



# **BÀI TẬP**

## **CHƯƠNG TRÌNH LẬP TRÌNH VIÊN JAVA**

---

### **MODULE 1: JAVA CƠ BẢN**



# BÀI 1: Tổng quan



*Mục tiêu chính:*

*Làm quen với môi trường làm việc với **Java** (Eclipse - Cài đặt gói **Windows Builder Pro** để tạo giao diện cho ứng dụng/ NetBeans)*

*Cách tổ chức ứng dụng Java*

*Tạo ứng dụng đơn giản*

**Quy ước đặt tên: (chỉ chọn một trong hai phong cách và sử dụng từ đầu đến cuối)**

## Phong cách 1:

- Tên của thể hiện: Th\_Ten\_the\_hien
- Tên của xử lý: Xl\_Ten\_xu\_ly
- Tên biến: Viết hoa ký tự đầu của từ đầu tiên, các từ sau viết thường, các từ cách nhau bằng dấu '\_' với biến có từ hai từ trở lên (Ví dụ: Chieu\_dai, Dien\_tich), viết thường nếu có một từ (Ví dụ: dai, rong, s, p)
- Tên hằng: Viết hoa tất cả các ký tự, các từ cách nhau bằng dấu gạch '\_' (Ví dụ: PI, TY\_GIA)
- Tên Class/Object: Viết hoa ký tự đầu tiên của từng từ, mỗi từ cách nhau bằng dấu '\_'
- Tên hàm: Viết hoa ký tự đầu của từ đầu tiên, các từ sau viết thường, các từ cách nhau bằng dấu '\_' với hàm có từ hai từ trở lên (Ví dụ: Dien\_tich, Tinh\_tong), viết thường nếu có một từ (Ví dụ: tong, hieu)

## Phong cách 2 (theo Camel Case):

- Tên của thể hiện: thTenTheHien
- Tên của xử lý: xlTenXuLy



- Tên biến: Viết thường ký tự đầu của từ đầu tiên, các từ sau viết hoa đầu từ (Ví dụ: chieuDai, dienTich), viết thường nếu có một từ (Ví dụ: dai, rong, s, p)
- Tên hằng: Viết hoa tất cả các ký tự, các từ cách nhau bằng dấu gạch '\_' (Ví dụ: PI, TY\_GIA)
- Tên Class/Object: Viết hoa ký tự đầu tiên của tất cả các từ (Ví dụ: SinhVien, ContactBook)
- Tên hàm: Viết thường ký tự đầu của từ đầu tiên, các từ sau viết hoa đầu từ (Ví dụ: dienTich, tinhTong), viết thường nếu có một từ (Ví dụ: tong, hieu)

### Quy ước viết tắt các điều khiển - prefix (chung):

- form: frm (vd: phong cách 1: frm\_Dang\_nhap, phong cách 2: frmDangNhap)
- combobox, dropdown listbox: cbo
- checkbox: chk
- image: img
- lable: lbl
- list/menu: lst/mnu
- textfield/textArea : txt
- button: btn
- radio button: rdo

## 1.1. Vẽ "HELLO" ra màn hình console

- ✓ **Yêu cầu:** Viết chương trình xuất "Hello" như hình vẽ ra màn hình console

```

**  **  *****  **  **  **
**  **  **  **  **  **  **
**  **  **  **  **  **  **
*****  *****  **  **  **
**  **  **  **  **  **  **
**  **  **  **  **  **  **
**  **  *****  *****  *****  **

```



✓ **Hướng dẫn sử dụng:**

- Xuất màn hình console in ra " HELLO" như hình trên

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Không có

▪ **Xuất:**

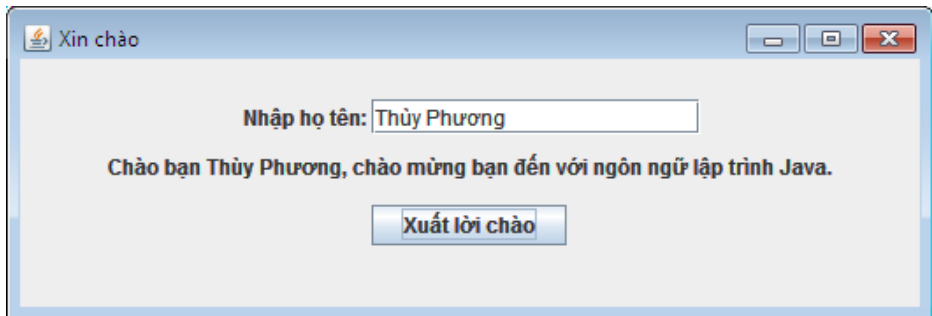
- HELLO

✓ **Hướng dẫn**

- Sử dụng lệnh `System.out.println("Nội dung");` xuất nội dung ra màn hình console

## 1.2. Xuất câu chào

✓ **Yêu cầu: Xây dựng màn hình Xuất câu chào**



✓ **Hướng dẫn sử dụng:**

- Nhập họ tên Sau đó nhấn nút "Xuất câu chào"

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

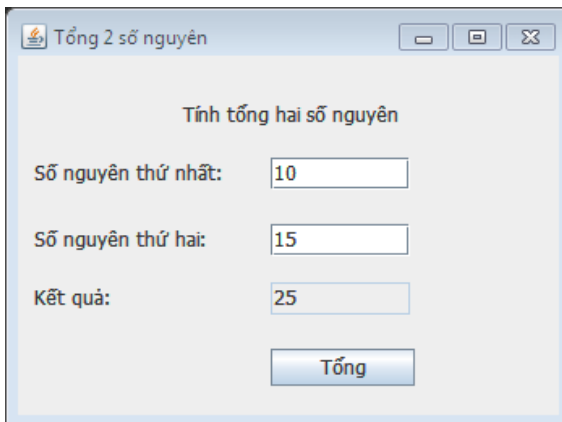
- `frmManHinhXuatCauChao`: `FrmXuatCauChao` (extends từ `JFrame`)

\* (Các thể hiện phía dưới đều nằm trong *Frame*)

- txtHoTen: JTextField (nhập liệu)
- lblCauchao: JLabel (Kết xuất)
- btnXuatCauChao: JButton
- **Nhập:**
  - Họ tên
- **Xuất:**
  - Câu chào
- ✓ **Thuật giải**
  - Khai báo biến HoTen nhận giá trị của txtHoTen
  - Kết xuất Câu chào + HoTen

### 1.3. Tính tổng hai số nguyên

- ✓ **Yêu cầu: Xây dựng màn hình viết chương trình tính tổng hai số nguyên**



- ✓ **Hướng dẫn sử dụng:**
  - Nhập số nguyên thứ nhất, số nguyên thứ hai sau đó nhấn nút “Tổng”
- ✓ **Tóm tắt yêu cầu**



- **Thiết kế giao diện người dùng:**

- frmManHinhTong2SoNguyen: JFrame

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtSoNguyenThuNhat: JTextField (nhập liệu)
- txtSoNguyenThuHai: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTong: JButton (xử lý tính tổng hai số nguyên)

- **Nhập:**

- Số nguyên thứ nhất
- Số nguyên thứ hai

- **Xuất:**

- Kết quả (Tổng hai số nguyên)

- **Qui tắc xử lý:**

- Tổng hai số nguyên = Số nguyên thứ nhất + Số nguyên thứ hai

✓ **Thuật giải**

- Khai báo biến soNguyenThuNhat nhận giá trị của txtSoNguyenThuNhat
- Khai báo biến soNguyenThuHai nhận giá trị của txtSoNguyenThuHai
- Chuyển giá trị 2 chuỗi sang 2 số nguyên
- Khai báo biến tongHaiSoNguyen
- Xử lý tính Tổng hai số nguyên ( $tongHaiSoNguyen = soNguyenThuNhat + soNguyenThuHai$ )
- Kết xuất tongHaiSoNguyen ra txtKetQua

✓ **Hướng dẫn**

- Hàm chuyển từ chuỗi sang số nguyên:  
**Integer.parseInt(String)**

## 1.4. Tính tiền hàng

- ✓ **Yêu cầu: Xây dựng màn hình tính tiền hàng theo công thức:**  
**Thành tiền = Số lượng \* Đơn giá**



- ✓ **Hướng dẫn sử dụng:**
  - Nhập số lượng và đơn giá. Nhấn nút “Tính tiền”
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - frmManHinhTinhTien: FrmTinhTien (extends từ JFrame)
    - \* (Các thể hiện phía dưới đều nằm trong Frame)
      - txtSoLuong: JTextField (nhập liệu)
      - txtDonGia: JTextField (nhập liệu)
      - txtThanhTien: JTextField (kết xuất, chỉ đọc)
      - btnTinhTien: JButton (Xử lý tính Thành Tiền)
  - **Nhập:**
    - Số lượng



- Đơn giá
- **Xuất:**
  - Thành tiền
- **Qui tắc xử lý:**
  - Thành tiền = Số lượng x Đơn giá
- ✓ **Thuật giải**
  - Khai báo biến SoLuong nhận giá trị của txtSoLuong
  - Khai báo biến DonGia nhận giá trị của txtDonGia
  - Chuyển đổi dữ liệu từ kiểu chuỗi sang kiểu số
  - Khai báo biến ThanhTien
  - Xử lý tính thành tiền ( $\text{ThanhTien} = \text{SoLuong} * \text{DonGia}$ )
  - Kết xuất ThanhTien ra txtThanhTien
- ✓ **Hướng dẫn**
  - Hàm chuyển từ chuỗi sang số nguyên:  
`Integer.parseInt(String)`



## BÀI 2: Các kiểu dữ liệu cơ sở



*Mục tiêu chính:*

*Cung cấp kiến thức và kỹ năng thao tác với các kiểu dữ liệu cơ sở*

*Nắm được cú pháp ngôn ngữ Java trong lập trình ứng dụng*

*Ôn lại các toán tử, các cấu trúc điều khiển, cách xử lý lỗi*

### 2.1. Giải Phương trình bậc I

- ✓ **Yêu cầu:** Xây dựng màn hình giải phương trình bậc nhất  $ax + b = 0$ , với  $a \neq 0$



- ✓ **Hướng dẫn sử dụng:**
  - Nhập vào a, b. Nhấn nút “Giải PT” => Hiện thị nghiệm của phương trình.
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - frmManHinhGiaiPhuongTrinh: FrmGiaiPhuongTrinh JFrame



*\* (Các thể hiện phía dưới đều nằm trong Frame)*

- txtGiaTriA: JTextField (nhập liệu)
- txtGiaTriB: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)
- btnNhapLai: JButton (Xử lý xóa hết giá trị trong các field)
- **Nhập:**
  - Giá trị a
  - Giá trị b
- **Xuất:**
  - Kết quả
- **Qui tắc xử lý :**
  - $x = -b/a$
- ✓ **Thuật giải**
  - Khai báo biến giaTriA nhận giá trị của txtGiaTriA
  - Khai báo biến giaTriB nhận giá trị của txtGiaTriB
  - Chuyển giá trị kiểu chuỗi thành số (Float)
  - Khai báo biến ketQua
  - Xử lý tính kết quả ( $x = -b / a$ )
  - Kết xuất dữ liệu

## 2.2. Tính giai thừa của một số

- ✓ **Yêu cầu:** Xây dựng màn hình tính giai thừa của một số



✓ **Hướng dẫn sử dụng:**

- Nhập vào x. Nhấn nút “Tính giai thừa” => Hiển thị kết quả

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmManHinhTinhGiaiThua: FrmTinhGiaiThua (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtGiaTriX: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)
- btnNhapLai: JButton (Xử lý xóa hết giá trị trong các field)

▪ **Nhập:**

- Nhập giá trị x, là số nguyên dương

▪ **Xuất:**

- Kết quả tính giai thừa của x

▪ **Qui tắc xử lý :**



- $x! = 1 * 2 * 3 * \dots * (x - 1) * x$
- $0! = 1$

✓ **Thuật giải**

- Khai báo biến giaTriX nhận giá trị của txtGiaTriX
- Chuyển giá trị chuỗi sang số nguyên
- Khai báo biến ketQua
- Xử lý tính kết quả
- Kết xuất dữ liệu

✓ **Hướng dẫn**

```
private Long tinhGiaiThua(int x){
    Long kq = 1l;

    if(x < 0){
        return -1l;
    }

    for(int i = 1; i <= x; i++){
        kq = kq * i;
    }

    return kq;
}
```

## 2.3. Tìm kiếm chuỗi

- ✓ **Yêu cầu:** Xây dựng màn hình Tìm kiếm chuỗi



✓ **Hướng dẫn sử dụng:**

- Nhập vào chuỗi thứ nhất và chuỗi thứ hai. Nhấn nút “Tìm kiếm” => Xuất ra kết quả cho biết chuỗi thứ hai có nằm trong chuỗi thứ nhất hay không.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmManHinhTimKiemChuoi: FrmTimKiemChuoi (extends từ JFrame)

*\* (Các thể hiện phía dưới đều nằm trong Frame)*

- txtChuoi1: JTextField (nhập liệu)
- txtChuoi2: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTiemKiem: JButton (Xử lý giá trị biểu thức)

▪ **Nhập:**

- Chuỗi 1
- Chuỗi 2

▪ **Xuất:**

- Kết quả



▪ **Qui tắc xử lý :**

✓ **Thuật giải**

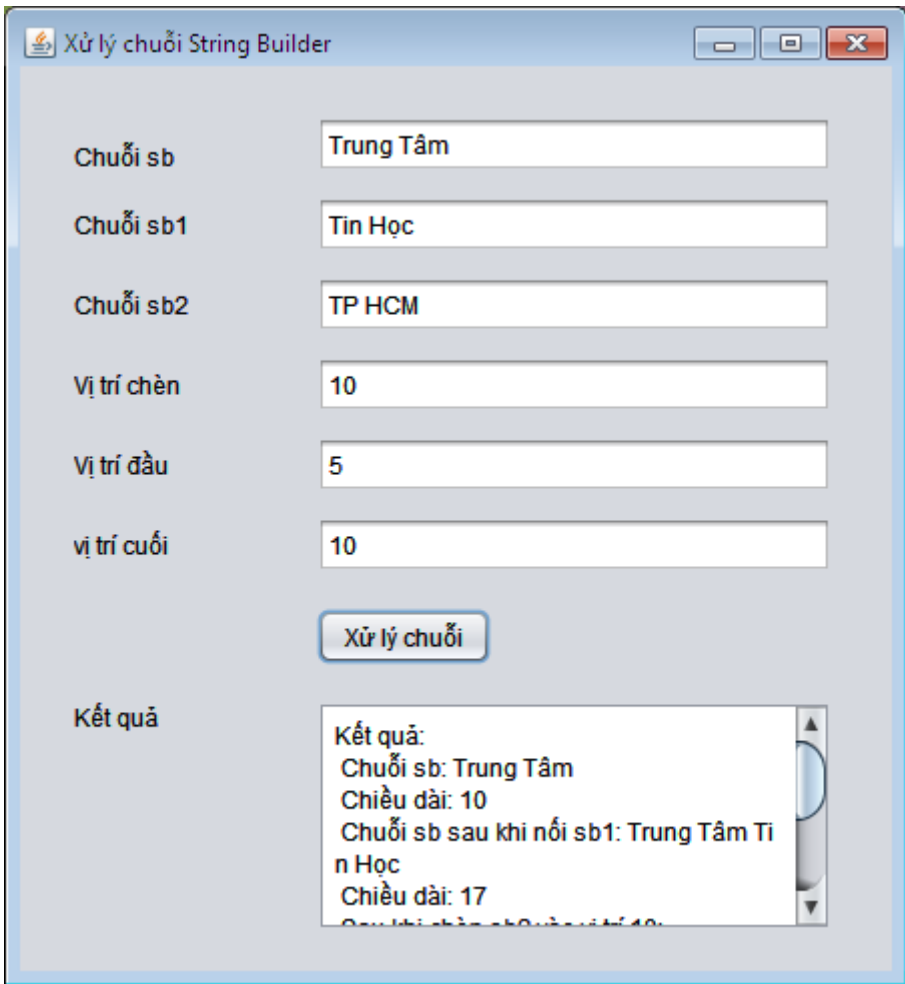
- Khai báo biến `chuoai1` nhận giá trị của `txtChuoai1`
- Khai báo biến `chuoai2` nhận giá trị của `txtChuoai2`
- Khai báo biến `ketQua`
- Xử lý tìm kiếm chuỗi và gán giá trị cho biến `ketQua`
- Kết xuất dữ liệu `ketQua` ra `txtKetQua`

✓ **Hướng dẫn**

- Dùng hàm `contains` của lớp `String` để xử lý

## 2.4. Xử lý chuỗi `StringBuilder`

- ✓ **Yêu cầu: Viết chương trình xử lý chuỗi bằng cách sử dụng lớp `StringBuilder` và các phương thức như sau:**



**Xử lý chuỗi String Builder**

Chuỗi sb: Trung Tâm

Chuỗi sb1: Tin Học

Chuỗi sb2: TP HCM

Vị trí chèn: 10

Vị trí đầu: 5

Vị trí cuối: 10

Xử lý chuỗi

Kết quả:

Kết quả:  
 Chuỗi sb: Trung Tâm  
 Chiều dài: 10  
 Chuỗi sb sau khi nối sb1: Trung Tâm Tin Học  
 Chiều dài: 17  
 Chuỗi sb sau khi nối sb2: Trung Tâm Tin Học TP HCM  
 Chiều dài: 23

✓ **Hướng dẫn sử dụng:**

- Nhập chuỗi nguồn sb, chuỗi sb1, chuỗi sb2, sb3, sb4, nhập vị trí chèn, vị trí đầu, vị trí cuối, bấm "Xử lý chuỗi" => thực hiện và in kết quả.

✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**



- frmXuLyChuoi: JFrame Form
- \* *(Các thể hiện phía dưới đều nằm trong form)*

- txtSb: TextField (nhập liệu)
- txtSb1: TextField (nhập liệu)
- txtSb2: TextField (nhập liệu)
- txtViTri: TextField (nhập liệu)
- txtBatDau: TextField (nhập liệu)
- txtKetThuc: TextField (nhập liệu)
- txtaResult: TextArea (kết xuất)
- btnXuLy: Button (xử lý chuỗi)

▪ **Nhập:**

- Chuỗi sb
- Chuỗi sb1 (nối vào sau sb)
- Chuỗi sb2 (chèn vào vị trí chèn trong sb)
- Vị trí chèn
- Vị trí đầu, vị trí cuối (để xóa)

▪ **Xuất:**

- Chiều dài chuỗi sb
- Chuỗi sb sau khi nối chuỗi sb1
- Chuỗi sb sau khi chèn sb2 vào tại vị trí chèn
- Chuỗi sb sau khi bị xóa nội dung từ vị trí đầu đến vị trí cuối
- Chuỗi sb sau khi bị đảo ngược

✓ **Hướng dẫn**

▪ **Trong phần xử lý của nút “Xử lý chuỗi”:**

- Khai báo các biến kiểu chuỗi: sb, sb1, sb2, viTri, batDau, ketThuc để nhận giá trị người dùng nhập trên màn hình





```
String sb = txtSb.getText();
String sb1 = txtSb1.getText();
String sb2 = txtSb2.getText();

int viTri = Integer.parseInt(txtViTri.getText());
int batDau = Integer.parseInt(txtBatDau.getText());
int ketThuc = Integer.parseInt(txtKetThuc.getText());
```

- Khai báo biến kiểu chuỗi ketQua để nhận các kết quả xuất ra TextArea.

```
String ketQua = "Kết quả:";
```

- Khai báo biến kiểu StringBuilder str > đưa sb vào, tính chiều dài > đưa sb1 vào, tính chiều dài > chèn sb2 vào str tại viTri > xóa str từ vị trí batDau đến vị trí ketThuc > đảo ngược chuỗi str

```
StringBuilder str = new StringBuilder(200);
str.append(sb);
ketQua += "\n Chuỗi sb: " + str;
ketQua += "\n Chiều dài: " + str.length();
str.append(sb1);
ketQua += "\n Chuỗi sb sau khi nối sb1: " + str;
ketQua += "\n Chiều dài: " + str.length();
str.insert(viTri, sb2);
ketQua += "\n Sau khi chèn sb2 vào vị trí " + viTri + ": \n" + str;
str.delete(batDau, ketThuc);
ketQua += "\n Sau khi xóa: \n" + str;
str.reverse();
```

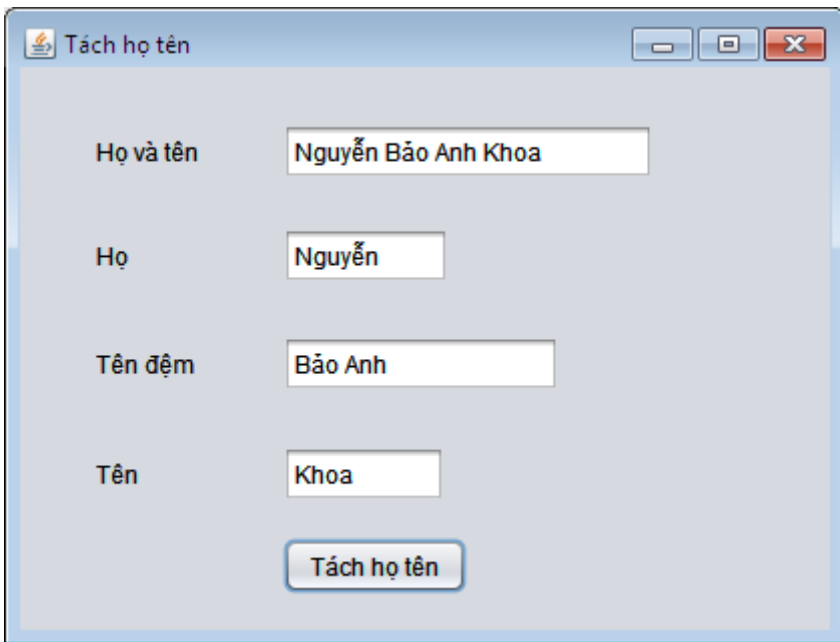
```
ketQua += "\n Sau khi đảo ngược: \n" + str;
```

- Xuất kết quả ra màn hình

```
txtaKetQua.setText(ketQua);
```

## 2.5. Phân tách họ tên

- ✓ **Yêu cầu:** Viết chương trình phân tách họ tên như sau:



*Phân tách họ tên*

- ✓ **Hướng dẫn sử dụng:**
  - Nhập họ tên => Nhấn "Tách họ tên" => Hiện thị Họ, Tên đệm, Tên
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**



- frmTachHoTen: JFrame (chính thuộc tính defaultCloseOperation của JFrame là DISPOSE)

*\* (Các thể hiện phía dưới đều nằm trong form)*

- txtHoTen: Text Field (nhập liệu)
- txtHo: Text Field (kết xuất)
- txtTenDem: Text Field (kết xuất)
- txtTen: Text Field (kết xuất)
- btnTach: Button (xử lý tách chuỗi và hiển thị)
- **Nhập:**
  - Họ tên
- **Xuất:**
  - Họ
  - Tên đệm
  - Tên

#### ✓ Thuật giải

- Sử dụng lớp StringTokenizer
  - Họ (token đầu tiên)
  - Tên đệm (các token ở giữa)
  - Tên (token cuối cùng)

## 2.6. Tính số ngày trong tháng

- ✓ **Yêu cầu:** Xây dựng màn hình Tính số ngày trong tháng



✓ **Hướng dẫn sử dụng:**

- Nhập vào tháng và năm. Nhấn nút “Tính” => Hiển thị số ngày trong tháng

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmManHinhTinhNgayTrongThang: FrmTinhNgayTrongThang (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtThang: JTextField (nhập liệu)
- txtNam: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)

▪ **Nhập:**

- Tháng
- Năm

▪ **Xuất:**

- Ngày trong tháng

▪ **Qui tắc xử lý :**



- Tháng 1, 3, 5, 7, 8, 10, 12 có 31 ngày
- Tháng 4, 6, 9, 11 có 30 ngày
- Tháng 2 năm thường có 28 ngày, năm nhuận có 29 ngày
- Năm nhuận là năm chia hết cho 4 và không chia hết cho 100 hoặc là năm chia hết cho 400

### ✓ Thuật giải

- Khai báo biến thang nhận giá trị của txtThang
- Khai báo biến nam nhận giá trị của txtNam
- Kiểm tra giá trị nhập
- Chuyển giá trị chuỗi sang số nguyên
- Khai báo biến ketQua
- Xử lý tính ngày theo tháng và năm sau đó gán giá trị cho biến ketQua. Lưu ý trường hợp năm nhuận.
- Kết xuất dữ liệu ra txtKetQua

### ✓ Hướng dẫn

Tính ngày trong tháng

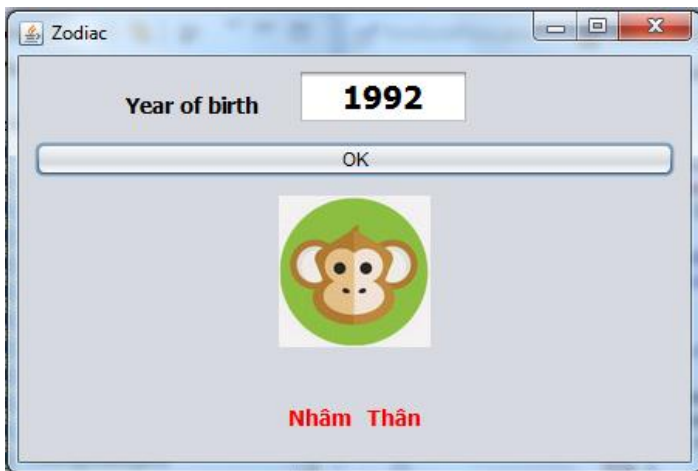
```
private int tinhNgayTrongThang(int thang, int nam){
    switch (thang) {
        case 1: case 3: case 5: case 7: case 8: case 10: case
12:
            return 31;
        case 4: case 6: case 9: case 11:
            return 30;
        case 2:
            if(laNamNhuan(nam)){
                return 29;
            }
            return 28;
    }
    return 0;
}
```

Kiểm tra năm nhuận

```
private boolean laNamNhuon(int nam){
    if((nam % 4 == 0 && nam % 100 != 0) || nam % 400 == 0){
        return true;
    }
    return false;
}
```

## 2.7. Tính năm âm lịch từ năm dương lịch

- ✓ **Yêu cầu:** Xây dựng chương trình tính năm âm lịch từ năm dương lịch



- ✓ **Hướng dẫn sử dụng:**
  - Khi người dùng nhập năm sinh, bấm nút “OK” thì hiển thị năm âm lịch (có Can chi) và hiển thị hình ảnh tương ứng
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmZodiac: form
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - txtYear: TextField (nhập liệu)
      - lblImage: Label (kết xuất, hiển thị ảnh kết quả)



- lblCan: Label (kết xuất, chỉ đọc)
- lblChi: Label (kết xuất, chỉ đọc)
- btnOk: Button (xử lý tính can chi)

▪ **Nhập:**

- Năm sinh

▪ **Xuất:**

- Năm âm lịch (Tên can chi)
- Hình ảnh đại diện của năm âm lịch

▪ **Qui tắc xử lý :**

- Can là lấy năm chia lấy dư cho 10

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

- Chi là lấy năm chia lấy dư cho 12

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mão	Thìn	Tỵ	Ngọ	mùi

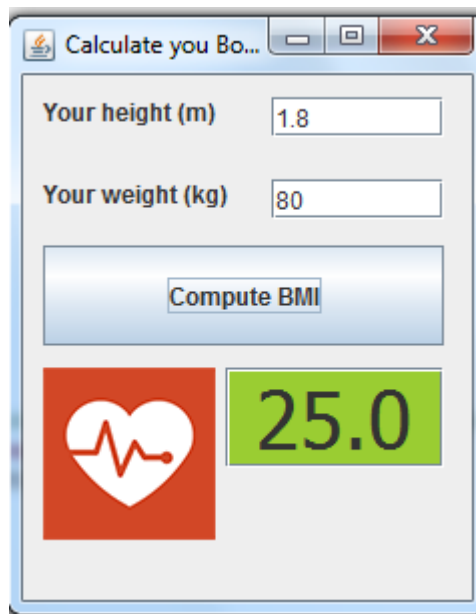
✓ **Hướng dẫn**

- *Chú ý: Các hình ảnh sử dụng trong module này được cung cấp sẵn trong thư mục images – HV có thể download trên cổng thông tin*
- Sử dụng: lblImage.**setIcon(new ImageIcon("./src/images/dau.jpg"))**; để hiển thị hình ảnh lên lable
- Sử dụng: lblCan.**setText("Nội dung")**; để thiết lập nội dung cho lable

- Viết phương thức tính can **tinhCan(int year)** có kết quả trả về là chuỗi can, tính chi **tinhChi(int year)** có kết quả trả về là chuỗi chi
- Ứng với chi => hiển thị hình ảnh tương ứng
- Sử dụng cấu trúc switch ... case để giải quyết bài toán theo quy tắc xử lý trên.

