: //so nguyen
33                     Integer.parseInt(s:txtField.getText());
34                     txtField.setBackground(bg:Color.WHITE);
35                     break;
36                 case 1: //so thuc
37                     Double.parseDouble(s:txtField.getText());
38                     txtField.setBackground(bg:Color.WHITE);
39                     break;
40             }
41             }catch(Exception e){
42                 System.out.println("ERROR Validator isNumber: " +e.toString());
43                 str.append(str:msg).append(str:"\n");
44                 txtField.setBackground(bg:Color.red);
45             return false;
46         }
47     }
48     return true;
49 }

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

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 nháy 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             String str = new StringBuilder();  
308             str.append("ID: ");  
309             str.append(txtID.getText()); str.append("\n");  
310             str.append("Name: ");  
311             str.append(txtName.getText()); str.append("\n");  
312             str.append("Age: ");  
313             str.append(txtAge.getText()); str.append("\n");  
314             str.append("Giới tính: ");  
315             if(rdFemale.isSelected()){  
316                 str.append("Nữ"); str.append("\n");  
317             }  
318             if(rdMale.isSelected()){  
319                 str.append("Nam"); str.append("\n");  
320             }  
321             str.append("Sở thích: ");  
322             if(chkDancing.isSelected()){  
323                 str.append("Dancing"); str.append("; ");  
324             }  
325             if(chkReading.isSelected()){  
326                 str.append("Reading"); str.append("; ");  
327             }  
328             if(chkSinging.isSelected()){  
329                 str.append("Singing"); str.append("\n");  
330             }  
331             str.append("\nTrạng thái: ");  
332             str.append(cbQualification.getSelectedItem().toString());  
333             str.append("\n");  
334             JOptionPane.showMessageDialog(parentComponent: this, message: str.toString());  
335         }  
336     }  
337 }
```

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("");  
342         txtName.setText("");  
343         txtAge.setText("");  
344         txtMark.setText("");  
345         txtAddress.setText("");  
346         rdMale.setSelected( b: true); //Mac dinh  
347         chkDancing.setSelected( b: false);  
348         chkReading.setSelected( b: false);  
349         chkSinging.setSelected( b: false);  
350     }  
351 }
```