

LAB 4: NGOẠI LỆ

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng lệnh try...catch để kiểm tra tính hợp lệ của dữ liệu
- ✓ Xây dựng thư viện xử lý các lỗi thường gặp

PHẦN I

Bài 1 (2 điểm)

Thiết kế giao diện như hình sau



- ✓ Đặt tên theo qui ước
- ✓ Đưa form ra giữa màn hình

Bài 2 (2 điểm)

Viết mã cho nút [KIỂM TRA] để kiểm tra các lỗi sau:

- ✓ Không để trống tất cả các ô nhập
- ✓ Ngày sinh phải đúng định dạng dd-MM-yyyy
- ✓ Lương phải là số dương
- ✓ Tô nền vàng các ô gặp lỗi và đưa ra thông báo phù hợp

HƯỚNG DẪN: Code tại nút Kiểm tra

- ✓ Sử dụng getText() để lấy dữ liệu nhập vào form
- ✓ Sử dụng phương thức length() để kiểm tra xem chuỗi có rỗng hay không
- ✓ Sử dụng Double.parseDouble() để đổi chuỗi thành số kết hợp với try...catch
- ✓ Sử dụng SimpleDateFormat để đổi chuỗi thành Date kết hợp với try...catch

Kiểm tra thông tin nhập không được bỏ trống txtHoTen (tương tự cho các text khác)

```
String name = txtHoTen.getText();
if (name.length()==0) {
    JOptionPane.showMessageDialog(this, "Ho va ten khong bo trong!");
    txtHoTen.setBackground(Color.yellow);
    txtHoTen.requestFocus();
    return;
}else{
    txtHoTen.setBackground(Color.white);
}
```

Kiểm tra thông tin nhập txtNgaSinh, txtLuong

```
try{
    SimpleDateFormat formater = new SimpleDateFormat();
    formater.applyPattern("dd-MM-yyyy");
    Date ngay = formater.parse(txtNgaySinh.getText());
    txtNgaySinh.setBackground(Color.white);
    double luong = Double.parseDouble(txtLuong.getText());
    txtLuong.setBackground(Color.white);
}catch(NumberFormatException ex){
    JOptionPane.showMessageDialog(this, "Luong khong hop le!");
    txtLuong.setBackground(Color.yellow);
    txtLuong.requestFocus();
    return;
}catch(Exception ex){
    JOptionPane.showMessageDialog(this, "Ngày sinh khong hop le!");
    txtNgaySinh.setBackground(java.awt.Color.yellow);
    txtNgaySinh.requestFocus();
    return;
}
JOptionPane.showMessageDialog(this, "Them thanh cong!");
```

PHẦN II

Bài 3 (2 điểm)

Xây dựng thư viện tiện ích có tên XDate chứa hàm parse để chuyển đổi String sang Date theo định dạng. Hàm parse() được khai báo như sau

```
public class XDate {  
    private static SimpleDateFormat formater = new SimpleDateFormat();  
    public static Date parse(String text, String pattern) throws RuntimeException{  
        try {  
            formater.applyPattern(pattern);  
            return formater.parse(text);  
        }  
        catch (Exception e) {  
            throw new RuntimeException(e);  
        }  
    }  
    public static Date parse(String text) throws RuntimeException{  
        return XDate.parse(text, "dd-MM-yyyy");  
    }  
}
```

Bài 4 (2 điểm) Sử dụng thư viện XDate ở bài 3 để kiểm tra lỗi ngày sinh ở bài số 2.

```
try {  
    SimpleDateFormat formater = new SimpleDateFormat();  
    formater.applyPattern("dd-MM-yyyy");  
    Date ngaySinh = formater.parse(birthday);  
}  
catch (Exception e) {  
    JOptionPane.showMessageDialog(this, "Không đúng định dạng ngày!");  
    txtHoTen.setBackground(Color.YELLOW);  
}
```

```
try {  
    Date ngaySinh = XDate.parse(birthday, "dd-MM-yyyy");  
}  
catch (Exception e) {  
    JOptionPane.showMessageDialog(this, "Không đúng định dạng ngày!");  
    txtHoTen.setBackground(Color.YELLOW);  
}
```

Hết