## 2.8. Tính chỉ số BMI

- ✓ **Yêu cầu: Xây dựng chương trình tính chỉ số BMI**



- ✓ **Hướng dẫn sử dụng:**
  - Khi người dùng nhập chiều cao, cân nặng , bấm nút "Compute BMI" thì hiển thị chỉ số BMI
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmCalculateYourBMI: form





*\* (Các thể hiện phía dưới đều nằm trong form)*

- txtHeight: TextField (nhập liệu)
- txtWeight: TextField (nhập liệu)
- lblImage: Label (Hiển thị ảnh)
- lblBmi: Label (kết xuất, chỉ đọc)
- btnCompute: Button (xử lý tính BMI)
- **Nhập:**
  - Chiều cao (m)
  - Cân nặng (kg)
- **Xuất:**
  - Chỉ số BMI của cơ thể
- **Qui tắc xử lý :**
  - $BMI = Weight / (Height * Height)$

## 2.9. Xem bói tình yêu (bài làm thêm)

- ✓ **Yêu cầu: Xây dựng chương trình xem bói tình yêu**



✓ **Hướng dẫn sử dụng:**

- Khi người dùng nhập 2 tên, bấm nút “Compute” thì hiển thị chỉ số

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- FrmXemBoiTinhYeu: form

\* (Các thể hiện phía dưới đều nằm trong form)

- txtName1: TextField (nhập liệu)
- txtName2: TextField (nhập liệu)
- lblImageBG: Label (Hiển thị ảnh nền)
- btnCompute: Button (xử lý tính)

▪ **Nhập:**

- Tên 2 người



▪ **Xuất:**

- Chỉ số phù hợp từ 2 tên

▪ **Quy tắc xử lý :**

- Xử lý dựa trên các chuỗi tên
- Điểm ban đầu = 30
- Nếu chiều dài của tên 1 > chiều dài của tên 2 thì điểm = điểm ban đầu + (chiều dài của tên 1 - chiều dài của tên 2)\*10. Ngược lại thì điểm = điểm ban đầu + (chiều dài của tên 2 - chiều dài của tên 1)\*10 + 15
- Nếu tên 1 chứa tên 2 thì điểm = điểm ban đầu + 25
- Nếu tên 2 chứa tên 1 thì điểm = điểm ban đầu + 30
- Nếu tên 1 giống tên 2 thì điểm = điểm ban đầu + 5

✓ **Hướng dẫn**

- Xây dựng và gọi sử dụng phương thức tính chỉ số **loveLevelCompute(name1, name2)** trong đó tính toán và trả về kết quả như quy tắc xử lý phía trên
- Sử dụng phương thức **name1.length()** để tính chiều dài của một chuỗi
- Sử dụng phương thức **name1.contains(name2)** để kiểm tra chuỗi có chứa trong chuỗi hay không
- Sử dụng phương thức **name1.equals(name2)** để so sánh xem nội dung 2 chuỗi có giống nhau hay không

## BÀI 3: Giao diện




*Mục tiêu chính:*

*Sử dụng các điều khiển đơn giản để thiết kế các màn hình.*

*Hiển thị thông tin*

### 3.1. Thông tin Liên hệ

- ✓ **Yêu cầu:** Thiết kế màn hình hiển thị thông tin Liên hệ có các thông tin sau:
  - Tên
  - ĐTDĐ
  - Hình ảnh


Thông tin liên hệ

Họ tên

Hoa Hường Dương

ĐTĐĐ

0989753951

Hình ảnh

src/images/contact/aodai1.jpg

Hiển thị

---

Họ tên

Hoa Hường Dương

ĐTĐĐ

0989753951

Hình ảnh



✓ **Hướng dẫn sử dụng:**

- Nhập vào các ô nhập liệu các thông tin: tên, điện thoại di động, hình ảnh. Nhấn nút “Hiển thị” => Các thông tin vừa nhập sẽ được hiển thị.



✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

– ThManHinhHienThi: JFrame

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (Nhập liệu)
- txtDtdd: JTextField (Nhập liệu)
- txtHinhAnh: JTextField (Nhập liệu)
- lblHienThiTen: JLabel (Read Only)
- lblHienThiDtdd: JLabel (Read Only)
- lblHienThiHinhAnh: JLabel (Read Only) (phải hiển thị hình ảnh)
- btnHienThi: JButton

▪ **Nhập:**

- Tên
- Điện thoại di động
- Hình ảnh

▪ **Xuất:**

- Tên
- Điện thoại di động
- Hình ảnh

▪ **Qui tắc xử lý:**

- Không có

✓ **Thuật giải**

- Khai báo biến ten nhận giá trị của txtTen
- Khai báo biến dtdd nhận giá trị của txtDtdd

- Khai báo biến hình nhận giá trị của txtHinhAnh
- Kiểm tra giá trị nhập hợp lệ
- Xử lý button hiển thị. Hiển thị hình ảnh dạng icon của JLabel
- Kết xuất dữ liệu

✓ **Hướng dẫn**

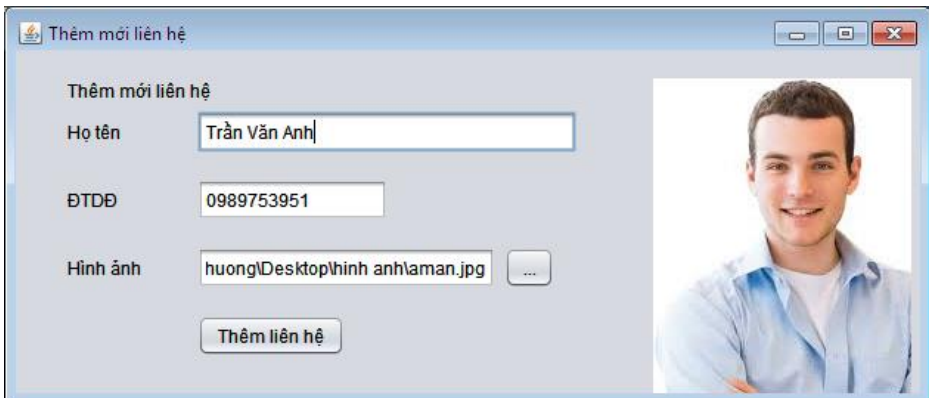
- Load hình ảnh:

```
ImageIcon icon = new ImageIcon("đường dẫn đến image");
lblHinhAnh.setIcon(icon);
```

### 3.2. Thêm mới Liên hệ

- ✓ **Yêu cầu: Thiết kế màn hình Thêm mới Liên hệ có các thông tin sau:**

- Tên
- ĐTDĐ
- Hình ảnh



- ✓ **Hướng dẫn sử dụng:**
- ✓ **Tóm tắt yêu cầu**
- **Thiết kế giao diện người dùng:**



– frmManHinhThem: FrmThemLienHe (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

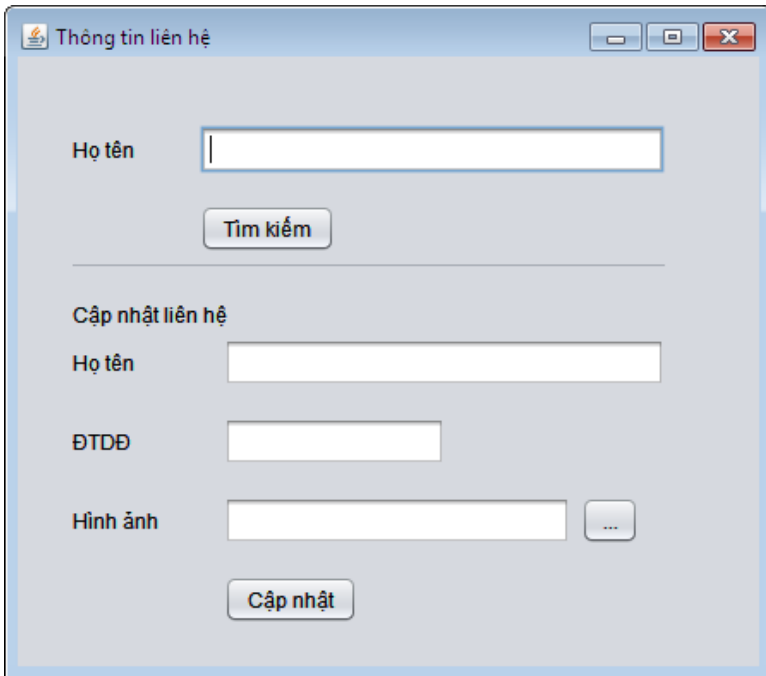
- txtTen: JTextField (nhập liệu)
- txtDtdd: JTextField (nhập liệu)
- fchHinhAnh: JFileChooser (chọn) (hoặc dùng TextField và button mở FileChooser )
- btnThem: JButton

### 3.3. Cập nhật liên hệ

✓ **Yêu cầu: Thiết kế màn hình Cập nhật Liên hệ có các thông tin sau:**

- Tên
- ĐTDĐ
- Hình ảnh





✓ **Hướng dẫn sử dụng:**

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmManHinhCapNhatLienHe: FrmCapNhatLienHe (extends từ JFrame)

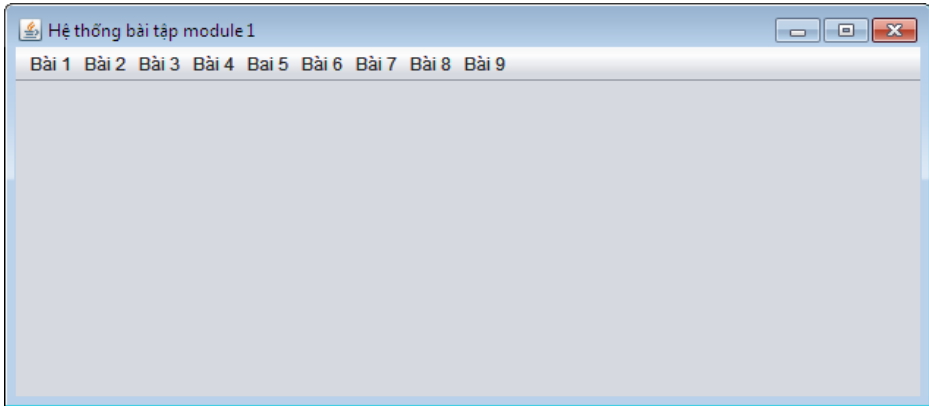
\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTenTim: JTextField (nhập liệu)
- btnTim: JButton (Xử lý tìm)
- txtTen: JTextField (cập nhật)
- txtDtdd: JTextField (cập nhật)
- lblHinhAnh: JLabel (read only)
- fchHinhAnh: JFileChooser (chọn)

- btnCapNhat: JButton (Xử lý cập nhật)

### 3.4. Hệ thống bài tập - MenuBar

- ✓ **Yêu cầu: Xây dựng ứng dụng JFrame có MenuBar**



*Hệ thống bài tập môn Java*

- ✓ **Hướng dẫn sử dụng:**
    - Click chọn các mục trên menu
  - ✓ **Tóm tắt yêu cầu**
    - **Thiết kế giao diện người dùng:**
      - frmHeThongBaiTap: JFrame
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - mnbMain: MenuBar
- Trong MenuBar có các menu item con sau:
- mniBai1: jMenuItem
  - mniBai1: jMenuItem
  - mniBai1: jMenuItem
  - mniBai1: jMenuItem
  - mniBai1: jMenuItem

- mniBai1: jMenu
- mniBai1: jMenu
- mniBai1: jMenu
- mniBai1: jMenu

### 3.5. Hiển thị hình ảnh được chọn

- ✓ **Yêu cầu: Thiết kế form hiển thị hình ảnh được chọn**



- ✓ **Hướng dẫn sử dụng:**
  - Click các button hình ảnh => Hiển thị hình ảnh to tương ứng với button được chọn
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**



- FrmImageView: form
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - lblImage: Label (Hiển thị ảnh)
      - btn1: Button
      - btn2: Button
      - btn3: Button
  - **Nhập:**
    - Đường dẫn hình ảnh
  - **Xuất:**
    - Hình ảnh lên button và lable
  - **Qui tắc xử lý :**
    - Mỗi lần btn1 hoặc btn2 hoặc btn3 được click thì sẽ đặt lại thuộc tính icon cho lblImage bằng hình ảnh tương ứng
- ✓ **Hướng dẫn**
- Sử dụng ImageIcon myImageIcon= **new ImageIcon("./src/img/Doraemon.jpg");** để thiết lập image icon

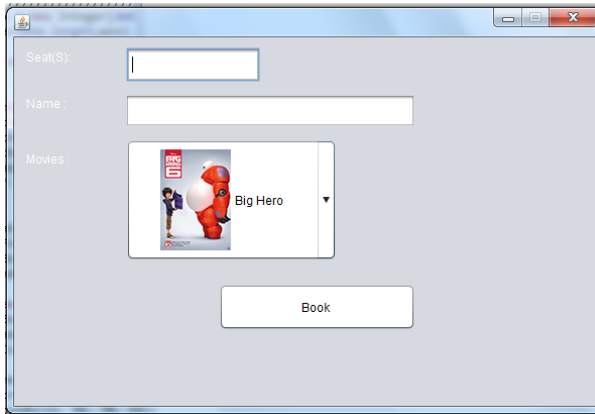
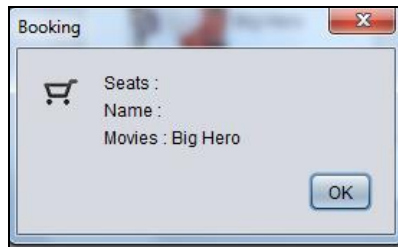
```

JButton btnDoremon = new JButton("");
btnDoremon.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        ImageIcon myImageIcon= new
        ImageIcon("./src/img/Doraemon.jpg");

        lblShowImage.setIcon(myImageIcon);
    }
});
btnDoremon.setIcon(new ImageIcon("./src/img/icon-
Doraemon.jpg"));
btnDoremon.setBounds(10, 10, 70, 70);
frmImageView.getContentPane().add(btnDoremon);
    
```

### 3.6. Đặt vé xem phim

- ✓ **Yêu cầu: Thiết kế form đặt vé xem phim như sau:**

- ✓ **Hướng dẫn sử dụng:**
  - Nhập tên, số ghế, chọn phim từ combobox nhấn button “OK”  
=> hiển thị thông tin tên, số ghế, phim vừa điền và chọn
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmBookingMovieTicket: form
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - txtName: TextField (Nhập liệu)
      - txtSeat: TextField (Nhập liệu)



- cbbMovies: ComboBox (Load danh sách phim từ file txt)
- btnBook: Button (Xử lý lấy ra thông tin)
- **Nhập:**
  - Tên, số ghế
  - Chọn phim
- **Xuất:**
  - Thông tin: tên, số ghế, tên phim
- ✓ **Hướng dẫn**
  - cbbMovies load tự động danh sách phim chứa trong file txt. Và cbbMovies được custom lại qua thuộc tính setRenderer và truyền vào đó đối tượng thuộc lớp ImageRender
  - btnSave xử lý lấy thông tin và xuất ra messagebox và messagebox có thêm thuộc tính icon
  - Cung cấp sẵn lớp ReadOrWriteFile và phương thức **String[] getFilmList()** trả về danh sách phim như sau:

```
private String[] getFilmList(){

    File myFile = new File("./src/FilmList.txt");
    String s;
    try {
        s = ReadOrWriteFile.readTXT(myFile);
        String temps[] = s.split("\n");

        return temps;
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```
  - Lớp ImageRender (được cung cấp)

```

1 import java.awt.Component;
2
3 @SuppressWarnings({ "serial", "rawtypes" })
4 public class ImageRender extends JLabel implements ListCellRenderer
5 {
6     String daftar[];
7     public ImageRender(String gmb[])
8     {
9         daftar = gmb;
10    }
11    @Override
12    public Component getListCellRendererComponent(JList list, Object value,
13        int index, boolean isSelected, boolean cellHasFocus)
14    {
15        // TODO Auto-generated method stub
16        int idx = ((Integer)value).intValue();
17        if(isSelected)
18        {
19            setBackground(list.getSelectionBackground());
20            setForeground(list.getSelectionForeground());
21        }
22        else
23        {
24            setBackground(list.getBackground());
25            setForeground(list.getForeground());
26        }
27        ImageIcon img = new ImageIcon("./src/img/" + daftar[idx].toLowerCase() + ".jpg" );
28        String var = daftar[idx];
29        setIcon(img);
30        setText(var);
31        setHorizontalAlignment(CENTER);
32        return this;
33    }
34 }

```

#### – Xử lý cbbMovies

```

59 Integer[] idx = new Integer[data.length];
60 for(int a=0;a<data.length;a++)
61 {
62     idx[a] = new Integer(a);
63 }
64
65 cbbMovies = new JComboBox(idx);
66 cbbMovies.setRenderer(new ImageRender(data));
67 cbbMovies.setBackground(new Color(255, 255, 255));
68 cbbMovies.setBounds(111, 101, 209, 120);
69 contentPane.add(cbbMovies);

```

### 3.7. Thay đổi màu sắc cánh hoa (bài làm thêm)

- ✓ Yêu cầu: Viết chương trình thay đổi màu sắc cánh hoa



✓ **Hướng dẫn sử dụng:**

- Bấm nút "Choose color" chọn màu để thay đổi màu cho cánh hoa

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- FrmColorChooser: form

\* (Các thể hiện phía dưới đều nằm trong form)

LayeredPane \*(thể hiện bên dưới nằm trong LayeredPane)

- pnlColor: Panel (hiển thị màu)
- lblImage: Label (hiển thị hình ảnh)
- btnColorChooser: Button (xử lý chọn và đổi màu cánh hoa)

▪ **Nhập:**





- Không có
- **Xuất:**
  - Cánh hoa đổi màu
- ✓ **Thuật giải**
  - btnColorChooser (Layer 2) khi click bật lên JColorChooser chọn màu, gán vào biến
  - pnlColor (Layer 0) để thuộc tính màu nền bằng với biến màu chọn được
  - lblImage (Layer 1) đặt icon là hình được cung cấp (hình hoa và có màu trong suốt ở vùng cánh hoa, file .png)
- ✓ **Hướng dẫn**

```
final JButton btnColorChooser = new JButton("Choose color");
btnColorChooser.setIcon(new ImageIcon("./src/img/color-chooser-
icon.png"));
layeredPane.setLayer(btnColorChooser, 2);
btnColorChooser.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        Color c = new Color(0); //default color gray
        c = JColorChooser.showDialog(null, "Pick a color",
c);
        btnColorChooser.setForeground(c);
        pnlBGColor.setBackground(c);
        layeredPane.add(pnlBGColor, 0);
    }
});
```

## BÀI 4: Truy xuất tập tin

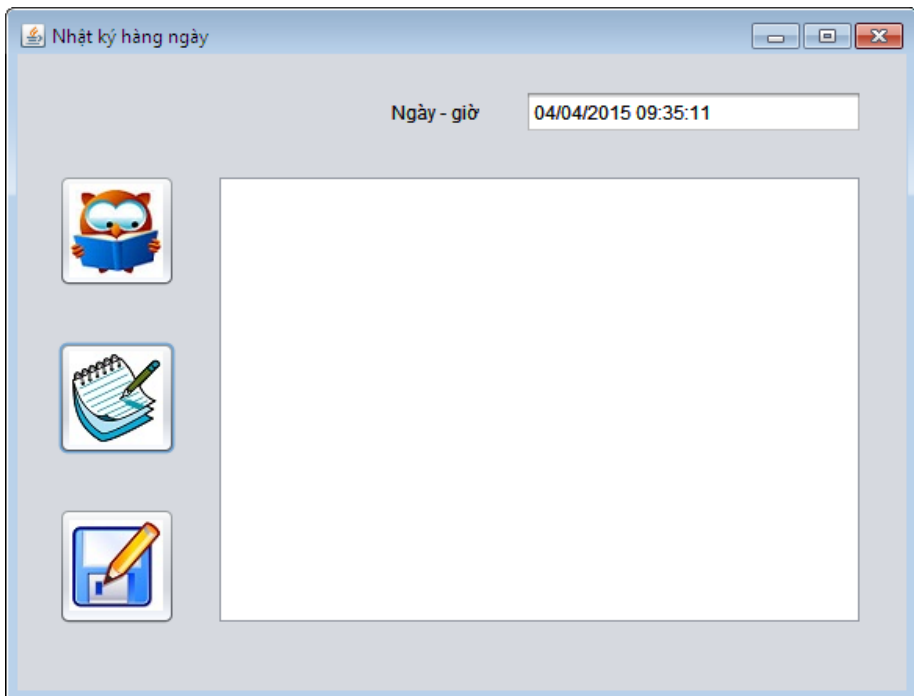


*Mục tiêu chính:*

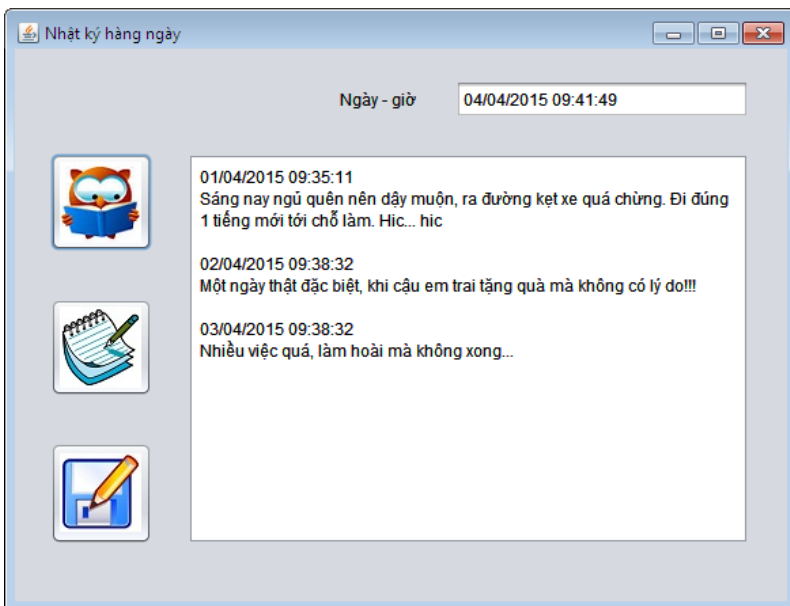
*Thao tác trên tập tin để đọc và xử lý nội dung tập tin*

**ký**

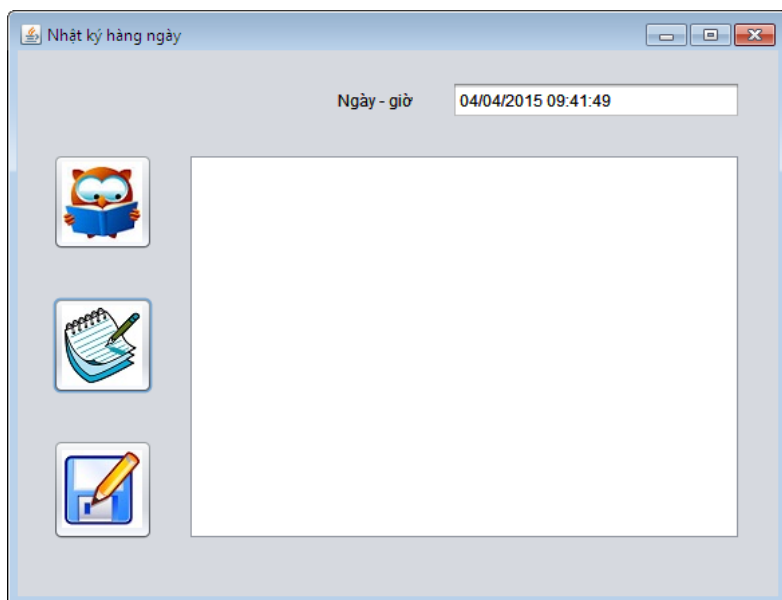
- ✓ **Yêu cầu:** Xây dựng ứng dụng Nhật ký hàng ngày như sau:



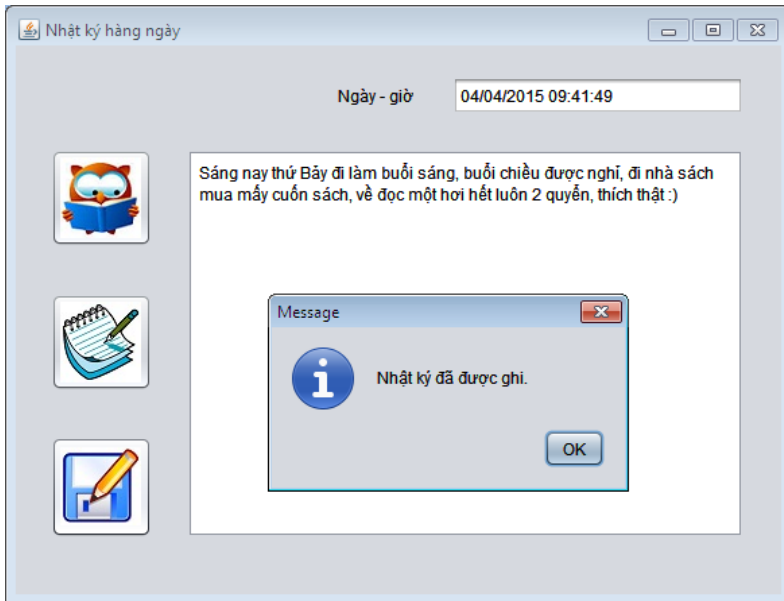
*Khi thực thi ứng dụng*



*Khi nhấn nút có icon đọc*



*Khi nhấn nút có icon viết*



*Khi nhấn nút có icon ghi*

✓ **Hướng dẫn sử dụng:**

- Nhấn nút có icon đọc => Hiển thị toàn bộ các nhật ký đã viết
- Nhấn nút có icon viết => Xóa hết các nội dung hiện có ở khung nhập liệu
- Nhập nội dung, nhấn nút có icon ghi => Hiển thị thông báo ghi thành công

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmNhatKy: JFrame (chỉnh thuộc tính defaultCloseOperation của JFrame là DISPOSE)

\* (Các thể hiện phía dưới đều nằm trong form)



- txtNgay: Text Field (kết xuất)
- txtaNoiDung: Text Area (nhập liệu, kết xuất)
- btnOpen: Button (xử lý mở tập tin, đọc và hiển thị nội dung tập tin)
- btnNew: Button (xóa nội dung trên Text Area)
- btnSave: Button (lưu nội dung trên Text Area vào tập tin)
- **Nhập:**
  - Nội dung
- **Xuất:**
  - Toàn bộ nhật ký
- ✓ **Hướng dẫn**
  - Tạo tập tin daily.txt ở trong thư mục của ứng dụng để lưu nhật ký.
  - Khi khởi động frmNhatKy: lấy ngày/ giờ hiện hành của hệ thống, định dạng theo "dd/MM/yyyy HH:mm:ss", xóa nội dung và thiết lập chế độ editable cho txtaNoiDung.

```
public frmNhatKy() {
    initComponents();
    DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss");
    Date date = new Date();
    txtNgay.setText(dateFormat.format(date));
    txtaNoiDung.setEditable(true);
    txtaNoiDung.setText("");
}
```



- Xử lý sự kiện cho nút có icon đọc: Sử dụng lớp `DataInputStream`, `BufferedInputStream`, `FileInputStream` để mở và đọc nội dung tập tin > hiển thị nội dung tập tin

```
private void btnOpenActionPerformed(java.awt.event.ActionEvent evt) {
    String nhatKy = "";
    String ngay;
    String noiDung;

    try (DataInputStream in = new DataInputStream(new
BufferedInputStream(new
FileInputStream("src/Bai6/daily.txt")))) {
        while (true) {
            ngay = in.readUTF();
            noiDung = in.readUTF();
            nhatKy += ngay + "\n" + noiDung + "\n\n";
        }
    } catch (EOFException e) {
    } catch (IOException ex) {
        Logger.getLogger(frmNhatKy.class.getName()).log(Level.SEVERE, null,
        ex);
    }

    txtaNoiDung.setText(nhatKy);
    txtaNoiDung.setEditable(false);
}
```



- Xử lý sự kiện cho nút có icon viết: lấy lại thời gian trên hệ thống > xóa nội dung và thiết lập chế độ editable cho txtaNoiDung

```
private void btnNewActionPerformed(java.awt.event.ActionEvent evt) {
    DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss");

    Date date = new Date();
    txtNgay.setText(dateFormat.format(date));

    txtaNoiDung.setEditable(true);
    txtaNoiDung.setText("");
}
```

- Xử lý sự kiện cho nút có icon ghi: Sử dụng lớp DataOutputStream, BufferedOutputStream, FileOutputStream để ghi nội dung tập tin: mở tập tin > lấy thông tin người dùng nhập ghi vào tập tin > xuất thông báo sau khi ghi xong

```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    String ngay = txtNgay.getText();
    String noiDung = txtaNoiDung.getText();

    try (DataOutputStream out = new DataOutputStream(
        new BufferedOutputStream (new
        FileOutputStream("src/Bai6/daily.txt", true)))) {
        out.writeUTF(ngay);
        out.writeUTF(noiDung);
        JOptionPane.showMessageDialog(rootPane, "Nhật ký đã
```

được ghi.");

**out.close();**

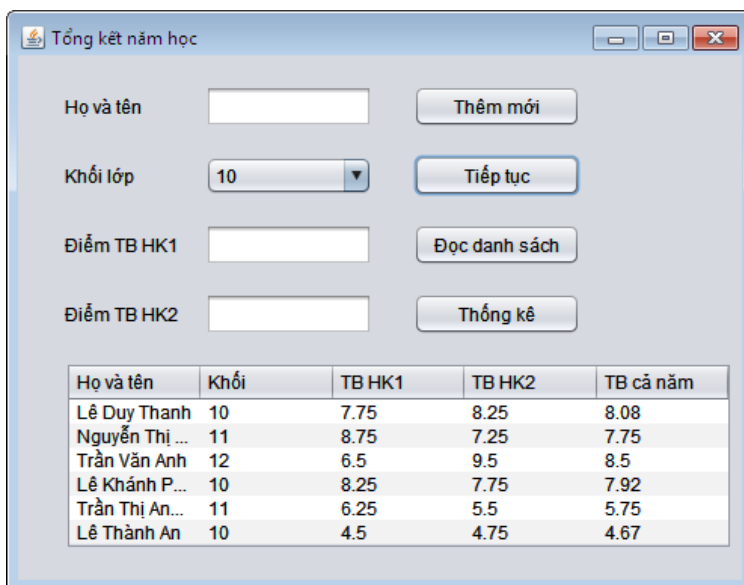
```

    } catch (FileNotFoundException ex) {
        Logger.getLogger(frmNhatKy.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(frmNhatKy.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

## 4.2. Tổng kết năm học

- ✓ **Yêu cầu:** Xây dựng ứng dụng tổng kết năm học như sau:



Họ và tên	Khối	TB HK1	TB HK2	TB cả năm
Lê Duy Thanh	10	7.75	8.25	8.08
Nguyễn Thị ...	11	8.75	7.25	7.75
Trần Văn Anh	12	6.5	9.5	8.5
Lê Khánh P...	10	8.25	7.75	7.92
Trần Thị An...	11	6.25	5.5	5.75
Lê Thành An	10	4.5	4.75	4.67

*Khi nhấn nút Đọc danh sách*



Tổng kết năm học

Họ và tên:

Khối lớp:

Điểm TB HK1:

Điểm TB HK2:

Họ và tên: Lê Duy Thanh, Nguyễn Thị ..., Trần Văn Anh, Lê Khánh P..., Trần Thị An..., Lê Thành An

Điểm TB HK1: 8.08, 7.5, 6.5, 8.25, 5.75, 4.67

Điểm TB HK2: 7.5, 7.5, 8.5, 7.92, 5.75, 4.67

Điểm TB cả năm: 7.5, 7.5, 8.5, 7.92, 5.75, 4.67

Message: Kết quả của học sinh đã được ghi.

OK

*Khi nhấn nút Thêm mới*

Tổng kết năm học

Họ và tên:

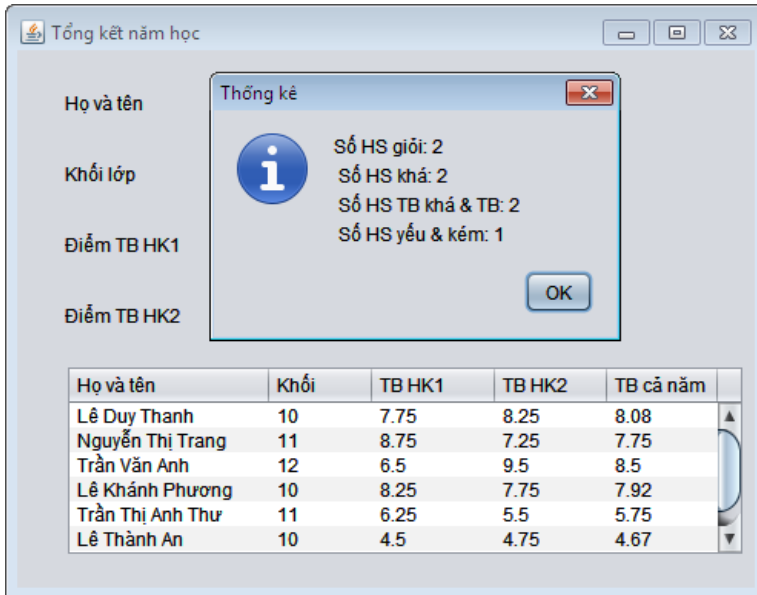
Khối lớp:

Điểm TB HK1:

Điểm TB HK2:

Họ và tên	Khối	TB HK1	TB HK2	TB cả năm
Lê Duy Thanh	10	7.75	8.25	8.08
Nguyễn Thị Trang	11	8.75	7.25	7.75
Trần Văn Anh	12	6.5	9.5	8.5
Lê Khánh Phương	10	8.25	7.75	7.92
Trần Thị Anh Thư	11	6.25	5.5	5.75
Lê Thành An	10	4.5	4.75	4.67

*Khi nhấn nút Tiếp tục*



Họ và tên	Khối	TB HK1	TB HK2	TB cả năm
Lê Duy Thanh	10	7.75	8.25	8.08
Nguyễn Thị Trang	11	8.75	7.25	7.75
Trần Văn Anh	12	6.5	9.5	8.5
Lê Khánh Phương	10	8.25	7.75	7.92
Trần Thị Anh Thư	11	6.25	5.5	5.75
Lê Thành An	10	4.5	4.75	4.67

*Khi nhấn nút Thống kê*

✓ **Hướng dẫn sử dụng:**

- Nhấn nút “Đọc danh sách” => Hiển thị toàn bộ học sinh và kết quả đang có
- Nhập họ và tên, chọn khối lớp, nhập điểm TB HK1, điểm TB HK2, nhấn nút “Thêm mới” => Hiển thị kết quả sau khi thêm.
- Nhấn nút “Tiếp tục” => Xóa hết các nội dung hiện có ở khung nhập liệu
- Nhấn nút “Thống kê” => Hiển thị thống kê số lượng học sinh giỏi, khá, trung bình khá & trung bình, yếu & kém

✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**



- frmTongKet: JFrame (chính thuộc tính defaultCloseOperation của JFrame là DISPOSE)

*\* (Các thể hiện phía dưới đều nằm trong form)*

- txtHoTen: Text Field (nhập liệu)
- cbbKhoi: Combo Box (chọn)
- txtTBHK1: Text Field (nhập liệu)
- txtTBHK2: Text Field (nhập liệu)
- tblHocSinh: Jtable (kết xuất, chỉ đọc)
- btnAdd: Button (xử lý thêm mới)
- btnContinue: Button (xử lý xóa nội dung)
- btnOpen: Button (xử lý đọc, tính toán và hiển thị danh sách học sinh)
- btnReport: Button (xử lý đọc, tính toán và hiển thị thống kê học sinh)
- **Nhập:**
  - Họ và tên
  - Khối
  - TB HK1
  - TB HK2
- **Xuất:**
  - Danh sách học sinh
- **Qui tắc xử lý :**
  - Trung bình cả năm =  $(TB\ HK1 + (TB\ HK2 * 2))/3$  làm tròn 2 số lẻ
  - Thống kê:
    - Nếu Trung bình cả năm  $\geq 8$ : giỏi



- Ngược lại, nếu  $7 \leq \text{Trung bình cả năm} < 8$ : khá
- Ngược lại, nếu  $5 \leq \text{Trung bình cả năm} < 7$ : trung bình khá & trung bình
- Ngược lại, nếu  $\text{Trung bình cả năm} < 5$ : yếu & kém

### ✓ Hướng dẫn

- Tạo tập tin tongketnamhoc.txt ở trong thư mục của ứng dụng để lưu thông tin tổng kết của học sinh
- Xử lý sự kiện cho nút “Thêm mới”: Sử dụng lớp DataOutputStream, BufferedOutputStream, FileOutputStream để ghi nội dung tập tin: mở tập tin > lấy thông tin người dùng nhập ghi vào tập tin > xuất thông báo sau khi ghi xong

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    String hoten = txtHoTen.getText();
    int khoi = Integer.parseInt(cbbKhoi.getSelectedItem().toString());
    double tbhk1 = Double.parseDouble(txtTBHK1.getText());
    double tbhk2 = Double.parseDouble(txtTBHK2.getText());
```

```
    try (DataOutputStream out = new DataOutputStream(
        new BufferedOutputStream(new
FileOutputStream("src/Bai4/tongketnamhoc.txt", true)))) {
        out.writeUTF(hoten);
        out.writeInt(khoi);
        out.writeDouble(tbhk1);
        out.writeDouble(tbhk2);
        JOptionPane.showMessageDialog(rootPane, "Kết quả của học
```



**sinh đã được ghi.");**

**out.close();**

```

    } catch (FileNotFoundException ex) {
        Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (IOException ex) {
        Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
}

```

- Xử lý sự kiện cho nút "Tiếp tục": xóa nội dung ở các ô nhập liệu
- Xử lý sự kiện cho nút "Đọc danh sách": Sử dụng lớp `DataInputStream`, `BufferedInputStream`, `FileInputStream` để mở và đọc nội dung tập tin > Xử lý tính toán và hiển thị kết quả lên table

```

private void btnOpenActionPerformed(java.awt.event.ActionEvent evt) {
    String hoten;
    double tbhk1;
    double tbhk2;
    double tbcn;
    int khai;

    DefaultTableModel model = new DefaultTableModel();
}

```



```
model.addColumn("Họ và tên");
model.addColumn("Khối");
model.addColumn("TB HK1");
model.addColumn("TB HK2");
model.addColumn("TB cả năm");
```

```
try (DataInputStream in = new DataInputStream(new
BufferedInputStream(new
FileInputStream("src/Bai4/tongketnamhoc.txt")))) {
    while (true) {
        hoten = in.readUTF();
        khoi = in.readInt();
        tbhk1 = in.readDouble();
        tbhk2 = in.readDouble();
        tbcn = (tbhk1 + tbhk2 * 2) / 3;
        DecimalFormat df = new DecimalFormat("###.##");
        tbcn = Double.parseDouble(df.format(tbcn));
        model.addRow(new Object[]{hoten, khoi, tbhk1, tbhk2,
tbcn});
    }
    } catch (EOFException e) {
        } catch (FileNotFoundException ex) {
        Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (IOException ex) {
```



```

    Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE, null,
    ex);

    }

    tblKetQua.setModel(model);

}

```

- Xử lý sự kiện cho nút “Tổng kê”: thực hiện việc đọc danh sách, tính toán và xuất kết quả ra message box.

```

private void btnReportActionPerformed(java.awt.event.ActionEvent evt) {
    String hoten;
    double tbhk1;
    double tbhk2;
    double tbcn;
    int khai;
    // khởi tạo các biến phục vụ cho việc thống kê
    int soHSGioi = 0;
    int soHSKha = 0;
    int soHSTBKhaTB = 0;
    int soHSYeuKem = 0;

    try (DataInputStream in = new DataInputStream(new
    BufferedInputStream(new
    FileInputStream("src/Bai4/tongketnamhoc.txt")))) {
        while (true) {

```



```

        hoten = in.readUTF();
        khoi = in.readInt();
        tbhk1 = in.readDouble();
        tbhk2 = in.readDouble();
        tbcn = (tbhk1 + tbhk2 * 2) / 3;
        if (tbcn >= 8) {
            soHSGioi++;
        } else if (tbcn >= 7 && tbcn < 8) {
            soHSKha++;
        } else if (tbcn >= 5 && tbcn < 7) {
            soHSTBKhaTB++;
        } else {
            soHSYeuKem++;
        }
    }
} catch (EOFException e) {
    } catch (FileNotFoundException ex) {
        Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (IOException ex) {
        Logger.getLogger(frmTongKet.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
// khai báo chuỗi kết quả và gán giá trị

```





```
String ketQua = "Số HS giỏi: " + soHSGioi + "\n Số HS khá: " +
soHSKha + "\n Số HS TB khá & TB: " + soHSTBKhaTB + "\n Số HS yếu &
kém: " + soHSYeuKem;

// xuất kết quả ra hộp thoại thông báo

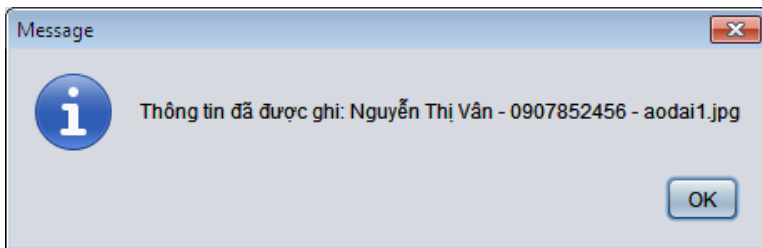
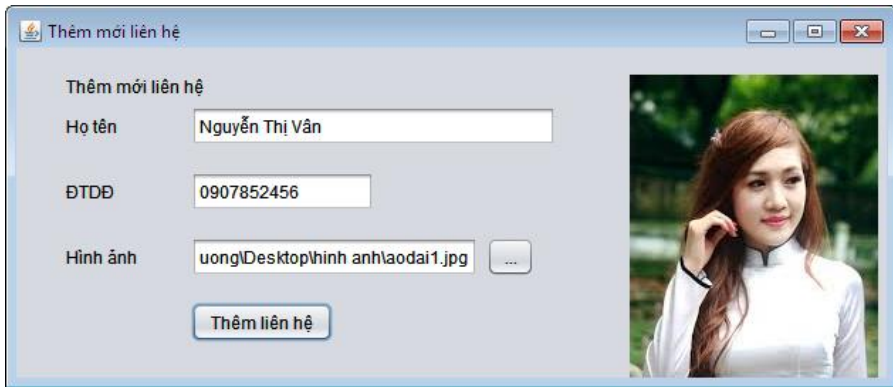
JOptionPane.showMessageDialog(rootPane, ketQua, "Thống kê", 1);
}
```

### 4.3. Thêm Liên hệ vào tập tin

- ✓ **Yêu cầu:** Xây dựng chương trình Thêm mới Liên hệ có các thông tin sau:

- Tên
- ĐTDĐ
- Hình ảnh

*Form Thêm liên hệ mới khi chạy*



*Sau khi thực hiện thao tác "Thêm liên hệ"*

✓ **Hướng dẫn sử dụng:**

- Nhập vào các ô nhập liệu các thông tin: tên, điện thoại di động, hình ảnh. Nhấn nút "Thêm" => Các thông tin vừa nhập sẽ được lưu vào tập tin LienHe.txt.

✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng (sử dụng giao diện đã thiết kế ở bài Giao diện):**

- frmManHinhThemLienHe: FrmThemLienHe (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (nhập liệu)
- txtDtdd: JTextField (nhập liệu)



- fchHinhAnh: JFileChooser (chọn)
- btnThem: JButton (Xử lý thêm)
- **Nhập:**
  - Tên
  - ĐTDĐ
  - Hình ảnh...
- **Xuất:**
  - “Thông tin Liên hệ đã được thêm vào tập tin” nếu thêm được hoặc “Không thể thêm Liên hệ”
- **Qui tắc xử lý:**
  - Tạo tập tin LienHe
  - Lấy thông tin được nhập và ghi vào tập tin LienHe.txt
- ✓ **Thuật giải**
  - Hiển thị đường dẫn file được chọn trong txtHinhAnh
  - Khai báo biến hoTen nhận giá trị từ txtHoTen
  - Khai báo biến dtdd nhận giá trị từ txtDtdd
  - Khai báo biến hinhAnh nhận giá trị từ txtHinhAnh
  - Xử lý đưa hình ảnh vào thư mục hình ảnh
  - Xử lý ghi vào file các thông tin hoTen, dtdd, hinhAnh
  - Xuất kết quả dưới dạng dialog thông báo
- ✓ **Hướng dẫn**
  - Copy lại form Thêm mới liên hệ đã làm ở bài 3 và đưa vào bài này để sử dụng lại
  - Tạo tập tin lienhe.txt để lưu trữ thông tin các liên hệ
  - Tạo thư mục images/contact để lưu trữ hình ảnh của liên hệ
  - Xử lý sự kiện cho nút “Thêm liên hệ”:



- Đưa hình ảnh vào thư mục contact trong thư mục images của ứng dụng
- Sử dụng lớp `DataOutputStream`, `BufferedOutputStream`, `FileOutputStream` để ghi nội dung tập tin: mở tập tin > lấy thông tin người dùng nhập ghi vào tập tin > xuất thông báo sau khi ghi xong.

```
private void btnHienThiActionPerformed(java.awt.event.ActionEvent evt) {
    // tạo một biến f kiểu File để lưu hình ảnh được chọn
    File f = new File(txtHinhAnh.getText());
    // chuẩn bị đường dẫn để đưa hình vào
    String path = "src/images/contact/";
    String duong_dan = path + f.getName();
    // đưa hình vào đường dẫn đã chuẩn bị và xuất thông báo
    try {
        Files.copy(f.toPath(), (new File(path +
f.getName())).toPath(), StandardCopyOption.REPLACE_EXISTING);
        JOptionPane.showMessageDialog(rootPane, " Đã đưa hình
liên hệ thư mục contact");
    } catch (IOException ex) {
        Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
        null, ex);
    }
    // lấy thông tin người dùng nhập
    String hoTen = txtHoTen.getText();
    String dtdd = txtDTDD.getText();
    String hinhAnh = f.getName();
    // đưa thông tin này vào tập tin lienhe.txt và thông báo
    try (DataOutputStream out = new DataOutputStream(
```



```

        new BufferedOutputStream(new
FileOutputStream("src/Bai4/lienhe.txt", true)))) {
            out.writeUTF(hoTen);
            out.writeUTF(dtdd);
            out.writeUTF(hinhAnh);
            String kq = hoTen + " - " + dtdd + " - " + hinhAnh;
            out.close();
            JOptionPane.showMessageDialog(rootPane, "Thông tin đã
được ghi: " + kq);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
            null, ex);
        } catch (IOException ex) {
            Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
            null, ex);
        }
    }
}

```

# BÀI 5: Mảng cơ sở

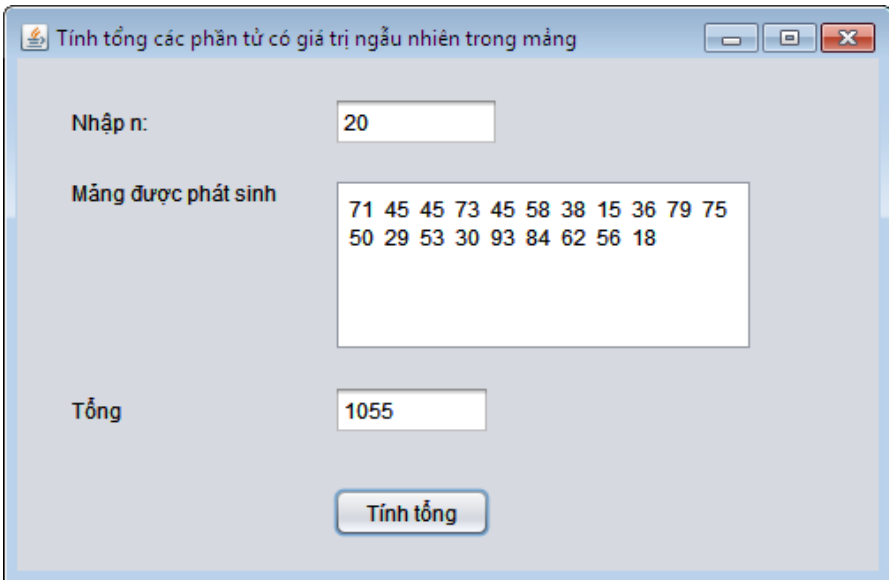


*Mục tiêu chính:*

*Xử lý mảng một chiều*

**Tính tổng các phần tử trong mảng**

- ✓ **Yêu cầu:** Viết chương trình Tính tổng các phần tử trong mảng có  $n$  phần tử, mỗi phần tử có giá trị ngẫu nhiên  $\leq 100$



Tính tổng các phần tử có giá trị ngẫu nhiên trong mảng

Nhập n: 20

Mảng được phát sinh

71	45	45	73	45	58	38	15	36	79	75
50	29	53	30	93	84	62	56	18		

Tổng: 1055

Tính tổng

- ✓ **Hướng dẫn sử dụng:**
  - Nhập vào số phần tử trong mảng  $n$ . Nhấn nút “Tính tổng”  
=> Hiển thị mảng và tổng các phần tử trong mảng
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - frmManHinhTinhTong: FrmTinhTong (extends từ JFrame)



\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtN: JTextField (nhập liệu)
- txtMang: JTextArea (kết xuất, chỉ đọc)
- txtTong: JTextField (kết xuất, chỉ đọc)
- btnTinhTong: JButton (Xử lý tính tổng và hiển thị)
- **Nhập:**
  - Nhập giá trị n
- **Xuất:**
  - Mảng số nguyên ngẫu nhiên
  - Xuất tổng
- **Qui tắc xử lý :**
  - Không có
- ✓ **Thuật giải**
  - Khai báo biến nhận giá trị nhập từ field txtN
  - Chuyển giá trị nhập từ chuỗi sang số
  - Phát sinh mảng ngẫu nhiên int[]
  - Xuất giá trị chuỗi của mảng vào txtMang
  - Tính tổng các phần tử trong mảng
  - Hiển thị kết quả tính lên txtTong
- ✓ **Hướng dẫn**
  - Xây dựng phương thức tạo mảng các số ngẫu nhiên  $\leq 100$  bằng cách sử dụng Random

```
private int[] mang(int n){
    int[] mang = new int[n];
    Random random = new Random();
```



```
for(int i = 0; i < n; i++){
    mang[i] = random.nextInt(100);
}
return mang;
}
```

- Xây dựng phương thức xuất mảng

```
private String mangChuoi(int[] mang){
    String chuoi = "";
    for(int i : mang ){
        chuoi += i + " ";
    }
    return chuoi;
}
```

- Xây dựng phương thức tính tổng các phần tử trong mảng

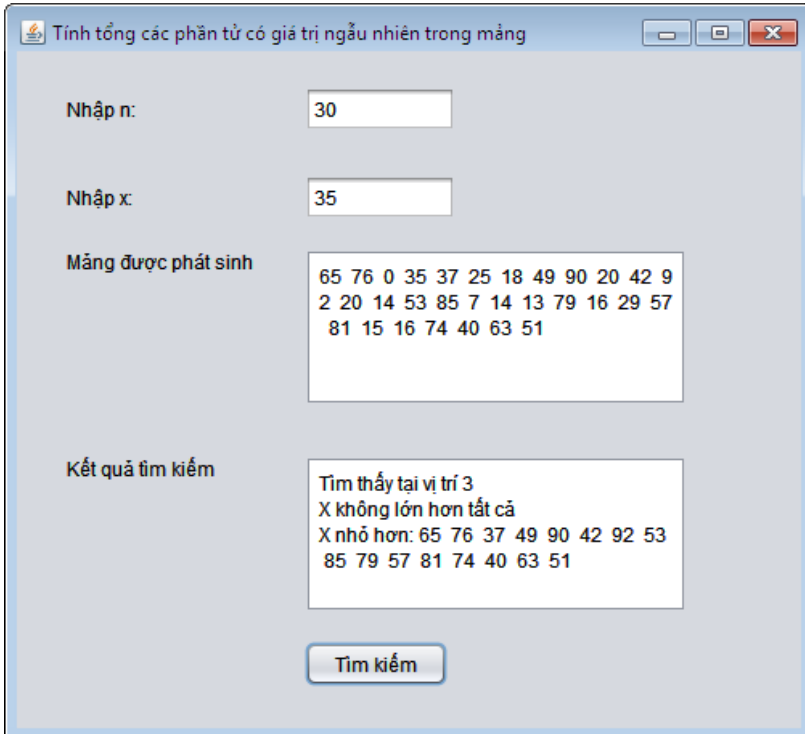
```
private int tong(int[] mang){
    int tong = 0;
    for(int i : mang){
        tong += i;
    }
    return tong;
}
```

- Gọi các phương thức đã xây dựng vào xử lý của nút “Tính tổng” > xuất kết quả

## 5.2. Tìm kiếm trong mảng (làm tiếp từ bài tính tổng)



- ✓ **Yêu cầu: Viết chương trình thực hiện việc xem xét giá trị của x so với các giá trị trong mảng**



Tính tổng các phần tử có giá trị ngẫu nhiên trong mảng

Nhập n: 30

Nhập x: 35

Mảng được phát sinh

65	76	0	35	37	25	18	49	90	20	42	9
2	20	14	53	85	7	14	13	79	16	29	57
81	15	16	74	40	63	51					

Kết quả tìm kiếm

Tìm thấy tại vị trí 3  
 X không lớn hơn tất cả  
 X nhỏ hơn: 65 76 37 49 90 42 92 53  
 85 79 57 81 74 40 63 51

Tìm kiếm

- Nhập vào một số x  
 => Chương trình sẽ trả lời những câu hỏi sau:
  - x có xuất hiện trong mảng không? Nếu có thì cho biết x xuất hiện lần đầu tiên ở vị trí nào?
  - x có lớn hơn tất cả các số trong mảng không?
  - x nhỏ hơn những số nào trong mảng? (In ra tất cả các số lớn hơn x)
- ✓ **Hướng dẫn sử dụng:**



- Nhập vào số phần tử trong mảng  $n > 0$ . Nhấn nút “Tìm kiếm” > Hiển thị kết quả.

✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng (copy giao diện của bài 5.1 và bổ sung thêm các điều khiển cần thiết):**
- **Nhập:**
  - $n, x$
- **Xuất:**
  - Kết quả tìm kiếm

✓ **Hướng dẫn**

- Sử dụng lại phương thức phát sinh và xuất mảng của bài trên.
- Xây dựng phương thức tìm kiếm  $x$  trong mảng

```
private int timX(int x, int[] mang){
    int vitri = -1;
    for(int i = 0; i < mang.length; i++){
        if(x==mang[i]){
            vitri = i;
            break;
        }
    }
    return vitri;
}
```

- Xây dựng phương thức kiểm tra  $x$  là số lớn nhất

```
private boolean xLonHon(int x, int[] mang ){
    boolean flag = true;
    for(int i : mang){
```



```

        if (x <= i) {
            flag = false;
            break;
        }
    }
    return flag;
}

```

- Xây dựng phương thức tìm và in ra các giá trị nhỏ hơn x trong mảng

```

private String xNhoHon(int x, int[] mang){
    String kq = "";
    for(int i : mang){
        if(x < i ) kq += i + " ";
    }
    return kq;
}

```

- Gọi các phương thức đã xây dựng vào xử lý của nút "Tìm kiếm" > xuất kết quả

```

private void btnTimActionPerformed(java.awt.event.ActionEvent evt) {
    int n = Integer.parseInt(txtN.getText());
    int x = Integer.parseInt(txtX.getText());
    int[] mang = mang(n);
    String kq = "";
    txtaMang.setText(mangChuoai(mang));
    int tim = timX(x, mang);
    kq += (tim == -1)? "Không tìm thấy" : "Tìm thấy tại vị trí " + tim;
    kq += "\n";
    kq += (xLonHon(x, mang))? "X lớn hơn tất cả" : "X không lớn hơn tất

```

```

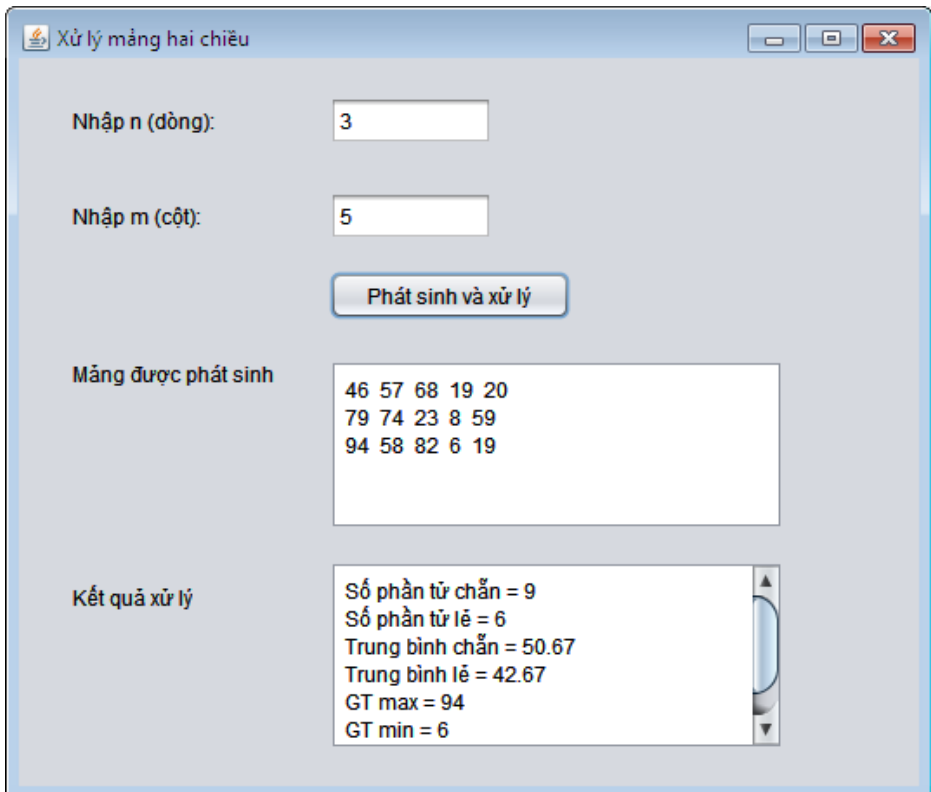
cả";

kq += "\n";

kq += "X nhỏ hơn: " + xNhoHon(x, mang);
txtaKetQua.setText(kq);
}
    
```

### 5.3. Mảng hai chiều

- ✓ **Yêu cầu:** Viết chương trình thực hiện việc xử lý tính toán trên mảng hai chiều



**Xử lý mảng hai chiều**

Nhập n (dòng):

Nhập m (cột):

**Phát sinh và xử lý**

Mảng được phát sinh

46	57	68	19	20
79	74	23	8	59
94	58	82	6	19

Kết quả xử lý

- Số phần tử chẵn = 9
- Số phần tử lẻ = 6
- Trung bình chẵn = 50.67
- Trung bình lẻ = 42.67
- GT max = 94
- GT min = 6



- Nhập  $n, m$  là số dòng và số cột trong mảng  $a$ .  
Chương trình sẽ thực hiện những công việc sau:
- Tạo mảng 2 chiều với giá trị ngẫu nhiên
- Xuất mảng hai chiều
- Đếm số phần tử chẵn/ phần tử lẻ trong mảng  $a$
- Tính giá trị trung bình các phần tử chẵn/ các phần tử lẻ trong mảng  $a$
- Tìm phần tử lớn nhất/ nhỏ nhất trong mảng  $a$

✓ **Hướng dẫn sử dụng:**

- Nhập số dòng, số cột > Nhấn “Phát sinh và xử lý”

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- Như hình trên

▪ **Nhập:**

- $n, m$

▪ **Xuất:**

- Số phần tử chẵn/ phần tử lẻ trong mảng  $a$
- Giá trị trung bình các phần tử chẵn/ các phần tử lẻ trong mảng  $a$
- Phần tử lớn nhất/ nhỏ nhất trong mảng  $a$

✓ **Hướng dẫn**

- Viết phương thức tạo mảng hai chiều với giá trị ngẫu nhiên

```
private int[][] mang(int n, int m) {
    int[][] mang = new int[n][m];
    Random random = new Random();
```



```

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            mang[i][j] = random.nextInt(100);
        }
    }
    return mang;
}

```

- Viết phương thức xuất mảng hai chiều

```

private String mangChuoi(int[][] mang) {
    String chuoi = "";
    for (int i = 0; i < mang.length; i++) {
        for (int j = 0; j < mang[i].length; j++) {
            chuoi += mang[i][j] + " ";
        }
        chuoi += "\n";
    }
    return chuoi;
}

```

- Viết phương thức xử lý mảng hai chiều với kết quả trả về là một mảng chứa các kết quả được yêu cầu

```

private int[] xulyMangHaiChieu(int[][] mang) {
    int[] mangKQ = new int[6];
    int soPTchan = 0;
    int soPTle = 0;
    int tongchan = 0;
}

```



```

int tongle = 0;
int max = mang[0][0];
int min = mang[0][0];

for (int i = 0; i < mang.length; i++) {
    for (int j = 0; j < mang[i].length; j++) {
        if (mang[i][j] % 2 == 0) {
            soPTchan++;
            tongchan += mang[i][j];
        }
        if (mang[i][j] % 2 != 0) {
            soPTle++;
            tongle += mang[i][j];
        }
        if (max < mang[i][j]) {
            max = mang[i][j];
        }
        if (min > mang[i][j]) {
            min = mang[i][j];
        }
    }
}

mangKQ[0] = soPTchan;
mangKQ[1] = soPTle;
mangKQ[2] = tongchan;
mangKQ[3] = tongle;
    
```



```
mangKQ[4] = max;
mangKQ[5] = min;
return mangKQ;
}
```

- Gọi các phương thức đã xây dựng vào xử lý của nút "Xử lý" > xuất kết quả

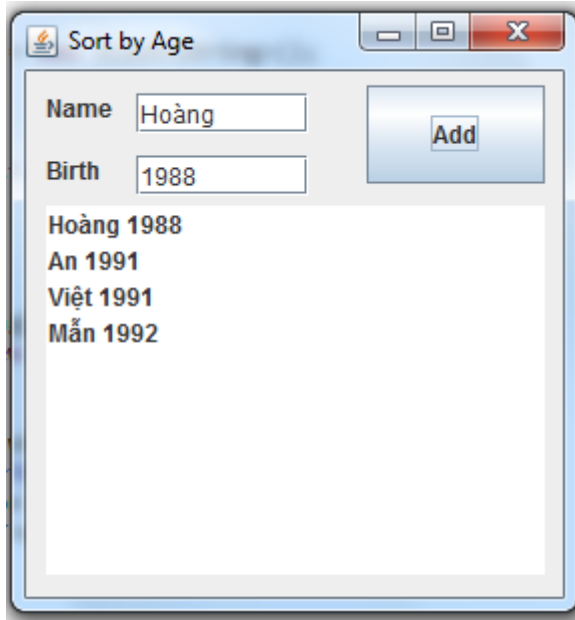
```
private void btnXuLyActionPerformed(java.awt.event.ActionEvent evt) {
    int n = Integer.parseInt(txtN.getText());
    int m = Integer.parseInt(txtM.getText());
    int[][] mang = mang(n, m);
    int[] mangKQ = xulyMangHaiChieu(mang);
    float tbchan = (float) mangKQ[2] / mangKQ[0];
    float tble = (float) mangKQ[3] / mangKQ[1];
    String kq = "";
    kq += "Số phần tử chẵn = " + mangKQ[0] + "\n";
    kq += "Số phần tử lẻ = " + mangKQ[1] + "\n";
    kq += "Trung bình chẵn = " + String.format("%.2f", tbchan) + "\n";
    kq += "Trung bình lẻ = " + String.format("%.2f", tble) + "\n";
    kq += "GT max = " + mangKQ[4] + "\n";
    kq += "GT min = " + mangKQ[5] + "\n";

    txtaMang.setText(mangChuoi(mang));
    txtaKetQua.setText(kq);
}
```

## 5.4. Sắp xếp tên và năm sinh



- ✓ **Yêu cầu: Viết chương trình để thêm tên và năm sinh của một người vào danh sách có sắp xếp theo thứ tự tuổi giảm dần**



- ✓ **Hướng dẫn sử dụng:**
  - Nhập tên và năm sinh của 1 người, bấm nút “Add” => tên và năm sinh sẽ đc thêm vào đúng vị trí đã sắp xếp như kết quả hình trên
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmSortByBirth: form
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - txtName: TextField (nhập liệu)
      - txtBirth: TextField (nhập liệu)
      - btnAdd: Button (xử lý thêm và sắp xếp)



- IstMain: List (hiển thị danh sách tên và năm sinh)
- **Nhập:**
  - Tên
  - Năm sinh
- **Xuất:**
  - Danh sách tên và năm sinh đã được sắp xếp

## 5.5. Bảng điểm

- ✓ **Yêu cầu: Viết chương trình tạo và sắp xếp và tìm kiếm tên và điểm trong bảng điểm**

- ✓ **Hướng dẫn sử dụng:**
  - Nhập tên và điểm, bấm nút "Add" để thêm, nút "Sort Name" hoặc "Sort Score" để sắp xếp bảng điểm theo tên hoặc theo điểm.



- Nhập vào điểm hoặc tên, bấm nút “Find” để tìm kiếm theo điểm hoặc theo tên.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- FrmScoreBoard: form

\* *(Các thể hiện phía dưới đều nằm trong form)*

- txtName: TextField (Nhập liệu)
- txtScore: TextField (Nhập liệu)
- txtKeyWord: TextField (Nhập liệu)
- btnAdd: Button (xử lý thêm)
- btnSortName: Button (xử lý sắp xếp theo tên)
- btnSortScore: Button (xử lý sắp xếp theo điểm)
- btnSearch: Button (xử lý tìm kiếm theo tên/ theo điểm)
- lstMain: List (hiển thị danh sách tên và điểm)

▪ **Nhập:**

- Tên
- Điểm

▪ **Xuất:**

- Danh sách tên và điểm đã được sắp xếp

## BÀI 6: Đối tượng



*Mục tiêu chính:*

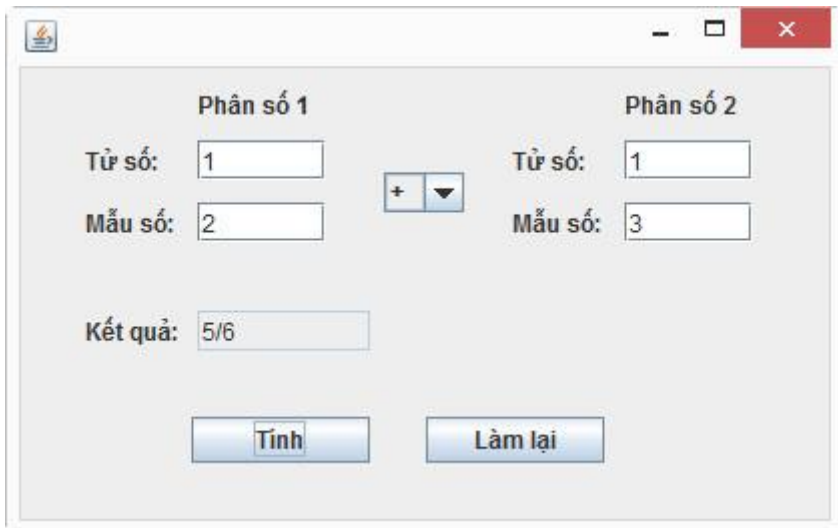
*Làm việc với đối tượng*

### hiện tính toán hai phân số

- ✓ **Yêu cầu: Xây dựng chương trình Tính toán hai phân số**

Gồm có:

- phân số thứ nhất (tử số, mẫu số) và phân số thứ hai (tử số, mẫu số)
- combobox 4 phép tính: +, -, \*, / cho người dùng chọn



- ✓ **Hướng dẫn sử dụng:**

- Nhập vào phân số thứ nhất (tử số, mẫu số) và phân số thứ 2 (tử số, mẫu số), chọn 1 trong 4 phép tính. Nhấn nút "Tính"  
=> Hiện thị kết quả lên phân số kết quả (tử số, mẫu số)

- ✓ **Tóm tắt yêu cầu**



- **Thiết kế giao diện người dùng:**

- frmManHinhTinhToanPhanSo: FrmTinhToanPhanSo (extend từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTuSo1: JTextField (nhập liệu)
- txtMauSo1: JTextField (nhập liệu)
- txtTuSo2: JTextField (nhập liệu)
- txtMauSo2: JTextField (nhập liệu)
- cmbPhepTinh: JComboBox (chọn)
- txtKetQua: JTextField (nhập liệu)
- btnTinh: JButton (Xử lý tính)
- btnLamLai: JButton (Xử lý làm lại)

- **Nhập:**

- Tử số phân số 1
- Mẫu số phân số 1
- Tử số phân số 2
- Mẫu số phân số 2
- Chọn phép tính

- **Xuất:**

- Kết quả tính toán 2 phân số

- **Qui tắc xử lý :**

- Xây dựng class PhanSo
- Xử lý tính toán trên hai đối tượng có kiểu là PhanSo

✓ **Thuật giải**

- Khai báo biến tuSo1 nhận kết quả từ txtTuSo1



- Khai báo biến mauSo1 nhận kết quả từ txtMauSo1
- Khởi tạo phanSo1 (kiểu PhanSo) từ tuSo1, mauSo1
- Khai báo biến tuSo2 nhận kết quả từ txtTuSo2
- Khai báo biến mauSo2 nhận kết quả từ txtMauSo2
- Khởi tạo phanSo2 (kiểu PhanSo) từ tuSo2, mauSo2
- Khai báo biến phepTinh nhận giá trị từ cmbPhepTinh
- Tính toán 2 phân số theo phép tính

✓ **Hướng dẫn**

- Xây dựng lớp PhanSo

```
public class PhanSo {
    int tuSo;
    int mauSo;
    public PhanSo(int tuSo, int mauSo) {
        this.tuSo = tuSo;
        this.mauSo = mauSo;
    }
    public PhanSo() {
    }
    public int getTuSo() {
        return tuSo;
    }
    public void setTuSo(int tuSo) {
        this.tuSo = tuSo;
    }
    public int getMauSo() {
        return mauSo;
    }
}
```



```

    }
    public void setMauSo(int mauSo) {
        this.mauSo = mauSo;
    }
}

```

- Xây dựng các phương thức tính toán trên 2 phân số:
- Tổng hai phân số

```

private PhanSo tinhTong(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo()
        + phanSo2.getTuSo() * phanSo1.getMauSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}

```

- Hiệu hai phân số

```

private PhanSo tinhHieu(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo()
        - phanSo2.getTuSo() * phanSo1.getMauSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}

```

- Tích hai phân số



```
private PhanSo tinhTich(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getTuSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}
```

- Thương hai phân số

```
private PhanSo tinhThuong(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getTuSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}
```

- Xây dựng phương thức tính toán trên hai phân số dựa vào phép tính mà người dùng chọn

```
private PhanSo tinh(String phepToan, PhanSo phanSo1,
    PhanSo phanSo2){
    if("+".equals(phepToan)){
        return tinhTong(phanSo1, phanSo2);
    }else if("-".equals(phepToan)){
        return tinhHieu(phanSo1, phanSo2);
    }else if("*".equals(phepToan)){
        return tinhTich(phanSo1, phanSo2);
    }
}
```





```


    }else if("/".equals(phepToan)){
        return tinhThuong(phanSo1, phanSo2);
    }
    return null;
}

```

- Gọi các phương thức đã xây dựng vào xử lý của nút "Tính" > xuất kết quả
  - Khai báo và khởi tạo 2 phân số
  - Gọi phương tinh(String phepToan, PhanSo phanSo1, PhanSo phanSo2) để tính toán
  - In kết quả

## 6.2. Quản lý CD

- ✓ **Viết chương trình xây dựng đối tượng CD và quản lý các CD như sau:**



Quản lý CD

Mã CD:  Tên CD:

Ca sĩ:  Số bài hát:  Giá thành:

cd001 - Chúc mừng Năm mới - Hồ Quang Hiếu - 10 - 120000.0  
 cd002 - Đêm giao thừa nghe khúc dân ca - Phương Mỹ Chi - 8 - 150000.0  
 cd003 - Ngày Hạnh Phúc - Cầm Ly - 10 - 120000.0

Tổng giá thành = 390000.0

✓ **Hướng dẫn sử dụng:**

- Khi người dùng nhập các thông tin của CD và nhấn nút “Thêm mới” thì ghi lại thông tin của CD này và đồng thời hiển thị danh sách các CD đang có kèm theo tổng giá thành của các CD.
- Khi người dùng nhấn nút “Tiếp tục” thì xóa nội dung đang có trong các textfield để người dùng nhập thông tin mới.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmQLCD: JFrame Form

\* (Các thể hiện phía dưới đều nằm trong form)

- txtMaCD: TextField (nhập liệu)
- txtTenCD: TextField (nhập liệu)



- txtCaSi: TextField (nhập liệu)
- txtSoBaiHat: TextField (nhập liệu)
- txtGiaThanh: TextField (nhập liệu)
- btnGhi: Button (xử lý lấy và xuất kết quả)
- btnTiep: Button (xử lý xóa nội dung trên các điều khiển)
- **Nhập:**
  - Mã CD: chuỗi ký tự
  - Tên CD: chuỗi ký tự
  - Ca sĩ: chuỗi ký tự
  - Số bài hát: số nguyên
  - Giá thành : số thực
- **Xuất:**
  - Các CD
  - Tổng giá thành của các CD

### ✓ **Hướng dẫn**

Trong project làm việc,

- Xây dựng lớp CD như sau:
  - Các thuộc tính gồm có: Mã CD, tên CD, ca sĩ, số bài hát, giá thành
  - Xây dựng phương thức khởi tạo cho lớp CD và các phương thức get, set

```
public class CD {
    public CD() {
    }
    String maCD;
```



```
String tenCD;
String caSi;
int soBaiHat;
double giaThanh;

public CD(String maCD, String tenCD, String caSi, int soBaiHat, double
giaThanh) {
    this.maCD = maCD;
    this.tenCD = tenCD;
    this.caSi = caSi;
    this.soBaiHat = soBaiHat;
    this.giaThanh = giaThanh;
}

public String getMaCD() {
    return maCD;
}

public void setMaCD(String maCD) {
    this.maCD = maCD;
}

public String getTenCD() {
    return tenCD;
}

public void setTenCD(String tenCD) {
    this.tenCD = tenCD;
}

public String getCaSi() {
    return caSi;
}
```



```

    }
    public void setCaSi(String caSi) {
        this.caSi = caSi;
    }
    public int getSoBaiHat() {
        return soBaiHat;
    }
    public void setSoBaiHat(int soBaiHat) {
        this.soBaiHat = soBaiHat;
    }
    public double getGiaThanh() {
        return giaThanh;
    }
    public void setGiaThanh(double giaThanh) {
        this.giaThanh = giaThanh;
    }
}

```

- Tạo lớp JFrame có tên là frmQLCD
  - Khai báo 2 biến toàn cục là chuỗi kết quả để lưu thông tin các CD và tổng giá thành để tính tổng giá thành của các CD được thêm vào

```

public double tong = 0;
String chuoiCD = "";

```

- Xử lý nút “Thêm mới”:
  - Lấy thông tin CD mà người dùng nhập đưa vào chuỗi kết quả

- Gán chuỗi kết quả cho TextArea đã chuẩn bị sẵn

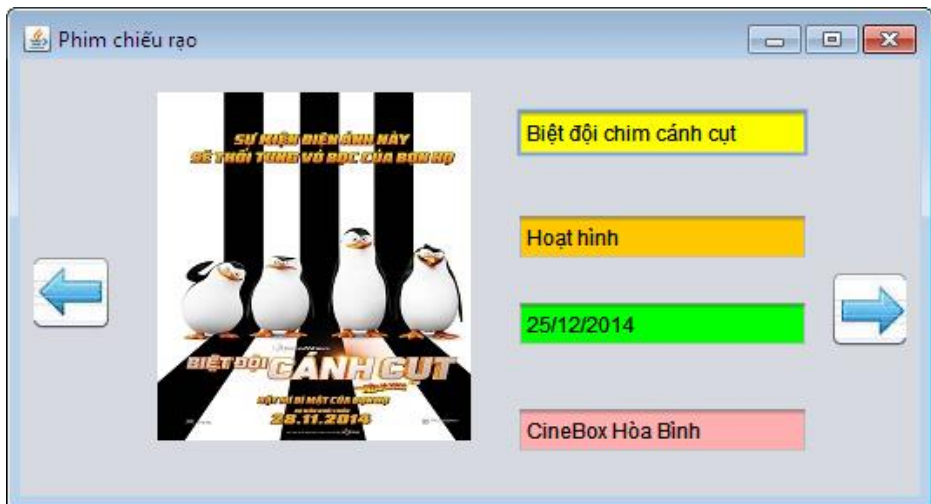
```
private void btnThemActionPerformed(java.awt.event.ActionEvent evt) {
    CD cd = new CD(txtMaCD.getText(), txtTenCD.getText(),
    txtCaSi.getText(),
    Integer.parseInt(txtSoBaiHat.getText()), Double.parseDouble(
    txtGiaThanh.getText()));

    chuoicd += cd.getMaCD() + " - " + cd.getTenCD() + " - " +
    cd.getCaSi() + " - " + cd.getSoBaiHat() + " - " + cd.getGiaThanh() + "\n";
    tong += cd.giaThanh;
    txtaMangCD.setText(chuoicd + "\n Tổng giá thành = " + tong );
}
```

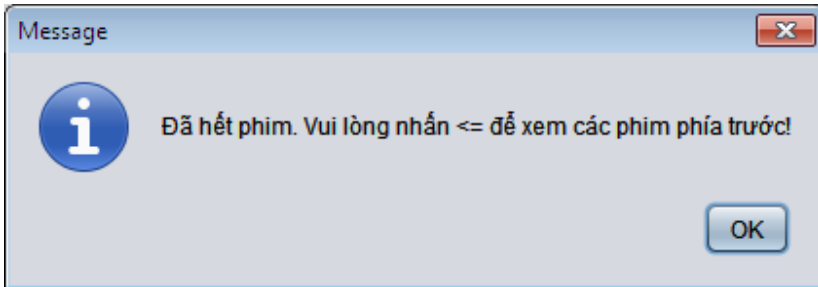
- Xử lý nút "Tiếp tục": xóa nội dung của các điều khiển nhập liệu để người dùng tiếp tục nhập thông tin

### 6.3. Phim chiếu rạp

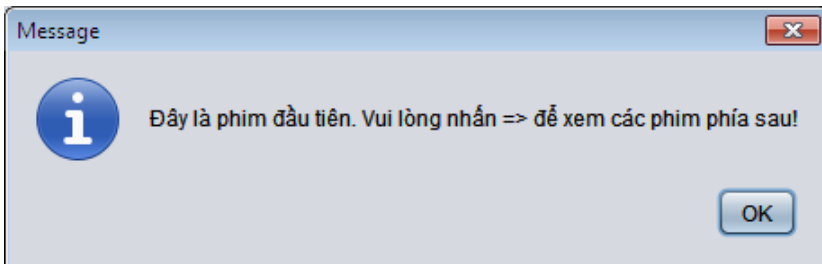
- ✓ **Yêu cầu:** Viết chương trình hiển thị danh sách Phim chiếu rạp như sau:



Khi khởi tạo chương trình



Tiếp tục nhấn "<=>" khi đã đến phim cuối cùng



Tiếp tục nhấn "<=" Khi ở phim đầu tiên

✓ **Hướng dẫn sử dụng:**

- Bấm nút tới hoặc lùi để hiển thị thông tin phim tiếp theo/ trước đó

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmDSPhim: form

\* (Các thể hiện phía dưới đều nằm trong form)

- lblHinhAnh: Label (hiển thị hình ảnh của phim)
- txtTenPhim: TextField (kết xuất)
- txtTheLoai: TextField (kết xuất)
- txtNgayChieu: TextField (kết xuất)



- txtRap: TextField (kết xuất)
- btnNext: Button (xử lý hiển thị thông tin phim tiếp theo)
- btnBack: Button (xử lý hiển thị thông tin phim phía trước)
- **Nhập:**
  - Không có
- **Xuất:**
  - Danh sách Phim
- ✓ **Hướng dẫn**
  - Trong project làm việc:
  - Chuẩn bị các hình ảnh và đưa vào thư mục images (HV tự chọn hình ảnh với kích cỡ tương đối khoảng W = 180 pixel và H = 200 pixel)
  - Xây dựng lớp Phim như sau:
    - Các thuộc tính gồm có: tên phim, thể loại, ngày chiếu, rap, đường dẫn hình
    - Phương thức khởi tạo mặc định và phương thức khởi tạo có tham số truyền vào, các phương thức get và set

```
public class Phim {
    String tenPhim;
    String theLoai;
    String ngayChieu;
    String rap;
    String duongDanHinh;

    public Phim(String tenPhim, String theLoai, String ngayChieu, String rap,
```





```
String duongDanHinh) {
    this.tenPhim = tenPhim;
    this.theLoai = theLoai;
    this.ngayChieu = ngayChieu;
    this.rap = rap;
    this.duongDanHinh = duongDanHinh;
}

public Phim() {
}

public String getTenPhim() {
    return tenPhim;
}

public void setTenPhim(String tenPhim) {
    this.tenPhim = tenPhim;
}

public String getTheLoai() {
    return theLoai;
}

public void setTheLoai(String theLoai) {
    this.theLoai = theLoai;
}

public String getNgayChieu() {
    return ngayChieu;
}

public void setNgayChieu(String ngayChieu) {
    this.ngayChieu = ngayChieu;
}
```



```

    }

    public String getRap() {
        return rap;
    }

    public void setRap(String rap) {
        this.rap = rap;
    }

    public String getDuongDanHinh() {
        return duongDanHinh;
    }

    public void setDuongDanHinh(String duongDanHinh) {
        this.duongDanHinh = duongDanHinh;
    }
}

```

– Tạo lớp JFrame có tên là frmDSPhim

- Tạo 2 biến toàn cục là vị trí = 0 và mảng phim có 5 phần tử.

```

public Phim[] phim = new Phim[5];
public int flag = 0;

```

- Trong phương thức khởi tạo của form: Tạo 5 phim tùy thích (kiểu Phim) và đưa 5 phim này vào mảng. thiết lập các thuộc tính cho các điều khiển trên màn hình hiển thị. Khi form load lên thì hiển thị phim đầu tiên trong danh sách (vị trí = 0)

```

public frmPhimChieuRap() {
    initComponents();
    Phim p1 = new Phim("Biệt đội chim cánh cụt", "Hoạt hình",

```



```
"25/12/2014", "CineBox Hòa Bình", "/images/bietdoichimcanhcut.jpg");

    Phim p2 = new Phim("Big Heros 6", "Hoạt hình", "10/01/2015", "Galaxy
Nguyễn Du", "/images/bighero6.jpg");

    Phim p3 = new Phim("Chàng trai năm ấy", "Tình cảm", "15/01/2015",
"CineBox Lý Chính Thắng", "/images/changtrainamay.jpg");

    Phim p4 = new Phim("Cuộc chiến chống Pharaon", "Giả sử",
"20/01/2015", "Galaxy Nguyễn Trãi", "/images/cuocchienchongpharaon.jpg");

    Phim p5 = new Phim("Đề mai tính", "Hài", "25/01/2015", "Galaxy Quang
Trung", "/images/demaitinh.jpg");

    phim[0] = p1;
    phim[1] = p2;
    phim[2] = p3;
    phim[3] = p4;
    phim[4] = p5;

    txtTenPhim.setBackground(Color.yellow);
    txtTenPhim.setText(phim[0].getTenPhim());

    txtRap.setBackground(Color.PINK);
    txtRap.setText(phim[0].getRap());

    txtNgayChieu.setBackground(Color.GREEN);
    txtNgayChieu.setText(phim[0].getNgayChieu());

    txtTheLoai.setBackground(Color.ORANGE);
    txtTheLoai.setText(phim[0].getTheLoai());

    lblHinh.setIcon(new
```



```
javax.swing.ImageIcon(getClass().getResource(phim[0].getDuongDanHinh()))
);
}
```

- btnNext hiển thị thông tin phim kế tiếp (vị trí ++)

```
private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    if (flag < 4) {
        flag++;
        txtTenPhim.setText(phim[flag].getTenPhim());
        txtRap.setText(phim[flag].getRap());
        txtNgayChieu.setText(phim[flag].getNgayChieu());
        txtTheLoai.setText(phim[flag].getTheLoai());
        lblHinh.setIcon(new
        javax.swing.ImageIcon(getClass().getResource(phim[flag].getDuongDanHinh(
        ))));
    } else {
        JOptionPane.showMessageDialog(rootPane, "Đã hết phim. Vui lòng
        nhấn <= để xem các phim phía trước!");
    }
}
```

- btnBack hiển thông tin phim trước đó (vị trí --)

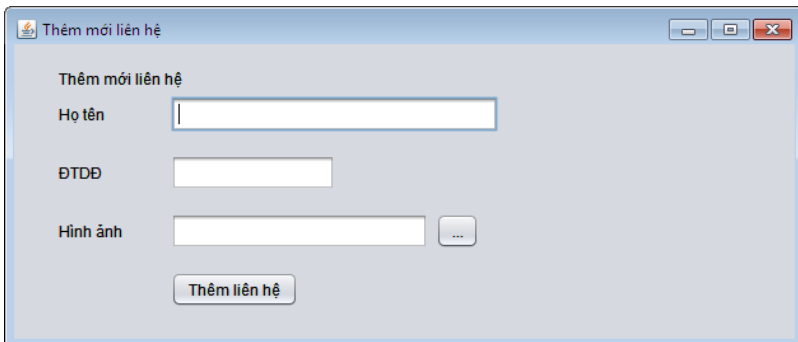
```
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    if (flag == 0) {
        JOptionPane.showMessageDialog(rootPane, "Đây là phim đầu tiên.
        Vui lòng nhấn => để xem các phim phía sau!");
    } else {
        flag--;
    }
}
```

```
txtTenPhim.setText(phim[flag].getTenPhim());
txtRap.setText(phim[flag].getRap());
txtNgayChieu.setText(phim[flag].getNgayChieu());
txtTheLoai.setText(phim[flag].getTheLoai());
lblHinh.setIcon(new
javax.swing.ImageIcon(getClass().getResource(phim[flag].getDuongDanHinh(
))));
}
}
```

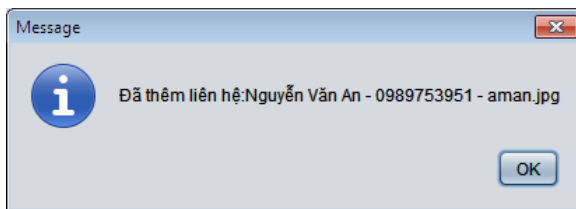
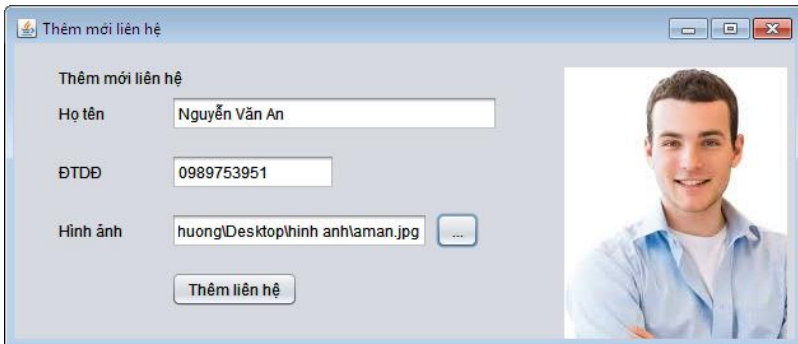
## 6.4. Thêm liên hệ mới

- ✓ **Yêu cầu:** Xây dựng chương trình Thêm mới Liên hệ có các thông tin sau:

- Tên
- ĐTDĐ
- Hình ảnh



*Form Thêm liên hệ mới khi chạy*



*Sau khi thực hiện thao tác "Thêm liên hệ"*

✓ **Hướng dẫn sử dụng:**

- Nhập vào các ô nhập liệu các thông tin: tên, điện thoại di động, hình ảnh. Nhấn nút "Thêm" => Các thông tin vừa nhập sẽ được lưu vào tập tin LienHe.txt.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng (sử dụng giao diện đã thiết kế ở bài Giao diện):**

- frmManHinhThemLienHe: FrmThemLienHe (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (nhập liệu)
- txtDtdd: JTextField (nhập liệu)
- fchHinhAnh: JFileChooser (chọn)
- btnThem: JButton (Xử lý thêm)



▪ **Nhập:**

- Tên
- ĐTDĐ
- Hình ảnh...

▪ **Xuất:**

- “Thông tin Liên hệ đã được thêm vào tập tin” nếu thêm được hoặc “Không thể thêm Liên hệ”

✓ **Thuật giải**

- Tạo lớp LienHe
- Hiển thị đường dẫn file được chọn trong txtHinhAnh
- Khai báo biến hoTen nhận giá trị từ txtHoTen
- Khai báo biến dtdd nhận giá trị từ txtDtdd
- Khai báo biến hinhAnh nhận giá trị từ txtHinhAnh
- Xử lý đưa hình ảnh vào thư mục hình ảnh
- Khai báo và khởi tạo biến lh kiểu LienHe lấy thông tin từ các biến trên
- Xử lý ghi vào file đối tượng lh
- Xuất kết quả dưới dạng dialog thông báo

✓ **Hướng dẫn**

- Copy lại form Thêm mới liên hệ đã làm ở bài 3 và đưa vào bài này để sử dụng lại
- Tạo tập tin lienhe.txt để lưu trữ thông tin các liên hệ
- Xây dựng lớp LienHe với các thuộc tính: tên, dtdd, hình, các phương thức khởi tạo, getter và setter

```
public class LienHe implements Serializable{
    String hoTen;
```



```
String dtdd;
String hình;
public LienHe(String hoTen, String dtdd, String hình) {
    this.hoTen = hoTen;
    this.dtdd = dtdd;
    this.hình = hình;
}
public LienHe() {
}
public String getHoTen() {
    return hoTen;
}
public void setHoTen(String hoTen) {
    this.hoTen = hoTen;
}
public String getDtdd() {
    return dtdd;
}
public void setDtdd(String dtdd) {
    this.dtdd = dtdd;
}
public String getHinh() {
    return hình;
}
public void setHinh(String hình) {
    this.hình = hình;
}
```





```
}
```

```
}
```

- Tạo thư mục images/contact để lưu trữ hình ảnh của liên hệ
- Xử lý sự kiện cho nút "Thêm liên hệ":
  - Đưa hình ảnh vào thư mục contact trong thư mục images của ứng dụng
  - Sử dụng lớp ObjectOutputStream, FileOutputStream để ghi đối tượng liên hệ vào tập tin > xuất thông báo sau khi ghi xong.

```
private void btnHienThiActionPerformed(java.awt.event.ActionEvent evt) {
    // đưa tập tin hình ảnh được chọn vào thư mục
    File f = new File(txtHinhAnh.getText());
    String path = "src/images/contact/";
    String duong_dan = path + f.getName();
    try {
        Files.copy(f.toPath(),
            (new File(path + f.getName())).toPath(),
            StandardCopyOption.REPLACE_EXISTING);
        System.out.println("Đã đưa hình liên hệ thư mục contact");
    } catch (IOException ex) {
        Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
            null, ex);
    }
    // lấy dữ liệu gán cho đối tượng liên hệ sau đó ghi vào tập tin lienhe.txt
    String hoTen = txtHoTen.getText();
    String dtdd = txtDTDD.getText();
    String hinhAnh = f.getName();
}
```



```

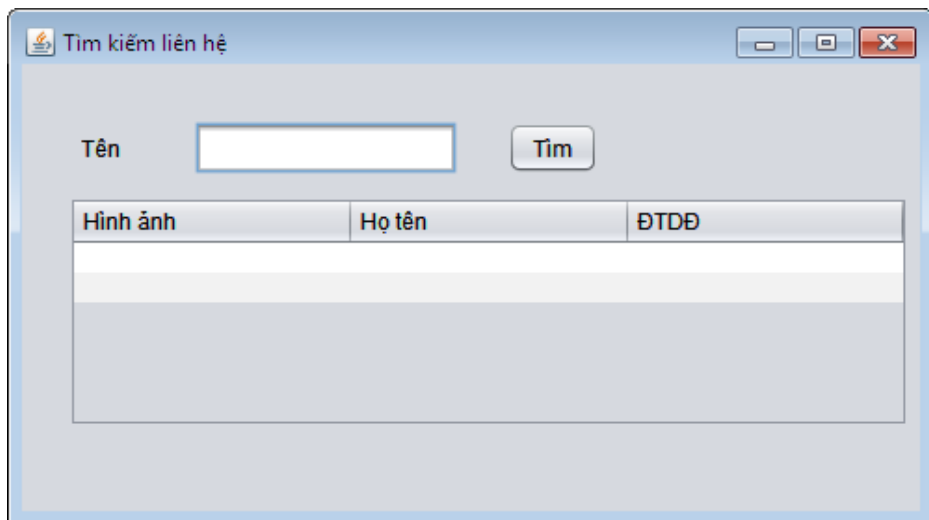
LienHe h = new LienHe(hoTen, dtdd, hinhAnh);
ObjectOutputStream out;
try {
    out = new ObjectOutputStream(new
FileOutputStream("src/bai6/lienhe.txt", true)) {
        @Override
        protected void writeStreamHeader() throws IOException {
            reset();
        }
    };
    out.writeObject(h);
    out.close();
    JOptionPane.showMessageDialog(rootPane, "Đã thêm liên hệ:" +
h.hoTen + " - " + h.dtdd + " - " + h.hinh);
} catch (FileNotFoundException ex) {
    Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
null, ex);
} catch (IOException ex) {
    Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
null, ex);
}
}

```

- Chú ý: Để tránh trường hợp không ghi tiếp được vào tập tin, có thể đọc tất cả các LienHe đang có trong tập tin ra trước và đưa vào List<LienHe>, khi người dùng nhập thông tin thì khởi tạo liên hệ mới và đưa thông tin này vào. Tiếp đó đưa liên hệ này vào list. Và ghi tất cả các liên hệ trong list vào tập tin.

## 6.5. Tìm kiếm liên hệ

- ✓ **Yêu cầu: Xây dựng chương trình tìm kiếm liên hệ như sau:**

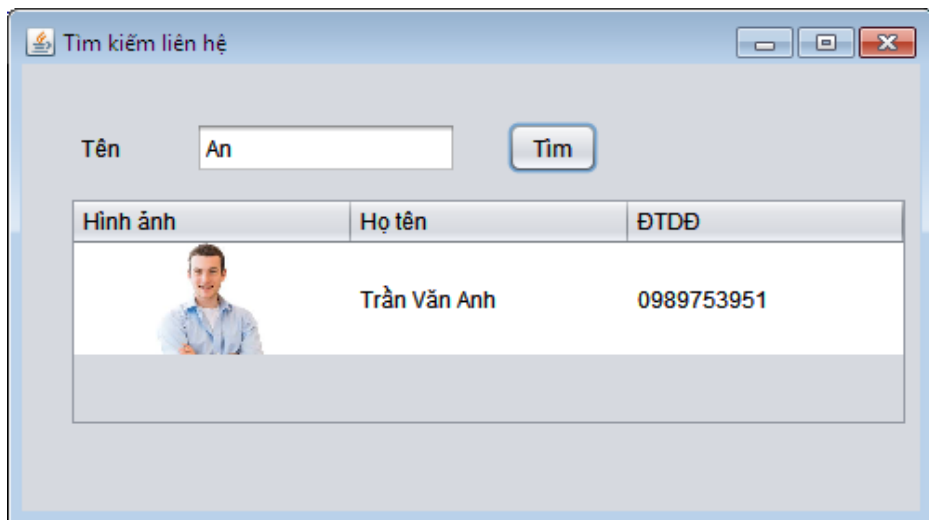


Tên

Tìm


Hình ảnh	Họ tên	ĐTDD

*Form tìm kiếm liên hệ khi chạy*

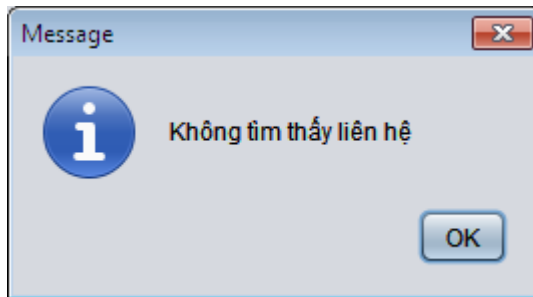


Tên

Tìm

Hình ảnh	Họ tên	ĐTDD
	Trần Văn Anh	0989753951

*Sau khi nhập tên và nhấn tìm kiếm nếu tìm thấy*



*Nếu không tìm thấy*

✓ **Hướng dẫn sử dụng:**

- Nhập vào tên => Nhấn nút "Tìm" => Hiển thị kết quả tìm kiếm lên table

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng**

- frmManHinhTimLienHe: FrmTimLienHe (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (nhập liệu)
- tblKetQua: Jtable (kết xuất)
- btnTim: JButton (Xử lý tìm kiếm và hiển thị kết quả)

▪ **Nhập:**

- Tên

▪ **Xuất:**

- Kết quả tìm kiếm

✓ **Hướng dẫn**

- Sử dụng lớp ImageTableCellRenderer để hiển thị hình ảnh như sau (Được cung cấp sẵn - HV download từ cổng thông tin):



```
public class ImageTableCellRenderer extends DefaultTableCellRenderer {
    private static final long serialVersionUID = -7959113581100753271L;
    private int imageWidth;
    private int imageHeight;
    public ImageTableCellRenderer(int imageWidth, int imageHeight) {
        this.imageWidth = imageWidth;
        this.imageHeight = imageHeight;
    }
    @Override
    public Component getTableCellRendererComponent(JTable table,
        Object value, boolean isSelected, boolean hasFocus, int row, int column) {
        super.getTableCellRendererComponent(table, value,
            isSelected, hasFocus, row, column);
        try {
            BufferedImage image = null;
            URL url = new URL("file:" + value);
            image = getScaledImages(ImageIO.read(url),
                imageWidth, imageHeight);
            setIcon(new ImageIcon(image));
            setHorizontalAlignment(JLabel.CENTER);
            setText("");
        } catch (Exception e) {
            setText("[No image]");
            e.printStackTrace();
        }
        return this;
    }
}
```



```
private BufferedImage getScaledImages(BufferedImage in, int
WIDTH, int HEIGHT) {
    BufferedImage out = new BufferedImage(WIDTH, HEIGHT,
        BufferedImage.TYPE_INT_RGB);
    Graphics2D g2 = out.createGraphics();
    g2.setColor(Color.white);
    g2.fillRect(0, 0, WIDTH, HEIGHT);
    double width = in.getWidth();
    double height = in.getHeight();
    double xScale = WIDTH / width;
    double yScale = HEIGHT / height;
    double scale = 1.0;

    scale = Math.min(xScale, yScale); // scale to fit

    double x = (WIDTH - width * scale) / 2;
    double y = (HEIGHT - height * scale) / 2;
    AffineTransform at = AffineTransform.getTranslateInstance(x,
y);

    at.scale(scale, scale);
    g2.drawRenderedImage(in, at);
    g2.dispose();
    return out;
}
}
```

- Khai báo biến toàn cục là danh sách Liên hệ lưu thông tin các liên hệ



```
List<LienHe> lst;
```

- Khi khởi tạo form, đọc toàn bộ các liên hệ trong tập tin và đưa vào danh sách

```
public frmTimKiemLienHe() {
    initComponents();
    lst = new ArrayList<>();
    try (ObjectInputStream in = new ObjectInputStream(
        new BufferedInputStream(new
        FileInputStream("src/bai5/lienhe.txt")))) {
        while(true){
            LienHe b = (LienHe) in.readObject();
            lst.add(b);
        }
    } catch (ClassNotFoundException | IOException ex) {
        System.out.println(ex.getMessage());
    }
}
```

- Xử lý cho nút "Tìm": kiểm tra trong danh sách liên hệ và tìm ra những liên hệ mà tên có chứa từ tìm kiếm mà người dùng nhập trên màn hình. Tạo một DefaultTableModel để chứa các liên hệ này > Hiển thị model này vào tblKetQua (Hiển thị hình ảnh bằng cách tạo TableCellRenderer cho cột "Hình ảnh")

```
private void btnTimActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("Hình ảnh");
    model.addColumn("Họ tên");
```



```

model.addColumn("ĐTDĐ");
boolean flag = false;
for (LienHe lh : lst) {
    if (lh.getHoTen().contains(txtTen.getText())) {
        ImageIcon img = new ImageIcon("src/images/contact/"
+ lh.hinh);
        model.addRow(new Object[]{img, lh.hoTen, lh.dtdd});
        flag = true;
    }
}
if (!flag) {
    JOptionPane.showMessageDialog(rootPane, "Không tìm thấy
liên hệ");
} else {
    tblKetQua.setRowHeight(60);
    tblKetQua.setModel(model);
    tblKetQua.getColumnModel().getColumn(0).setCell
Renderer(new Bai5.ImageTableCellRenderer(60, 60));
}
}

```

## 6.6. Mua sắm online (làm thêm)

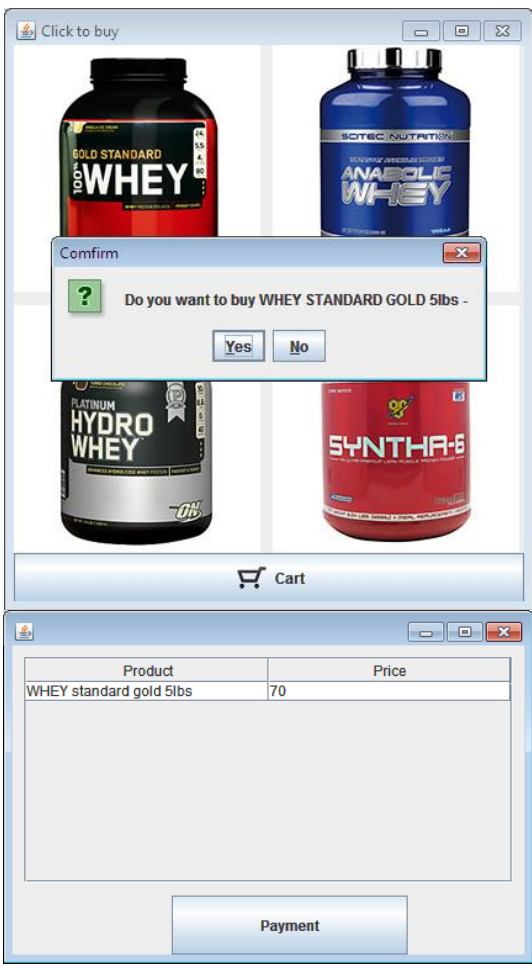
- ✓ Yêu cầu: viết chương trình mua sắm online



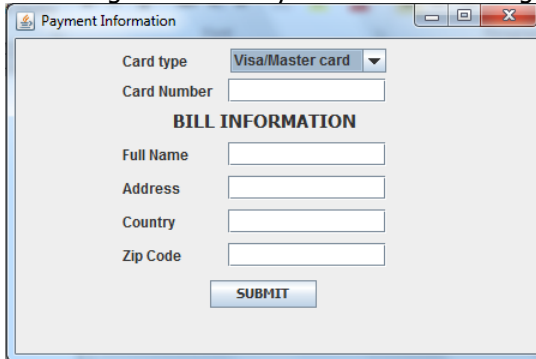


✓ **Hướng dẫn sử dụng:**

- Click vào hình để mua hàng, sau đó bấm nút "Cart" để thanh toán



- Cuối cùng bấm nút “Payment” để điền thông tin thanh toán



✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- FrmShopping: form
  - \* (Các thể hiện phía dưới đều nằm trong form)
    - lblProduct1: Label (hiển thị sản phẩm)
    - lblProduct2: Label (hiển thị sản phẩm)
    - lblProduct3: Label (hiển thị sản phẩm)
    - lblProduct4: Label (hiển thị sản phẩm)
    - btnCart: Button (xử lý hiển thị FrmInCart có sản phẩm đã chọn)
- FrmInCart: form
  - \* (Các thể hiện phía dưới đều nằm trong form)
    - tblProducts: Table (danh sách sản phẩm đã chọn)
    - btnPayMent: Button (xử lý hiển thị FrmPayment)
- FrmPayment: form
  - \* (Các thể hiện phía dưới đều nằm trong form)
    - txtName: TextField (nhập liệu)
    - txtAdd: TextField (nhập liệu)



- txtContry: TextField (nhập liệu)
- txtZipCode: TextField (nhập liệu)
- txtCardNumber: TextField (nhập liệu)
- cbbTypeCard: ComboBox
- btnSubmit: Button (xử lý xác nhận mua hàng)
- **Nhập:**
  - FrmShopping: Chọn sản phẩm
  - FrmInCart: không có
  - FrmPayment: chọn loại thẻ, số thẻ, tên, địa chỉ, thành phố, zip code
- **Xuất:**
  - FrmShopping: không có
  - FrmInCart: sản phẩm được chọn
  - FrmPayment: không có
- ✓ **Hướng dẫn**
  - Học viên tạo thư mục images/ trong project làm việc sau đó tùy ý tìm hình thích hợp cho 4 sản phẩm và gán cho các label

# BÀI 7: Lập trình hướng đối tượng nâng cao



*Mục tiêu chính:*

*Hiện thực ứng dụng áp dụng kế thừa*

*Áp dụng kiến thức về Abstract class và Interface để giải quyết các bài toán*

## 7.1. Animal say hello

- ✓ **Yêu cầu:** Xây dựng chương trình hiển thị thông tin của một số con vật



Animal says hello

Name: Mimi

Animal: Cat

Say Hello

Mimi says Meo meo meo

- ✓ **Hướng dẫn sử dụng:**



- Nhập tên và chọn một con vật, nhấn button "Say hello", chương trình xuất tên, tiếng kêu và hình ảnh của con vật được chọn.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmAnimalSayHello: FrmAnimalSayHello (extends từ JFrame)

\* (Các thể hiện phía dưới đều nằm trong Frame)

- txtName: JTextField (nhập liệu)
- cmbAnimal: JComboBox (chọn Dog/Cat/Pig)
- lblHello: JLabel (kết xuất)
- lblImage: JLabel (kết xuất)
- btnSayHello: JButton (Xử lý hiển thị thông tin)

▪ **Nhập:**

- Tên
- Chọn 1 con vật

▪ **Xuất:**

- Tên, tiếng kêu, hình ảnh

▪ **Qui tắc xử lý:**

- Không có

✓ **Thuật giải**

- Không có

✓ **Hướng dẫn**

- Xây dựng lớp Animal, có các thuộc tính name, image. Trong lớp này có phương thức sayHello() trả về chuỗi "Hello".

```
public class Animal {
```



```
String name;
String image;
public String getName() {
    return name;
}
public String getImage() {
    return image;
}
public Animal() {
}
public Animal(String name, String image) {
    this.name = name;
    this.image = image;
}
public String sayHello() {
    return "Hello!";
}
}
```

- Xây dựng các lớp Dog/ Cat/ Pig ... kế thừa từ lớp Animal có phương thức sayHello() trả về chuỗi chào tương ứng với từng loại, ví dụ như: "Gâu gâu gâu", "Meo meo meo", "Oii oii oii"

```
public class Dog extends Animal {
    @Override
    public String sayHello() {
        return "Gâu gâu gâu";
    }
    public Dog() {
    }
    public Dog(String name, String image) {
```



```
        super(name, image);
    }
}
```

// Học viên làm tương tự cho lớp Cat/ Pig

- Trên form, khởi tạo lớp và xuất kết quả tương ứng với lựa chọn của người dùng.

```
private void btnSayActionPerformed(java.awt.event.ActionEvent evt) {
    String name = txtName.getText();
    String imagePath = "src/images/";
    ImageIcon img = null;
    switch (cbbAnimal.getSelectedItem().toString()) {
        case "Dog":
            Dog d = new Dog(name, "1_dog.png");
            lblHello.setText(d.name + " says " + d.sayHello());
            img = new ImageIcon(imagePath + d.getImage());
            lblImage.setIcon(img);
            break;
        case "Cat":
            Cat c = new Cat(name, "1_cat.png");
            lblHello.setText(c.name + " says " + c.sayHello());
            img = new ImageIcon(imagePath + c.getImage());
            lblImage.setIcon(img);
            break;
        default:
            Pig p = new Pig(name, "1_pig.jpg");
            lblHello.setText(p.name + " says " + p.sayHello());
            img = new ImageIcon(imagePath + p.getImage());
    }
}
```





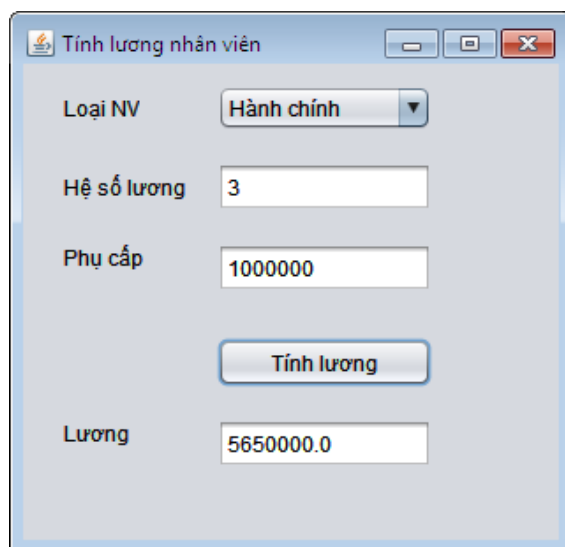
```
        lblImage.setIcon(img);
        break;
    }
}
```

## 7.2. Tính lương nhân viên

✓ **Yêu cầu: Xây dựng ứng dụng tính lương nhân viên**

**Lương được tính theo công thức sau:**

- Lương cơ bản cố định: 1.550.000
- Nhân viên = lương cơ bản \* hệ số
- Nhân viên hành chính = lương cơ bản \* hệ số + phụ cấp
- Nhân viên kỹ thuật = lương cơ bản \* hệ số + (số dự án hoàn thành trong tháng x 1.000.000)
- Nhân viên kinh doanh = lương cơ bản \* hệ số + (số sản phẩm \* 20.000)
- Nhân viên lãnh đạo = lương cơ bản \* hệ số + 10% tổng lương (tiền trách nhiệm)



Tính lương nhân viên

Loại NV: Hành chính

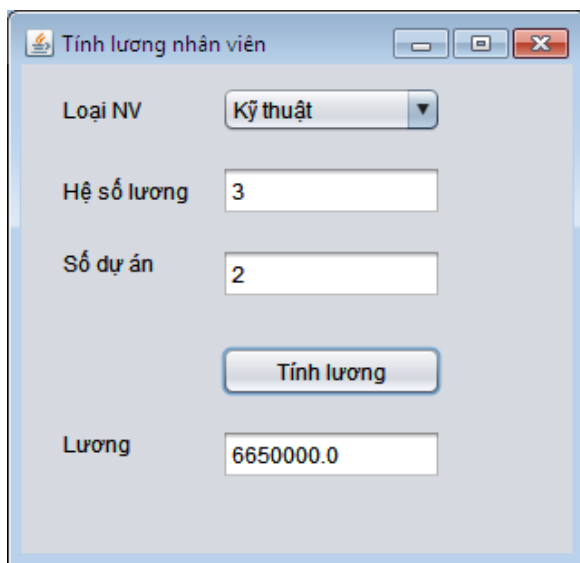
Hệ số lương: 3

Phụ cấp: 1000000

Tính lương

Lương: 5650000.0

Khi chọn "Hành chính" và điền các số liệu



Tính lương nhân viên

Loại NV: Kỹ thuật

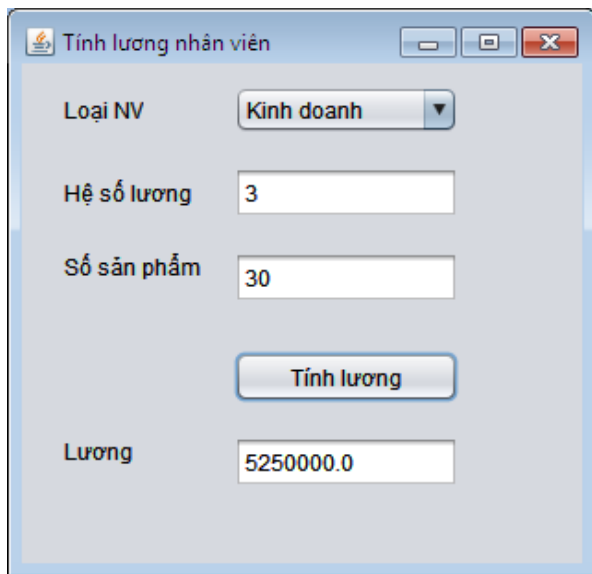
Hệ số lương: 3

Số dự án: 2

Tính lương

Lương: 6650000.0

Khi chọn "Kỹ thuật" và điền các số liệu



Tính lương nhân viên

Loại NV: Kinh doanh

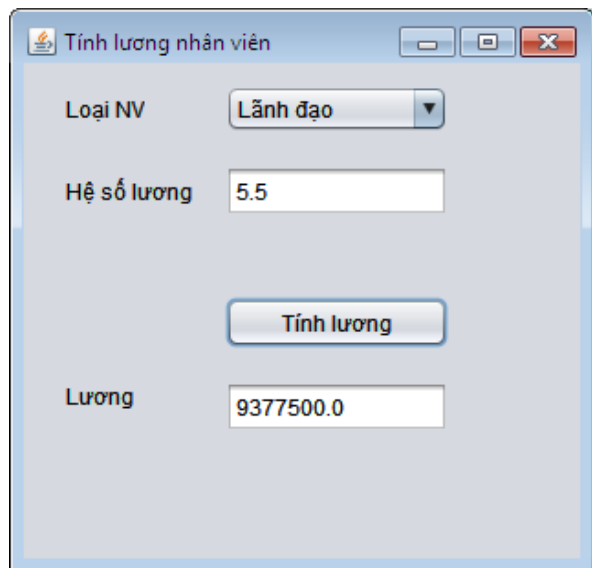
Hệ số lương: 3

Số sản phẩm: 30

Tính lương

Lương: 5250000.0

Khi chọn "Kinh doanh" và điền các số liệu



Tính lương nhân viên

Loại NV: Lãnh đạo

Hệ số lương: 5.5

Số sản phẩm: 30

Tính lương

Lương: 9377500.0

Khi chọn "Lãnh đạo" và điền các số liệu



✓ **Hướng dẫn sử dụng:**

- Chọn loại nhân viên
- Nhập các giá trị theo loại nhân viên
- Nhấn nút “Tính lương”

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmTinhLuong: FrmTinhLuong

*\* Học viên tự xây dựng form như hình trên, lưu ý cần thay đổi nội dung label sau khi chọn loại nhân viên cho phù hợp*

▪ **Nhập:**

- Các thông tin tương ứng với từng loại nhân viên

▪ **Xuất:**

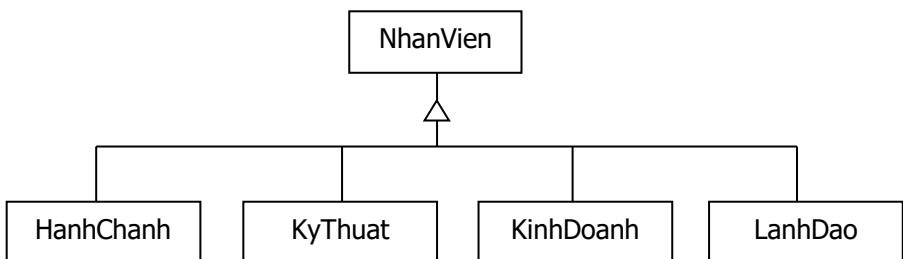
- Lương của nhân viên

▪ **Qui tắc xử lý:**

- Tính lương như yêu cầu trên

✓ **Thuật giải**

Thiết kế lớp theo sơ đồ sau:



✓ **Hướng dẫn**

- Xây dựng lớp NhanVien có lương cơ bản = 1.550.000 là hằng số và phương thức tinhLuong()



```
public class NhanVien {
    final double LUONGCOBAN = 1550000;
    double hesoLuong;
    public NhanVien(double hesoLuong) {
        this.hesoLuong = hesoLuong;
    }
    public double getHesoLuong() {
        return hesoLuong;
    }
    public void setHesoLuong(double hesoLuong) {
        this.hesoLuong = hesoLuong;
    }
    public double tinhLuong(){
        return this.LUONGCOBAN * this.hesoLuong;
    }
}
```

- Xây dựng các lớp HanhChanh có thêm thuộc tính phụ cấp và phương thức tinhLuong() trả về lương cơ bản \* hệ số + phụ cấp

```
public class HanhChinh extends NhanVien {
    double phuCap;
    public HanhChinh(double phuCap, double hesoLuong) {
        super(hesoLuong);
        this.phuCap = phuCap;
    }
}
```



```

public double getPhuCap() {
    return phuCap;
}

public void setPhuCap(double phuCap) {
    this.phuCap = phuCap;
}

@Override
public double tinhLuong() {
    return super.tinhLuong() + this.phuCap;
}
}
    
```

- Thực hiện tương tự cho các lớp KyThuat, KinhDoanh, LanhDao kế thừa từ lớp NhanVien có thuộc tính riêng và phương thức tinhLuong() trả về lương tương ứng với từng loại nhân viên theo công thức tính lương trên.
- Trên form, khởi tạo lớp và tính lương ứng với lựa chọn của người dùng.

```

private void btnTinhActionPerformed(java.awt.event.ActionEvent evt) {
    double hesoLuong = Double.parseDouble(txtHeSo.getText());
    int thongSo = Integer.parseInt(txtThongSo.getText());

    switch (cbbLoaiNV.getSelectedIndex()) {
        case 0: // hanh chinh
            HanhChinh hc = new HanhChinh(thongSo, hesoLuong);
    }
}
    
```




```

        txtLuong.setText(String.valueOf(hc.tinhLuong()));
        break;
    case 1: // ky thuat
        Kythuat kt = new Kythuat(thongSo, hesoLuong);
        txtLuong.setText(String.valueOf(kt.tinhLuong()));
        break;
    case 2: // kinh doanh
        KinhDoanh kd = new KinhDoanh(thongSo, hesoLuong);
        txtLuong.setText(String.valueOf(kd.tinhLuong()));
        break;
    case 3: // lanh dao
        LanhDao ld = new LanhDao(hesoLuong);
        txtLuong.setText(String.valueOf(ld.tinhLuong()));
        break;
    }
}

```

### 7.3. Quản lý sách thư viện

- ✓ **Yêu cầu:** Xây dựng ứng dụng quản lý sách thư viện


 Quản lý sách

Thông tin chung

Mã sách

gk002

Tên sách

PHP & MySQL tập 1

Ngày nhập

26/03/2015

NXB

ĐHQG

Đơn giá

36000

Số lượng

50

☒ Sách giáo khoa
 ☐ Sách tham khảo

Tình trạng

Cũ

Thuế (1-100)

Nhập sách

Tiếp tục

Sách giáo khoa

Sách tham khảo

Tổng thành tiền

5100000.0

gk001 - Quy trình phát triển phần mềm - 26/03/2015 - ĐHQG - 100 - 42000.0 - 4200000.0  
 gk002 - PHP & MySQL tập 1 - 26/03/2015 - ĐHQG - 50 - 36000.0 - 900000.0

*Quản lý các loại sách – tab Sách giáo khoa*



Quản lý sách

Thông tin chung

Mã sách

tk002

Tên sách

Giáo trình Dream Weaver

Ngày nhập

30/03/2015

NXB

Khoa học

Đơn giá

52000

Số lượng

150

☐ Sách giáo khoa
 ☒ Sách tham khảo

Tình trạng

Cũ

Thuế (1-100)

5

Nhập sách

Tiếp tục

Sách giáo khoa

Sách tham khảo

Tổng thành tiền

1.864E7

Đơn giá trung bình

73500.0

tk001 - Kiểm thử phần mềm cơ bản - 28/03/2015 - ĐHQG - 10 - 1.045E7

tk002 - Giáo trình Dream Weaver - 30/03/2015 - Khoa học - 5 - 8190000.0

### *Quản lý các loại sách – tab Sách tham khảo*

Mô tả: Thư viện X quản lý danh sách các loại sách. Thông tin về các loại sách:

- Sách giáo khoa: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ)
  - Nếu tình trạng sách là mới thì: thành tiền = số lượng \* đơn giá
  - Nếu tình trạng sách là cũ thì: thành tiền = số lượng \* đơn giá \* 50%
- Sách tham khảo: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, thuế



- Thành tiền = (số lượng \* đơn giá) + (số lượng \* đơn giá) \* thuế (với thuế có giá trị từ 1% đến 100%)

Dựa vào mô tả trên, hãy:

- Tạo lớp Sach với các thuộc tính và phương thức chung.
- Tạo lớp SachGiaoKhoa và lớp SachThamKhao kế thừa từ lớp sách với các thuộc tính riêng và phương thức cần thiết.
- Nhập xuất danh sách các loại sách.
- Tính tổng thành tiền cho từng loại.
- Tính trung bình cộng đơn giá của các sách tham khảo.

#### ✓ Hướng dẫn sử dụng:

- Nhập thông tin chung về sách, sau đó chọn loại sách và thông tin chi tiết cho loại sách được chọn > Nhấn “Nhập sách” => Hiện thị thông tin các sách trong thư viện với tổng thành tiền của từng loại và đơn giá trung bình của sách tham khảo.
- Nhấn “Tiếp tục” => Các ô nhập liệu sẽ được xóa nội dung để người dùng nhập dữ liệu mới.

#### ✓ Tóm tắt yêu cầu

##### ▪ Thiết kế giao diện người dùng:

- frmQuanLySach: JFrame

*\* (Các thể hiện phía dưới đều nằm trong form)*

- txtMaSach: Text Field (nhập liệu)
- txtTenSach: Text Field (nhập liệu)
- txtNgayNhap: Text Field (kết xuất)
- txtNXB: Text Field (nhập liệu)
- txtDonGia: Text Field (nhập liệu)
- txtSoLuong: Text Field (kết xuất)



- rbtgLoaiSach: Radio Button Group (nhóm 2 radio button Sách giáo khoa và Sách tham khảo)
  - rbtSGK: Radio Button (chọn)
    - cbbTìnhTrang: Combo Box (chọn)
  - rbtSTK: Radio Button (chọn)
    - txtThue: Text Field (nhập liệu)
- btnNhap: Button (xử lý tính toán và in kết quả)
- btnTiepTuc: Button (xử lý xóa nội dung của các điều khiển trên màn hình để người dùng tiếp tục nhập liệu)
- tpIThongKe: Tab Panel (chứa thông tin thống kê)
  - pnlSGK: Panel (chứa thông tin Sách giáo khoa)
    - txtTienSGK: Text Field (kết xuất)
    - lstSGK: List (kết xuất)
  - pnlSTK: Panel (chứa thông tin Sách tham khảo)
    - txtTienSTK: Text Field (kết xuất)
    - txtTBCDonGia: Text Field (kết xuất)
    - lstSTK: List (kết xuất)
- **Nhập:**
  - Mã sách
  - Tên sách
  - Ngày nhập
  - Nhà xuất bản
  - Đơn giá
  - Số lượng



- Sách giáo khoa (Tình trạng)/ Sách tham khảo (Thuế)

- **Xuất:**

- Danh sách sách
- Tổng thành tiền sách giáo khoa/ sách tham khảo
- Đơn giá trung bình sách tham khảo

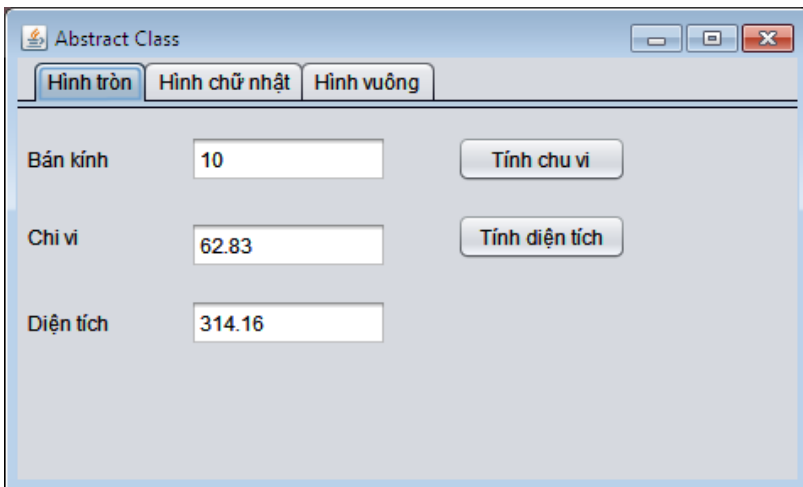
✓ **Hướng dẫn**

- Tạo một lớp có tên là Sach gồm có
  - Các thuộc tính: maSach, tenSach, ngayNhap, NXB, donGia, soLuong
  - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính trên
  - Phương thức tinhThanhTien() với thành tiền = số lượng \* đơn giá
- Tạo một lớp có tên là SachGiaoKhoa kế thừa lớp Sach gồm có
  - Các thuộc tính: tinhTrang
  - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính ứng với các thuộc tính
  - Phương thức tinhThanhTien() với thành tiền = số lượng \* đơn giá nếu sách mới và thành tiền = số lượng \* đơn giá \* 50% nếu là sách cũ
- Tạo một lớp có tên là SachThamKhao kế thừa từ lớp Sach gồm có:
  - Các thuộc tính: thue
  - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính ứng với các thuộc tính
  - Phương thức tinhThanhTien() với thành tiền = số lượng \* đơn giá + (số lượng \* đơn giá \* thuế) / 100

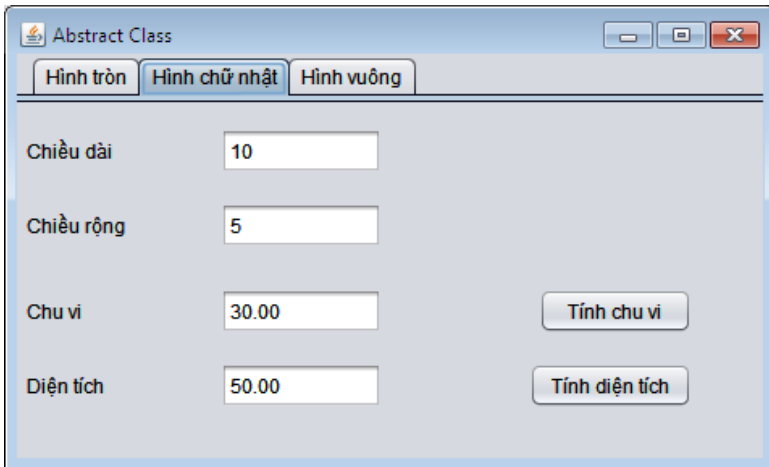
- Thực hiện những hàm xử lý sự kiện cho các nút nhấn:
  - Khi chọn nút “Nhập sách”, thực hiện:
    - Gọi sử dụng class sách tương ứng với thông tin nhập
    - Đưa sách vừa nhập vào danh sách Sách giáo khoa hoặc Sách tham khảo
    - Tính toán
    - Xuất kết quả ra các điều khiển
  - Khi chọn nút “tiếp tục”, thực hiện:
    - Đặt lại giá trị rỗng cho các điều khiển để người dùng tiếp tục nhập sách

## 7.4. Tính chu vi & diện tích các hình

- ✓ **Yêu cầu:** Viết chương trình tính chu vi và diện tích của một số hình như sau:



*Hình tròn*



Abstract Class

Hình tròn   **Hình chữ nhật**   Hình vuông

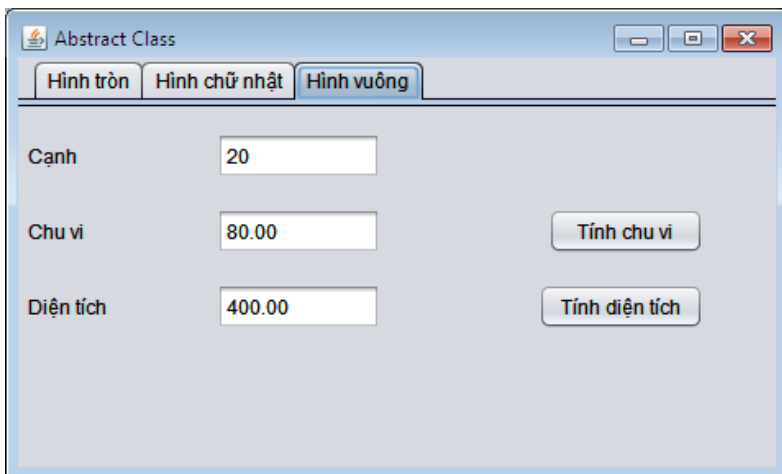
Chiều dài: 10

Chiều rộng: 5

Chu vi: 30.00   **Tính chu vi**

Diện tích: 50.00   **Tính diện tích**

*Hình chữ nhật*



Abstract Class

Hình tròn   Hình chữ nhật   **Hình vuông**

Cạnh: 20

Chu vi: 80.00   **Tính chu vi**

Diện tích: 400.00   **Tính diện tích**

*Hình vuông*

✓ **Hướng dẫn sử dụng:**

- Nhập bán kính hình tròn hoặc chiều dài và chiều rộng hình chữ nhật hoặc cạnh hình vuông, nhấn "Tính chu vi"/ "Tính diện tích" => hiển thị kết quả tương ứng



## ✓ Tóm tắt yêu cầu

### ▪ Thiết kế giao diện người dùng:

- frmAbstractClass: JFrame (chính thuộc tính defaultCloseOperation của JFrame là DISPOSE)

*\* (Các thể hiện phía dưới đều nằm trong form)*

- tptHinh: Tab Panel (chứa 3 panel hình tròn, hình chữ nhật, hình vuông)
  - pnlCircle: Panel (chứa thông tin hình tròn)
    - txtR: Text Field (nhập liệu)
    - txtPTron: Text Field (kết xuất)
    - txtSTron: Text Field (kết xuất)
    - btnTinhPTron (xử lý tính chu vi hình tròn và in kết quả)
    - btnTinhSTron (xử lý tính diện tích hình tròn và in kết quả)
  - pnlRectangle: Panel (chứa thông tin hình chữ nhật)
    - txtCanhA: Text Field (nhập liệu)
    - txtCanhB: Text Field (nhập liệu)
    - txtPCN: Text Field (kết xuất)
    - txtSCN: Text Field (kết xuất)
    - btnTinhPCN (xử lý tính chu vi hình chữ nhật và in kết quả)
    - btnTinhSCN (xử lý tính diện tích hình chữ nhật và in kết quả)
  - pnlSquare: Panel (chứa thông tin hình vuông)
    - txtCanh: Text Field (nhập liệu)
    - btnTinhPV (xử lý tính chu vi hình vuông và in kết quả)



- btnTinhSV (xử lý tính diện tích hình vuông và in kết quả)

▪ **Nhập:**

- Bán kính/ chiều dài & chiều rộng/ cạnh

▪ **Xuất:**

- Chu vi
- Diện tích

✓ **Hướng dẫn**

- Xây dựng lớp trừu tượng Shape như sau:

```
public abstract class Shape {
    public abstract double perimeter();
    public abstract double area();
}
```

- Xây dựng lớp Circle kế thừa lớp Shape

```
public class Circle extends Shape {
    public double r;
    public void setR(double r) {
        this.r = r;
    }
    public double getR() {
        return r;
    }
    @Override
```





```
public double perimeter() {
    return 2* Math.PI * r;
}

@Override
public double area() {
    return Math.PI * Math.pow(r,2);
}
}
```

- Làm tương tự cho các lớp còn lại
- Khởi tạo và gọi sử dụng các lớp này trong form để tính chu vi, diện tích và xuất kết quả

```
// tính chu vi hình tròn
private void
brnTinhPTronActionPerformed(java.awt.event.ActionEvent evt) {
    double r = Double.parseDouble(txtR.getText());
    Circle circle = new Circle();
    circle.r = r;
    txtPTron.setText(String.format("%.2f",square.perimeter()));
}

// tính diện tích hình tròn
private void
btnTinhSTronActionPerformed(java.awt.event.ActionEvent evt) {
    double r = Double.parseDouble(txtR.getText());
```

```

Circle circle = new Circle();

circle.r = r;

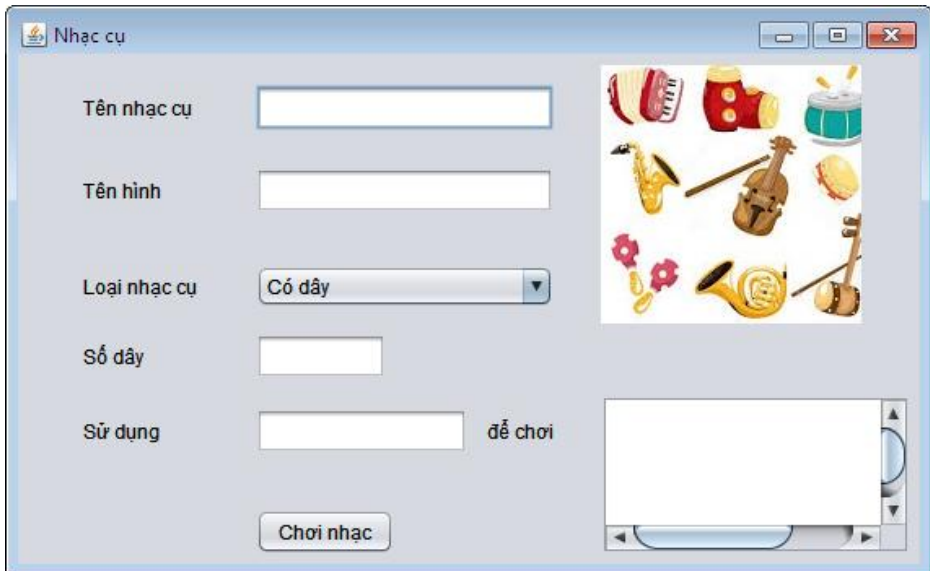
txtSTron.setText(String.format("%.2f",circle.area()));

}
    
```

- Làm tương tự cho các hình còn lại

## 7.5. Chơi nhạc

- ✓ **Yêu cầu: Viết chương trình chơi nhạc cụ như sau:**



*Khởi động chương trình*




Nhạc cụ

Tên nhạc cụ

Tên hình

Loại nhạc cụ

Có dây ▼

Số dây

Chơi nhạc



Đàn guitar có 6 dây

*Nhạc cụ có dây*


Nhạc cụ

Tên nhạc cụ

Tên hình

Loại nhạc cụ

Không dây

Sử dụng

để chơi

Chơi nhạc



Phát ra âm thanh khi tác động v  
ào phím đàn

*Nhạc cụ không dây*



Mô tả: Chương trình có 2 loại nhạc cụ:

- Nhạc cụ có dây: tên nhạc cụ, tên hình, số dây.
  - Chơi nhạc được thực hiện như sau: Tên đàn + số dây
- Nhạc cụ không dây: tên nhạc cụ, tên hình, sử dụng
  - Chơi nhạc được thực hiện như sau: “Phát ra âm thanh khi tác động vào” + sử dụng

Dựa vào mô tả trên, hãy:

- Tạo lớp trừu tượng `NhacCu` với thuộc tính chung và phương thức trừu tượng `choiNhac()`
- Tạo lớp `NhacCuCoDay` kế thừa từ lớp nhạc cụ với các thuộc tính riêng và hiện thực phương thức `choiNhac()`.
- Tạo lớp `NhacCuKhongDay` kế thừa từ lớp nhạc cụ với các thuộc tính riêng và hiện thực phương thức `choiNhac()`
- Hiển thị hình ảnh nhạc cụ và thông tin chơi nhạc

✓ **Hướng dẫn sử dụng:**

- Nhập Tên nhạc cụ, Tên hình
- Chọn Loại nhạc cụ “Có dây”, nhập Số dây  
Hoặc
- Chọn Loại nhạc cụ “Không dây”, nhập Sử dụng
- Nhấn “Chơi nhạc” -> hiển thị kết quả

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- `frmChoiNhac`: `JFrame`

\* (Các thể hiện phía dưới đều nằm trong form)

- `txtTen`: Text Field (nhập liệu)



- txtHinh: Text Field (nhập liệu)
- cbbLoai: Combo Box (chọn Có dây/ không dây)
- txtSoDay: Text Field (nhập liệu)
- txtSuDung: Text Field (nhập liệu)
- lblHinh: Label (kết xuất)
- txtaChoiDan: Text Area (kết xuất)
- btnChoi: Button (xử lý và in kết quả)
- **Nhập:**
  - Tên nhạc cụ, Tên hình
  - Loại nhạc cụ “Có dây”, Số dây
  - Hoặc
  - Loại nhạc cụ “Không dây”, Sử dụng
- **Xuất:**
  - Hình ảnh nhạc cụ
  - Thông tin chơi nhạc
- ✓ **Hướng dẫn**
  - Tạo một trừu tượng có tên là NhạcCu

```
public abstract class NhạcCu {
    String ten;
    String hinh;
    abstract public String chơiNhac();
    public NhạcCu() {
    }
}
```



```
public NhạcCu(String ten, String hinh) {
    this.ten = ten;
    this.hinh = hinh;
}
public String getTen() {
    return ten;
}
public void setTen(String ten) {
    this.ten = ten;
}
public String getHinh() {
    return hinh;
}
public void setHinh(String hinh) {
    this.hinh = hinh;
}
}
```

– Tạo lớp NhạcCuCoDay kế thừa lớp NhạcCu

```
public class NhạcCuCoDay extends NhạcCu {
    int soDay;
    public NhạcCuCoDay(int soDay, String ten, String hinh) {
        super(ten, hinh);
    }
}
```



```

        this.soDay = soDay;
    }

    public int getSoDay() {
        return soDay;
    }

    public void setSoDay(int soDay) {
        this.soDay = soDay;
    }

    @Override
    public String chonNhac() {
        return this.getTen() + " có " + this.soDay + " dây";
    }
}

```

- Làm tương tự cho lớp NhạcCuKhongDay
- Gọi sử dụng các lớp trong form, xử lý và hiển thị kết quả

```

private void btnChoiActionPerformed(java.awt.event.ActionEvent evt) {
    String chon = "";
    String hinh = "src/images/" + txtDuongDan.getText();

    if(cbbLoai.getSelectedIndex()==0){ // nhac cu co day
        NhạcCuCoDay nccd = new
        NhạcCuCoDay(Integer.parseInt(txtSoDay.getText()), txtTen.getText(), hinh);
        chon = nccd.chonNhac();
    }
}

```



```

    }
    else{
        NhạcCuKhongDay nckd = new
        NhạcCuKhongDay(txtSuDung.getText(), txtTen.getText(),hinh);
        choi = nckd.choiNhac();
    }
    txtaChoiDan.setText(choi);
    ImageIcon iconLogo = new ImageIcon(hinh);
    lblHinh.setIcon(iconLogo);
}

```

- Chú ý: cần tạo thư mục images trong thư mục src của project và chuẩn bị sẵn một số hình ảnh cần thiết cho bài tập này.

## 7.6. Phương tiện giao thông và động cơ

- ✓ **Yêu cầu:** Xây dựng ứng dụng tính vận tốc và trung bình tiêu thụ nhiên liệu như sau:





Phương tiện giao thông - động cơ

Thông tin chung

Quãng đường (km)  Thời gian (giờ)  Nhiên liệu (lít)

☒ Xe hơi ☐ Tàu thủy

Tên xe  Trọng lượng (kg)

Hãng sản xuất  Sức tải (kg)

Vận tốc nước xuôi dòng (km/h)

Vận tốc nước ngược dòng (km/h)

Nhiên liệu tiêu hao (lít)

Vận tốc (km/giờ)  Tiêu thụ nhiên liệu (km/lít)

*Xe hơi*

Phương tiện giao thông - động cơ

Thông tin chung

Quãng đường (km)  Thời gian (giờ)  Nhiên liệu (lít)

☐ Xe hơi ☒ Tàu thủy

Tên xe  Trọng lượng (kg)

Hãng sản xuất  Sức tải (kg)

Vận tốc nước xuôi dòng (km/h)

Vận tốc nước ngược dòng (km/h)

Nhiên liệu tiêu hao (lít)

Vận tốc (km/giờ)  Tiêu thụ nhiên liệu (km/lít)

*Tàu thủy*



Mô tả: Chương trình có 2 loại phương tiện giao thông:

- Xe hơi: tên xe, hãng sản xuất.
  - Vận tốc xe hơi (km/giờ) = Quãng đường/ Thời gian
  - Tiêu hao nhiên liệu xe hơi (km/lít) = Quãng đường/ Số lít
- Tàu thủy: Trọng lượng, Sức tải, Vận tốc nước xuôi dòng, Vận tốc nước ngược dòng, Nhiên liệu tiêu hao
  - Vận tốc tàu thủy (km/giờ)= Quãng đường/ Thời gian + Vận tốc nước xuôi dòng – Vận tốc nước ngược dòng
  - Tiêu thụ nhiên liệu tàu thủy (km/lít) = Quãng đường / (Số lít – nhiên liệu tiêu hao)

Dựa vào mô tả trên, hãy:

- Tạo interface DongCo có phương thức trừu tượng tinhTieuThuNhiemLieu(double QuangDuong, double SoLit)
- Tạo interface PhuongTienGiaoThong có phương thức trừu tượng tinhVanToc(double QuangDuong, double ThoiGian)
- Tạo lớp XeHoi implements 2 interface trên và hiện thực các phương thức trừu tượng.
- Tạo lớp TauThuy implements 2 interface trên và hiện thực các phương thức trừu tượng.
- Tính toán và hiển thị vận tốc, tiêu thụ nhiên liệu tương ứng với từng loại

✓ **Hướng dẫn sử dụng:**

- Nhập Quãng đường, Thời gian, Nhiên liệu
- Chọn “Xe hơi” => nhập Tên xe, Hãng sản xuất => Nhấn “Tính vận tốc – Tiêu thụ nhiên liệu” => Hiển thị kết quả



Hoặc

- Chọn “Tàu thủy” => nhập Trọng lượng, Sức tải, Vận tốc nước xuôi dòng, Vận tốc nước ngược dòng, Nhiên liệu tiêu hao => Nhấn “Tính vận tốc – Tiêu thụ nhiên liệu” => Hiển thị kết quả

### ✓ Tóm tắt yêu cầu

#### ▪ Thiết kế giao diện người dùng:

- frmInterface: JFrame (chính thuộc tính defaultCloseOperation của JFrame là DISPOSE)

*\* (Các thể hiện phía dưới đều nằm trong form)*

- txtQuangDuong: Text Field (nhập liệu)
- txtThoiGian: Text Field (nhập liệu)
- txtNhiemLieu: Text Field (nhập liệu)
- btnngTauXe: Button Group
  - rbTXe: RadioButton (chọn)
    - txtTenXe: Text Field (nhập liệu)
    - txtHSX: Text Field (nhập liệu)
  - rbTTau: Radio Button (chọn)
    - txtTrongLuong: Text Field (nhập liệu)
    - txtSucTai: Text Field (nhập liệu)
    - txtXuoi: Text Field (nhập liệu)
    - txtNguoc: Text Field (nhập liệu)
    - txtTieuHao: Text Field (nhập liệu)
- txtVanToc: Text Field (kết xuất)
- txtTieuThuNL: Text Field (kết xuất)
- btnVanToc\_TieuThuNL: Button (xử lý tính toán và in kết quả)

#### ▪ Nhập:



- Quãng đường
  - Thời gian
  - Nhiên liệu
  - Chọn “Xe hơi”
    - Tên xe
    - Hãng sản xuất
  - Chọn “Tàu thủy”
    - Trọng lượng
    - Sức tải
    - Vận tốc nước xuôi dòng
    - Vận tốc nước ngược dòng
    - Nhiên liệu tiêu hao
  - **Xuất:**
    - Vận tốc
    - Tiêu thụ nhiên liệu
- ✓ **Hướng dẫn**
- Tạo Interface `PhuongTienGiaoThong`

```
public interface PhuongTienGiaoThong {  
    double tinhVanToc(double QuangDuong, double ThoiGian);  
}
```

- Tạo interface `DongCo`:

```
public interface DongCo {
```



```
double tinhTieuThuNhiemLieu(double QuangDuong, double SoLit);
}
```

- Tạo class XeHoi implements 2 interface trên:
  - Thuộc tính: tenXe, hangSanXuat
  - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thuộc tính trên
  - Phương thức: hiện thực các phương thức của 2 interface trên

```
public class XeHoi implements PhuongTienGiaoThong, DongCo{
    String tenXe;
    String hangSanXuat;
    public XeHoi(String tenXe, String hangSanXuat) {
        this.tenXe = tenXe;
        this.hangSanXuat = hangSanXuat;
    }
    public String getTenXe() {
        return tenXe;
    }
    public void setTenXe(String tenXe) {
        this.tenXe = tenXe;
    }
    public String getHangSanXuat() {
        return hangSanXuat;
    }
}
```



```

    }

    public void setHangSanXuat(String hangSanXuat) {
        this.hangSanXuat = hangSanXuat;
    }

    /**
     *
     * @param QuangDuong
     * @param ThoiGian
     * @return
     */
    @Override
    public double tinhVanToc(double QuangDuong, double ThoiGian) {
        return QuangDuong/ThoiGian;
    }

    /**
     *
     * @param QuangDuong
     * @param SoLit
     * @return
     */
    @Override
    public double tinhTieuThuNhiemLieu(double QuangDuong, double SoLit) {
        return QuangDuong/SoLit;
    }

```



```
}

```

```
}

```

- Tạo class `TauThuy` implements 2 interface trên:
  - Thuộc tính: `trongLuong`, `sucTai`, `vanTocNuocXuoiDong`, `vanTocNuocNguocDong`, `nhienLieuTieuHao`
  - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thuộc tính trên
  - Phương thức: hiện thực các phương thức của 2 interface trên

```
public class TauThuy implements PhuongTienGiaoThong, DongCo{
    double trongLuong;
    int sucTai;
    double vanTocNuocXuoiDong;
    double vanTocNuocNguocDong;
    double nhienLieuTieuHao;

    public TauThuy(double trongLuong, int sucTai, double
vanTocNuocXuoiDong, double vanTocNuocNguocDong, double
nhienLieuTieuHao) {
        this.trongLuong = trongLuong;
        this.sucTai = sucTai;
        this.vanTocNuocXuoiDong = vanTocNuocXuoiDong;
        this.vanTocNuocNguocDong = vanTocNuocNguocDong;
    }
}
```



```

        this.nhienLieuTieuHao = nhienLieuTieuHao;
    }

    public double getTrongLuong() {
        return trongLuong;
    }

    public void setTrongLuong(double trongLuong) {
        this.trongLuong = trongLuong;
    }

    public int getSucTai() {
        return sucTai;
    }

    public void setSucTai(int sucTai) {
        this.sucTai = sucTai;
    }

    public double getVanTocNuocXuaiDong() {
        return vanTocNuocXuaiDong;
    }

    public void setVanTocNuocXuaiDong(double vanTocNuocXuaiDong) {
        this.vanTocNuocXuaiDong = vanTocNuocXuaiDong;
    }

    public double getVanTocNuocNguocDong() {
        return vanTocNuocNguocDong;
    }

```





```

    }

    public void setVanTocNuocNguocDong(double vanTocNuocNguocDong) {
        this.vanTocNuocNguocDong = vanTocNuocNguocDong;
    }

    public double getNhienLieuTieuHao() {
        return nhienLieuTieuHao;
    }

    public void setNhienLieuTieuHao(double nhienLieuTieuHao) {
        this.nhienLieuTieuHao = nhienLieuTieuHao;
    }

    @Override
    public double tinhVanToc(double QuangDuong, double ThoiGian) {
        double vanToc = (QuangDuong/ThoiGian) + this.vanTocNuocXuaiDong -
this.vanTocNuocNguocDong;

        return vanToc;
    }

    @Override
    public double tinhTieuThuNhienLieu(double QuangDuong, double SoLit) {
        return QuangDuong/(SoLit - this.nhienLieuTieuHao);
    }
}

```

- Khởi tạo và sử dụng 2 lớp vừa xây dựng trong form, tính toán và hiển thị kết quả

## BÀI 8: Sử dụng dịch vụ

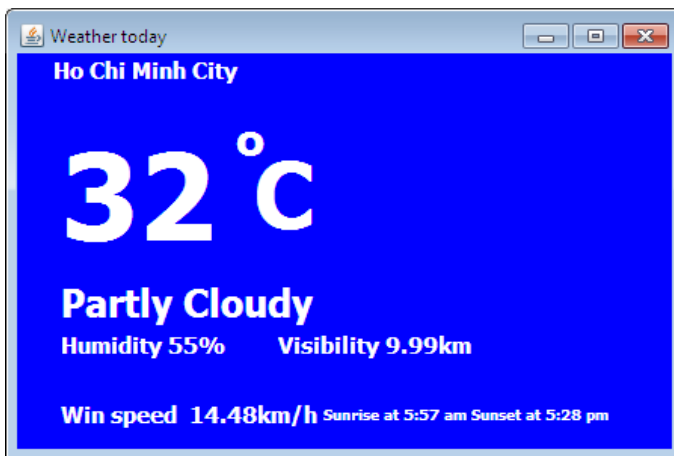


Mục tiêu chính:

Sử dụng dịch vụ trong ứng dụng

Thiết kế hôm nay

- ✓ **Yêu cầu:** Viết ứng dụng xem thời tiết hôm nay



- ✓ **Hướng dẫn sử dụng:**
  - Load frm hiện lên thông tin thời tiết hôm nay ở TP.HCM
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmWeatherToDay: form
      - \* (Các thể hiện phía dưới đều nằm trong form)
        - Các Label tương ứng với thông tin đọc được
  - **Nhập**
    - Không có



▪ **Xuất**

- Thông tin thời tiết hôm nay ở TP.HCM

▪ **Quy tắc xử lý:**

- Không có

✓ **Thuật giải**

- Load frm lên, đọc nội dung xml từ đường dẫn  
(<http://weather.yahooapis.com/forecastrss?w=1252431&u=c>)
- Lấy những thông tin cần thiết và gán cho Label tương ứng

✓ **Hướng dẫn**

```
public void showWeather() throws SAXException, IOException{
try {
    URL url = new
URL("http://weather.yahooapis.com/forecastrss?w=1252431&u=c");
    Document doc = ReadOrWriteXML.readXMLFromURL(url);
    NodeList tempElement =
doc.getElementsByTagName("yweather:condition");
    for (int i = 0; i < tempElement.getLength(); i++) {
        Node n = tempElement.item(i);
        if (n.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element)n;
            String s = eElement.getAttribute("temp");
            lbltemperature.setText(s);
            s = eElement.getAttribute("text");
            lblCondition.setText(s);
        }
    }
    NodeList atmosphereElement =
doc.getElementsByTagName("yweather:atmosphere");
    for (int i = 0; i < atmosphereElement.getLength(); i++)
    {
        Node n = atmosphereElement.item(i);
        if (n.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element)n;
            String s =
eElement.getAttribute("humidity");
```



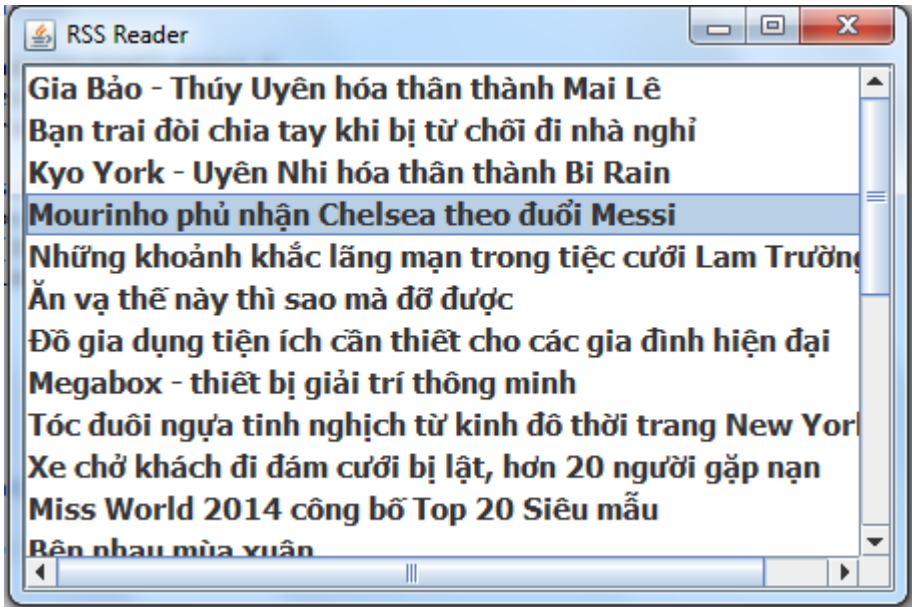
```

        lblHumidity.setText("Humidity " + s + "%");
        s = eElement.getAttribute("visibility");
        lblVisibility.setText("Visibility " + s + "km");
    }
}
NodeList winElement =
doc.getElementsByTagName("yweather:wind");
    for (int i = 0; i < winElement.getLength(); i++) {
        Node n = winElement.item(i);
        if (n.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element)n;
            String s = eElement.getAttribute("speed");
            lblWind.setText("Win speed " + s +
"km/h");
        }
    }
NodeList sunElement =
doc.getElementsByTagName("yweather:astronomy");
    for (int i = 0; i < sunElement.getLength(); i++) {
        Node n = sunElement.item(i);
        if (n.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element)n;
            String s = eElement.getAttribute("sunset");
            String r =
eElement.getAttribute("sunrise");
            lblSun.setText("Sunrise at " + r + " Sunset
at " + s);
        }
    }
} catch (MalformedURLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

```

## 8.2. Xem tin tức

✓ **Yêu cầu: Viết ứng dụng xem nhanh tin tức**



✓ **Hướng dẫn sử dụng:**

- Load frm hiện lên tin tức ngày hôm nay

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- FrmRssReader: form

\* (Các thể hiện phía dưới đều nằm trong form)

- IstMain: List (load tất cả các title vào list)

▪ **Nhập**

- Không có

▪ **Xuất**

- Tin tức ngày hôm nay



✓ **Thuật giải**


- Load frm lên, đọc nội dung xml từ đường dẫn (http://vnexpress.net/rss/tin-moi-nhat.rss)
- Lấy các title gán vào list

✓ **Hướng dẫn**

- Xây dựng phương thức loadRSS() để load danh sách các tiêu đề tin tức mới trong ngày

```
public void loadRSS() throws SAXException, IOException {
    DefaultListModel<String> myDefaultLst = new
DefaultListModel<String>();
    URL url = new URL("http://vnexpress.net/rss/tin-moi-
nhat.rss");
    Document doc = ReadOrWriteXML.readXMLFromURL(url);
    NodeList itemList = doc.getElementsByTagName("item");
    for (int i = 0; i < itemList.getLength(); i++) {
        Element eElement = (Element)itemList.item(i);
        String s =
eElement.getElementsByTagName("title").item(0).getTextContent()
;
        myDefaultLst.addElement(s);
    }
    lstMain.setModel(myDefaultLst);
}
```

# BÀI 9: Tổng kết



Mục tiêu chính:

Ôn tập các kiến thức của module 1

**Cập nhật – Xóa liên hệ**

- ✓ **Yêu cầu:** Xây dựng chương trình cập nhật – xóa liên hệ như sau:

Cập nhật - Xóa Liên Hệ

Tên




ĐTDD

Hình

...

Cập nhật

Xóa




Hình ảnh	Họ tên	ĐTDD
	Trần Văn Anh	0989753951
	Trần Thùy Dương	0989753951
	Lê Bảo Khuyên	0913852147

Hình 9.1: Form Cập nhật – Xóa liên hệ khi chạy

**Cập nhật - Xóa Liên Hệ**

Tên:  ĐTDĐ:

Hình:  ...




Hình ảnh	Họ tên	ĐTDĐ
	Trần Văn Anh	0989753951
	Trần Thùy Dương	0989753951
	Lê Bảo Khuyến	0913852147

*Sau khi chọn một liên hệ muốn cập nhật*

**Cập nhật - Xóa Liên Hệ**

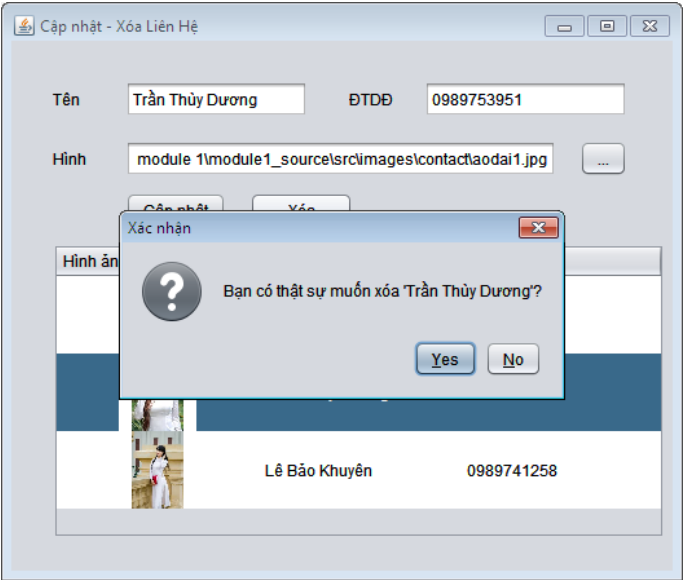
Tên:  ĐTDĐ:

Hình:  ...

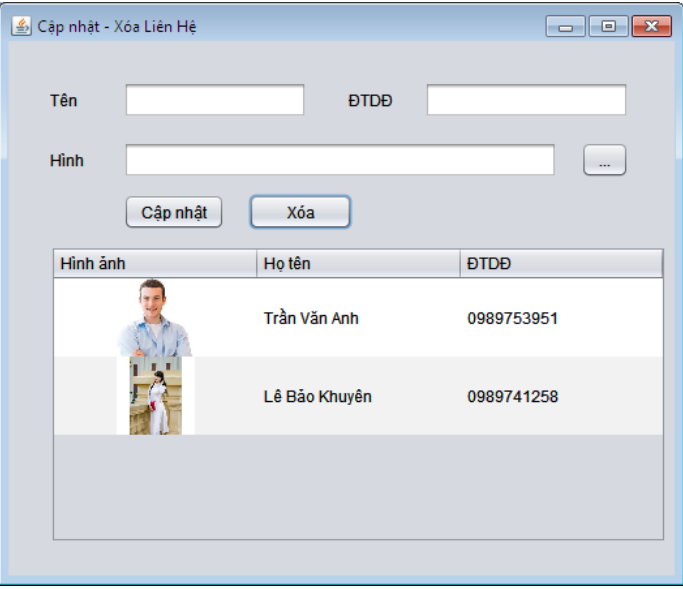
Hình ảnh	Họ tên	ĐTDĐ
	Trần Văn Anh	0989753951
	Trần Thùy Dương	0989753951
	Lê Bảo Khuyến	0989741258



*Sau khi cập nhật thông tin của liên hệ*



*Sau khi xóa thông tin của liên hệ*





✓ **Hướng dẫn sử dụng:**

- Chọn 1 liên hệ trên lưới > Nội dung sẽ được hiển thị > Cập nhật lại thông tin và nhấn "Cập nhật" để ghi hoặc nhấn "Xóa" để xóa.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng**

- frmCapNhatXoaLienhe: (extends từ JFrame)

*\* Học viên thiết kế form như hình 9.1*

✓ **Hướng dẫn**

- Khai báo biến toàn cục lst kiểu List<LienHe> để chứa danh sách liên hệ và biến model kiểu DefaultTableModel để chứa các liên hệ trước khi đưa vào table

```
List<LienHe> lst;
```

```
DefaultTableModel model = new DefaultTableModel();
```

- Ở phương thức khởi tạo form: Đọc tập tin lienhe.txt, đưa các liên hệ vào lst => chuyển vào model => hiển thị lên table:

```
public frmCapNhatXoaLienHe() {
    initComponents();
    lst = new ArrayList<>();
    try (ObjectInputStream in = new ObjectInputStream(
        new BufferedInputStream(new
        FileInputStream("src/Bai5/lienhe.txt")))) {
        while (true) {
            LienHe b = (LienHe) in.readObject();
            System.out.println(b.hoTen + " - " + b.dtdd + " - " + b.hinh);
            lst.add(b);
        }
    }
```



```
} catch (ClassNotFoundException | IOException ex) {
    System.out.println(ex.getMessage());
}
```

```
model.addColumn("Hình ảnh");
model.addColumn("Họ tên");
model.addColumn("ĐTDD");
```

```
for (LienHe lh : lst) {
    ImageIcon img = new ImageIcon("src/images/contact/" + lh.hinh);
    model.addRow(new Object[]{img, lh.hoTen, lh.dtdd});
}
tblKetQua.setRowHeight(60);
tblKetQua.setModel(model);
```

```
tblKetQua.getColumnModel().getColumn(0).setCellRenderer(new
Bai5.ImageTableCellRenderer(60, 60));
}
```

- Xử lý sự kiện của MouseClicked table để lấy thông tin dòng được chọn và đưa nội dung vào các điều khiển để chuẩn bị cho việc cập nhật:

```
private void tblKetQuaMouseClicked(java.awt.event.MouseEvent evt) {
    String path =
String.valueOf(model.getValueAt(tblKetQua.getSelectedRow(), 0));
    File f = new File(path);
    txtHinh.setText(f.getAbsolutePath());
    txtTen.setText(String.valueOf(model.getValueAt(tblKetQua.getSelectedRow(),
1)));
}
```



```
txtDTDD.setText(String.valueOf(model.getValueAt(tblKetQua.getSelectedRow(
), 2)));
}
```

– Xử lý sự kiện cho nút “Cập nhật”

```
private void btnCapNhatActionPerformed(java.awt.event.ActionEvent evt) {
    // Đưa hình ảnh cập nhật vào nơi lưu trữ
    File f = new File(txtHinh.getText());
    String path = "src/images/contact/";
    String duong_dan = path + f.getName();
    try {
        Files.copy(f.toPath(), (new File(path + f.getName())).toPath(),
        StandardCopyOption.REPLACE_EXISTING);
        System.out.println("Đã đưa lại hình liên hệ thư mục contact");
    } catch (IOException ex) {
        Logger.getLogger(frmThemLienHeMoi.class.getName()).log(Level.SEVERE,
        null, ex);
    }
    // Ghi lại thông tin của liên hệ sau khi người dùng cập nhật
    for (LienHe lh : lst) {
        if (lh.hoTen.equals(txtTen.getText())) {
            lh.dtdd = txtDTDD.getText();
            lh.hinh = f.getName();
        }
    }
    // ghi danh sách liên hệ vào tập tin lienhe.txt sau khi người dùng cập nhật
    try (ObjectOutputStream out = new ObjectOutputStream(
```



```

        new BufferedOutputStream(new
        FileOutputStream("src/Bai5/lienhe.txt")))) {
            out.reset();
            for (LienHe lh : lst) {
                out.writeObject(lh);
            }
            out.flush();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        // hiển thị lại nội dung đã cập nhật lên dòng trên table
        ImageIcon img = new ImageIcon(duong_dan);
        model.setValueAt(img, tblKetQua.getSelectedRow(), 0);
        model.setValueAt(txtTen.getText(), tblKetQua.getSelectedRow(), 1);
        model.setValueAt(txtDTDD.getText(), tblKetQua.getSelectedRow(), 2);
    }

```

– Xử lý sự kiện cho nút “Xóa”

```

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // Xác nhận thông tin xóa, nếu chọn đồng ý xóa
    int reply = JOptionPane.showConfirmDialog(rootPane, "Bạn có thật sự
    muốn xóa " + txtTen.getText() + "?", "Xác nhận",
    JOptionPane.YES_NO_OPTION);
    if (reply == JOptionPane.YES_OPTION) {
        // tạo list mới chứa các liên hệ sau khi bỏ đi liên hệ đã chọn xóa
        List<LienHe> lstAfter = new ArrayList<>();
        for (LienHe lh : lst) {

```



```

        if (lh.hoTen.equals(txtTen.getText())) {
            continue;
        }
        lstAfter.add(lh);
    }
    // ghi list mới này vào tập tin lienhe.txt
    try (ObjectOutputStream out = new ObjectOutputStream(
        new BufferedOutputStream(new
FileOutputStream("src/Bai5/lienhe.txt")))) {
        out.reset();
        for (LienHe lh : lstAfter) {
            out.writeObject(lh);
        }
        out.flush();
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
    // xóa các liên hệ trong lst và đưa các liên hệ trong list mới vào
    lst.clear();
    boolean addAll = lst.addAll(lstAfter);
    // xóa dòng đã chọn trên table
    model.removeRow(tblKetQua.getSelectedRow());
    // xóa nội dung trên các điều khiển để người dùng tiếp tục thao tác
    txtTen.setText("");
    txtHinh.setText("");
    txtDTDD.setText("");

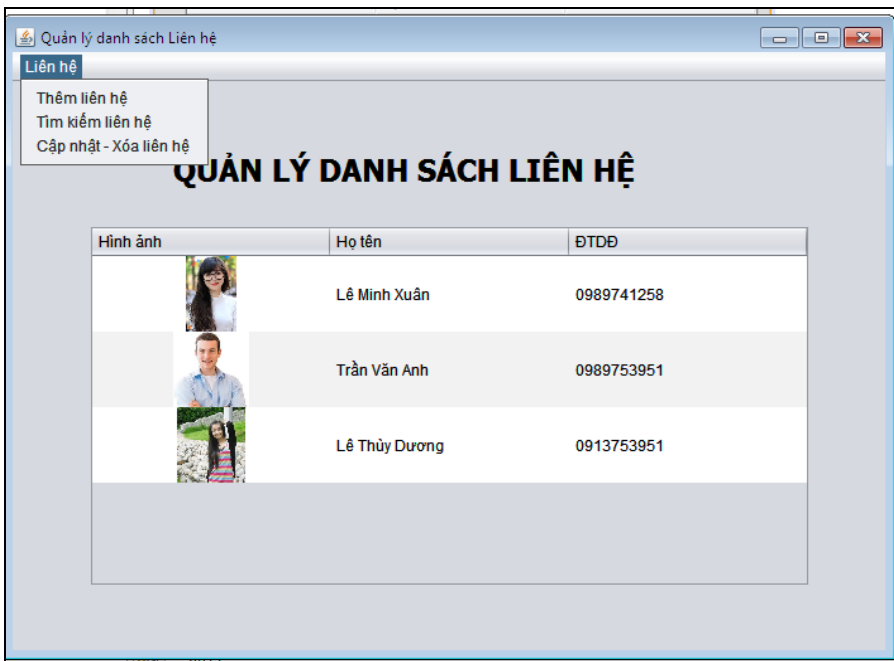
```

}

}

## 9.2. Hoàn chỉnh ứng dụng

- ✓ **Yêu cầu: Hoàn chỉnh ứng dụng Quản lý danh bạ điện thoại**
  - Gồm có các màn hình: Hiển thị danh bạ, Thêm mới liên hệ, Tìm kiếm liên hệ, Cập nhật/ Xóa Liên Hệ
  - Kỹ thuật: thao tác với tập tin, List, đối tượng (class)



Màn hình chính

- ✓ **Hướng dẫn sử dụng:**
  - Chọn các công việc trên danh bạ để thực hiện
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**



- Tạo formMain để quản lý các form, có Code > Generate Center: check
- Tạo Menu và các menuItem
- Học viên gắn các màn hình đã xây dựng vào các menuItem
- Thiết lập thuộc tính default close Operation là DISPOSE, Code > Generate Center: check: cho các form Hiển thị danh bạ, Thêm mới liên hệ, Tìm kiếm liên hệ, Cập nhật/ Xóa Liên Hệ

### ✓ Hướng dẫn

- Khi khởi tạo formMain, đọc và hiển thị các liên hệ trong tập tin lienhe.txt

```
public frmMainLienHe() {
    initComponents();
    List<LienHe> lst;
    DefaultTableModel model = new DefaultTableModel();
    lst = new ArrayList<>();
    try (ObjectInputStream in = new ObjectInputStream(
        new BufferedInputStream(new
        FileInputStream("src/Bai5/lienhe.txt")))) {
        while (true) {
            LienHe b = (LienHe) in.readObject();
            System.out.println(b.hoTen + " - " + b.dtdd + " - " + b.hinh);
            lst.add(b);
        }
    } catch (ClassNotFoundException | IOException ex) {
        System.out.println(ex.getMessage());
    }
    model.addColumn("Hình ảnh");
```





```

model.addColumn("Họ tên");
model.addColumn("ĐTĐĐ");
for (LienHe lh : lst) {
    ImageIcon img = new ImageIcon("src/images/contact/" + lh.hinh);
    model.addRow(new Object[]{img, lh.hoTen, lh.dtdd});
}
tblKetQua.setRowHeight(60);
tblKetQua.setModel(model);
tblKetQua.getColumnModel().getColumn(0).setCellRenderer(new
Bai5.ImageTableCellRenderer(60, 60));
}

```

- Xử lý sự kiện cho các menuItem để mở form:

```

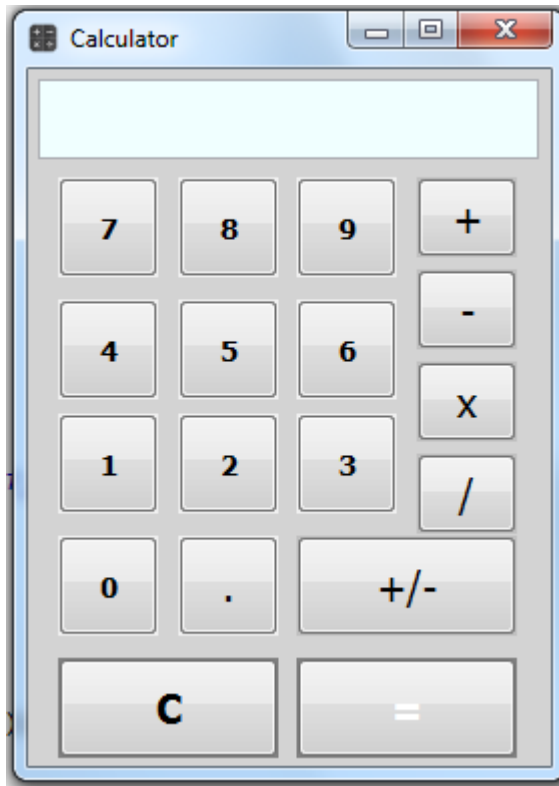
private void mniThemActionPerformed(java.awt.event.ActionEvent evt) {
    frmThemLienHeMoi frm = new frmThemLienHeMoi();
    frm.setVisible(rootPaneCheckingEnabled);
}

```

- Làm tương tự cho các menuItem khác

### 9.3. Máy tính bỏ túi (bài làm thêm)

- ✓ **Yêu cầu:** Xây dựng ứng dụng máy tính bỏ túi



- ✓ **Hướng dẫn sử dụng:**
  - Sử dụng như máy tính bỏ túi đơn giản
- ✓ **Tóm tắt yêu cầu**
  - **Thiết kế giao diện người dùng:**
    - FrmCalculator: form
    - \* (Các thể hiện phía dưới đều nằm trong form)
      - Học viên được cung cấp giao diện và bài làm gợi ý
  - **Nhập**
    - Số, phép tính...



- **Xuất**
  - Kết quả tính toán



# Mục lục

<b>BÀI 1: Tổng quan</b>	<b>2</b>
1.1. Vẽ "HELLO" ra màn hình console	3
1.2. Xuất câu chào	4
1.3. Tính tổng hai số nguyên	5
1.4. Tính tiền hàng	7
<b>BÀI 2: Các kiểu dữ liệu cơ sở</b>	<b>9</b>
2.1. Giải Phương trình bậc I	9
2.2. Tính giai thừa của một số	10
2.3. Tìm kiếm chuỗi	12
2.4. Xử lý chuỗi StringBuilder	14
2.5. Phân tách họ tên	18
2.6. Tính số ngày trong tháng	19
2.7. Tính năm âm lịch từ năm dương lịch	22
2.8. Tính chỉ số BMI	24
2.9. Xem bói tình yêu (bài làm thêm)	25
<b>BÀI 3: Giao diện</b>	<b>28</b>
3.1. Thông tin Liên hệ	28



<b>3.2. Thêm mới Liên hệ.....</b>	<b>31</b>
<b>3.3. Cập nhật liên hệ .....</b>	<b>32</b>
<b>3.4. Hệ thống bài tập - MenuBar .....</b>	<b>34</b>
<b>3.5. Hiện thị hình ảnh được chọn.....</b>	<b>35</b>
<b>3.6. Đặt vé xem phim .....</b>	<b>37</b>
<b>3.7. Thay đổi màu sắc cánh hoa (bài làm thêm) .....</b>	<b>39</b>
<b>BÀI 4: Truy xuất tập tin .....</b>	<b>42</b>
<b>4.1. Nhật ký.....</b>	<b>42</b>
<b>4.2. Tổng kết năm học.....</b>	<b>48</b>
<b>4.3. Thêm Liên hệ vào tập tin .....</b>	<b>57</b>
<b>BÀI 5: Mảng cơ sở.....</b>	<b>62</b>
<b>5.1. Tính tổng các phần tử trong mảng .....</b>	<b>62</b>
<b>5.2. Tìm kiếm trong mảng (làm tiếp từ bài tính tổng) .....</b>	<b>64</b>
<b>5.3. Mảng hai chiều .....</b>	<b>68</b>
<b>5.4. Sắp xếp tên và năm sinh .....</b>	<b>72</b>
<b>5.5. Bảng điểm .....</b>	<b>74</b>
<b>BÀI 6: Đối tượng.....</b>	<b>76</b>
<b>6.1. Thực hiện tính toán hai phân số .....</b>	<b>76</b>



<b>6.2. Quản lý CD .....</b>	<b>81</b>
<b>6.3. Phim chiếu rạp .....</b>	<b>86</b>
<b>6.4. Thêm liên hệ mới.....</b>	<b>93</b>
<b>6.5. Tìm kiếm liên hệ.....</b>	<b>98</b>
<b>6.6. Mua sắm online (làm thêm).....</b>	<b>104</b>
<b>BÀI 7: Lập trình hướng đối tượng nâng cao .....</b>	<b>109</b>
<b>7.1. Animal say hello .....</b>	<b>109</b>
<b>7.2. Tính lương nhân viên.....</b>	<b>113</b>
<b>7.3. Quản lý sách thư viện.....</b>	<b>119</b>
<b>7.4. Tính chu vi &amp; diện tích các hình .....</b>	<b>125</b>
<b>7.5. Chơi nhạc .....</b>	<b>130</b>
<b>7.6. Phương tiện giao thông và động cơ .....</b>	<b>136</b>
<b>BÀI 8: Sử dụng dịch vụ .....</b>	<b>146</b>
<b>8.1. Thời tiết hôm nay .....</b>	<b>146</b>
<b>8.2. Xem tin tức.....</b>	<b>148</b>
<b>BÀI 9: Tổng kết.....</b>	<b>151</b>
<b>9.1. Cập nhật – Xóa liên hệ.....</b>	<b>151</b>
<b>9.2. Hoàn chỉnh ứng dụng .....</b>	<b>159</b>



### **9.3. Máy tính bỏ túi (bài làm thêm) ..... 161**